Proceedings of the 4th Workshop on Exploring Beyond the Worst Case In Computational Social Choice (EXPLORE 2017)

Held as part of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017)

Sao Paulo, Brazil, May 9th, 2017

EXPLORE 2017 Website: <u>http://www.explore-2017.preflib.org/</u>

AAMAS 2016 Website: <u>http://www.aamas2017.org/</u>

EXPLORE-2017

Preface

This volume contains the papers presented at EXPLORE-2017: The 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice held on May 9th, 2016 in Sao Paulo, Brazil. There were 14 papers accepted for presentation at the workshop and one invited talk from Jerome Lang from CNRS LAMSADE and the Universite Paris-Dauphine, France.

This is the third installment of the EXPLORE workshop to be held at AAMAS. Our conference webpage at http://www.explore-2017.preflib.org/ has links to past iterations of the conference. More infomation about empirical testing in Social Choice and data can be found at www.PrefLib.org.

We thank all members of the ComSoc Community for their continued support of EXPLORE and PrefLib. The conference was managed with EasyChair.

April, 2017 Yorktown Heights, New York Haris Aziz John P. Dickerson Omer Lev Nicholas Mattei

Program Committee

Haris Aziz Markus Brill John Dickerson Edith Elkind Gabor Erdelvi Piotr Faliszewski Rupert Freeman Serge Gaspers Umberto Grandi Hadi Hosseini Aleksandr M. Kazachkov Jérôme Lang Kate Larson Omer Lev Yoad Lewenberg David Manlove Nicholas Mattei Reshef Meir Nina Narodytska Dominik Peters Maria Silvia Pini Jeffrey S. Rosenschein Piotr Skowron Rohit Vaish Mark Wilson Lirong Xia

Data61, CSIRO and UNSW University of Oxford UMD University of Oxford University of Siegen AGH University of Science and Technology Duke University UNSW Australia and Data61, CSIRO University of Toulouse Rochester Institute of Technology Carnegie Mellon University CNRS, LAMSADE, Université Paris-Dauphine University of Waterloo University of Toronto The Hebrew University of Jerusalem University of Glasgow **IBM** Research Technion-Israel Institute of Technology Samsung Research America University of Oxford University of Padova The Hebrew University of Jerusalem University of Oxford Indian Institute of Science University of Auckland RPI

Table of Contents

Working Together: Committee Selection and the Supermodular Degree	1
DYCOM: A Dynamic Truthful Budget Balanced Double-sided Combinatorial Market Rica Gonen and Ozi Egri	8
Proxy Voting for Revealing Ground Truth Gal Cohensius and Reshef Meir	18
Committee Scoring Rules, Banzhaf Values, and Approximation Algorithms Edith Elkind, Piotr Faliszewski, Martin Lackner, Dominik Peters and Nimrod Talmon	24
Optimal Decision Making with CP-nets and PCP-nets Sibel Adali, Sujoy Sikdar and Lirong Xia	32
Natural Interviewing Equilibria in Matching Settings Joanna Drummond, Omer Lev, Allan Borodin and Kate Larson	41
Budgeted Online Assignment in Crowdsourcing Markets: Theory and Practice Pan Xu, Aravind Srinivasan, Kanthi Sarpatwar and Kun-Lung Wu	50
If you like it, then you should a put a sticker on it: A Model for Strategic Timing in Voting $Alan\ Tsang\ and\ Kate\ Larson$	59
Approximation and Parameterized Complexity of Minimax Approval Voting Marek Cygan, Lukasz Kowalik, Arkadiusz Socała and Krzysztof Sornat	68
Propositionwise Opinion Diffusion with Constraints Sirin Botan, Umberto Grandi and Laurent Perrussel	76
Thwarting Vote Buying Through Decoy Ballots David Parkes, Paul Tylkin and Lirong Xia	85
Judgment Aggregation in Dynamic Logic of Propositional Assignments Arianna Novaro, Umberto Grandi and Andreas Herzig	92
Approval in the Echo Chamber	101
Practical Algorithms for Computing STV and Other Multi-Round Voting Rules	109

Working Together: Committee Selection and the Supermodular Degree

Rani Izsak Department of Computer Science and Applied Mathematics Weizmann Institute of Science Rehovot, Israel ran.izsak@weizmann.ac.il

ABSTRACT

We introduce a voting rule for committee selection that captures **positive correlation** (synergy) between candidates. We argue that positive correlation can naturally happen in common scenarios that are related to committee selection. For example, in the movies selection problem, where prospective travelers are requested to choose the movies that will be available on their flight, it is reasonable to assume that they will tend to prefer voting for a movie in a series, only if they can watch also the former movies in that series. In elections to the parliament, it can be that two candidates are working extremely well together, so voters will benefit from being represented by both of them together.

In our model, the preferences of the candidates are represented by set functions, and we would like to maximize the total satisfaction of the voters. We show that although computing the best solution is \mathcal{NP} -hard, there exists an approximation algorithm with approximation guarantees that deteriorate gracefully with the amount of synergy between the candidates. This amount of synergy is measured by a natural extension of the supermodular degree [Feige and Izsak, ITCS 2013] that we introduce – the joint supermodular degree.

1. INTRODUCTION

Consider the following scenario (see, e.g., [9, 21]). An airline wishes to increase the satisfaction of the travelers by letting them choose the set of movies that will be available on their flight. It is decided to store on the airplane some fixed number k of movies. The airline surveys the preferences of the prospective passengers of the flight, and aims to make the best decision given their preferences. Two questions arise. First, how should the preferences of the prospective travelers be modeled? Second, given the preferences of the travelers, how should the set of movies be chosen? This problem of choosing some fixed number of candidates to the satisfaction of the voters is a fundamental problem. Generally speaking, in the k-COMMITEE SELEC-TION problem, we have a set V of n voters and a set C of m candidates, and we would like to select k candidates out of the m, such that the voters will be most satisfied. The

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

answers to the two questions above vary in the literature. For example, by the Chamberlin-Courant rule we have a value for each of the candidates, by each of the voters, and the satisfaction of a voter is measured by the highest value she has for any elected candidate. The overall satisfaction is either the sum of the values of the voters or the value of the least satisfied voter (utilitarian [5] or egalitarian [2] variant, respectively). Other possibilities are to aggregate for every voter her value for every elected candidate or to give higher weight for candidates ranked higher by her (e.g. Borda rule). In a recent work, Skowron, Faliszewski and Lang [21] introduce an elegant model that captures the latter examples as well as others. They model the preferences of each voter by an intrinsic value for each of the candidates. Then, they calculate the value of a possible set of k candidates by a voter, by ordering her k intrinsic values for the k candidates, and multiplying them by some weight that corresponds to their rank in the order. This vector of weights is called "OWA operator" (Ordered Weighted Average). Skowron, Faliszewski and Lang [21] study their model for different restrictions on the OWA vector. Among their results, they show a (1 - 1/e)-approximation algorithm for the case of non-increasing weights OWA vectors, by showing it is captured by submodular set functions.¹

However, none of the models above capture positive correlation (i.e. synergy) between specific candidates (see Section 2.2 for further discussion). Positive correlation can happen in various cases: from two candidates to the parliament that are working great together (see Woolley et al. [22] for a research about collective intelligence), to a series of movies that people tend to prefer watching the latter parts only after watching the former parts. In this paper we suggest a voting rule that captures positive correlation between specific candidates. Specifically, our answers to the two questions above are:

- The preferences of each of the candidates are modeled by a non-decreasing monotone set function from subsets of candidates to non-negative real numbers.
- A set of k candidates that maximizes the sum of values of the voters is elected.

We formally present our model in Section 4. In order to measure the amount of synergy between different candi-

¹A submodular set function is a function $f: 2^M \to \mathbb{R}^+$, such that for every $S' \subseteq S \subseteq M$, and every $j \in M$, $f(j \mid S') \ge f(j \mid S)$, where $f(j \mid S) = f(\{j\} \cup S) - f(S)$ is the marginal value of j with respect to S. That is, the marginal values are monotone non-increasing

dates, we extend the supermodular degree [10], by introducing the joint supermodular degree (Section 4.1).

We also study applications for the model. In Section 5, we justify the naturalness of the joint supermodular degree from an applicative view point. In Section 6, we demonstrate how preference elicitation can be practically done.

Finally, in Section 7, we study the computability of our voting rule. On the bright side, we show that although computing the optimum is, generally, \mathcal{NP} -hard, one can approximate the optimum with a guarantee that depends on the amount of synergy between different candidates, as measured by the joint supermodular degree. On the flip side, we show that the same results cannot be achieved for the supermodular degree.

2. PRELIMINARIES

The definitions below are taken from the works [10, 11]. Let C be a set of items (e.g. candidates in election, movies to watch on an airplane) and let $f: 2^C \to \mathbb{R}^+$ be a set function (e.g. of preferences of one of the voters). The following definition is standard.

DEFINITION 1. Let $c \in C$. The marginal set function $f_c : 2^{C \setminus \{c\}} \to \mathbb{R}^+$ is a function mapping each subset $S \subseteq C \setminus \{c\}$ to the marginal value of c given S:

$$f_c(S) \stackrel{\text{def}}{=} f(S \cup \{c\}) - f(S)$$

We denote the marginal value $f_c(S)$ by f(c | S). For $S' = \{c_1, \ldots, c_{|S'|}\} \subseteq C$ and $S \subseteq C \setminus S'$ we also use either of the notations $f(c_1, \ldots, c_{|S'|} | S)$ or f(S' | S) to indicate $f(S \cup S') - f(S)$.

The following definitions were introduced by Feige and Izsak [10].

DEFINITION 2. Let $c \in C$. The supermodular dependency set of c by f is the set of all items $c' \in C$ such that there exists $S \subseteq C \setminus \{c, c'\}$ such that $f(c \mid S \cup \{c'\}) > f(c \mid S)$. We denote the supermodular dependency set of c by $\mathcal{D}_{f}^{+}(c)$. We sometimes omit f, when it is clear from the context.

DEFINITION 3. The supermodular degree of f is defined as $\mathcal{D}_{f}^{+} \stackrel{def}{=} \max_{c \in C} |\mathcal{D}_{f}^{+}(c)|.$

2.1 Representation of set functions

Let $f: 2^C \to \mathbb{R}^+$ be a set function. Then, f associates values to $2^{|C|}$ possible subsets. If we want our algorithms to run in time polynomial in |C|, they, of course, cannot read an input that is exponential in C. Therefore, it is crucial to consider the representation of set functions. One common way to represent set functions is by queries. Another is by an explicit representation. In this section, we mention both.

Queries

The arguably simplest queries are the following.

DEFINITION 4. Value queries for f are defined as follows: Input: A subset $S \subseteq C$. Output: f(S).

That is, if we assume our algorithm has access to value queries for a given set function, we merely assume it can ask for the value of a subset by the function. Another type of queries that we use (see [10]) is the following. DEFINITION 5. Supermodular queries for f are defined as follows:

Input: An item (i.e. a candidate) $c \subseteq C$. Output: $\mathcal{D}_{f}^{+}(c)$.

That is, given a candidate we can ask with whom she has a positive correlation as defined by the supermodular dependencies. In the context of movies, we can ask for a movie that is part of a series, what are the other movies in that series. See Section 5 for further discussion.

An explicit representation

Another way to represent set functions is by an explicit representation. For example, any set function can be represented in a unique way by a hypergraph with weighted edges (see [1, 6, 8]). In this representation, a vertex is introduced for each of the items in the ground set of f. The weights in the sub-hypergraph induced by a set of vertices sum up exactly to the value of the subset with the respective items, by f. To see how weights can be allocated, consider the following iterative process. To hyperedges of size 1, we allocate weights that are the values of the respective singleton subsets. Note that this allocation of weights to hyperedges is unique. Then, for hyperedges of size 2, we allocate weights that are the difference between the value of the respective subset and the sum of the weights of their two singleton subsets. Note that this allocation is unique, as well. Also note that after iteration ℓ , the values by the hypergraph representation are correct for subsets of size up to ℓ . We proceed iteratively till we arrive to the unique edge of size |C|, and then we have a representation of the set function for any size of subset.

A succinct representation.

We say that a representation of a set function is succinct if its size is polynomially bounded by the size of the ground set of the function. Note that in the hypergraph representation, we can list only the edges of value different from 0. So, sometimes this representation can be succinct. In particular, for additive set functions we clearly allocate non-zero values only for the hyperedges of size 1.

2.2 Related work

We list here some of the voting rules from the literature, mostly based on Masthoff [16], and also on the works [7, 9, 15, 16, 17, 21].

- Plurality: When electing a single candidate, plurality means selecting the candidate who is ranked first among the candidates, for the highest number of voters. When "ranked first" can mean that by the voting rule, preferences are ranks of candidates, or alternatively, that there are values for the candidates by the different voters that are used in order to get the candidates' ranks. In order to use this rule for choosing k candidates, one can just repeat it k times, while removing the winner at each iteration.
- Utilitarian: Each voter has a value for each of the candidates, and these values are summed up. The k candidates with the largest sums win.
- Borda [4]: This voting rule assumes the preferences of the candidates are modeled as a list of ranks, and

it converts this list to values, with higher values for higher ranks: $m - 1, m - 2, \ldots, 0$ for ranks $1, \ldots, m$, respectively (*m* is the number of candidates). These values are summed up and highest scores win, similarly to the utilitarian rule above.

- Copeland: The score of a candidate is the number of pairwise elections she wins (by plurality) minus the number of pairwise elections she loses (ties do not count). Values are again, summed up, and higher scores win.
- Maximin: The score of a candidate c with respect to a candidate c' is the number of voters that prefer c over c' (we denote it by score_c(c'). The score of a candidate c is the minimum score of c with respect to a candidate (i.e. argmin_{c'} score_c(c')). For example, if for a candidate c, there exists a candidate that is preferred by all of the voters, then c will get a value of 0. If for a candidate c, all the voters prefer it over all the candidates, then (and only then) she will get the maximal score of n (i.e. the number of voters).
- Approval voting: Each voter either approves or disapproves every candidate. The *k* candidates with largest number of approvals win.

Positional scoring.

Positional scoring is a bunch of voting rules, where the preferences of the voters are just an ordering of the candidates and the rule is defined by a vector of size m of values corresponding to positions by the voters. The total value of a candidate is the sum of these values of the voters. Note that plurality is a positional scoring rule with the vector $(1, 0, \ldots, 0)$ and Borda is a positional scoring rule with the vector $(m-1, m-2, \ldots, 0)$. There is also a rule called "Veto" where the vector is $(1, \ldots, 1, 0)$, so a voter actually chooses one candidate she prefers *not* to include in the selected committee.

Weighted aggregation of preferences of a voter.

Skowron, Faliszewski and Lang [21] introduced the following family of voting rules for choosing k out of m candidates. The preferences of the voters are intrinsic values for the different candidates, and additionally, there is a vector of size k that is called OWA (ordered weighted average). When calculating the value for a set of k candidates by the preferences of a single voter, we do the following. We order the k candidates by their values according to the voter, in an increasing order of values, and then we sum up the values multiplied by the OWA vector (inner product). That is, every value is multiplied by a weight appearing in the OWA vector that corresponds to the rank of the candidate by the voter. To calculate the overall value of a subset of k candidates, we sum up the values of this set of candidates by the voters (utilitarian model). Skowron, Faliszewski and Lang [21] show that when the OWA vector is non-increasing (that is higher ranked candidates by a voter are multiplied by higher (or equal) weights), then the preferences of the voters can be represented by a submodular set function, and therefore a (1 - 1/e)-approximation guarantee can be achieved in polynomial time, by using the classical algorithm of Fisher, Nemhauser and Wolsey [14].

When the OWA vector is not non-increasing, some positive correlation between the candidates can happen, but not between specific candidates. For example, in the min OWA vector $(0, \ldots, 0, 1)$, only the worst candidate in the selected committee counts. This means, roughly speaking, that all the candidates should be adequate by a voter in order to have an adequate score by her. In terms of set functions, it means as follows. The marginal value of a candidate is 0 with respect to any committee that contains a worse (or equal) candidate. The marginal value of a candidate with respect to a committee that contains only better candidates is the difference between the intrinsic values of the new candidate and of the worst candidate in the committee. For example, adding a candidate with an intrinsic value of 1 to a committee, when the worse candidate in it has an intrinsic value of 10 means a marginal value of (-9). On the other hand, if there is also a candidate with an intrinsic value of 2 in the committee, then the marginal value of the new one will be (-1). That is, the marginal value of the new candidate increased because of the inclusion of the candidate with a value of 2. However, it is clear that this does not model synergy between these two candidates. Moreover, positive correlation between *specific* candidates cannot be modeled using OWA vectors, as described above, since they cannot relate to specific candidates differently. This means that in scenarios like the movies example described earlier, a positive correlation within a series of movies cannot be modeled. In this sense, our model adds new possibilities with respect to the model of Skowron, Faliszewski and Lang [21].

Another relevant model was studied by Fishburn and Pekec [13]. Fishburn and Pekec [13] studied an approval voting model, where each of the voters can approve a few candidates, and a committee is approved by a voter if it contains a sufficient number of candidates that are approved by the voter.

3. OUR CONTRIBUTION

This paper introduces a new model for voting rules, based on set functions, together with the required conceptual framework. Our model can be used to model both synergy between candidates (i.e. compliments) and substitutes (e.g., two candidates that each of them is worth 1 and both of them together are worth 1, as well). Since general set functions might be highly complex, we introduce the joint supermodular degree, which we see as a natural extension of the supermodular degree [10]. We demonstrate applications for our model in Section 5. In particular, we suggest practical preference elicitation that is tailored for the joint supermodular degree in Section 6.

Finally, in Section 7, we show how the joint supermodular degree enables one to easily use existing algorithms for function maximization that are tailored for the supermodular degree to achieve approximations for our voting rule. Since there exist such algorithms both for offline and online settings, one can use either and immediately get approximation guarantees for our voting rule in the corresponding setting. Moreover, future algorithms for the supermodular degree can also be easily used by our framework, to get computational results for committee selection. Conceptually speaking, the result of the approximation algorithms can also be seen as the voting rule itself (see Skowron, Faliszewski and Lang [21]). We complement our algorithmic result with a proof of computational hardness.

To the best of our knowledge, our results represent the

first voting rules that capture synergy between specific candidates.

4. THE MODEL

We formally define our model. Let $V = \{v_1, \ldots, v_n\}$ be a set of n voters, let C be a set of m candidates and let k be an integer. Let $f_1, \ldots, f_n : 2^C \to \mathbb{R}^+$ be preference (set) functions, associated with the voters v_1, \ldots, v_n , respectively. We assume that the preferences functions are normalized (i.e., $\forall_i f_i(\emptyset) = 0$) and non-decreasing monotone (i.e., $\forall_{i,S' \subseteq S \subseteq M} f_i(S') \leq f_i(S)$). Our aim is to choose a set $C_{max} \subseteq C$ of size k that maximizes the satisfaction of the voters by their personal preferences:

$$C_{max} = \operatorname*{argmax}_{S \subseteq C ||S|=k} \sum_{i=1}^{n} f_i(S) \; .$$

We refer to this problem as (the) k-COMMITEE SELECTION problem and to the selected subset as the selected committee. Note that this problem can be seen as a voting rule. Alternatively, an approximation algorithm to this problem can be seen as the voting rule (see also Skowron, Faliszewski and Lang [21]).

4.1 The joint supermodular degree

We introduce the following natural extensions of the definitions of Feige and Izsak [10] to a collection of set functions.

DEFINITION 6. Let f_1, \ldots, f_t be set functions for some $t \in \mathbb{N}$ and let $c \in C$. The joint supermodular dependency set of c by f_1, \ldots, f_t is $\bigcup_{i=1}^t \mathcal{D}_{f_i}^+(c)$.

DEFINITION 7. The joint supermodular degree of f_1, \ldots, f_t is the maximum cardinality among the cardinalities of joint dependency sets of items of C by f_1, \ldots, f_t .

The main property of the joint supermodular degree that we use is that the sum function of functions with joint supermodular degree of at most d has supermodular degree of at most d.

We think this definition is natural for voting rules, since it means that positive correlation between the candidates can be modeled, when it is inherent to the candidates themselves, and not to the perspective of the voters about them.

For example, if a candidate is working well together with 2 other candidates, then each of the voters has the possibility to give these 3 candidates or any subset of them a score that is higher than the sum of their individual scores. However, if a candidate does not work well with some other candidate, then none of the voters has the possibility to give them together a score that is higher than the sum of their individual scores. That is, the set of other candidates that the candidate has synergy with depends on her. The decision of whether to take this into account depends on each of the voters. So, the supermodular dependency set of a candidate c, by any of the preference functions of the voters, will contain only other candidates that have synergy (i.e. are working well together) with c.

We discuss applications of our model with respect to the joint supermodular degree in Section 5. In particular, we suggest preference elicitation in Section 6.

5. APPLICATIONS

We discuss in this section applications of our model, together with the joint supermodular degree. Specifically, we demonstrate its merits for two real world examples (see [9]).

- Parliamentary elections: In voting to the parliament, it is possible that candidates complement each other, and work better together. It was actually shown by Woolley et al. [22] that there is a measure for the collective intelligence of a group of people that is different from the intelligence quantities of different people in the group. So, it seems reasonable to allow the voters to give extra value for choosing *together* a pair of candidates that are known to work well together on, e.g., suggesting complex laws in the parliament. Note that the fact that two candidates are working well together is related to the candidates and not to the voters, and indeed, the joint supermodular degree of the voters will reflect the synergies between the candidates.
- Movie selection: Consider the problem of choosing kmovies to be available on an airplane (passengers can watch on their flight movies from the selected set). It seems reasonable that people would prefer to watch latter parts of a series only after the former. Moreover, it might be unreasonable to consider a series of movies as one movie, if, e.g., physical storage is a limitation. Then, it is plausible to give the prospective passengers the possibility to give higher values for movies in the series, given that all the former are selected, as well. Additionally, movie selection can admit submodular behaviour (i.e. substitutes). For example, since the time of the flight is bounded, the number of movies one can watch out of the k selected movies is bounded, as well. This means that, if for example, k = 100 and the time of the flight allows one passenger to watch up to 5 movies, then any movie out of the k that is not among the 5 best for that passenger is redundant for her. So her value will not increase given that we add to the selected set other great movies. On the other hand, we do want to allow k to be large enough to allow different passengers to enjoy different movies. The latter behaviour is submodular. Synergy between selected movies is supermodular. Our model enables one to express such preferences. Furthermore, submodularity does not hurt the approximation guarantees, since it does not increase the joint supermodular degree of the preference functions (see Section 7).

6. PREFERENCE ELICITATION

Consider the movies selection example. When a prospective passenger is asked to express her preferences about possible movies, it seems unreasonable to require her to specify her values for all the exponentially many possibilities. We briefly demonstrate a simple user interface to elicit users' preferences in that case, while enabling them to benefit from the possibility of expressing positive correlations.

The user interface will be as follows. Each of the prospective passengers will be able to give a value for each of the possible movies (these are the values of the singleton subsets). In addition, the prospective passengers will be able to add for each of the movies other values – the marginal values of a movie, with respect to a subset of its joint supermodular dependency set (i.e., other movies in the same series). In order to select such a subset of the movies, a list of the movies in the joint supermodular dependency set will be presented, and a passenger will be able to select the relevant movies (e.g. by checking them by a 'V'). In order to enforce the preference functions of the prospective passengers to be well defined (i.e. a single value for each of the subsets), we will let the prospective passengers check by a 'V' only the movies that were former to a movie in a series.

Note that the supermodular dependency is symmetric (see [10] for a proof). So, in a series of movies, also the former movies are dependent on the latter movies. As an example, one can think of two movies, where each of them is worth 1, but the second one is worth 10 with respect to the first. Then, both movies together are worth 11, and the marginal contribution of each of them with respect to the other is 10, instead of 1 (as it is with respect to the empty set).

Generally speaking, this example interface can be extended in any way that enforces the preference functions to be well defined (e.g. by ordering the items and letting the prospective passengers to check a dependency by 'V' only if it is before the current item in that ordering).

To see the power of combining supermodular dependencies with submodular behaviour, note that we can also ask each passenger how many movies she would like to watch in her flight (with a maximum that depends on the duration of the flight), and then calculate as her preference, the best subset of that number of movies, from any input subset of movies.

Note that it is easy to emulate both value and supermodular queries using such a representation, and then to use the algorithms of Feldman and Izsak [11], as described in Section 7.

7. COMPUTATIONAL RESULTS

The following theorem shows that there exists an approximation algorithm with approximation guarantee that is linear in the amount of synergy between the candidates, as measured by the joint supermodular degree of the preference functions of the voters. For submodular set functions, the result described by the theorem coincides with the optimal result for submodular set functions of Fisher, Nemhauser and Wolsey [14] that is used by Skowron, Faliszewski and Lang [21].

THEOREM 1. When the joint supermodular degree of the preferences functions of the voters is d, the k-committee selection problem admits an approximation algorithm with guarantee $(1 - e^{-1/(d+1)}) \ge 1/(d+2)$. The algorithm gets access to the preference functions by value queries and supermodular queries, and its running time is $Poly(n, m, 2^d)$.

Note that the above result captures the example of movies selection from the introduction (see Section 5 for further discussion). Note also that the proof of the above result applies to the case of committee selection subject to a general matroid constraint (cardinality constraint is a special case of a matroid constraint), but with an approximation guarantee of 1/(d+2), by using the respective algorithm of Feldman and Izsak [11].

Moreover, one can use the algorithms of Feldman and Izsak [12] in order to get an online (secretary like) version of Theorem 1, when the candidates arrive one by one in an online fashion, and we need to decide on the spot, irrevocably, whether to elect a candidate or not, based on the preferences of the voters (for exact details of the model, see [12]). As an example, consider hiring a team to a project, where each of the candidates meets with a few interviewers. Then, an optimal team of candidates should be hired, according to the preferences of the interviewers.

By using the algorithm of Feldman and Izsak [12] for a cardinality constraint, one gets an approximation guarantee polynomial in the joint supermodular degree. Any approximation guarantee that depends only on the joint supermodular degree gives a constant approximation guarantee, if the candidates admit synergy only with a constant number of other candidates (e.g. if there is a positive correlation only within series of movies, and all the series suggested are of length up to 3). See also Oren and Lucier [18] for a different secretary like model.

Additionally, we show a hardness result for the case of non-bounded joint supermodular degree, even when the supermodular degree of all the set functions is bounded by 1. For this, we use a reduction from the k-dense subgraph problem (see e.g. Bhaskara el al. [3]).

DEFINITION 8. The k-dense subgraph problem is the following. We are given as input a graph G = (V, E) and an integer $k \in \mathbb{N}$, and our aim is to select k vertices such that the number of edges in their induced subgraph is maximized.

This problem is NP-hard and it is highly believed it is hard to approximate it within any constant guarantee. Actually, no efficient algorithm is currently known that approximates it within a guarantee better than n^c , for some constant c(see e.g. [3, 19, 20]).

THEOREM 2. The k-committee selection problem is at least as hard as the k-dense subgraph problem, even if the supermodular degree of the set functions is 1, and even if an explicit representation of the preference functions is given. This means, in particular, that it is NP-hard² and SSEhard (see [19] and also [20]).

PROOF OF THEOREM 1. Let V be the set of n voters, let C be the set of m candidates, let k be the requested number of elected candidates and let $f_1, \ldots, f_n : 2^C \to \mathbb{R}^+$ be the preference functions of the voters. We prove that since the joint supermodular degree of f_1, \ldots, f_n is upper bounded by d, then the supermodular degree of their summation function $f_{\Sigma}(S) \stackrel{\text{def}}{=} \sum_{i=1}^n f_i(S)$ is upper bounded by d, as well. Note that this would not be necessarily true if only the supermodular degree of f_1, \ldots, f_n was bounded by d (or even by 1). Actually, Theorem 2 serves as a counter example to the latter for d = 1.

To prove the bound on the supermodular degree of the summation function f_{Σ} , we show that every supermodular dependency by f_{Σ} induces the same supermodular dependency by one of the f_i s in the sum. Let $c, c' \in C$ and $S \subseteq C$ be such that $f_{\Sigma}(c \mid S \cup \{c'\}) > f_{\Sigma}(c \mid S)$. Then, by the definition of f_{Σ} , $\sum_{i=1}^{n} f_i(c \mid S \cup \{c'\}) > \sum_{i=1}^{n} f_i(c \mid S)$. So, $\exists_{1 \leq i \leq n}$ s.t. $f_i(c \mid S \cup \{c'\}) > f_i(c \mid S)$, as claimed.

Now, we can just use the algorithm of [11] for monotone function maximization subject to uniform matroid constraint (i.e. cardinality constraint) on the function f_{Σ} with a constraint k. Note that the latter algorithm gives an optimal approximation guarantee for submodular set functions, and generally its guarantee deteriorates linearly with the supermodular degree. Moreover, its running time is as required by the Theorem. This concludes the proof of Theorem 1. \Box

 ^{2}NP -hardness is actually true also for submodular set functions, i.e. supermodular degree of 0.

PROOF OF THEOREM 2. The proof is somewhat similar to the proof of $SS\mathcal{E}$ -hardness for maximizing set function subject to cardinality constraint, given by [11]. Given an algorithm for solving the k-committee selection problem within approximation guarantee α , we show how to solve any input instance of the k-dense subgraph problem within approximation guarantee α . Let G = (S, E) be an instance of the k-dense graph problem. Then, our set of candidates C will be S (the set of vertices of G). We also introduce a voter v_e for every edge $e = \{v_{e1}, v_{e2}\} \in E$ and let $V = \bigcup_{e \in E} \{v_e\}$. For every voter v_e , her preference set function is:

$$f_e = \begin{cases} 1 & \text{if } v_{e_1} \text{ and } v_{e_2} \text{ are both elected} \\ 0 & \text{otherwise} \end{cases}$$

That is, in this instance of the k-committee selection problem, our aim is to find a subset of k candidates (where the set of candidates corresponds exactly to the set S of vertices of G), such that the number of pairs of candidates, that correspond to the preference functions of the voters, is maximized (where these pairs of candidates are exactly the edges E of G). This is exactly the k-dense subgraph problem. That is, given a solution to this instance of k-committee selection problem, we just output the subset of vertices of S that corresponds to the candidates in C that were selected, as a solution to the input instance of the k-dense subgraph problem. This gives us a feasible solution with the same value, and thus with the same approximation guarantee α . This concludes the proof of Theorem 2. \Box

8. CONCLUSIONS

We suggest a new voting rule for committee selection that enables the voters to express positive correlation between the candidates. We also introduce the joint supermodular degree that enables us to use existing computational results for the supermodular degree, and get efficient approximation algorithms for our voting rule. We see our work as a proof of concept, and hope that it will lead to further study of committee selection with positive correlation between the candidates.

Acknowledgments

Work supported in part by the Israel Science Foundation (grant No. 1388/16). I would like to thank Uri Feige for many useful discussions and for his contributions to this paper. I would like to thank Nimrod Talmon for useful discussions and for directing me to the paper of Skowron, Faliszewski and Lang [21]. I would also like to thank Moshe Babaioff, Shahar Dobzinski and Moshe Tennenholtz for useful discussions.

REFERENCES

- I. Abraham, M. Babaioff, S. Dughmi, and T. Roughgarden. Combinatorial auctions with restricted complements. In *EC*, pages 3–16, New York, NY, USA, 2012. ACM.
- [2] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. *Artificial Intelligence Research*, 47:475–519, 2013.
- [3] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an

 $O(n^{1/4})$ approximation for densest k-subgraph. In Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010, pages 201–210, 2010.

- [4] J. Borda. Memmoire sur les elections au scrutine. Histoire de l'Academie Royale des Sciences, 1781.
- [5] B. Chamberlin and P. Courant. Representative deliberations and representative decisions: Proportional representation and the borda rule. *American Political Science Review*, 77:718–733, 1983.
- [6] Y. Chevaleyre, U. Endriss, S. Estivie, and N. Maudet. Multiagent resource allocation in k-additive domains: preference representation and complexity. Annals of Operations Research, 163:49–62, 2008.
- [7] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? J. ACM, 54(3), 2007.
- [8] V. Conitzer, T. Sandholm, and P. Santi. Combinatorial auctions with k-wise dependent valuations. In AAAI, pages 248–254, 2005.
- [9] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of multiwinner voting rules. In AAMAS, pages 53–60, 2014.
- [10] U. Feige and R. Izsak. Welfare maximization and the supermodular degree. In *ITCS*, pages 247–256, 2013.
- [11] M. Feldman and R. Izsak. Constrained monotone function maximization and the supermodular degree. In APPROX-RANDOM, pages 160–175, 2014.
- [12] M. Feldman and R. Izsak. Building a good team: Secretary problems and the supermodular degree. In SODA, 2017.
- [13] P. Fishburn and A. Pekec. Approval voting for committees: Threshold approaches. 2017. working paper.
- [14] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions – II. In *Polyhedral Combinatorics*, volume 8 of *Mathematical Programming Study*, pages 73–87. North-Holland Publishing Company, 1978.
- [15] J. Goldsmith, J. Lang, N. Mattei, and P. Perny. Voting with rank dependent scoring rules. In AAAI, pages 698–704, 2014.
- [16] J. Masthoff. Group modeling: Selecting a sequence of television items to suit a group of viewers. User Model. User-Adapt. Interact., 14(1):37–85, 2004.
- [17] J. Masthoff. Group recommender systems: Combining individual models. In *Recommender Systems Handbook*, pages 677–702. 2011.
- [18] J. Oren and B. Lucier. Online (budgeted) social choice. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada., pages 1456–1462, 2014.
- [19] P. Raghavendra and D. Steurer. Graph expansion and the unique games conjecture. In STOC, pages 755–764, 2010.
- [20] P. Raghavendra, D. Steurer, and M. Tulsiani. Reductions between expansion problems. In *IEEE Conference on Computational Complexity*, pages 64–73, 2012.
- [21] P. Skowron, P. Faliszewski, and J. Lang. Finding a collective set of items: From proportional

multi representation to group recommendation. In $AAAI,\, {\rm pages}~2131{-}2137,\,2015.$

[22] A. W. Woolley, C. F. Chabris, A. Pentland, N. Hashmi, and T. W. Malone. Evidence for a collective intelligence factor in the performance of human groups. *Science*, 330:265–294, 2010.

DYCOM: A Dynamic Truthful Budget Balanced Two-sided Combinatorial Market

Rica Gonen Dept. of Management and Economics The Open University of Israel 1 University Road, Raanana 43107 ricagonen@gmail.com

ABSTRACT

Recently, there has been increased attention on finding solutions for two-sided markets with strategic buying and selling agents. However, the known literature largely focuses on solutions in settings where there exists a single commodity for sale and agents ask/offer one unit of the commodity.

In this paper we present and evaluate a general solution that matches agents in a dynamic, two-sided combinatorial market. Multiple commodities, each with multiple units, are bought and sold in different bundles by agents that arrive over time.

Our solution, DYCOM, provides the first dynamic twosided combinatorial market that allows truthful and individuallyrational behavior for both buying and selling agents, keeps the market budget balanced and approximates social welfare efficiency. We experimentally examine the allocative efficiency of DYCOM under variety of distributions of bids and market demand. The experimental results are given with respect to our proven theoretical bounds and with respect to other known (dynamic and non-dynamic) two-sided markets with a single commodity as well as a non-dynamic combinatorial market. DYCOM performs well by all benchmarks and in many cases improves on previous mechanisms.

CCS Concepts

 $\bullet Information \ systems \rightarrow Web \ applications; \ \bullet Applied \ computing \ \rightarrow \ Electronic \ commerce;$

Keywords

Strategic agents, Electronic commerce, combinatorial exchanges

1. INTRODUCTION

One-sided auctions have long been studied in economics and computer science. In particular, such auctions see use in the multi-agent planning domain for purposes such as task allocation [12], robot exploration [24], and resource allocation [9]. One-sided auctions aim to find high-social welfare (SWF) (an efficient) allocation of a commodity to a set of agents, while ensuring that a truthful reporting of the Ozi Egri Dept. of Mathematics and Computer Science The Open University of Israel 1 University Road, Raanana 43107 ozieg@hotmail.com

agents' input is their best strategy. An important extension of one-sided auctions are one-sided combinatorial auctions where multiple commodities are offered for sale. Agents bid on bundles of commodities which allows agents to express complex preferences over subsets of commodities (see [8] for many examples within). An elegant and well-studied class of combinatorial one-sided auctions are the sequential posted price auctions in which the agents are presented sequentially with a vector of prices and must choose their preferred bundle given the price vector (among the first studied are [1, 20]). One-sided combinatorial auctions have been applied to various problems, including airport time slot allocation [19], distributed query optimization [23] and transportation service procurement [22].

Recent years have brought increased attention to the problems that arise in two-sided markets, in which the set of agents is composed of buying and selling agents. As opposed to one-sided auctions where the auctioneer initially holds the commodity or the commodities and is not considered strategic, in the two-sided market the commodities are initially held by the set of selling agents, who have costs for the commodities they hold and are expected to behave strategically. The market maker's role is to match buying agents with selling agents as well as to determine what price each matched buying agent pays the market and what price the market pays each selling agent.

The cornerstone method in auction theory for high-SWF (efficient) allocation and incentivizing agents' truth-telling strategy is the Vickrey-Clarke-Groves (VCG) mechanism [25, 6, 13]. In addition to motivating agents to report their true input VCG is also individually rational (IR) in many settings. IR requires that no agent can lose by participating in the mechanism. In two-sided markets, a further important requirement is budget-balance (BB) meaning that the market does not end up with a loss. VCG is not BB except in special cases [14]. It is well known from [18] that maximizing SWF while maintaining IR and truthfulness perforce runs a deficit (is not BB) even in the bilateral trade setting, i.e., when there are just two agents trading with each other. Well-known circumventions of [18]'s impossibility in the setting of double sided auctions with a single commodity (and unit demand and supply) are [15, 16], which relax efficiency in return for maintaining the other properties of truthfulness, IR and BB. Other circumventions of [18]'s impossibility include relaxing determinism in addition to efficiency, i.e., are randomized solutions some in the simple setting of a single-commodity single-unit market [21] and some in the extended setting of combinatorial market [3]. [7, 11, 27] cir-

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

cumvents [18] in the setting of single-commodity single-unit, multi-commodity single-unit and single-commodity multiunit respectively.

The growing interest in two-sided markets is motivated by the numerous examples of applications such as stock exchanges, online advertising exchanges, pollution rights and the recent US FCC effort to reallocate electromagnetic spectrum from UHF television broadcasting to use for wireless broadband services. Many of these examples represent dynamic and uncertain environments, and thus require dynamic markets where agents arrive over time. Moreover, the examples emphasize the need for solutions that involve multiple commodities and agents that can buy and sell the multiplicity of those commodities, i.e., two-sided combinatorial markets as opposed to unit demand/supply. On the one hand due to the complex design requirements of such twosided combinatorial markets, practical solutions for those dynamic environments such as the recent US incentive auctions circumvent the dynamic aspect of the problem by employing an iterative process [17]. And on the other hand, to our knowledge, the theoretical solutions of dynamic twosided markets in the literature focuses on a single commodity for sale and agents ask/offer one unit of the commodity [4, 2].

Wurman et al. [26] presented a dynamic two-sided solution incentivizing truthful reporting from either the buyers or the sellers but not simultaneously from both. A different dynamic solution given by Blum et al. [2] maximizes the SWF of buyers and non-selling sellers in the single commodity unit demand setting. Finally, Bredin et al. [4] present a truthful dynamic double-sided auction that is constructed from a truthful offline double-sided auction rule also in the single commodity unit demand setting.

In this paper we present and evaluate a general solution that dynamically matches agents in a two-sided combinatorial market. Multiple commodities, each with multiple units, are bought and sold in different bundles by agents that arrive over time. Our solution, DYCOM, provides the first dynamic two-sided combinatorial market that allows truthful and IR behavior for both buying and selling agents, keeps the market BB and approximates SWF efficiency.

The main idea behind our DYCOM solution is the transformation of the two-sided combinatorial market into a onesided combinatorial auction. The transformation of the market into an auction makes use of a novel principle: each selling agent is a buying agent of his own commodities. Thus all our dynamic market's selling agents become virtual buying agents who buy in a dynamic one-sided combinatorial auction along with our market's actual buying agents. DY-COM is a primal-dual sequential posted-price mechanism that builds upon a combinatorial auction studied in the literature [5]. However, DYCOM incorporates solutions to the design challenges imposed by the simulation process such as higher initial price constraints and payment computations for virtual buying agents. Much like other sequential postedprice mechanisms DYCOM does not require any assumption on agents' arrival order.

To validate the performance of our suggested solution, we experimentally tested the SWF efficiency of DYCOM under variety of agents' bid distributions and agents' demand against a number of benchmarks. Some of the benchmarks were dynamic and some were non-dynamic. The most notable of DYCOM's results were when compared with:

- An optimal non-dynamic and non-truthful allocation algorithm (simplex), where DYCOM's approximation approaches 0.5 of the market SWF.
- McAfee [16]'s non-dynamic single commodity unit demand market. Here DYCOM's approximation approaches 1 though DYCOM is tailored for a completely general combinatorial setting and it is dynamic unlike [16] and as such it was not expected to perform as well as [16].
- [3]'s randomized non-dynamic combinatorial market. In this comparison DYCOM's approximation approaches 10 times that of [3]'s SWF in large markets even though DYCOM is deterministic and dynamic unlike [3] and as such it was not expected to perform better than [3].

The paper's contributions are threefold. First, we provide the first dynamic two-sided combinatorial market that is truthful, IR and BB for all agents that approximates SWF efficiency. Second, our experimental tests show that our dynamic two-sided combinatorial market is a general and practical platform as it performs as well as the known McAfee [16]'s non-dynamic single-commodity unit-demand two-sided market and performs better than the randomized non-dynamic combinatorial market with limited valuations and cost domains [3]. Third, our two-sided combinatorial market transformation into a one-sided combinatorial auction is of independent interest for future work on simplifying other forms of multi-sided exchanges to the well studied form of one-sided auctions.

2. PRELIMINARIES

Consider a dynamic market model in which agents arrive over time. Agents are either buyers or sellers interested in trading multiple units of multiple commodities in bundles. Commodities are sold by selling agents and allocated to buying agents irrevocably.

Let m be the total number of non-identical commodities offered by all selling agents accumulatively. Each commodity $j \in \{1, \ldots, m\}$ has a_i identical units (or copies). Though in our model selling agents arrive dynamically we assume that a_j is a priori known to the market. The assumption that the number of a commodity's units is a priori known to the market was made by almost all previous literature on dynamic markets see $([2, 7, 26])^1$. There are practical examples where the quantity of commodities expected in the market is a priori known to the market maker. For instance consider a securities Exchange with no short sells. The number of shares of each stock issued by its company is pre-known to the exchange yet buyers and sellers arrive dynamically. Another example where the quantity of commodities expected in the market is pre-known, though without dynamic arrivals of buyers and sellers, are the newly run FCC incentive auctions where the broadcast frequencies are pre-known to the government.

A bundle of commodities, s, is defined as vector $(d_{s,1}...d_{s,m})$, where $0 \le d_{s,j} \le a_j$ is the number of units of commodity j in the bundle. We say that $s \le s'$ if the vector of s is at most the vector of s' coordinate-wise. There are l agents who are interested in selling commodities. Each selling agent thas a bundle of commodities $S_t = (d_{S_t,1}...d_{S_t,m})$ he initially

¹except for the work by [4] which assumed an alternative assumption of agents bounded patience.

owns and a cost function c_t that assigns a non-negative cost for each bundle $s_t \leq S_t$ of commodities, $c_t : \{0...S_{t1}\} \times ... \times \{0...S_{tm}\} \rightarrow R^+$ and any other bundle is assigned zero. We denote by c a vector of declared costs c_1, \ldots, c_l and by c_{-t} a vector of declared costs $c_1, \ldots, c_{t-1}, c_{t+1}, \ldots, c_l$. There are n agents who are interested in buying commodities. Each buying agent *i* has a valuation function v_i that assigns a non-negative value for each bundle of commodities, $v_i: \{0...a_1\} \times ... \times \{0...a_m\} \to R^+$. We denote by v a vector of declared valuations $v_1, ..., v_n$ and by v_{-i} a vector of declared valuations $v_1, ..., v_{i-1}, v_{i+1}, ..., v_n$. For simplicity of notations we denote $v_i(s)$ by $v_{i,s}$ and $c_t(s)$ by $c_{t,s}$. We assume the standard assumption in combinatorial auctions literature that commodities can be perishable and the valuation function is monotonic non-decreasing. That is, for each buying agent i and $s \leq s'$, $v_{i,s} \leq v_{i,s'}$ and for each selling agent t and $s \leq s', c_{t,s} \leq c_{t,s'}$. Also, for any $i, v_i(\emptyset) = 0$ and for any $t, c_t(\emptyset) = 0$ (normalization).

Bundle s is denoted as *feasible bundle* for buying agent i (selling agent t) if there is no bundle $s' \leq s$, $v_{i,s'} = v_{i,s}$, $(c_{t,s'} = c_{i,s})$. Intuitively, bundles that are not feasible contain commodities that are perishable. Let S(i) (S(t)) be the set of feasible bundles for buying agent i (selling agent t). We assume that there are known bounds

$$1 < \Theta \le \min_{\{i,j, s_i \in S(i), s_t \in S(t)\}} \left\{ \frac{a_j}{d_{s_t,j}}, \frac{a_j}{d_{s_i,j}} \right\} \text{ and}$$

 $\theta \geq \max_{\{i,j, s_i \in S(i), s_t \in S(t)\}} \left\{ \frac{a_j}{d_{s_t,j}}, \frac{a_j}{d_{s_i,j}} \right\}$. That is, for each bundle $s_i \in S(i)$ and $s_t \in S(t)$ and commodity j, the number of commodities j in the bundle, $d_{s_i,j}$ ($d_{s_t,j}$), is at least $1/\theta$ and at most $1/\Theta$ fraction of the total number of commodities j, a_j . The θ, Θ demand bounds are parameters in our SWF approximation ratio, as will be shown in section 3. The SWF approximation ratio improves as the agents' demand decreases relative to the supply of commodities in the market. Intuitively the effect of the above parameters can be understood as improving the algorithm's performance when each participant represents a bounded share of the demand in the market. Accordingly, the algorithm performs better for large markets than thin markets as will be seen in section 4. This characteristic makes the algorithm practical and desirable for use in large markets.

Our agents are assumed to have a demand (supply) oracle representation of their valuations (costs) (a common assumption in the combinatorial auction literature e.g. [10] for valuation oracle).

DEFINITION 2.1. (demand (supply) oracle) For every buying agent i (selling agent t), a demand oracle for valuation (cost) v (c) accepts a vector of commodity prices $(p_i^{(1)}...p_i^{(m)})$ as input and outputs the demand for (supply of) the commodities at these prices, i.e. it outputs the vector $(d_{s,1},...,d_{s,m})$, $s \in S(i)$ ($s \in S(t)$) that maximizes i's utility $\max_{s,s \in S(i)} v_{i,s} \sum_{j=1}^m d_{s,j} p_i^{(j)}$ (t's utility $\max_{s,s \in S(t)} \sum_{j=1}^m d_{s,j} p_t^{(j)} - c_{t,s}$).

In a concrete market implementation the valuations (costs) will be given in some "bidding language" and our market will operate in polynomial time as long as the bidding language allows polynomial-time computation of answers to demand (supply) oracle queries. Note that these types of oracle queries can be easily answered for the case where each agent puts forward an arbitrary list of mutually exclusive bids for packages.

Let $A = \sum_{j=1}^{m} a_j$ be the total number of commodities. Let $s_{\max} = \max_{i,t,(s \in S(i) \land s \in S(t))} \{\sum_{j=1}^{m} d_{s,j}\}$ be the largest bundle requested (offered) in the market, note that, $s_{\max} \leq A$ and that we do not assume that s_{\max} is pre-known.

An allocation for a two-sided market can be represented as a pair of vectors $(X, Y) = ((X_1, ..., X_n), (Y_1, ..., Y_l))$ such that the sum of the union of $X_1, ..., X_n, Y_1, ... Y_l$ is A, and $X_1, ..., X_n, Y_1, ..., Y_l$ are mutually non-intersecting. The goal of the market maker is to dynamically match the agents such that each buying agent i interested in buying a bundle is allocated with available commodities of selling agents t, so as to maximize $\sum_{i=1}^n v_i(X_i) + \sum_{i=1}^l c_t (\cup s \in S(t) \setminus Y_t)$. This goal is referred to as SWF or efficiency (of trading buyers and remaining commodities).

We transform the two-sided combinatorial market into a one-sided combinatorial auction where selling agents are reduced to virtual buying agents of their own offered commodities. The one-sided combinatorial auction used to host the two-sided combinatorial market is inspired by [5]'s primal dual combinatorial auction. The goal of the auctioneer in the one-sided combinatorial auction is to partition the available commodities by allocating each buying agent *i* a bundle s_i , so as to maximize $\sum_{i=1}^{n} v_i(s_i)$. This goal is referred to as maximizing SWF (or efficiency).

We say that a mechanism is *truthful* if reporting the true value and cost is a dominant strategy for each agent regardless of the other agents' reports.

We say that a mechanism is *individually rational (IR)* if no agent can receive a negative utility by participating.

We say that a market is *budget balanced* (BB) if the sum of the prices paid by the buying agents is at least as high as the sum of the prices paid to the selling agents.

2.1 The one-sided combinatorial auction formulation as a linear programming problem

Our proposed one-sided combinatorial mechanism is based on solving a linear relaxation of the problem in a dynamic fashion. Let us first introduce an integer formulation for the one-sided combinatorial mechanism problem.

Let $y_{i,s} \in \{0,1\}$ be a variable indicating that bundle s is allocated to buying agent i. Constraint (1) suggests that each buying agent is allocated at most one bundle. Constraint (2) suggests that the number of units sold from commodity j is at most a_j .

We relax the integrality constraints $y_{i,s} \in \{0, 1\}$ in order to achieve the below linear program formulation that upper bounds the maximum SWF.

$$[Dual] \qquad \max \sum_{i=1}^{n} \sum_{s \in S(i)} v_{i,s} y_{i,s}$$

$$\sum_{s \in S(i)} y_{i,s} \le 1 \forall 1 \le i \le n \tag{1}$$

$$\frac{1}{a_j} \sum_{i=1}^n \sum_{s \in S(i)} d_{s,j} y_{i,s} \le 1 \forall 1 \le j \le m \qquad (2)$$
$$y_{i,s} > 0 \forall 1 < i < n, s \in S(i)$$

Note that the number of variables may be exponential. However, our algorithm never solves this linear formulation explicitly. We refer to this formulation as the dual program [Dual]. To obtain the corresponding primal program [Primal], we define variable z_i for each buying agent *i*, and variable x_j for each commodity *j*. The primal linear formulation is as follows. [Primal]

$$\min \sum_{i=1}^{n} z_i + \sum_{j=1}^{m} x_j$$
$$z_i + \sum_{j=1}^{m} \frac{d_{s,j}}{a_j} x_j \ge v_{i,s} \forall 1 \le i \le n, s \in S(i) \quad (3)$$
$$z_i, x_j \ge 0 \forall 1 \le i \le n, 1 \le j \le m$$

Note that the dual problem described above is not the integer formulation of the traditional combinatorial auction problem but rather a linear programming formulation that upper bounds the maximum social welfare and is never solved explicitly by the solution we present in the paper. Our solution solves the primal problem presented and therefore it is presented as primal.

3. DYCOM AND THE SIMULATION OF TWO-SIDED COMBINATORIAL MARKET AS A ONE-SIDED

In this section we first discuss how to transform a twosided combinatorial market into a one-sided combinatorial auction such that one can conclude the allocation and prices of the buying agents as well as the allocation of the selling agents and the payments they receive. We then present DY-COM and prove its economic properties and approximation.

Consider a dynamic market in which agents arrive over time and prices increase with demand (we make the common assumption in online mechanism design literature that the order of arrival is arbitrary and agents have no control over it. This assumption can also be found in [1], [5] and many citations within.). Agents are either buyers or sellers which arrive once and are faced with a vector of prices. Agents can demand/supply a bundle of their choice in the given prices immediately or leave permanently. Selling agents that supply a bundle stay at the market until their supply is sold (or return to them in market closing time). For every arriving agent t which is interested in selling bundles of commodities S(t) and initially owns commodities S_t , we construct a virtual agent i that is interested in buying some of selling agent t's commodities. In order to simulate a virtual buying agent i that represents selling agent t's interests we need to allow virtual agent *i* to buy the commodities that are not beneficial for selling agent t to sell. For example if selling agent t has one unit of commodity 1 and one unit of commodity 2, his cost function is $c_{t,\{1\}} = 10$, $c_{t,\{2\}} = 5$, $c_{t,\{1,2\}} = 14$ and he is presented with prices $p_t^{(1)} = 8$, $p_t^{(2)} = 7$ then he is not interested in selling commodity 1. Therefore the virtual buying agent that represents him will buy commodity 1. Since all that we have access to is the agent's demand(supply) oracle, in order to simulate selling agent t as a buying agent of t's commodities we query each selling agent t's supply oracle as he arrives and allocate the created virtual buying agent with the commodities $S_t \setminus s_t$ where s_t is the bundle answered by t's supply oracle. Selling agent t's commodities that were not bought by its virtual buying agent are offered to the "regular" (non virtual) buying agents that arrive in the time periods that follow.

We assume a priori knowledge of the values v_{\max} and c_{\min} such that $v_{\max} \ge \max_{i,s} \{v_{i,s}\}, c_{\min} \le \min_{t,s} \{c_{t,s}\}$ and $v_{\max} > c_{\min}$. It is easy to verify that v_{\max} and c_{\min} knowledge is necessary in order to obtain non-trivial approximation ratio. First we consider the a priori knowledge of v_{\max} . If v_{\max} is unknown to the algorithm, then any deterministic algorithm has an unbounded efficiency approximation

ratio even if there is only a single commodity (with multiple units). To see this, consider selling agents with cost zero for all commodities and consider the following simple adversarial sequence. In each iteration the next buying agent would like a single unit of the (single) commodity and his bid is the smallest value of the remaining buying agents that still need to arrive. If there is no such value then certainly the algorithm has no bounded efficiency. Otherwise, the algorithm always allocates all units, and after allocating all units then, the next buying agent has value that is very large compared to all previous bids. Similar argument can be made for the necessity of $c_{\rm min}$.

As the assumption of a priori knowledge of v_{max} and c_{min} is necessary in order to obtain a non-trivial approximation all previous literature on dynamic markets even ones with single commodity assume similar a priori knowledge of the max, min values (see [2, 7, 26]). The only previous work on dynamic markets that does not assume a similar assumption to the max, min values, is the work by [4]. However [4]'s work assumes an alternative assumption: agents bounded patience, that without it no reasonable efficiency can be achieved.

Let $s_{\max}^i = \max_{s|y_{i,s}=1} \{\sum_{j=1}^m d_{s,j}\}$ be the maximal size of any bundle allocated by DYCOM until agent *i*'s arrival (including *i*) and let $\psi = \frac{\ln(1+s_{\max}^i(w_{\max} - c_{\min}))}{1-1/\Theta}$.

DYCOM is composed of initialization stage (the first two for loops) and a running loop that handles dynamically arriving agents. The loop for the arriving agents has 5 steps.

- Step (1) update the prices of all commodities for the new agent that arrived.
- Step (2) query the arriving agent for his demand or supply (depending on the type of agent) of commodities given the current prices.
- Step (3) handle selling agents by converting each arriving selling agent in to a virtual buying agent. The virtual buying agent is configured to buy the commodities that the selling agent is better off keeping and not selling given the current market prices, i.e., his total commodities bundle S_i minus the bundle of commodities that are most beneficial for him to sell according to his supply oracle s_i . Payment to the arriving agent is made every time his commodities are bought by future arriving buying agents. The payments are computed according to the prices that were presented to the selling agent at his arrival².
- Step (4) handle buying agents by allocating each arriving buying agent his requested bundle at current prices and charging him according to those prices. In this step DYCOM pays selling agents for commodities that were bought by the currently arriving buying agent.

²Note that the IR property is not affected by the later payments since if units of commodities in s_i are not sold in the market by its closing time then those units can be returned to seller *i*. Also note that similarly to [4] we could have changed our algorithm to pay the arriving selling agents instantaneously and have the market "hold" the commodities until bought, however such approach will lead to market deficits during the market run as occurs in [4]'s market.

Step (5) updates the parameters given the new allocation and in particular updates the primal parameter xⁱ_j for the next arriving agent's prices. The update formula, xⁱ⁺¹_j, is motivated by the idea that no commodity j is allocated more than a_j times. This can be achieved by increasing the price of every commodity j such that after the allocation of at least (1 - 1/Θ)a_j units of the commodity the price reaches the level of v_{max}. At this high price no agent can afford to buy the commodity. Meaning, no more units of the commodity to an agent is at most 1/Θ of the commodity, no more than a_j allocations of the commodity can occur in total.

DYCOM

For each commodity
$$j$$
 set $x_j^1 = c_{\min}a_j$, $s_{\max}^1 = 0$
For each buying and selling agent $1 \le i \le n + l$
set $z_i = 0$, $y_{i,s} = 0$
For each arriving agent $i = 1...n + l$
(1)for each commodity j set the price $p_i^{(j)} = x_j^i/a_j$
(2)input current prices $p_i^{(1)} ... p_i^{(m)}$ to agent i's
demand/supply oracle and
output demand/supply bundle $d_{s,1}...d_{s,m}$, $s_i \in S(i)$
(3)if i is a selling agent then construct a
virtual buying agent i by:
allocating him the bundle $s = S_i \setminus s_i$
paying him the future payment determine at (4)
query i on $c_{i,s}$, set $v_{i,s} = c_{i,s}$
(4)if i is a buying agent
allocate i with bundle $s = s_i$
charge $i p_i = \sum_{j=1}^m d_{s,j} p_i^{(j)}$
query i on $v_{i,s}$
for $k = 1...i - 1$
for every unit of commodity j
of virtual buying agent $(i - k)$
that is allocated to agent i in $d_{s,j}$,
pay agent $(i - k)$ the price $p_{i-k}^{(j)}$
(5)Update:
 $y_{i,s} = 1, z_i = v_{i,s}$, recompute s_{\max}^i
for all j : $x_j^{i+1} \leftarrow x_j^i \exp\left(\frac{d_{s,j}y_{i,s}}{a_j}\psi\right) - 1$)

3.1 Analysis

In this section we analyze the performance of the DYCOM solution as a truthful, IR, BB and SWF maximizing market. Our analysis first shows that the market is truthful and IR both for buying and selling agents and does not run a deficit. Then we focus on the analysis of the SWF approximation ratio.

LEMMA 3.1. DYCOM is truthful and IR for buying and selling agents and is a BB market.

PROOF. We start by claiming that DYCOM is truthful. Since agents have no control over their arrival order they can not affect the commodities prices they are faced with. Nevertheless agents can potentially misreport their demand/supply bundle or can misreport its value/cost. We first claim the buying agents are weakly better off reporting their true demand bundle and their true value for it. Assume for the contrary that a buying agent requested a bundle s' that is not the bundle s that was recommended to him by his demand oracle. As the demand oracle outputs the bundle that maximizes the agent's utility given the price vector, when allocated s', the agent can not gain a higher utility than s. Thus the buying agent is (weakly) better off reporting his true demand bundle. Any declaration of s's value can not change the allocation (and therefore can not change the buying agent's utility) as the allocation is determined by the bundle demand. Moreover buying agent's lie will be immediately exposed if he reports the value of s such that it is less than the total price of the bundle s as his demand oracle is utility maximizing.

We continue by claiming that the selling agents are weakly better off by reporting their true supply bundle and their true cost for the bundle. Assume to the contrary that a selling agent t requested a bundle s' that is not the bundle s that was recommended to him by his supply oracle. First assume the case that there exists a unit of commodity jthat is in s however it is not in s'. That means that the unit of commodity j will be allocated to the virtual buying agent constructed of selling agent t and t will not be paid for it. However we know that given the prices presented to t and his supply oracle, his utility will increase if we will not keep the unit of commodity j and will get paid for it, in its presented price. Thus t is better of requesting bundle s. Now assume that there exists a unit of commodity j that is in s' however it is not in s. That means that the unit of commodity j will not be allocated to the virtual buying agent constructed of selling agent t and t will be paid for it. However since selling agent t's supply oracle is utility maximizing, we know that t's utility will increase (or at lease will not decrease) by not selling the unit of commodity j and not get paid for it. Thus the selling agent is (weakly) better off reporting his true supply bundle. Any declaration of s's cost can not change the allocation (and therefore can not change the selling agent's utility) as the allocation is determined by the supply of bundles. Moreover, a selling agent's lie will be immediately exposed if he reports the cost of s such that it is more than the total price of the bundle sas his supply oracle is utility maximizing.

We continue by claiming that DYCOM is IR. For buying agents DYCOM is IR since a buying agent only pays for units of commodities he is allocated and his payment is computed based on the commodities price vector presented to him. As each buying agent's demand oracle is utility maximizing no allocation will result in a negative utility for a buying agent. For selling agents DYCOM is IR since a selling agent only gets paid for units of commodities that his virtual buying agent is not allocated (which is exactly his supply bundle) and the mechanism's payment to him is computed based on the commodities price vector presented to him. As each selling agent's supply oracle is utility maximizing no sell will result in a negative utility for a selling agent.

Now we claim that DYCOM does not run a deficit, i.e., DYCOM is BB. Since selling agents get paid according to the commodities prices that are presented when they arrive and buying agents pay according to the current prices they see, and since prices are non-decreasing between arrival times, every buying agent payment on every unit of a commodity will be at least as high as the payment for its selling agent.

We continue by analyzing the SWF approximation ratio.

LEMMA 3.2. DYCOM approximates the SWF of the trading buying agent and the remaining commodities with in $O(\Theta[(1 + s_{\max}(v_{\max} - c_{\min}))^{\frac{1}{\Theta-1}} - 1] + \theta).$

Before we present the proof of our approximation claim we like to compare DYCOM's approximation ratio with that of the other known combinatorial two sided market by [3]. [3]'s approximation for the SWF of the trading buying agent and the remaining commodities in a randomized mechanism and if all valuations and costs are subadditive³ is $8H_{s_{\text{max}}}$ where $H_{s_{\text{max}}}$ is the s_{max} harmonic number. Their mechanism assumes distributional knowledge of the median value of each selling agent's Θ , θ bounds. Figure 1 shows that for large markets DYCOM achieves better theoretical approximation ration than [3] even though [3]'s solution is randomized non-dynamic and the approximation ration is only guaranteed for the cases where valuations and costs are subadditive and not generated for the general case as ours⁴.

PROOF. In order to show DYCOM's SWF of the trading buying agent and the remaining commodities' approximation ratio, it is enough to show the SWF approximation ratio for buying and virtual buying trading agents as the last ones are allocated the remaining commodities.

Let $\Delta Primal$ be the change in the value of the primal solution and let $\Delta Dual$ be the change in the value of the dual solution. After each agent's arrival DYCOM updates a primal solution [*Primal*] and a dual solution [*Dual*]. In order to show the approximation ratio we need to prove that (i) the primal solution produced by DYCOM is feasible. (ii) the dual solution output by DYCOM is feasible. (iii) after each agent arrival, $\Delta Primal$ is at most w times $\Delta Dual$, where w would have been our desired approximation ratio if both the primal and dual solutions were initially 0. In that case we would have achieved an approximation of at least 1/w times the feasible primal solution we produce. Since our primal solution is not initially 0 but $\sum_{j=1}^{m} c_{min}a_j$, we need to reduce $\phi = \sum_{j=1}^{m} c_{min}a_j$ from [*Primal*] and we conclude that our approximation is at least $\frac{1}{w + \frac{\phi}{[Dual]}}$ times the feasible primal solution we produce. The lemma's claim follows directly by weak duality.

Primal feasibility: It is easy to verify that the primal solution produced by DYCOM satisfies all primal constraints. We omit the details due to space limitations.

Primal-Dual relation: We need to show the relations of $\Delta Primal$ and $\Delta Dual$ created by the arrival of agent *i*. Denote $\Delta x_j = x_j^{i+1} - x_j^i$. Let $q = \left(\exp\left(\frac{d_{s,j}}{a_j}\psi\right) - 1\right)$ and let



Figure 1: DYCOM's theoretical SWF approximation ratio under different bounds of agents' demand/supply level with respect to the overall supply of commodities in the market vs. Blumrosen and Dobzinski 2014 theoretical SWF approximation ratio. When demand/supply of each agent is bounded by at least 1/400 of total market units DYCOM's theoretical SWF approximation ratio is better than Blumrosen and Dobzinski 2014.

 $q' = \left(\exp\left(\frac{\psi}{\Theta}\right) - 1\right)$ for the ease of presentation.

$$\Delta \quad Primal = z_i + \sum_{j=1}^m \Delta x_j = v_{i,s}$$

$$+ \quad \sum_{j=1}^m \left[x_j \left(\exp\left(\frac{d_{s,j}}{a_j}\psi\right) - 1 \right) + \left(\frac{a_j}{s_{\max}^i} - a_j c_{\min}\right) q \right]$$

$$= \quad v_{i,s} + \sum_{j=1}^m \left(x_j \cdot \frac{d_{s,j}}{a_j} + \frac{d_{s,j}}{s_{\max}^i} - d_{s,j} c_{\min} \right) \cdot \frac{a_j}{d_{s,j}} q$$

$$\leq \quad v_{i,s} + \sum_{j=1}^m \left(x_j \cdot \frac{d_{s,j}}{a_j} + \frac{d_{s,j}}{s_{\max}^i} - d_{s,j} c_{\min} \right) \cdot \Theta q' \qquad (4)$$

$$\leq v_{i,s} + (v_{i,s} + 1 - c_{\min}) \cdot \Theta q' \tag{5}$$

$$< v_{i,s} + (v_{i,s} + 1) \cdot \Theta q' \tag{6}$$

$$\leq (1 + 2\Theta q') \Delta Dual$$
 (7)

Inequality (4) follows as for every $x, \psi \ge 1 x(e^{\frac{\psi}{x}}-1)$ is monotonic decreasing. Inequality (5) follows as $\sum_{j=1}^{m} \frac{d_{s,j}}{s_{\max}^{i}} \le 1$ $(s_{\max}^{i}$ is the size of the maximal bundle allocated until current agent's arrival) and as $\sum_{j=1}^{m} d_{s,j} \frac{x_{j}}{a_{j}} = \sum_{j=1}^{m} d_{s,j} p_{i}^{(j)} \le$ $v_{i,s}$. Also, since the minimal size allocated bundle is a single unit of one item type then $\sum_{j=1}^{m} d_{s,j} \ge 1$. Inequality (6) follows since $c_{\min} > 0$. Finally, Inequality (7) follows since $\Delta Dual = v_{i,s}$. Note that s_{\max}^{i} and ψ are non-decreasing throughout the agents' arrivals. Last but not least is bound-

³Subadditive - roughly speaking, the value of bundle A plus the value of bundle B is greater than the value of their union. ⁴Figure 1 shows its finding under subadditive valuations and costs. When we generate data removing this assumption DYCOM gives even better theoretical guarantees for the presented large markets. This graph was omitted due to page limitations

ing from above $\phi/[Dual]$.

$$\frac{\phi}{[Dual]} = \frac{\sum_{j=1}^{m} c_{\min} a_j}{\sum_{i=1}^{n} v_{i,s} y_{i,s}} \leq \frac{\sum_{j=1}^{m} c_{\min} a_j}{v_{i,s}} \leq \frac{\sum_{j=1}^{m} x_j^i}{v_{i,s}}$$
(8)

$$\leq \frac{\sum_{j=1}^{m} \left(x_{j}^{i} \frac{d_{s,j}}{a_{j}}\right) \cdot \frac{a_{j}}{d_{s,j}}}{v_{i,s}}$$

$$\leq \frac{\sum_{j=1}^{m} x_{j}^{i} \frac{d_{s,j}}{a_{j}} \cdot \theta}{v_{i,s}} \leq \theta \tag{9}$$

Inequality (9) follows since $\sum_{j=1}^{m} x_j^i \frac{d_{s,j}}{a_j} \leq v_{i,s}$. Substituting ψ and s_{\max}^i we achieve the approximation ra-

Substituting ψ and s_{\max}^i we achieve the approximation ratio of:

the of: $1+2\Theta q'+\theta \le 1+2\Theta \left(\exp\left(\frac{\ln(1+s_{\max}(v_{\max}-c_{\min}))}{\Theta-1}\right)-1\right)+\theta = O(\Theta[(1+s_{\max}(v_{\max}-c_{\min}))^{\frac{1}{\Theta-1}}-1]+\theta).$ Dual feasibility: DYCOM's solution is the solution produced by the dual solution. In order to prove dual feasibility we need to show that no agent is allocated more than a single bundle and that no commodity j is allocated more than a_j times. Since each arriving agent is asked to declare a bundle of interest through a demand oracle and the demand oracle outputs a single bundle as an out come, the single bundle constraint is satisfied. In order to prove the commodity constraint we prove that for every commodity j the price reaches the level of v_{max} after the allocation of at least $(1 - 1/\Theta)a_j$ units of the commodity. At the resulting high price no agent can afford to buy the commodity any more. Meaning, no more units of the commodity will be sold after the price reached the v_{\max} level . As each allocation of a commodity for an agent is at most $1/\Theta$ of the commodity, no more than a_j allocations of the commodity can occur in total. We look for a price expression such that when agent g arrives $\frac{1}{a_j}\sum_{i=1}^{g-1}\sum_{s\in S(i)} d_{s,j}y_{i,s} \geq 1 - 1/\Theta$, then the price is at least v_{\max} . We show that the price computed by DYCOM is such. We prove by induction that the price of one unit of commodity j at the arrival time of agent g is as follows:

Let $Q = \frac{\ln(1+s_{\max}^g(v_{\max}-c_{\min}))}{(1-1/\Theta)a_j}$ for the ease of presentation and let s_{\max}^g is the maximal bundle allocated by DYCA up to the arrival of agent g.

$$p_g^{(j)} = \frac{x_j^g}{a_j} \ge \frac{1}{s_{\max}^g} \left(\exp\left(Q \cdot \sum_{i=1}^{g-1} \sum_{s \in S(i)} d_{s,j} y_{i,s}\right) - 1 \right) + c_{\min}(10)$$

We omit the details of the induction proof due to space limitations.

4. EXPERIMENTAL RESULTS

We conduct an empirical evaluation of our suggested solution's performance against a range of known market benchmarks. We compare the allocative SWF efficiency of buying agents and unallocated commodities of DYCOM with the known non-dynamic single commodity unit-demand solution by [16] (Figure 2) and the dynamic single-commodity unitdemand solution by [2] (Figure 5)⁵. We also compare the allocative SWF efficiency DYCOM's buying agents and unallocated commodities with [3]'s randomized non-dynamic combinatorial market. (Figure 6)⁶. Our experimental results show that though DYCOM is dynamic, combinatorial and more general it can perform in practice as well as the above known solutions that were tailored for limited market settings and perform even better for some of the above known solutions for large markets. While [2]'s solution's performance mainly depends on the size of the valuation/cost range in the market (which may be large in an electronic global market), our DYCOM solution performs best on large markets where no buying or selling agents control a large portion of the demand or supply (See Figure 5). As was seen in Subsection 3.1 Figure 1 DYCOM theoretically performs better than [3] in large markets where the agents' values and costs are taken over a large spread and each agent's demand/supply is bounded by at least 1/400 of total market units. Interestingly the performance gap improves favorably towards DYCOM in the practical comparison. Figure 6 shows that even if each agent's demand/supply is bounded by at least 1/200 of total market units, DYCOM performs better. Figure 6 shows its finding under subadditive valuations and costs. When we generate date removing this assumption DYCOM performs even better with respect to [3] under the same size market's demand/supply bounds. The graph was omitted due to page limitations.

All results presented were averaged over 1000 trials. The comparisons with [16, 2] were performed on a market with 800 units of a commodity. In all the experiments we found minimal to no qualitative differences between the use of different distributions. We also compared DYCOM's practical performance with that of the theoretical results for multiple commodities (Figure 3 and Figure 4) and found that in practice DYCOM's SWF approximation ratio is improved by an order of magnitude and converges to half the SWF of an optimal non-dynamic combinatorial solution (simplex). We note that the theoretical approximation ratio proven in Subsection 3.1 converges to 0.06 in the runs we performed (Figure 4).

5. CONCLUSION AND DISCUSSION

In this paper we present and evaluate DYCOM the *first dynamic two-sided combinatorial market* that allows truthful and IR behavior for both buying and selling agents, keeps the market BB and approximates SWF efficiency. DYCOM is a general solution that dynamically matches agents that arrive over time in a two-sided combinatorial market with multiple commodities of multiple units.

The main idea behind our DYCOM solution is a transformation of the two-sided combinatorial market into a onesided combinatorial auction. The transformation of the market into an auction makes use of a novel principle that each selling agent is a buying agent of his own commodities. DY-COM is a primal-dual sequential posted-price mechanism

⁵We omitted the comparison of DYCOM with [4]'s solution as they conclude that their Chain mechanism performs essentially the same as [2]'s mechanism in practice.

 $^{^{6}}$ the comparison is done such that all valuations and costs are subadditive as [3] assumes such valuations and costs as part of their approximation ratio bound.



Figure 2: McAfee 2008's SWF vs. DYCOM's SWF. DYCOM seems to converge to an identical SWF approximation ration as McAfee as the valuation and cost range grows.

and its economic properties as well as its approximation guarantee are theoretically proven.

To validate the performance of our DYCOM solution, we experimentally tested the SWF efficiency of DYCOM under variety of agents' bid distributions and agents' demand against a number of benchmarks. Our experimental tests show that DYCOM is a general and practical platform as 1) it performs as well as the known McAfee [16]'s non dynamic single-commodity unit-demand two-sided market though DYCOM is tailored for a completely general combinatorial setting and it is dynamic unlike [16] and 2) it's approximation approaches 10 times that of [3]'s market's SWF in large markets though DYCOM is deterministic and dynamic unlike [3] which is randomized and non dynamic.

In addition to providing a practical solution to the important problem of a dynamic two-sided combinatorial market, we believe that our two-sided combinatorial market transformation into a one-sided combinatorial auction is of independent interest for future work on reducing other forms of multi-sided exchanges to the well studied form of one-sided auctions.

REFERENCES

- Y. Bartal, R. Gonen, and N. Nisan. Incentive compatible multi unit combinatorial auctions. In *TARK*, pages 72–87, 2003.
- [2] A. Blum, T. Sandholm, and M. Zinkevich. Online algorithms for market clearing. In SODA, pages 971–980, 2002.
- [3] L. Blumrosen and S. Dobzinski. Reallocation mechanisms. In EC, pages 617–640, 2014.
- [4] J. Bredin, D. Parkes, and Q. Duong. Chain: A dynamic double auction framework for matching patient agents. *Journal of Artificial Intelligence*



Figure 3: DYCOM's practical SWF approximation ratio under different bounds of agents' demand/supply level with respect to the overall supply of commodities in the market. The red line represents demand/supply of each agent that is bounded by 1/10 of total market units. The pink line represents demand/supply of each agent that is bounded by 1/800 of total market units. In large markets, i.e. where the demand/supply bound is smaller DYCOM's practical SWF approximation ratio converges to 0.5.



Figure 4: DYCOM's theoretical SWF approximation ratio under different bounds of agents' demand/supply level with respect to the overall supply of commodities in the market. The red line represents demand/supply of each agent that is bounded by 1/10 of total market units. The pink line represents demand/supply of each agent that is bounded by 1/800 of total market units. In large markets, i.e. where the demand/supply bound is smaller DY-COM's theoretical SWF approximation ratio converges to 0.06.

8



Figure 5: Blum et al.'s SWF vs. DYCOM's SWF. DYCOM seems to converge to an identical SWF approximation ration as Blum et al. as the valuation and cost range grows.



Figure 6: DYCOM's SWF approximation ratio under different bounds of agents' demand/supply level with respect to the overall supply of commodities in the market vs. Blumrosen and Dobzinski 2014 SWF approximation ratio. When demand/supply of each agent is bounded by at least 1/200 of total market units DYCOM's SWF approximation ratio is better than Blumrosen and Dobzinski 2014 even under Blumrosen and Dobzinski limiting assumption of subadditive valuations and costs.

Research, 30:133-179, 2007.

- [5] N. Buchbinder and R. Gonen. Incentive compatible mulit-unit combinatorial auctions: A primal dual approach. *Algorithmica*, 72(1):167–190, 2015.
- [6] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 2:17–33, 1971.
- [7] R. Colini-Baldeschi, B. de Keijzer, S. Leonardi, and S. Turchetta. Approximately efficient double auctions with strong budget balance. In SODA, pages 1424–1443, 2016.
- [8] P. Cramton, Y. Shoham, and R. Shoham. Combinatorial auctions. MIT press, 2006.
- [9] D. Dolgov and E. Durfee. Computationally-efficient combinatorial auctions for resource allocation in weakly- coupled mdps. In AAMAS, pages 657–664, 2005.
- [10] S. Dughmi, T. Roughgarden, and Q. Yan. From convex optimization to randomized mechanisms: toward optimal combinatorial auctions. In Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011, pages 149–158, 2011.
- [11] X. Feng, Y. Chen, J. Zhang, Q. Zhang, and B. Li. TAHES: A truthful double auction mechanism for heterogeneous spectrums. *IEEE Trans. Wireless Communications*, 11(11):4038–4047, 2012.
- [12] B. P. Gerkey and M. J. Mataric. Sold!: Auction methods for multi robot coordination. *IEEE Transactions on Robotics and Automation*, 18:758–768, 2002.
- [13] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.
- [14] M. Guo, E. Markakis, K. R. Apt, and V. Conitzer. Undominated groves mechanisms. J. Artif. Intell. Res. (JAIR), 46:129–163, 2013.
- [15] R. P. McAfee. dominant strategy double auction. Journal of Economic Theory, 56:434–450, 1992.
- [16] R. P. McAfee. The gains from trade under fixed price mechanisms. Applied Economics Research Bulletin, 1, 2008.
- [17] P. Milgrom, L. Ausubel, J. Levin, and I. Segal. Incentive auction rules option and discussion. In Appendix C to the FCC's Notice of Proposed Rulemaking, GN Docket No 12-268, October 2, 2012.
- [18] R. B. Myerson and M. A. Satterthwaite. Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29:265–281, 1983.
- [19] S. J. Rassenti, V. L. Smith, and R. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *The Bell Journal of Economics*, pages 402–417, 1982.
- [20] T. Sandholm and A. Gilpin. Sequences of take-it-or-leave-it offers: Near-optimal auctions without full valuation revelation. In AAMAS, pages 73–91, 2006.
- [21] E. Segal-Halevi, A. Hassidim, and Y. Aumann. SBBA: A strongly-budget-balanced double-auction mechanism. In SAGT, pages 260–272, 2016.
- [22] J. Song and A. Regan. Combinatorial auctions for transportation service procurement: The carrier perspective. *Transportation Research Record: Journal*

of the Transportation Research Board, 1833:40–46, 2003.

- [23] M. Stonebraker, R. Devine, M. Kornacker, W. Litwin, A. Pfeffer, A. Sah, and C. Staelin. An economic paradigm for query processing and data migration in mariposa. In *PDIS*, pages 58–67, 1994.
- [24] C. Tovey, M. Lagoudakis, and S. Koenig. The generation of bidding rules for auction-based robot coordination, volume In Multi-Robot Systems. From Swarms to Intelligent Automata Volume III. Springer, 2005.
- [25] W. Vickrey. Counterspeculation, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.
- [26] P. Wurman, W. Walsh, and M. Wellman. Flexible double auctions for electronic commerce: Theory and implementation. *Decision Support Systems*, 24:17–27, 1998.
- [27] H. Xu, J. Jin, and B. Li. A secondary market for spectrum. In *INFOCOM*, pages 31–35, 2010.

Proxy Voting for Revealing Ground Truth

Gal Cohensius and Reshef Meir Technion—Israel Institute of Technology

ABSTRACT

We consider a social choice problem where only a small subset of voters actually votes. The outcome of a vote with low participation rate could be far from the outcome reached by a vote with full participation. A possible solution to a vote with low participation rate is allowing voting by proxies. Proxy voting is a scenario which enables the voters that do not vote to transfer their voting rights to another voter.

In some voting settings voters try to discover or agree upon some ground truth while each voter gets a noisy signal about that truth [1, 7]. From this viewpoint, different voting scenarios can be compared upon the expected distance of the aggregated outcome from the truth. By comparing voting with and without proxies, we try to define the conditions under which proxy voting helps to get closer to the truth. A specific model of proxy voting was suggested and studied in [4]. In this paper we apply this model to the case where a ground truth exists. We analyze datasets of social choice and multiple-choice questions and show that Proxy voting can be beneficial in order to find an outcome that is closer to the ground truth. When the participation rate is low enough, proxy voting is always beneficial. In some instances, proxy voting can even get closer to the truth than a vote with full participation. This is a bit surprising since proxy voting uses strictly less information than full participation vote.

1. INTRODUCTION

In the model of proxy voting suggested by [4], it is shown that in various domains, allowing proxy voting results in an outcome that is closer to the aggregated opinion of the entire population. This means that proxy voting improves the social outcome when the outcome reached by the whole population is assumed to be good. In contrast to [4], where it is assumed that the aggregated vote of the entire population is optimal, in this paper we consider voting profiles that are derived from some *ground truth*. Thus the criterion for successful voting mechanisms is finding an outcome that is close to this ground truth. This paper compares proxy voting to full participation vote and to partial participation vote, and categorizes the conditions for which proxy voting is beneficial i.e. it results in an outcome that is closer to the ground truth according to some natural metric.

For example, when votes are orders (permutations) over a set, a natural metric is the Kendall tau distance. We use this distance to assign proxies (i.e., an inactive voter will relegate her voting rights to the voter with nearest vote). Then we aggregate votes to a single order using some standard social welfare function (voting rule), where only active voters participate, and are weighted by their number of followers. Finally, we measure the Kendall tau distance from the aggregated outcome and the ground truth.

1.1 Contribution

We apply the proxy voting model from [4] both to synthetic datasets generated from Mallow's distribution model with a given ground truth order, and to empirical datasets in two natural domains. The first is from a crowd sourcing experiment where subjects were requested to order four items according to their correct order [6], and the second is from Pisa standard tests take by Israeli students.

We show that for all voting rules we tried and for every sample size, it is beneficial to allow inactive voters to use proxies, in the sense that the aggregated vote becomes closer to the ground truth on expectation. For some datasets, proxy voting can be even better than a full participation vote.

We analyze the reasons for this improvement by looking at the distribution of proxies' weights, and suggest a preliminary theoretical result that explains why better proxies get higher weights.

2. PRELIMINARIES

We follow the model and the definitions of [4] as described below.

Domains.

 \mathcal{X} is the *space*, or set of possible votes, or voter types. Consider some finite set of alternatives $A = \{a_1, \ldots, a_l\}$. In this paper we will consider two Domains:

- 1. multiple discrete issues $\mathcal{X} = A^k$ with the Hamming distance
- 2. ordinal preferences $\mathcal{X} = \Pi(A)$ with the Kendall tau distance.

distances.

The Hamming distance between two agents' positions v_1, v_2 is the number of issues on which the agents disagree on. For example, the Hamming distance between $v_1 = (1, 5, 0, 0), v_2 = (2, 1, 0, 1)$ is 3 since the agents' positions are different in issues j = 1, 2, 4.

The Kendall tau distance between two ranking is the number of pairwise disagreements between them. Kendall tau distance is also called bubble-sort distance since it is equivalent to the number of swaps that the bubble sort algorithm would make to place one list in the same order as the other list. For example, the Kendall tau distance between (a,b,c,d) and (b,c,a,d) is 2, since two swaps are

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

needed in order to get from the first ranking to the second. Note that Hamming distance between agents' positions equals the Kendall tau distance between their ranking.

Ground Truth and profiles.

The *Truth* $T \in \mathcal{X}$ is a particular point in space (either an *m*-size vector in domain (1), or an order over alternatives in domain (2)). The model does not assume a priori any dependence between the truth and the voting profiles.

 S_N is the voting profile of the set of voters N of size n. The interesting cases are when society have some idea about the truth, that is to say, there is some dependency between S_N and the truth T.

Mechanisms.

A mechanism $\mathbf{g}: \mathcal{X}^n \to \mathcal{X}$ (also called a voting rule) is a function that maps any profile (set of positions) to a winning position.

For the binary issues we will focus on a simple *Majority mechanism* that aggregates each issue independently according to the majority of votes. That is, $(\mathbf{mj}(S))^{(j)} = 1$ if $|\{i : s_i^{(j)} = 1\}| > |\{i : s_i^{(j)} = 0\}|$ and 0 otherwise, where $s^{(j)}$ is the *j*'th entry of position vector *s*. For example, say that the number of voters *n* is 7, for any issue *j* the outcome of the **mj** mechanism will be '1' if at least 4 of the voters vote '1', else the outcome will be '0'. In all mechanisms we break ties uniformly at random.

For the ordinal preferences we use pairwise majority (**mj**) in addition to four different voting rules: Kemeny, Borda, Plurality and Veto which are denoted respectively by **km**, **bo**, **pl** and **vt**. In order to aggregate pairwise majority, ordinal votes are converted into issues by checking for each pair of alternatives (α , β) whether α is preferred over β . For example assume votes are a strict ranking of 4 alternatives, then a conversion into issues will result in $\binom{4}{2} = 6$ discrete binary issues. On each issue a majority voting rule is resolute.

All the voting rules (mechanisms) that we use naturally extend to weighted finite populations, by considering voting with w_i copies of voter i

Scenarios.

We label the 'everyone vote' scenario as E, the basic scenario as B and the proxy scenario as P. In scenario E, all voters votes and the result is $g(S_N)$. In scenario B, only the subset of active voters $M \subseteq N$ votes, while inactive voters abstain. The result is $g(S_M)$. In scenario P, active voters vote, while each unavailable voter grant her voting right to an active voter. Given a set M of active agents, the decisions of inactive voters are specified by a mapping $J_M : \mathcal{X} \to M$, where $J_M(x) \in M$ is the proxy of any voter located at $x \in \mathcal{X}$. Thus the results in scenario P is $g(S_M, \mathbf{w}_M)$, where for each $j \in M$, $w_j = |\{i \in N : J(s_i) = j\}|$, i.e. the weight of proxy j is the number of inactive voters who select proxy j, plus himself.

Without further constraints, we will assume that the proxy of a voter at x is always its nearest active agent, i.e. the agent whose position (or vote) are most similar to x. Thus for every subset of active agents M, we get a partition of \mathcal{X} . We can compute according to the metric we choose the *weight* of each active agent j. This is done by summing the number of inactive agent that j is their closest active agent. Formally, $J_M(x) = \operatorname{argmin}_{j \in M} ||x - s_j||$ and $w_j = |\{i \in N : J(s_i) = j\}|$. For example, voter i' is at location $x_{i'}$ and is following the closest active voter to her, which is $J_M(x_{i'}) = \operatorname{argmin}_{j \in M} ||x_{i'} - s_j|| = j'$, the weight of proxy j' is increased by one. If there are several proxies at the same distance, voter i selects one of them at random.

To recap, an *instance* is defined by a profile S_N and a truth T. Each instance produce an outcome according to the scenario $Q \in \{E, B, P\}$, mechanism $\mathbf{g} \in \{\mathbf{km}, \mathbf{bo}, \mathbf{pl}, \mathbf{vt}, \mathbf{mj}\}$, and the sample size |M| = m.

Evaluation.

We want to measure how close is $\mathbf{g}^Q(S_N)$ to the *truth* T. We define the *error* as the distance between $\mathbf{g}^Q(S_N)$ and the truth. Note that the Kendall tau distance is the Hamming distance over the induced binary vectors of pairwise preferences (where each pair of alternatives in A induces a binary issue). Thus the distance between any two votes $\mathbf{s}, \mathbf{s}' \in \mathcal{X}$ can be written as $\|\mathbf{s} - \mathbf{s}'\|$ (since these are binary vectors it does not matter which norm is used). In particular, the error of \mathbf{g} on S_N in scenario Q is $\|\mathbf{g}^Q(S_N) - T\|$.

The *loss* of a mechanism g is calculated according to its mean square error (MSE)—the expected squared distance from the truth—over all samples of m available voters.

$$\mathcal{L}^{Q}(T, S_{N}, m) = \mathbb{E}_{M \sim U[N^{m}]} \left[\| \mathbf{g}^{Q}(S_{N}) - T \|^{2} \right], \quad (1)$$

where the mechanism g can be inferred from the context, and the expectation is over all subsets of m positions sampled uniformly without repetitions from S_N (sometimes omitted from the subscript).

3. SIMULATIONS

The experiments were designed to test two hypothesis:

- 1. $\mathcal{L}^P < \mathcal{L}^B$ for every voting rule. That is, whether for random samples of a given size *m*, proxy voting always yields an outcome which is closer to the truth than an outcome yield by unweighted vote with the same set of proxies.
- 2. there is some setting where $\mathcal{L}^P < \mathcal{L}^E$. That is, under certain parameters, taking a sample of active voters and use them as proxies will yield an outcome which is closer to the truth than the outcome reached by aggregating all votes.

3.1 Datasets

3.1.1 Generative model of votes

We generate synthetic profiles, by sampling rankings from Mallow's model. Mallow's distribution model is a distance-base ranking model, which is parametrized by a true order T and a dispersion parameter $\phi \in (0, 1]$. For any ranking $r \in \Pi(A)$, the Mallows model specifies:

$$\Pr(r) = \Pr(r|T, \phi) = \frac{1}{Z}\phi^{d(r,T)}$$

Where *d* is the Kendall tau distance and $Z = \sum_{r' \in \omega} \phi^{d(r',T)}$ is a normalization constant. When $\phi = 1$ the distribution is uniform over all permutations (very noisy), when $\phi \ll 1$ almost all the mass is concentrated at *T* (small amount of noise). Using synthetic datasets helps understanding the role of each parameter while fixing the others. Mallow's distribution model is one of the two most popular noise models in the machine learning community together with Plackett-Luce [5].

3.1.2 Natural experiments

We used two ranking datasets made by [6] using crowd-sourcing. One is referred to as the dots dataset. In that test voters were shown four pictures with dots and were asked to ranked the pictures by the number of dots from least to most. The number of



Figure 1: $\mathcal{L}^{\{E,B,P\}}$, under generative model of preference, using five voting rules. Note that for Mallow's model with low dispersion $\mathcal{L}^E < \mathcal{L}^P < \mathcal{L}^B$. For Mallow's model with high dispersion $\phi = 0.95$, (high mistakes probability) $\mathcal{L}^P < \mathcal{L}^E$.

dots in the pictures were $\{200, 200 + i, 200 + 2i, 200 + 3i\}$ for $i = \{3, 5, 7, 9\}$. The data contain at each level of noise, i, 40 preference profiles. Each profile contain about 20 voters. This test has been suggested as a benchmark task for human computation in [2]. In the second dataset, refereed as the sliding puzzle, voters had four situations of a size 8 sliding puzzle, and they had to order those puzzles by the minimal number of moves left to solve them. The number of minimal moves were $\{d, d+3, d+6, d+9\}$, for $d = \{5, 7, 9, 11\}$. Again, 40 preference profiles, each containing 20 voters, were gathered. Those dataset were valuable for us since they contain a true order. A comprehensive explanation on how the data was gathered in the two experimental datasets can be found at [6]. Another dataset that we examined is the results of a test called the program for international student Assessment (PISA). This test evaluate education systems worldwide by testing the skills and knowledge of 15-year-old students. We where looking for a dataset of multi choice questions (without missing data), thus we included only the students who answered all of the first 18 multiple choice questions. This data match the first domain (multiple discrete issues). Our dataset contain n = 571 student (voters), answering k = 18 questions (issues), each question has 4 possible answers.

3.1.3 Method

The simulation start by creating a profile of votes, either by sampling from Mallow's model or by loading the empirical datasets. Then, a ranking profile of active voters was simulated for each scenario: The E scenario used the original profile with N voters. The B scenario sample uniformly at random, a given size m < n of active voters while the other abstained. In P scenario M were active voters while each of the other voters $N \setminus M$ becomes a clone of her closest agent in M. Thirdly an aggregated order was calculated for each scenario using five popular voting rules: $\{\mathbf{km}, \mathbf{bo}, \mathbf{pl}, \mathbf{vt}, \mathbf{mj}\}$. Lastly, for each voting rule, we compared the Kendall tau distances (Hamming distance for Pairwise majority) from the truth T to the order obtained by each scenario.



Figure 2: $\mathcal{L}^{\{E,B,P\}}$, for puzzle and dots datasets, using five voting rules. Proxy voting is doing better than random sample, i.e. $\mathcal{L}^{E} < \mathcal{L}^{P} < \mathcal{L}^{B}$.

3.2 Results

Our simulations on the generative model shows that $\mathcal{L}^{P}(\mathbf{g}) < \mathcal{L}^{B}(\mathbf{g})$ for all five voting rules \mathbf{g} , in all datasets, and for almost every sample size m (Figs. 1, 2 and 5). Same results are obtain analyzing the Pisa dataset (Fig. 9) with the Majority voting rule (In this dataset the domain is multiple discrete issues, thus the voting rule is majority.) This results supports our conjecture that proxy voting reveals ground truth better than a random sample, and often considerably better.

The big difference in the settings from [4] is that when there is some hidden ground truth, the outcome gathered from full participation vote (scenario E) is *not necessary optimal*. On some votes the best active voter get much closer than the aggregated decision of the entire population, thus there is hope that with the appropriate weights, proxy voting can do better than *E*. Indeed for some datasets proxy voting is even better than a full participation vote, e.g., Mallow's model with 4 alternatives, 20 voters and $\phi = 0.95$ (Fig. 1). This is also true for about half the crowd-sourcing datasets (Fig.3). This is an interesting phenomenon since proxy voting uses strictly less information than a full participation vote. In future work we will try to characterize the conditions for that to happen. For now we only found some rules of thumb:

- Amount of noise should be high enough in order to make the *E* scenario do pretty bad. If ordering is too easy, *E* will have almost no loss, thus no scenario can do better, this is the situation in [4] where the outcome reached by scenario *E* is the optimal one.
- There should be high variance in the individual performance of voters. If all voters have roughly the same accuracy then proxy voting does not help much.

3.3 Analysis

Denote by $R_i = ||\mathbf{s}_i - T||$ the distance of voter *i* from the truth. The reason that proxy voting gets closer to the truth than a random sample lies in the weight distribution of the proxies. While in scenario *B* the weights are uniform by definition, at scenario *P* the weights are roughly decreasing in their ratio of mistakes R_i , that is



Figure 3: \mathcal{L}^E vs. \mathcal{L}^P , examining the 40 dots datasets (noise level i=3), under Borda voting rule and m = 10 proxies. Markers under the 45 degree line are datasets where P is closer to the ground truth than E. A small random noise was added to \mathcal{L}^E in order to better visualize close outcomes.



Figure 4: \mathcal{L}^E vs. \mathcal{L}^P shown for the 40 datasets of d = 11 puzzle.

to say, better proxies get more voting weight. When dispersion is low, the distribution of the weights is monotonic decreasing in R_i . When dispersion raises, some of the voting weight moves towered the worst proxy, resulting a single dip distribution, with peaks at the best and worst proxies, see Fig 7.

4. EXPLAINING PROXY WEIGHTS

In the previous section, we observed empirically that better proxies (i.e. ones closer to the ground truth T) tend to get more followers and thus higher weight. We are interested in a theoretical model that explains this. One such result was given in [4] for the limit case of $k \to \infty$ binary issues, where essentially *all* inactive voters select either the best or the worst proxy, according to which one is closer. However, in realistic scenarios (including our datasets), the number of issues is much smaller.

We model a simplified version of the problem, where there is one follower which is requested to choose a proxy from two active voters. A priori, we only know the distribution of votes, and we



Figure 5: The expected error in scenarios $\{E, B, P\}$. Using Borda voting rule. Examining d = 11 puzzle dataset . $\mathcal{L}^P < \mathcal{L}^E$ for a subset of active voters M large enough.



Figure 6: P is decreasing faster than B and better approximating E.

want to estimate the probability that the follower would choose the better proxy.

Following [4], we model each agent with a fixed error probability P_i . Consider two active agents with error probabilities $P_i < P_j < 0.5$, and an inactive agent with error probability Z < 0.5. W.l.o.g. T = 0. Thus $\mathbf{s}_i, \mathbf{s}_j$ and \mathbf{z} are random binary vectors of length k, whose entries are '1' with respective probabilities of P_i, P_j , and Z.

Fix the values of the best proxy $P_i = P$, an inactive agent Z, and the number of issues k. Denote by $\epsilon = P_j - P_i > 0$ the difference in the quality of proxy j from the best proxy. We want to understand better how the probability of selecting the worse proxy P_j behaves as ϵ and k vary. Note that this probability is taken in expectation over all realizations of $\mathbf{s}_i, \mathbf{s}_j, \mathbf{z}$, as in each such realization the decision of the inactive voter is deterministic (up to tie-breaking).

[4] showed that z is more likely to be closer to s_i , and that the probability of being closer to j drops exponentially with the number of issues k. Let $q_i = Pr(s_i^{(t)} \neq z^{(t)}) = P_i(1-Z) + (1-Z)$



Figure 7: Weight per proxy, ordered by the ratio of wrong answers R_i in increasing order. The dots dataset.



Figure 8: Weight per proxy, ordered by the ratio of wrong answers R_i in increasing order. Mallow's model with 4 alternatives, 20 voters, and dispersion parameter $\phi = 0.95$.

 $P_i Z = P_i + Z - 2P_i Z$. Indeed, they showed

$$Pr(||z - s_i|| > ||z - s_j||) \cong \Phi\left(\frac{\sqrt{k}(q_j - q_i)}{\sqrt{q_i(1 - q_j) + q_j(1 - q_i)}}\right),$$

where $\Phi(x) = Pr_{X \sim N(0,1)}(X > x)$, and the approximation is due to the Binomial-to-Normal approximation.

Note that $q_j - q_i = P_j + Z - 2P_jZ - (P_i + Z - 2P_iZ) = (P_j - P_i)(1 - 2Z) = \epsilon(1 - 2Z)$. Thus

$$Pr(\|z - s_i\| > \|z - s_j\|) \cong \Phi\left(\frac{\sqrt{k}(q_j - q_i)}{\sqrt{q_i(1 - q_j) + q_j(1 - q_i)}}\right)$$

$$=\Phi\left(\frac{\sqrt{k}(1-2Z)\epsilon}{\sqrt{(P+Z-2PZ)(1-(P+\epsilon+Z-2(P+\epsilon)Z))}}\right)$$



Figure 9: The expected error in scenarios $\{E, B, P\}$. Examining the Pisa dataset. The expected error of E is 3, meaning that the aggregated majority vote made 3 mistakes out of 18 questions. The expected error for both scenarios B, P is roughly decreasing in the number of active voters m but is substantially lower for P.

$$= \Phi\left(\frac{\sqrt{k}(1-2Z)\epsilon}{\sqrt{(4Z^2-8PZ^2+8PZ-4Z+2P+1)\epsilon}} \right)$$
$$= \Phi\left(\frac{C_1\epsilon}{\sqrt{C_2\epsilon+C_3}}\right) = \Phi\left(\Theta(\sqrt{\epsilon})\right) = exp(-\Theta(\epsilon)).$$

For some constants $C_1 > 0, C_2 > 0, C_3$

That is, the probability that j is selected decreases exponentially fast in the distance between the error rate of j and that of the best proxy. The drop is exponential when the distance is large enough and there are enough issues. Another observation is that if we fix $\epsilon < P < 0.5$ and Z approaches 0.5 (i.e. an ignorant inactive agent), then the term in brackets approaches 0. In other words, ignorant agents spread their weight roughly evenly over all active voters, whereas smart agents are substantially more likely to give their vote to a good active voter.

This supports the intuitive argument from [3] regarding the "Anna Karenina principle" (as good agents are indeed similar to one another), and thus at least partially explains the weight distribution of active agents. To see if this is a sufficient explanation, one needs to compare the actual weight distribution, and specifically $\frac{w_j}{w_1}$, to the expression above.

REFERENCES

- Ioannis Caragiannis, Ariel D Procaccia, and Nisarg Shah. When do noisy votes reveal the truth? In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 143–160. ACM, 2013.
- [2] Lydia B Chilton, John J Horton, Robert C Miller, and Shiri Azenkot. Task search in a human computation market. In

Proceedings of the ACM SIGKDD workshop on human computation, pages 1–9. ACM, 2010.

- [3] Gal Cohensius, Shie Manor, Reshef Meir, Eli Meirom, and Ariel Orda. Proxy voting for better outcomes. *arXiv preprint arXiv:1611.08308*, 2016.
- [4] Gal Cohensius, Shie Manor, Reshef Meir, Eli Meirom, and Ariel Orda. Proxy voting for better outcomes. In *Proceedings* of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2017), 2017.
- [5] Tyler Lu and Craig Boutilier. Effective sampling and learning for mallows models with pairwise-preference data. *Journal of Machine Learning Research*, 15(1):3783–3829, 2014.
- [6] Andrew Mao, Ariel D Procaccia, and Yiling Chen. Better human computation through principled voting. In *AAAI*. Citeseer, 2013.
- [7] H Peyton Young. Condorcet's theory of voting. *American Political science review*, 82(04):1231–1244, 1988.

Committee Scoring Rules, Banzhaf Values, and Approximation Algorithms

Edith Elkind University of Oxford Oxford, UK Piotr Faliszewski AGH University Krakow, Poland Martin Lackner University of Oxford Oxford, UK

Dominik Peters University of Oxford Oxford, UK Nimrod Talmon Weizmann Institute Rehovot, Israel

ABSTRACT

We consider committee scoring rules (a family of multiwinner voting rules) and define a class of cooperative games based on elections held according to these rules. We show that there is a polynomial-time algorithm for computing the Banzhaf value for a large subclass of these games and we show, using this Banzhaf value, an appealing heuristic algorithm for computing winning committees. We evaluate this algorithm experimentally for the case of the Chamberlin-Courant voting rule.

CCS Concepts

 $\label{eq:computing methodologies} \bullet Multi-agent systems; \\ Cooperation and coordination;$

Keywords

committee scoring rules, approximation algorithms, Banzhaf value, greedy algorithms, winner determination

1. INTRODUCTION

The goal of a multiwinner election is to choose a subset (a committee) of presented items (candidates) based on the preferences of a group of agents (the voters). Committee elections are a natural model for various tasks [12], ranging from shortlisting [4], through numerous business applications [13, 19, 20, 27], to tasks involving proportional representation, such as parliamentary elections [1,6].

In consequence, there is a great number of very diverse multiwinner voting rules, based on many different principles. For example, Kilgour [18] discusses various approvalbased rules (see also the work of Aziz et al. [2] for a more computational persective), Gehrlein [16] and Ratliff [25] discuss elections in the ordinal model that are based on the Condorcet principle¹, and many rules can be seen as extensions of single-winner scoring rules (e.g., the Bloc rule, the k-Borda rule [9], or the Chamberlin–Courant [8] and Monroe [22] rules). Elkind et al. [12] recently provided the formalism of committee scoring rules that captures many of the rules from this last group.

In this paper, we focus on the family of committee scoring rules (with a particular focus on decomposable committee scoring rules rules [13] in the theoretical part of the paper, and on the Chamberlin–Courant rule², in the experimental part). We show that committee scoring rules can be used to define a class of cooperative games, consider the Banzhaf value of this game, and show that it leads to improved approximation algorithms for our rules.

Committee Scoring Rules. We consider elections where the voters express preferences over the candidates by ranking them from the most to the least desired ones (we refer to such preference orders as votes), and where the goal is to select a group of candidates, i.e., a committee, of a given size k. A single-winner scoring function associates each position in a preference order with a score. For example, the Borda scoring function for m candidates, β_m , associates value m-iwith position i (so the top ranked candidate has Borda score m-1, the next one m-2, and so on). A committee scoring function associates a position of a committee in a preference order (i.e., a sequence of the positions of the committee members, sorted in the increasing order) with a numerical score, and the score of a committee in an election is the sum of the scores it receives from all the voters. For example, the k-Borda rule uses the committee scoring function that sums up the Borda scores of the committee members (within a vote), whereas the Chamberlin-Courant rule uses a scoring function under which the score of a committee (in a given vote) is the Borda score of the top-ranked committee member (referred to as the representative of this voter in the committee). In consequence, the k-Borda rule tends to select very similar candidates, whereas the Chamberlin-Courant rule tries to select a very diverse committee [11].

Cooperative Games for Committee Scoring Rules. Multiwinner elections and committee scoring rules define a certain natural class of cooperative games. Let us consider an election E (with candidate set C and voter collection V) and some committee scoring rule \mathcal{R}_f with underlying committee

¹In single-winner voting, a candidate is a Condorcet winner if he or she is preferred to every other candidate by a majority of the voters (albeit, possibly a different majority in each case).

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

 $^{^{2}}$ Naturally, we intend to extend these results, but for the current—preliminary report of our work—in the experimental evaluation we limit ourselves to this rule only.

scoring function f (see the preliminaries for exact formal definitions).

We define a cooperative game $G(E, \mathcal{R}_f)$, where the candidates are the players and the value of a coalition S is the score assigned by f to committee S. Given such a game, we can use the full set of tools developed within cooperative game theory to analyze properties of the underlying election; here, in particular, we consider the Banzhaf values of the candidates, where the Banzhaf value of a player *i* in a cooperative game is its average marginal contribution to the value of a coalition, where the coalitions are chosen uniformly at random from the set of all possible coalitions that do not include i. Intuitively, the Banzhaf value of a candidate (a player) in our game measures the importance of this candidate. There are, however, two issues to resolve. First, the definition of Banzhaf value requires us to consider all coalitions, while in committee elections the size of the committee is usually constrained. Second, we may already know some committee members (see below) and the definition of the Banzhaf value does not account for this. Thus, we consider a variant of the Banzhaf value where we measure the average marginal contribution of a player to a randomly chosen coalition that (a) contains some given preselected players, and (b) is of a required size.

We show that our generalized variant of the Banzhaf value is polynomial-time computable for a large class of committee scoring rules (including the Chamberlin–Courant rule, but also several more involved rules).

Application of the Banzhaf Value. Unfortunately, many committee scoring rules are NP-hard to compute [5, 19, 24, 27]. One way to deal with this problem is to use approximation algorithms, the first of which was proposed by Lu and Boutilier [19] for the case of the Chamberlin–Courant rule. This algorithm, which is a natural incarnation of the celebrated greedy approximation algorithm for maximizing submodular functions of Nemhauser et al. [23], achieves approximation ratio of $(1 - \frac{1}{e})$. Indeed, the algorithm was shown to be applicable to a very large subclass of committee scoring rules [13] (and to date it is the only general algorithm that applies to a wide class of committee scoring rules and provides guarantees regarding the output quality).

The greedy algorithm proceeds as follows. We start with an empty committee and in a sequence of k iterations (where k is the number of candidates that we want to select) we keep adding to the committee those candidates that increase its score by the largest value. While this algorithm seems to be achieving very good results in practice (see the works of Lu and Boutilier [19] and Skowron et al. [28] for some experimental results), it also has some drawbacks. For example, if it is used to select an approximate Chamberlin-Courant committee, then in the first iteration it always selects the candidate with the highest individual Borda score (i.e., the so-called Borda winner), even though in many cases this candidate does not belong to the optimal committee. While it is clear that an approximation algorithm does not always choose the optimal committee, always selecting the Borda winner creates a huge systematic bias against some candidates that might otherwise be selected (this effect is illustrated in the full version of the work of Elkind et al. [11]). This bias is a strong argument against using the algorithm.

We modify the greedy algorithm so that instead of choosing the candidate that increases the score of the committee the most, it chooses the candidate with the highest Banzhaf value (for a given committee size and the candidates already selected in previous iterations).

We evaluate our algorithm experimentally for the case of the Chamberlin–Courant rule. In our experiments, the Banzhaf-based algorithm essentially always outperforms the original greedy algorithm, and often outperforms the algorithm of Skowron et al. [28] that underlies their polynomialtime approximation scheme (PTAS) for the Chamberlin– Courant rule (we use a variant of the algorithm improved by Elkind et al. [11]; see the full version of their paper).

Our approach is partially inspired by the line of work regarding the use of the Shapley value to extend centrality notions for networks (see the work of Michalak et al. [21] as a representative paper). We also model our problem (finding a good, approximate committee) using the language of cooperative game theory and use its tools to obtain improved results.

Organization of the Paper. In Section 2, we provide necessary background regarding elections, committee scoring rules, and cooperative games. Then, in Section 3, we discuss the Banzhaf value for the games we define as well as algorithms for computing it. In Section 4, we present the greedy algorithm based on the Banzhaf value, and in Section 5 we evaluate it experimentally for the Chamberlin–Courant rule. We discuss further research directions in Section 6.

2. PRELIMINARIES

In this section we present necessary background regarding elections and cooperative games. For a positive integer t, we write [t] to denote the set $\{1, \ldots, t\}$. For a logical expression Λ , by $[\Lambda]$ we mean 1 if Λ is true and 0 otherwise.

2.1 Multiwinner Elections

An election is a pair E = (C, V), where $C = \{c_1, \ldots, c_m\}$ is the set of candidates and $V = (v_1, \ldots, v_n)$ is a collection of voters. Each voter v is associated with a preference order \succ_v , i.e., with a ranking of the candidates from C (from best to worst). A single-winner voting rule is a function that, given an election, outputs a set of tied winners.³ In multiwinner elections we are interested in choosing whole committees of candidates of a given size k. A multiwinner voting rule \mathcal{R} is a function that, given an election E = (C, V)and a positive integer $k, k \leq |C|$, returns a family of size-kcommittees that tie as winners. Before we discuss committee scoring rules, used for multiwinner elections, we discuss single-winner scoring rules, on which they are based.

Single-Winner Scoring Rules. Let E be an election with m candidates. For a candidate c and a voter v, we write $\text{pos}_v(c)$ to denote the position of c in v's preference order. A singlewinner scoring function γ_m for m candidates, $\gamma_m : [m] \to \mathbb{R}$, is a non-increasing function that associates each possible position in a vote with a score. For example, $\beta_m(i) = m - i$ is the Borda scoring function; for each positive integer t, $\alpha_t(i) = [i \leq t]$ is the t-Approval scoring function. Typically,

 $^{^{3}\}mathrm{In}$ practice, it is necessary to have some tie-breaking scheme. We disregard this issue for convenience.

we consider families of scoring functions, e.g., $\gamma = (\gamma_m)_{m \in \mathbb{N}}$, with one function for each number of candidates. For such a family γ , the γ -score of candidate c in election E = (C, V)is γ -score_E $(c) = \sum_{v \in V} \gamma_{|C|}(\text{pos}_v(c))$. For each scoring function γ , we have a single-winner scoring rule, denoted by \mathcal{R}_{γ} , which is defined as the voting rule which outputs the candidates with the highest γ -score.

Committee Scoring Rules. Elkind et al. [12] generalize the idea of single-winner scoring functions to the committee setting. Consider an election E with m candidates and a given committee size k. We define the position of a committee S (i.e., of a size-k set of candidates) in some vote v to be the increasing sequence resulting from sorting the set {pos_v(c) | $c \in S$ }. We write $[m]_k$ to denote the set of all length-k increasing sequences of numbers from [m]. For two committee positions $I = (i_1, \ldots, i_k)$ and $J = (j_1, \ldots, j_k), I, J \in [m]_k$, we say that I dominates Jif $i_1 \leq j_1, \ldots, i_k \leq j_k$.

A committee scoring function $f_{m,k}$ for m candidates and committee size $k, f_{m,k}: [m]_k \to \mathbb{R}$, is a function that associates each committee position with a score in such a way that if some committee position I dominates some committee position J, then it holds that $f_{m,k}(I) \ge f_{m,k}(J)$.

DEFINITION 1 (ELKIND ET AL. [12]). Let $f = (f_{m,k})_{k \leq m}$ be a family of committee scoring functions. The committee scoring rule \mathcal{R}_f is a multiwinner rule that, given an election E = (C, V) and a committee size k, outputs those committees S that maximize the value:

$$f$$
-score_E $(S) = \sum_{v \in V} f_{|C|,k} (\text{pos}_v(S)).$

To distinguish single-winner scoring functions and committee scoring functions, we use Greek letters to denote the former and Latin letters to denote the latter.

Examples of Committee Scoring Rules. It turns out that the family of committee scoring rules is quite rich [13]; specifically, a number of well-known multiwinner voting rules are, in fact, committee scoring rules. For example, the k-Borda rule (which outputs committees of k candidates with the highest Borda scores) is the committee scoring rule defined via the following scoring function:

$$f_{m,k}^{k\text{-Borda}}(i_1,\ldots,i_k) = \beta_m(i_1) + \cdots + \beta_m(i_k)$$

. .

The SNTV and the Bloc rules are defined analogously, where, instead of the Borda scoring function, the former uses 1-Approval (known as the Plurality scoring function) while the latter uses the k-Approval scoring function. Committee scoring rules of this form are known as *weakly separable*. Identifying a winning committee under a weakly separable rule can be done in polynomial time (provided the underlying single-winner scoring functions are polynomialtime computable), since one can compute the corresponding score of each candidate independently from the others.

The Chamberlin–Courant rule (β -CC) is the committee scoring rule defined by scoring functions of the following form:

$$f_{m,k}^{\beta-\mathrm{CC}}(i_1,\ldots,i_k) = \beta_m(i_1).$$

Intuitively, under the Chamberlin–Courant rule, each voter is assigned a representative (the committee member that he or she ranks the highest, among all the selected committee members) and each voter increases the score of the committee by exactly the Borda score of his or her representative. The committees with the highest scores tie as co-winners. Elkind et al. [12] argue that this rule is useful where we aim to find a diverse committee in which each voter is well represented (but where each committee member can represent different numbers of voters). The Chamberlin–Courant rule is computationally hard [19,24] but there are good parameterized algorithms for it [5] as well as approximation algorithms and heuristics [15, 28].

Our final example of a committee scoring rule is the α_k -PAV rule, which is a variant of the *proportional approval voting* (*PAV*) rule, adapted to the format of committee scoring rules. It uses scoring functions of the following form:

$$f_{m,k}^{\alpha_k \text{-PAV}}(i_1, \dots, i_k) = \sum_{t=1}^k \frac{1}{t} \cdot \alpha_k(i_t)$$

Aziz et al. [1], Elkind et al. [11], Sánchez-Fernández et al. [26] and Brill et al. [6] provide strong evidence that the α_j -PAV rule is a very good choice if the goal is to elect a committee that represents voters proportionally. Winner determination for this rule is computationally hard, but there exists an approximation algorithm [27] and parameterized algorithms [14].

Decomposable Committee Scoring Rules. Throughout this paper we are mostly interested in *decomposable committee* scoring rules [13]. A committee scoring rule is decomposable if it can be defined through committee scoring functions of the following form:

$$f_{m,k}(i_1,\ldots,i_k) = \gamma_{m,k}^{(1)}(i_1) + \cdots + \gamma_{m,k}^{(k)}(i_k),$$

where $\gamma = (\gamma_{m,k}^{(t)})_{1 \le t \le k \le m}$ is a vector of single-winner scoring functions. All the committee scoring rules mentioned above are decomposable committee scoring rules; indeed, the subclass of decomposable committee scoring rules is quite a profound subclass of all committee scoring rules [13].

2.2 Cooperative Games

A cooperative game $G = (N, \nu)$ consists of a set of players $N = \{1, \ldots, n\}$ and a characteristic function $\nu: 2^N \to \mathbb{R}$ such that $\nu(\emptyset) = 0$. Intuitively, for each coalition N' of players $(N' \subseteq N), \nu(N')$ is the joint payoff that the players in N'receive for working together. We refer to subsets of players as coalitions. Throughout this paper we consider monotone games only, i.e., games where for each two coalitions N' and N'' such that $N' \subseteq N''$, it holds that $\nu(N') \leq \nu(N'')$.

There are many solution concepts in cooperative game theory that describe which coalitions may form and/or how to distribute the coalitions' payoffs among their members (see, e.g., the overview of Chalkiadakis et al. [7] for a computer science perspective on the theory of cooperative games). Among the solution concepts, we focus on the Banzhaf value [3, 10], as the basic notion which turns out to be useful in our context.

DEFINITION 2. Let $G = (N, \nu)$ be a cooperative game. The Banzhaf value of player $i \in N$ is defined as follows:

$$B_G(i) = \frac{1}{2^{|N-\{i\}|}} \sum_{S \subseteq N-\{i\}} \left(\nu(S \cup \{i\}) - \nu(S) \right).$$

In other words, the Banzhaf value of player i is its marginal contribution to a randomly selected coalition. Intuitively, we can view it as the player's importance: the higher the Banzhaf value, the more useful the player is for a (random) coalition.

3. COMMITTEE SCORING RULES AND BANZHAF VALUES

One of the contributions of this paper is in the following connection between multiwinner elections (committee scoring rules in particular) and cooperative games (Banzhaf values in particular). Next, we define a class of cooperative games associated with multiwinner elections and committee scoring rules.

DEFINITION 3. Let E = (C, V) be an election with $C = (c_1, \ldots, c_m)$ and let \mathcal{R}_f be a committee scoring rule defined through scoring functions $f = (f_{m,k})_{k \leq m}$. We define the game $G(E, \mathcal{R}_f) = (C, \nu)$, associated with an election E and an election rule \mathcal{R} so that for each coalition S of candidates (players) we have:

$$\nu(S) = \begin{cases} f\operatorname{-score}_E(S) & , \text{ if } S \neq \emptyset, \\ 0 & , \text{ otherwise.} \end{cases}$$

The above definition requires some explanations and comments. First, the most important aspect of this definition is that the candidates are the players. The payoff of a coalition S is simply the score that this coalition—interpreted as a committee—would obtain in the underlying election. Thus, we use the terms *committee* and *coalition* interchangeably, depending on the context. Second, the characteristic function encompasses the committee scoring functions for all committee sizes between 1 and m. While it may seem somewhat strange at first, we view it as a natural approach (for the cases where we want to focus on particular election sizes only, we simply limit ourselves to coalitions of this size).

3.1 Computing Banzhaf Values

Let us consider the Banzhaf value of the game G associated with election E = (C, V), where $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$, and with the committee scoring rule \mathcal{R}_f , where $f = (f_{m,k})_{k \leq m}$. Intuitively, the Banzhaf value of a candidate measures the importance of this candidate in the given election.

Our first goal is to show that, for decomposable committee scoring rules, computing the Banzhaf values of all candidates can be done in polynomial-time. Our polynomialtime algorithm builds on the following two observations: (1) the Banzhaf value of a candidate can be computed for each voter separately (this follows since committee scoring rules treat each voter separately), and (2) the Banzhaf value of a candidate with respect to a given voter depends only on the position of this candidate (and on the positions of committee members already fixed to be present; see below); this follows since committee scoring rules depend only on the positions of the committee members within each voter's preference order.

Before we formally describe our polynomial time algorithm, we need some definitions and observations. Later we will need a variant of the Banzhaf value that considers committees of a given size only, such that some committee members are fixed, so we define the following variant of the Banzhaf value. We let k be a committee size, c_i be the candidate we are interested in, and W be a coalition of size smaller than k (which does not include c_i):

$$B_G(c_i, k, W) = \sum_{S \subseteq C: W \subseteq S, |S|=k-1} \nu(S \cup \{c_i\}) - \nu(S).$$

Note that:

$$B_G(c_i) = \frac{1}{2^{m-1}} \sum_{k=1}^m B_G(c_i, k, \emptyset),$$

so it suffices to focus on computing the values $B(c_i, k, W)$.

Next we show observation (1), which says that instead of considering the whole election E, it suffices to focus on each vote separately. We write $G(v_j)$ to denote the game G where the voter set is restricted to v_j only. Specifically, for each candidate c_i it holds that:

$$B_G(c_i, k, W) = \sum_{j=1}^{n} B_{G(v_j)}(c_i, k, W).$$

Corresponding to observation (2), now we prove the following technical lemma.

LEMMA 1. Let $f = (f_{m,k})_{k \leq m}$ be a family of decomposable committee scoring rules (defined through polynomialtime computable single-winner scoring functions), let E = (C, V) be an election, let k be the committee size, and let $G = (\mathcal{R}_f, E)$ be the game associated with \mathcal{R}_f and E. Then, for each voter v in V, each candidate $c \in C$, and each set W such that $W \subseteq C - \{c\}$ and |W| < k, the value $B_{G(v)}(c, k, W)$ can be computed in polynomial time.

PROOF. We set m = |C|. Let $\gamma_{m,k}^{(1)}, \ldots, \gamma_{m,k}^{(t)}$ be the polynomial-time computable single-winner scoring functions such that:

$$f_{m,k}(i_1,\ldots,i_k) = \gamma_{m,k}^{(1)}(i_1) + \cdots + \gamma_{m,k}^{(k)}(i_k).$$

Let us rename the candidate set to $C = \{a_1, \ldots, a_x, c, b_1, \ldots, b_y\}$, such that voter v has the following preference order:

$$v: a_1 \succ \cdots \succ a_x \succ c \succ b_1 \succ \cdots \succ b_y.$$

Then, we partition W into two sets, W_A and W_B , such that v ranks all the candidates in W_A before c and all the candidates in W_B after c. Let v be the characteristic function associated with our game G(v). Our goal is to compute the following quantity:

$$B_{G(v)}(c,k,W) = \sum_{S \subseteq C : |W \subseteq S, |S| = k-1} \nu(S \cup \{c\}) - \nu(S).$$
(1)

To this end, for each candidate $d \in C - \{c\}$ and each integer $t \in [k]$, we write C(d, t) to denote the set of coalitions S such that: (a) $W \subseteq S$; (b) |S| = k - 1; (c) $d \in S$; and (d) voter v ranks d as his or her t'th most desirable member of S. We define r(d) to be 0 if v ranks d ahead of c and define it to be 1 otherwise. Further, we define:

$$\Delta(d) = \sum_{t=1}^{\kappa} \sum_{S \in \mathcal{C}(d,t)} \left(\gamma_{m,k}^{(t)}(\mathrm{pos}_{v}(d)) - \gamma_{m,k-1}^{(t+r(d))}(\mathrm{pos}_{v}(d)) \right)$$
$$= \sum_{t=1}^{k} |\mathcal{C}(d,t)| \cdot \left(\gamma_{m,k}^{(t)}(\mathrm{pos}_{v}(d)) - \gamma_{m,k-1}^{(t+r(d))}(\mathrm{pos}_{v}(d)) \right).$$

Intuitively, $\Delta(d)$ is the contribution of candidate d to the sum in Equation (1).

We define $\Delta(c)$ in a similar (but not identical) way. That is, for each $t \in [k]$, we let $\mathcal{C}(c,t)$ be the set of coalitions Ssuch that $W \subset S$, |S| = k - 1 and voter v ranks c as his or her t'th-best among the candidates in $S \cup \{c\}$. Further, we set:

$$\Delta(c) = \sum_{t=1}^{k} \sum_{S \in \mathcal{C}(c,t)} \gamma_{m,k}^{(t)}(\text{pos}_{v}(c))$$
$$= \sum_{t=1}^{k} |\mathcal{C}(c,t)| \cdot \gamma_{m,k}^{(t)}(\text{pos}_{v}(c))$$

As in the case of $\Delta(d)$, $\Delta(c)$ is the contribution of c to the sum in Equation (1). We conclude the following:

$$B_{G(v)}(c,k,W) = \Delta(c) + \sum_{d \in C - \{c\}} \Delta(d).$$

To complete the proof, it suffices to note that, for each candidate $e \in C$ and each $t \in [k]$, the value $|\mathcal{C}(e,t)|$ can be computed in polynomial time. For example, for $t > |W_A|$ we have the following:

$$|\mathcal{C}(c,t)| = \begin{pmatrix} pos_v(c) - 1 - |W_A| \\ t - 1 - |W_A| \end{pmatrix} \cdot \begin{pmatrix} m - pos_v(c) - |W_B| \\ t - k - |W_B| \end{pmatrix}$$

The idea behind the formula above is as follows. For c to be ranked on the t'th position among the candidates in $S \cup \{c\}$, S has to contain exactly t-1 candidates that v ranks ahead of c. S has to contain all members of W so it contains the $|W_A|$ members of W ranked ahead of t, and it suffices to add the missing $t-1-|W_A|$ in an arbitrary way (altogether there are $pos_v(c) - 1 - |W_A|$ candidates that do not belong to W and that v ranks ahead of c). We calculate the number of ways in which we can choose members of S that v ranks after c analogously. \Box

By our preceding reasoning, Lemma 1 immediately implies the following.

THEOREM 2. For each decomposable committee scoring rule \mathcal{R}_f defined through polynomial-time computable singlewinner scoring functions, there is a polynomial-time algorithm that computes the Banzhaf value for each candidate in a given election.

We conclude this section with the following important remark. Let \mathcal{R}_f be some committee scoring rule and let E = (C, V) be an election. Notice that, for each candidate $c \in C$ and each voter $v \in V$, the Banzhaf value of c in the game $G(\mathcal{R}_f, v)$ depends only on the position of c in v. This means that we can define a single-winner scoring rule γ (for |C| candidates) so that $\gamma(i)$ is the Banzhaf value of the candidate ranked on the *i*'th position among the |C| candidates in an election with a single vote. Then, the Banzhaf value of candidate c in the game $G(\mathcal{R}_f, E)$ is simply the γ -score of c in election E.

In particular, the above remark implies that forming a committee by choosing k candidates in the order of their decreasing Banzhaf values (with respect to some initial committee scoring rule) means simply using a weakly separable committee scoring rule (albeit, based on a fairly complicated single-winner scoring function).

Another consequence of the above remark is that for every committee scoring rule \mathcal{R}_f based on a family of committee scoring functions with values bounded by functions exponential in the number of candidates, the problem of computing $B_{G(\mathcal{R}_f,E)}(c)$ (for some election E and a candidate c) belongs to the complexity class P/poly (see, e.g., the book of Hemaspaandra and Ogihara [17] for an extensive catalog of complexity classes). Intuitively, the class P/poly contains those problems for which, given an instance I and a value h(|I|) (where |I| is the length of the encoding of I and h is some, not necessarily computable, function whose output is polynomially bounded in |I| it is possible to solve the problem in polynomial time. In our case, the value of h(I) would consist of the description of functions γ from the preceding two paragraphs. The consequence is that, under standard complexity-theoretic assumptions, the problem of computing $B_{G(\mathcal{R}_f,E)}(c)$ cannot be NP-hard (however this does not apply to the more general problem from Lemma 1).

4. BANZHAF VALUES AND APPROXIMA-TION ALGORITHMS

In this section we show how to use the ideas concerning Banzhaf values, as discussed above, in order to design good heuristic algorithms for computing winning committees under decomposable committee scoring rules (indeed, winner determination under such rules is typically NP-hard).

Lu and Boutilier [19] introduced a greedy algorithm for computing committees of a given size with score close to the optimal one (they did it for the β -CC rule, and later other authors applied the algorithm to further rules, with Faliszewski et al. [13] providing the most general application). Let E = (C, V) be the input election, let k be the committee size, and let \mathcal{R}_f be the committee scoring rule to use (defined by committee scoring functions $f = (f_{m,k})_{k \leq m}$). The algorithm starts with an empty committee S and then executes k iterations. In the *i*'th iteration, it selects a candidate $c \notin S$ that maximizes the value f-score $_E(S \cup \{c\})$.

Faliszewski et al. [13] invoke the classic result of Nemhauser et al. [23] to argue that this greedy algorithm outputs a committee whose score is at least a (1 - 1/e)fraction of the optimal one, provided that the underlying scoring rule is decomposable and based on single-winner scoring functions $\gamma = (\gamma_{m,k}^{(t)})_{1 \le t \le k \le m}$ such that $\gamma_{m,k}^{(t)}(i) \ge$ $\gamma_{m,k}^{(t+1)}(i)$ for each $m \in \mathbb{N}$ $(m > 0), k \in [m], t \in [k - 1],$ and $i \in [m]$. In particular, this applies both to β -CC and α_k -PAV.

Unfortunately, while this greedy algorithm has reasonably good approximation guarantee and it usually performs quite well in practice (in terms of its approximability [19, 28]), it has one serious drawback. Namely, it selects committees that are in some sense quite biased. For example, for β -CC, the greedy algorithm always starts by selecting the Borda winner, whereas for α_k -PAV it always starts by selecting a candidate with the highest k-Approval score.

We propose to rectify this issue by using a "non-myopic" variant of the algorithm. The only difference is that instead of selecting in the *i*'th iteration a candidate that maximizes the marginal increase of the given (partial) committee, we select a candidate with the highest Banzhaf value; specifically, in the *i*'th iteration, we compute Banzhaf values focusing on committees of size k, but under the assumption that the i - 1 committee members from the previous iterations

```
Notation: f = (f_{m,k})_{k \le m} \leftarrow \text{committee scoring}

functions.

E = (C, V) \leftarrow \text{input election.}

k \leftarrow \text{committee size.}

G \leftarrow G(\mathcal{R}_f, E)

S \leftarrow \emptyset

A \leftarrow C

for i \leftarrow 1 to K do

\begin{vmatrix} c \leftarrow \operatorname{argmax}_{a \in A} B_G(a, k, S) \\ S \leftarrow S \cup \{c\} \\ A \leftarrow S \setminus \{c\} \end{vmatrix}

return S
```

Algorithm 1: Banzhaf-based approximation algorithm.

are already belong to the coalition. The exact pseudocode is given as Algorithm 1.

As hinted above, and formally stated below, for many decomposable committee scoring rules (including all rules described in this paper), Algorithm 1 runs in polynomial time.

THEOREM 3. For a decomposable committee scoring rule \mathcal{R}_f defined by a family of polynomial-time computable singlewinner scoring functions $\gamma = (\gamma_{m,k}^{(t)})_{1 \leq t \leq k \leq m}$, Algorithm 1 is polynomial-time computable.

Proof. Follows by Lemma 1 and a simple analysis of the algorithm. $\hfill\square$

5. EXPERIMENTAL ANALYSIS

In this section we report on experiments we performed in order to assess the approximation quality of Algorithm 1. We tested our Banzhaf-based approximation algorithms for β -CC (we intend to extend the experiments to other rules; the current manuscript is a preliminary presentation of our ideas). We performed two experiments; in the first we investigated biases of the algorithm, whereas in the second one we compared the quality of the generated committees.

Histograms. In the first experiment, building on the work of Elkind et al. [11], we computed histograms which visually demonstrate the behavior of our Banzhaf-based approximation algorithm, and compared it to the greedy algorithm of Lu and Boutilier [19] (we note that the histogram for this algorithm was already presented in the full version of the paper of Elkind et al. [11]).

Specifically, we generated elections from the twodimensional Euclidean domain, where both the candidates and the voters are drawn uniformly at random from a uniform square (each candidate and each voter is a point; a voter ranks the candidates by sorting them in increasing order of the Euclidean distances). We generated 100 candidates and 100 voters for each election, and computed winning committees with 10 committee members each. We generated one histogram for the actual β -CC rule (using an ILP solver to find the optimal solutions), another histogram for the greedy approximation algorithm of Lu and Boutilier [19], and another histogram for our Banzhaf-based approximation algorithm. Each histogram was created by aggregating 10000 elections.



Figure 1: Comparison of the histograms for β -CC using the greedy algorithm (on the left), the Banzhaf-based one (in the center), and the optimal ILP-based one (on the right). The histograms were computed for 10000 elections, with 100 candidates and 100 voters each, for committee size k = 10.

Figure 1 shows the results of the first experiment. It is quite visible that, while the greedy algorithm of Lu and Boutilier [19] has a bias towards the center of the histogram (as explained above), the Banzhaf-based algorithm does not suffer from this problem. Further, the histogram of the Banzhaf-based algorithm looks very similarly to the histogram showing the optimal ILP-based algorithm for β -CC (the rangingCC algorithm, described below, also gets a histogram very similar to that for β -CC [11]).

Positions of the Representatives. In the second experiment, we checked the average position (in voters' preference orders) of the representatives chosen by various approximation rules. This is a more direct way of assessing the approximation quality of the algorithms, since β -CC minimizes this value. Indeed, β -CC maximizes the sum of the Borda scores of the representatives; we believe, however, that it is more useful to consider this "reversed" measure. There are two reason for this. First, measuring the direct approximation ratio suggests that all the algorithms have near-perfect performance (e.g., we have to choose between 0.98 and 0.99 approximation ratios; this is clearly possible but inconvenient). Second, the average position of the representative is a very intuitive measure from the point of view of the voters: A voter can more easily interpret information that "on the average he or she will be represented by someone he or she ranks as third best" than that "he or she will be represented by someone he or she prefers to 97 candidates, on average" (in particular, the former does not require the voter to know how many candidates there were in the election).

We considered two distributions of voters' preference orders. The first distribution is the one used in the first experiment (we generated candidates and voters as points on a square, uniformly at random, and the preference order of each voter is formed by sorting the candidates with respect to the distance from the voter). The second one followed the impartial culture assumption (each voter chose his or her preference order uniformly at random). For the first distribution we created 1000 elections, each with 100 voters and 100 candidates, and we varied the committee size k to be any integer between 2 and 30; we checked the average position of each voter's representative for β -CC (computed using an ILP solver), rangingCC,⁴ the greedy approximation

⁴RangingCC is a variant of Algorithm P [28], improved by Elkind et al. [11]. The algorithm proceeds as follows: Given an election E = (C, V) and a committee size k, it considers

algorithm, and our Banzhaf-based approximation algorithm. For the second distribution we generated only 250 elections for each committee size between 2 and 30 (with step 2), but otherwise the experiment was analogous (the reason for this restriction was that we ran out of time for our computations; we plan to have 1000 elections per data point for the final version of this paper).

Figure 2 shows the results of the second experiment. The results are normalized to β -CC, i.e., the figure shows the ratio between the average position of a representative under a given algorithm and the optimal average position (thus, the values are always greater or equal to 1). While rangingCC performs very well for small committee sizes, both the greedy algorithm and our Banzhaf-based approximation algorithm perform much better as the committee size increases. Further, our Banzhaf-based approximation algorithm consistently outperforms the greedy algorithm.

Running Times. In the analysis above we have disregarded the running times of our algorithms. Indeed, both rangingCC and greedyCC can be significantly faster than the Banzhaf-based algorithm (by an order of magnitude in our experiments)⁵. Thus, one might say that the Banzhafbased heuristic has unfair advantage over the greedy algorithm and, in particular, one might consider a greedy algorithm that picks two candidates in each iteration instead of one. Our very preliminary experiments suggest that this does not improve the performance of the algorithm, but in our setting—increases the running time by two orders of magnitude as compared to the classic greedy algorithm (resulting in, altogether, a 10 times slower algorithm than the Banzhaf-based one, with worse quality of results).

6. OUTLOOK

We considered multiwinner elections (held using committee scoring rules) as cooperative games and, building on this idea, were able to design improved approximation algorithms for winner determination for a rich class of multiwinner voting rules. We provided some preliminary experiments to assess the quality of our algorithms. While the experiments showed that in some cases the quality of approximation of these new approximation algorithms is quite good, more experiments are needed to fully understand when these algorithms are most useful. Specifically, one might consider further election distributions as well as real-world elections.

In this paper we concentrated on the Banzhaf value, as a fundamental solution concept in cooperative game theory. It might be interesting to consider other solution concepts,

⁵However, we should mention that we used a highlyoptimized variant of our algorithm. In particular, our algorithm never recomputed already-used values of binomial coefficients and used formulas from Lemma 1 optimized for β -CC, to not compute values that have to add up to zero.



(a) 2D model: uniform distribution on a square



(b) Impartial culture

Figure 2: Comparison of the average representative position for β -CC for two distributions: uniform square Euclidean domain (on the top) and impartial culture (on the bottom); for three algorithms: rangingCC, greedyCC, and BanzhafCC (our Banzhafbased algorithm). The values are computed for elections, with 100 candidates and 100 voters each (1000 elections per data point). For each committee size we present the ratio between the average position of a voter's representative under the given algorithm and under β -CC (computed using an ILP solver).

such as the Shapley value; guiding greedy algorithms by solution concepts other than the Banzhaf value might lead to efficient algorithms with a better quality of approximation.

Finally, in this paper we concentrated on committee scoring rules, especially decomposable committee scoring rules. While the subclass of decomposable committee scoring rules is quite rich, it is natural to wonder whether the ideas presented here can be useful for other multiwinner voting rules.

Acknowledgments Edith Elkind, Martin Lackner, and Dominik Peters were supported by the ERC grant 639945 (ACCORD). Piotr Faliszewski was supported by the National Science Centre, Poland, under project 2016/21/B/ST6/01509.

each threshold value $t \in [m]$; for a given threshold value t, it greedily finds a committee such that as many voters as possible rank some committee member among the top t positions (i.e., for a given value t, it first adds to the committee the candidate ranked among the top t positions by most voters, removes these voters from consideration, and repeats the process until k candidates are selected). Then, for each computed committee (one computed committee for each value of $t \in [m]$) it computes its β -CC score; then, it outputs the committee with the highest score. This algorithm is the basis of a PTAS for β -CC and has the highest theoretically-established approximation guarantee for this rule.

REFERENCES

- H. Aziz, M. Brill, V. Conitzer, E. Elkind, R. Freeman, and T. Walsh. Justified representation in approval-based committee voting. *Social Choice and Welfare*, 48(2):461–485, 2017.
- [2] H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, and T. Walsh. Computational aspects of multi-winner approval voting. In *Proceedings of the* 14th International Conference on Autonomous Agents and Multiagent Systems, pages 107–115, 2015.
- [3] J. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19:317–343, 1965.
- [4] S. Barberà and D. Coelho. How to choose a non-controversial list with k names. Social Choice and Welfare, 31(1):79–96, 2008.
- [5] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. *Journal of Artificial Intelligence Research*, 47:475–519, 2013.
- [6] M. Brill, J. Laslier, and P. Skowron. Multiwinner approval rules as apportionment methods. In Proceedings of the 31st AAAI Conference on Artificial Intelligence, pages 414–420, 2017.
- [7] G. Chalkiadakis, E. Elkind, and M. Wooldridge. Computational Aspects of Cooperative Game Theory. Morgan & Claypool, 2012.
- [8] B. Chamberlin and P. Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review*, 77(3):718–733, 1983.
- B. Debord. An axiomatic characterization of Borda's k-choice function. Social Choice and Welfare, 9(4):337–343, 1992.
- [10] P. Dubey and L. Shapley. Mathematical properties of the Banzhaf power index. *Mathematics of Operations Research*, 4(2):99–131, May 1979.
- [11] E. Elkind, P. Faliszewski, J. Laslier, P. Skowron, A. Slinko, and N. Talmon. What do multiwinner voting rules do? An experiment over the two-dimensional euclidean domain. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 494-501, 2017. Full version available at http://home.agh.edu.pl/~faliszew/2d.pdf.
- [12] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of multiwinner voting rules. *Social Choice and Welfare*, 48(3):599–632, 2017.
- [13] P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. Committee scoring rules: Axiomatic classification and hierarchy. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 250–256, 2016.
- [14] P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. Multiwinner analogues of the plurality rule: Axiomatic and algorithmic views. In *Proceedings of* the 30th AAAI Conference on Artificial Intelligence, pages 482–488, 2016.
- [15] P. Faliszewski, A. Slinko, K. Stahl, and N. Talmon. Achieving fully proportional representation by clustering voters. In *Proceedings of the 2016*

International Conference on Autonomous Agents & Multiagent Systems, pages 296–304, 2016.

- [16] W. Gehrlein. The Condorcet criterion and committee selection. *Mathematical Social Sciences*, 10(3):199–209, 1985.
- [17] L. Hemaspaandra and M. Ogihara. The Complexity Theory Companion. Springer-Verlag, 2002.
- [18] M. Kilgour. Approval balloting for multi-winner elections. In *Handbook on Approval Voting*. Springer, 2010. Chapter 6.
- [19] T. Lu and C. Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Proceedings of the 22nd International Joint Conference* on Artificial Intelligence, pages 280–286, 2011.
- [20] T. Lu and C. Boutilier. Value-directed compression of large-scale assignment problems. In *Proceedings of the* 29th AAAI Conference on Artificial Intelligence, pages 1182–1190, 2015.
- [21] T. Michalak, K. Aadithya, P. Szczepanski, B. Ravindran, and N. Jennings. Efficient computation of the shapley value for game-theoretic network centrality. *Journal of Artificial Intelligence Research*, 46:607–650, 2013.
- [22] B. Monroe. Fully proportional representation. American Political Science Review, 89(4):925–940, 1995.
- [23] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, December 1978.
- [24] A. Procaccia, J. Rosenschein, and A. Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.
- [25] T. Ratliff. Some startling inconsistencies when electing committees. Social Choice and Welfare, 21(3):433–454, 2003.
- [26] L. Sánchez-Fernández, E. Elkind, M. Lackner, N. Fernández, J. A. Fisteus, P. Basanta Val, and P. Skowron. Proportional justified representation. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 670–676, 2017.
- [27] P. Skowron, P. Faliszewski, and J. Lang. Finding a collective set of items: From proportional multirepresentation to group recommendation. *Artificial Intelligence*, 241:191–216, 2016.
- [28] P. Skowron, P. Faliszewski, and A. Slinko. Achieving fully proportional representation: Approximability result. Artificial Intelligence, 222:67–103, 2015.

Optimal Decision Making with CP-nets and PCP-nets

Sujoy Sikdar Rensselaer Polytechnic Inst. Dept. of Computer Science sikdas@rpi.edu Sibel Adalı Rensselaer Polytechnic Inst. Dept. of Computer Science sibel@cs.rpi.edu

Lirong Xia Rensselaer Polytechnic Inst. Dept. of Computer Science xial@cs.rpi.edu

ABSTRACT

Probabilistic conditional preference networks (PCP-nets) are a generalization of CP-nets for compactly representing preferences over multi-attribute domains. We introduce the notion of a loss function whose inputs are a CP-net and an outcome. We focus on the optimal decision-making problem for acyclic and cyclic CP-nets and PCP-nets. Our motivations are three-fold: (1) our framework naturally extends to allow reasoning on cyclic CP-nets and PCPnets for full generality, (2) in the multi-agent setting, we place no restriction on agents' preferences structure and voting rules under our framework have desirable axiomatic properties, (3) we generalize several previous approaches to finding the optimum outcome in individual and multi-agent contexts. We characterize the computational complexity of computing the loss of a given outcome and computing the outcomes with minimum loss for three natural loss functions: 0-1 loss, neighborhood loss, and global loss. While the optimal decision is NP-hard to compute for many cases, we give a polynomial-time algorithm for computing the optimal decision for tree-structured PCP-nets and profiles of CP-net preferences with a shared dependency structure, w.r.t. neighborhood loss function.

1. INTRODUCTION

Many decision-making problems involve choosing an optimal outcome from a *multi-attribute domain* where the alternatives are characterized by $p \ge 1$ variables and each variable corresponds to an attribute of the outcome. In combinatorial voting there are p issues, and the alternatives correspond to the decisions made on each issue. For example, a dinner menu can be characterized by two variables: the main dish **M** and the wine **W**. The main dish can be either beef (M_b) or fish (M_f) and the wine can be either white wine (W_w) or red wine (W_r) . We want to make an optimal (joint) decision for an agent or a group of agents with preferences over the alternatives. However, since the number of outcomes in a multi-attribute domain is exponentially large, it is impractical for the agents to express preferences as a full ranking over all outcomes.

A popular practical solution is to use a compact preference language to represent agents' preferences. Perhaps the most commonly used language for agents to represent their preferences over multi-attribute domains are CP-nets (conditional preference networks) [2]. In a CP-net, an agent can specify her local preferences over any attribute given the values of some other attributes (called its *parents*). Such preferences can arise from, and be decomposed into *ceteris paribus* statements of the form: "I prefer red wine to white wine, ceteris paribus, given that meat is served as the main dish." The dependency graph of a CP-net is a directed graph where the vertices are the variables and each variable has incoming edges from its parents.

For a single agent whose preferences are represented by a CPnet, a natural optimization objective is to identify *undominated* outcomes [3]. Informally, an outcome is undominated if no other outcome is preferred over it. The problem of computing undominated outcomes is well studied in the CP-net literature. For acyclic CPnets (CP-nets with acyclic dependency graphs), an undominated outcome always exists and is unique [2]. However, when we allow cyclic dependencies, undominated outcomes can be hard to compute [3, 9].

Recently, probabilistic conditional preference networks (PCPnets) have been introduced as a natural generalization of CP-nets [1, 7]. In a PCP-net, for any variable X and any valuation of its parents, there is a probability distribution over all rankings over X's value domain. A PCP-net can be used to represent a single agent's uncertain preferences over a set of CP-nets, or a preference profile of multiple CP-nets [8]. Given an acyclic PCP-net, [7] provides a polynomial-time algorithm for computing the outcome that is undominated with the highest probability. Despite this promising first step in decision making with PCP-nets, the optimal decision making problem for PCP-nets remains largely open. In particular, is there any other sensible and more quantitative optimality criterion beyond "being undominated" that we may consider for CP-nets as well as PCP-nets? If so, how can we compute them?

In the combinatorial voting setting, we are given a *profile*, a collection of multiple agents' individual CP-net preferences or *votes*. Several approaches [11, 20, 18, 14, 19, 15, 5, 12] have been proposed to aggregate preferences in this setting by extending standard voting rules and axiomatic properties. Additionally, [8] represents the profile with a single PCP-net, and [17] proposes mCP-nets to deal with partial CP-nets where agents may have preference over only a subset of the issues. However, much of the existing work focuses on certain special cases with rather severe restrictions on agents' preferences such as allowing only profiles with acyclic CP-nets, and dependencies that are compatible with a common order on the issues (*O*-legality). We design a new class of voting rules characterized by a loss function which takes as input any profile of CP-net preferences and outputs a set of loss minimizing outcomes.

1.1 Our Contributions

We take a decision-theoretic approach by modeling the optimality of an outcome by a *loss function*, whose inputs are an outcome (an assignment of values to attributes) and a single (acyclic

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.
Table 1: Complexity of *L*-LOSS w.r.t. acyclic and cyclic CP-nets. The complexity remains unchanged for the case of acyclic and cyclic PCP-nets.

Loss fn.	Acyclic	Cyclic
L_{0-1}	D (trivial)	P (Prop. 1)
L_N	r (urviar)	P (Prop. 2)
L_G	coNP-hard (Thm. 2)	coNP-hard

Table 2: Complexity of *L*-OPTDECISION w.r.t. acyclic and cyclic CP-nets and PCP-nets.

Loss fn.	Loss fn. Acyclic Cyclic	
L_{0-1} L_N	P [2]	NP-complete (Prop. 3)
L_G	- [-]	P (Prop. 4)

(a)	CP-ne	ets
·/		

Loss fn.	Acyclic	Cyclic		
L_{0-1}	NP-complete, P for trees [7]	NP-complete [7]		
L_N	NP-hard (Thm. 3), P for trees (Thm. 4)	NP-hard (Thm. 3)		
L_G	coNP-hard (Thm. 5)			

(b) PCP-nets

or cyclic) CP-net. In this paper we focus on multi-attribute domains where all variables are *binary* (although we emphasize that all our results also apply to multi-valued variables), and the following three natural loss functions for an outcome \vec{d} and a CP-net C.

- 0-1 loss function (L_{0−1}): the loss is 1 if d is dominated in C, and is 0 otherwise. This loss function corresponds to the most probable optimal outcome studied by [7].
- 2. Neighborhood loss (L_N) : the loss is the number of *neighbors* that dominate \vec{d} . A neighbor of \vec{d} differs from \vec{d} on only one attribute. This loss function corresponds to *local Condorcet* winner [5].
- 3. Global loss $(L_{\vec{G}})$: the loss is the total number of outcomes that dominate \vec{d} .

These loss functions can be naturally extended to evaluate the loss of an outcome in PCP-nets and profiles of CP-nets. We then consider the problem of computing an optimal decision in a loss minimization framework.

Given a loss function L, an outcome \vec{d} , a number k, and a CPnet (or PCP-net) C, in the **L**-LOSS problem we are asked whether the loss of \vec{d} in C is no more than k. Given a loss function L, a number k, and a CP-net (or PCP-net) C, in the **L**-OPTDECISION problem we are asked whether there exists an outcome \vec{d} whose loss is no more than k. Given a loss function L, a number k, and a profile P of CP-nets, in the **L**-OPTJOINTDECISION problem we are asked whether there exists an outcome \vec{d} whose loss for the entire profile P is no more than k. The results for L-LOSS are summarized in Table 1. Our main results on the problems L-OPTDECISION, and L-OPTJOINTDECISION are shown in Table 2 and Table 3 respectively.

One might be tempted to believe that PCP-nets are so complicated that all problems are hard to compute. This is not true. As we can see in Table 1, computing LOSS w.r.t. L_{0-1} and L_N can be done in polynomial time for PCP-nets. Another false belief could be that for the same loss function, LOSS is easier than OPTDE- Table 3: Complexity of *L*-OPTJOINTDECISION w.r.t. profiles of acyclic and cyclic CP-nets.

Loss fn.	Acyclic	Cyclic	
L_{0-1}	P (Thm. 6)	NP-complete (Thm. 6)	
L_N	NP-complete	e (Thm. 6),	
	P for shared tree-structured dependency graph.		
	(Thm. 7)		
L_G		coNP-hard (Thm. 8)	

CISION (or vice versa). Neither is true by comparing Table 2(a) and Table 1. L_G -LOSS is coNP-hard but L_G -OPTDECISION is in P for acyclic CP-nets. L_N -LOSS is in P but L_N -OPTDECISION is NP-complete for cyclic CP-nets. While it is hard to compute the optimal outcomes w.r.t. all three loss functions (Table 2), for tree-structured PCP-nets, we have a polynomial-time algorithm to compute the optimal outcome (Theorem 4). Similarly, while it is, hard to compute the optimal outcomes w.r.t. L_{0-1} for acyclic PCP-nets, a simple polynomial time algorithm allows us to compute the optimal outcome for a profile of acyclic CP-nets.

Finally, we show that every voting rule under our framework satisfies anonymity, category-wise neutrality, consistency and weak monotonicity.

1.2 RELATED WORK AND DISCUSSIONS

Since PCP-nets can be used to represent the preferences of a group of agents, our loss-minimization framework can naturally be used as a solution to group decision-making as done by [7] for L_{0-1} . However, among all three loss functions considered in this paper, only L_{0-1} has been studied for PCP-nets. All our computational results about L_N and L_G for PCP-nets are new.

Our loss-minimization framework is also related to other recent research agenda in aggregating CP-nets in multi-attribute domains [17, 11, 20, 18, 13, 14, 19, 15, 5, 6, 12, 4]. The main challenge is in the case where agents' preferences are represented by cyclic CP-nets, or there does not exist a common ordering over attributes that is compatible with all agents' CP-nets. In these cases even the optimality of an outcome is not clear. We handle cyclic CP-nets differently by introducing loss functions that work for cyclic CP-nets and PCP-nets. At a high level, our approach is similar to the idea of applying a positional scoring rule to profiles of *LP-trees* [12]. The difference is that an LP-tree represents a linear order over a multi-attribute domain but CP-nets generally represent a partial order. Therefore, positional scoring rules are not directly applicable to profiles of CP-nets.

2. PRELIMINARIES

Let $I = \{X_1, ..., X_p\}$ be a finite set of p variables with finite domains $D(X_i)$. Let $\mathcal{L}(D(X_i))$ denote the set of all linear orders over $D(X_i)$. For ease of presentation, we will assume that all variables are binary in this paper. An assignment (or outcome) \vec{d} is a vector in $\prod_{i \leq p} D(X_i)$. We use either d_{X_i} or d_i to denote the value of X_i in \vec{d} , and d_{-i} to denote the values of all other variables. For any subset of variables $S \subseteq I$, we let $D(S) = \prod_{X_i \in S} D(X_i)$, and $D(-S) = \prod_{X_i \in I \setminus S} D(X_i)$. We use \vec{d}_S to denote the assignment to the variables in S.

DEFINITION 1. [2] A CP-net C over the set of variables I is given by two components (i) a directed graph G = (I, E) called the dependency graph, and (ii) for each variable X_i , there is a conditional preference table $CPT(X_i)$ that contains a linear order $\succ_{C,\vec{u}}^i$ over $D(X_i)$ for each valuation \vec{u} of the parents of X_i

(denoted $Pa(X_i)$) in G.

When G is (a)cyclic we say that C is a (a)cyclic CP-net.

The partial order \succ_C induced by a CP-net C over the set of all possible assignments $\prod_{i \leq p} D(X_i)$ is the transitive closure of $\{(a_i, \vec{u}, \vec{z}) \succ (b_i, \vec{u}, \vec{z})\} : i \leq p; a_i, b_i \in D(X_i); \vec{u} \in$ $D(Pa(X_i)); \vec{z} \in D(-(Pa(X_i) \cup \{X_i\}))\}$. A CP-net is said to be consistent if \succ_C is asymmetric. Acyclic CP-nets are consistent but cyclic CP-nets are not necessarily consistent.

DEFINITION 2 (WEAK AND STRICT DOMINANCE). An assignment \vec{a} weakly dominates \vec{b} if $\vec{a} \succ_C \vec{b}$. An assignment \vec{a} strictly dominates \vec{b} if $\vec{a} \succ_C \vec{b}$ and $\vec{b} \not\prec_C \vec{a}$.

Dominance relations can also be described by *improving flip dynamics* [2]. If $\vec{d'}$ differs from \vec{d} in the value of exactly one variable X_i (i.e. $d'_i \neq d_i, d'_{-i} = d_{-i}$) and $d'_i \succ^i_{C,\vec{u}} d_i$ where $\vec{u} = \vec{d}_{Pa(X_i)}$, then the change from \vec{d} to $\vec{d'}$ via changing the value of X_i is an *improving flip*, and $\vec{d} \prec_C \vec{d'}$. For any pair of assignments \vec{a}, \vec{b} where $\vec{a} \succ_C \vec{b}$, there exists a sequence of such improving flips starting from \vec{a} by which we obtain \vec{b} . If $\vec{a} \neq_C \vec{b}$, then there is no such sequence of improving flips from \vec{a} to \vec{b} . In the case of cyclic CP-nets, it is possible to simultaneously have $\vec{a} \succ_C \vec{b}$ and $\vec{b} \succ_C \vec{a}$ and have a corresponding sequence of improving flips in either direction.



Figure 1: A CP-net representing preferences for dinner consisting of a main dish (**M**) and wine (**W**). The available choices are: For the main course, either beef (M_b) or fish (M_f) , and for wine, either red wine (W_r) or white wine (W_w) .

EXAMPLE 1. Figure 1 shows an agent's preferences over dinner represented as a CP-net and its hypercube representation [5]. In the hypercube representation there is an edge between every pair of neighboring assignments representing the agent's preferences. For example, the edge $M_bW_r \rightarrow M_bW_w$ means that $M_bW_r \succ M_bW_w$, and that we can obtain M_bW_r from M_bW_w by an improving flip. Serving beef along with red wine (i.e. the assignment M_bW_r) is the optimal decision and it strictly dominates every other configuration.

DEFINITION 3. A PCP-net [1, 7] Q over the set of variables I is given by (i) a directed graph G = (I, E), and (ii) for each variable X_i , there is a probabilistic conditional preference table $PCPT(X_i)$ that contains a probability distribution $f_{Q,\vec{u}}^i$ over $\mathcal{L}(D(X_i))$ for each valuation \vec{u} of the parents of X_i in G.

A CP-net C with dependency graph G = (V, E') is compatible with a PCP-net Q with a dependency graph G = (V, E) if $E' \subseteq E$. Any PCP-net Q represents a probability distribution over all CPnets that are compatible with Q. For any CP-net C compatible with a PCP-net Q, the probability of C, denoted by $f_Q(C)$, is calculated by multiplying the probabilities of all local preferences in C by looking up corresponding entries in PCPTs in Q. Formally,

$$f_Q(C) = \prod_{X_i} \prod_{\vec{u} \in D(Pa_Q(X_i))} f_{Q,\vec{u}}^i(\succ_{C,\vec{u}}^i)$$

EXAMPLE 2. Figure 2 illustrates a PCP-net Q and a CP-net C that is compatible with Q. We have $f_Q(C) = 0.3 \times 0.6 \times 0.3$. The first 0.3 is the probability of $M_f \succ M_b$ in C; the 0.6 is the probability of $W_r \succ W_w$ given M_b in C; the last 0.3 is the probability of $W_r \succ W_w$ given M_b in C.



Figure 2: PCP-net Q and a CP-net C it induces.

A profile $P = (P_1, ..., P_n)$ or n agents' CP-net preferences over a set of variables I is a collection of CP-nets $P_i, 1 \le i \le n$ over I, one for each agent i representing her vote. A profile P is said to be *O-legal* if there is some linear order O over the variables Isuch that for every CP-net P_i , every variable X_i , it holds that if $X_j \in Pa(X_i)$, then $X_j \succ_O X_i$ i.e that every parent of X_i appears before X_i in O. A voting rule r is a function that takes as input a profile and outputs a set of outcomes.

2.1 Loss Functions

In this paper we will focus on three loss functions. Each loss function L takes a CP-net C and an assignment \vec{d} as inputs and outputs a real number $L(C, \vec{d})$.

DEFINITION 4. The 0-1 loss function is defined as $L_{0-1}(C, \vec{d}) = \begin{cases} 1 & \text{if there exists } \vec{d'} \text{ such that } \vec{d'} \succ_C \vec{d}, \\ 0 & \text{otherwise} \end{cases}$

That is, the 0-1 loss function takes the value 0 if and only if \vec{d} is not weakly dominated by any other assignment in C.

DEFINITION 5. The neighborhood loss function is defined as $L_N(C, \vec{d}) = |\{\vec{d'}: \exists i: d'_i \succ_C d_i \text{ and } d'_{-i} = d_{-i}\}|.$

That is, the neighborhood loss of \vec{d} in C is the number of \vec{d} 's neighbors that can be obtained by a single improving flip from \vec{d} in C.

DEFINITION 6. The global loss function is defined as $L_G(C, \vec{d}) = |\{\vec{d'}: \vec{d'} \succ_C \vec{d}, and \vec{d} \neq_C \vec{d'}|.$

That is, the global loss of \vec{d} in C is the total number of assignments that strictly dominate \vec{d} in C.

For example, in the CP-net of Figure 1, $M_f W_r$ has a neighborhood loss of 2, and a global loss of 3. $M_b W_r$ has a global loss of 0 because no assignment strictly dominates it.

All loss functions can be naturally extended to PCP-nets by computing the expected loss of a given assignment w.r.t. the distribution f_Q over CP-nets represented by the given PCP-net Q. Similarly, the loss functions extend to a profile of CP-net preferences by computing the sum total of the loss of a given assignment w.r.t. each of the CP-nets in the profile.

3. COMPUTING THE LOSS OF ASSIGN-MENTS

We now formally define the decision problem of computing the loss of an assignment w.r.t. a loss function.

DEFINITION 7 (L-LOSS). Given a PCP-net Q, a loss function L, a decision \vec{d} , and a number $k \in \mathbb{R}$, in L-LOSS we are asked to compute whether $L(Q, \vec{d}) \leq k$.

OBSERVATION 1. Because CP-nets are a special case of PCPnets, any hardness results for CP-nets immediately extend to the case of PCP-nets. Conversely, if a problem is easy for PCP-nets then it is also easy for CP-nets.

We find that L_{0-1} -LOSS and L_N -LOSS are easy for even cyclic PCP-nets. By our previous observation, this also extends to acyclic PCP-nets and both acyclic and cyclic CP-nets.

PROPOSITION 1. L_{0-1} -LOSS is in P for possibly cyclic PCPnets.

We note that given a cyclic PCP-net Q, the 0-1 loss of \vec{d} in a CP-net C that is compatible with Q is 1 if and only if \vec{d} is less preferred than one of its neighbors. Therefore, we have

$$L_{0-1}(Q, \vec{d}) = 1 - \prod_{i=1}^{p} f_{Q, d_{Pa(X_i)}}^i (d_i \succ \bar{d}_i),$$

where \bar{d}_i is the complement of d_i , $f^i_{Q,d_{Pa(X_i)}}$ is the $PCPT(X_i)$ given that the parents of X_i take their values as in \vec{d} .

PROPOSITION 2. L_N -LOSS is in P for possibly cyclic PCPnets.

PROOF. It is not hard to check that $L_N(Q, \vec{d}) = \sum_{i=1}^p f_{Q,d_{Pa}(X_i)}^i (\vec{d}_i \succ d_i).$

THEOREM 1. L_G -LOSS is PSPACE-complete for inconsistent, cyclic CP-nets.

PROOF. We show a reduction from the PSPACE-complete problem WEAKLY NON-DOMINATED OUTCOME [9], where we are given a CP-net C and an assignment \vec{d} , and we are asked whether \vec{d} is weakly non-dominated. An assignment is weakly nondominated if there is no $\vec{d'} \succ_C \vec{d}$. It follows from the definitions that \vec{d} is weakly non-dominated if and only if $L_G(C, \vec{d}) = 0$, and not weakly non-dominated if and only if $L_G(C, \vec{d}) \ge 1$. This corresponds to a reduction to L_G -LOSS where k = 0. \Box

THEOREM 2. L_G -LOSS is coNP-hard for acyclic CP-nets but in PSPACE.

PROOF. We give a polynomial time reduction from 3-SAT to the complement of L_G -LOSS, denoted by $\overline{L_G}$ -LOSS, which is defined as: Given a CP-net C, a decision \vec{d} , and a number $k \in \mathbb{R}$, is $L_G(C, \vec{d}) > k$. Our construction is inspired by the one used in [2] to prove the hardness of dominance testing in acyclic graphs. In an instance of 3-SAT we are given a Boolean formula $\mathcal{F} = C_1 \wedge ... \wedge C_n$ in 3-CNF over a set of Boolean variables $\{x_1, ..., x_m\}$. We are asked whether there exists a truth assignment to the variables such that \mathcal{F} is satisfied. We construct an instance of $\overline{L_G}$ -LOSS (see Figure 3), beginning with the construction of a CP-net C as follows:

• $I = \{V_1, \overline{V}_1, ..., V_m, \overline{V}_m\} \cup \{C_1, ..., C_n\} \cup \{D_0, D_1, ..., D_{2m+n}\}$ is a set of binary variables. Each V_i, \overline{V}_i corresponds to



Figure 3: The CP-net used in the proof of Theorem 2.

a Boolean variable x_i involved in the 3-SAT instance. Each C_i corresponds to a clause C_i .

• Let x_{i1}, x_{i2}, x_{i3} be the variables involved in the clause C_i . Then, (a) for all $V_i, \bar{V}_i \in I$, we let $Pa(V_i) = Pa(\bar{V}_i) = \emptyset$, (b) $Pa(C_i) = \{V_{i1}, \bar{V}_{i1}, V_{i2}, \bar{V}_{i2}, V_{i3}, \bar{V}_{i3}\}$, and importantly, (c) for all $2 \le i \le n$, $Pa(C_i) = Pa(C_i) \cup \{C_{i-1}\}$.

• For all $1 \le i \le 2m + n$, we let $Pa(D_0) = C_n$ and $Pa(D_i) = \{D_0\}$.

We populate the associated CP-tables as follows:

• The CPTs for all V_i , \overline{V}_i are $1 \succ 0$.

• For all C_i , we add the entry $1 \succ 0$ for every assignment to $Pa(C_i)$ where there exists a $k \leq 3$ such that all the following conditions are satisfied: (1) $V_{i_k \neq} \overline{V}_{i_k}$, (2) $V_{i_k} = 1$ if x_{i_k} is in clause j, OR $V_{i_k} = 0$ if $\neg x_{i_k}$ is in C_j , and (3) $C_{i-1} = 1$ if i > 1. Add entry $0 \succ 1$ for all remaining assignments.

• For D_0 , $1 \succ 0$ if $C_n = 1$, $0 \succ 1$ otherwise.

• For all $i \leq 2m + n$, we let the $CPT(D_i)$ be $1 \succ 0$ if $D_0 = 1$, and $0 \succ 1$ otherwise.

Finally, we let $\vec{d} = \vec{0}$ and $k = 2^{2m+n}$.

CLAIM 1. \mathcal{F} is satisfiable if and only if $L_G(C, \vec{0}) > 2^{2m+n}$.

PROOF. Intuitively, starting from $\vec{0}$, D_0 acts as a switch that can only be flipped on when the variables V_i , V_i are set in a way so that the corresponding assignment to x_i 's satisfies \mathcal{F} , and only when all the clause variables C_i have flipped (sequentially) to 1. Once D_0 flips to 1, the variables $D_{1 \le i \le 2m+n}$ may flip to 1 independently. Together, they account for a loss of 2^{2m+n} . The formal proof works as follows.

⇒ Let ϕ be an assignment that satisfies \mathcal{F} . Then, by construction, there exists a sequence of improving flips starting from $\vec{0}$ as follows: For i = 1, ..., m, if $\phi_i = 1$, flip V_i to 1, otherwise, flip \bar{V}_i to 1. By construction, we can flip C_1 to 1 and subsequently, each $C_2, ..., C_n$ to 1 in this order. This enables the flip of D_0 to 1, and enables $D_1, ..., D_{2m+n}$ to be flipped to 1 in any order. Together with the flip of D_0 to 1, and C_n to 1, there are at least 2^{2m+n} assignments that are preferred over $\vec{0}$.

 $\label{eq:suppose \mathcal{F} be unsatisfiable. For sake of contradiction, suppose that $\vec{0}$ has a global loss $L_G(C, $\vec{0}$) > 2^{2m+n}. There are at most $2^{2m+n} - 1$ assignments that involve changes in the values of $2m + n$ variables $\{V_i, V_i\}_{i \leq m} \cup $\{C_i\}_{i \leq n}$. For the inequality to hold there must be a sequence of improving flips to an assignment where a variable D_i has value 1. Then there must be a sequence S from $\vec{0}$ to an assignment d' where $D_0 = 1$, and $C_1, ..., C_n must have already been flipped to 1 along S in turn. Consider the construction of an assignment ϕ to the Boolean variables as follows. By construction, $\forall C_i, there must exist an assignment in S obtained by flipping C_i from 0 to 1$. When the flip occurs, there must exist some $j: V_j \neq V_j, $V_j, $V_j \in Pa(C_i)$. If $V_j = 1$, $V_j = 0$, $V_j = 1$, set $\phi_j = 0$. Simultaneously, clause C_i must be satisfied. Once any of the variables $V_i, V_i is set to 1 in the sequence, P and P and$

it can never flip back to 0 in S subsequently (doing so would not be an improving flip). There never exists a pair of assignments e, e' in S such that $V_i = 1, \overline{V}_i = 0$ in e but $V_i = 0, \overline{V}_i = 1$ in e'. Therefore, when each C_i is flipped to 1 in S, the values of the variables $V_j, \overline{V}_j \in Pa(C_i)$ are consistent with the assignment of the corresponding variables x_j in ϕ that satisfies clause C_i . If we can flip C_n to 1 in this way, then ϕ is a satisfying assignment. \Box

It is easy to see that the problem is in PSPACE. We conjecture that the problem is PSPACE-complete.

4. COMPUTING OPTIMAL DECISIONS FOR PCP-NETS

We define the decision problem of computing optimal assignments L-OPTDECISION as follows.

DEFINITION 8 (L-OPTDECISION). Given a PCP-net Q, a loss function L, and a number $k \in \mathbb{R}$, does there exist an assignment \vec{d} such that $L(Q, \vec{d}) \leq k$?

PROPOSITION 3. L_{0-1} -OPTDECISION and L_N -OPTDECISION are NP-complete for cyclic CP-nets.

PROOF. We give a reduction from the problem EXISTENCE OF NON-DOMINATED OUTCOME [9]. An outcome is nondominated if it uniquely belongs to a maximal dominance class (i.e. there is no way to improve from \vec{d} to any other assignment). It follows from the definition that an assignment \vec{d} is a non-dominated outcome w.r.t. a CP-net *C* if and only if $L_{0-1}(C, \vec{d}) = 0$ (equivalently, $L_N(C, \vec{d}) = 0$). The problem of deciding the existence of a non-dominated outcome reduces to the checking if there is a decision \vec{d} with $L_{0-1}(C, \vec{d}) = 0$ (equivalently, $L_N(C, \vec{d}) = 0$). \Box

PROPOSITION 4. L_G -OPTDECISION can be solved in constant time for cyclic CP-nets.

PROOF. For any CP-net C, a weakly non-dominated outcome d always exists such that $L_G(C, \vec{d}) = 0$. \Box

PROPOSITION 5. L_{0-1} -OPTDECISION is in P for PCP-nets Q with a tree structured dependency graph but NP-complete in general for acyclic dependency graphs.

PROOF. $\arg \min_{\vec{d}} L_{0-1}(Q, \vec{d}) = \arg \min_{\vec{d}} (1 - \prod_{i=1}^{n} f_{Q,d_{Pa(X_i)}}^i)(\vec{d}_i \succ d_i)) = 1 - \arg \max_{\vec{d}} (\prod_{i=1}^{n} f_{Q,d_{Pa(X_i)}}^i)(\vec{d}_i \succ d_i))$. This problem is equivalent to finding most probable explanation (MPE) for a Bayesian network [7]. This problem is NP-complete in general for acyclic graphs but is in P for tree structured Bayesian networks [10]. \Box

THEOREM 3. L_N -OPTDECISION is NP-hard for acyclic PCPnets.

PROOF. We give a reduction from 3-SAT. Given a 3-SAT instance $\mathcal{F} = C_1 \wedge ... \wedge C_n$, we consider the following construction of an instance of L_N -OPTDECISION:

• $I = \{V_i, \bar{V}_i\}_{1 \le i \le m} \cup \{C_i\}_{1 \le i \le n} \cup \{D\}$ is a set of binary variables. Each V_i, \bar{V}_i corresponds to a Boolean variable x_i involved in the 3-SAT instance. Each C_i corresponds to the clause C_i in \mathcal{F} . • For all $C_i \in I$, let x_{i1}, x_{i2}, x_{i3} be the variables involved in clause C_i . Then, (a) for all $V_i, \bar{V}_i \in I$, we let $Pa(V_i) = Pa(\bar{V}_i) = \emptyset$, (b) $Pa(C_i) = \{V_{i1}, ..., \bar{V}_{i3}\}$, and importantly, (c) for all $2 \le i \le n$, we let $Pa(C_i) = Pa(C_i) \cup \{C_{i-1}\}$. • $Pa(D) = C_n$. We now define the PCP-tables.

- For all $V_i, \overline{V}_i, 1 \succ 0$ (whose probability is 0.5).
- For all C_i , we add entry $1 \succ 0$ (whose probability is 1) for every assignment to $Pa(C_i)$ that satisfies all the following conditions: (1) $V_{i_k \neq} \overline{V}_{i_k}$, (2) $V_{i_k} = 1$ if x_{i_k} in clause j, OR $V_{i_k} = 0$ if $\neg x_{i_k}$ in C_j , and (3) $C_{i-1} = 1$ if i > 1. Add entry $0 \succ 1$ (whose probability is 1) for all assignments to $Pa(C_i)$ that do not satisfy all conditions.

• For D: if $C_n = 1$, then we add an entry $1 \succ 0$ (whose probability is 1). Otherwise, add an entry $0 \succ 1$ (whose probability is 0.5).



Figure 4: Construction of PCP-net from 3-SAT instance for Theorem 3.

We show that \mathcal{F} is satisfiable if and only if there exists an assignment \vec{d} such that $L_N(Q, \vec{d}) \leq n$.

⇒ Let ϕ be an assignment to the Boolean variables that satisfies \mathcal{F} . Let \vec{d} be the assignment where if $\phi_i = 1$, $d_{V_i} = 1$, $d_{\bar{V}_i} = 0$, otherwise, $d_{V_i} = 0$, $d_{\bar{V}_i} = 1$, all $d_{C_i} = 1$, and $d_D = 1$. Now, consider any CP-net C induced by Q. The only variables that can change value in a single improving flip are the variables V_i , \bar{V}_i . The total expected neighborhood loss of \vec{d} is at most $0.5 \cdot 2n$.

⇐ Let \mathcal{F} be unsatisfiable, and for the sake of contradiction, let \vec{d} be an assignment with loss $L_N(Q, \vec{d}) \leq n$. Every assignment has neighborhood loss of at least $0.5 \cdot 2n$ contributed by the variables V_i, \vec{V}_i . If $d_{C_n} = 0$, then there is an improving flip in the value of D with probability 0.5. If $d_{C_n} = 1$, and $d_{C_i} = 1$ for all i < n, then either there is an improving flip in the value of some C_i or \mathcal{F} is satisfiable. If there is a $d_{C_i} = 0, i < n$, then there must exist a pair $C_j, C_{j+1}, j < n$ such that $d_{C_j} = 0, d_{C_{j+1}} = 1$. Again, there is a non zero probability that C_{j+1} has an improving flip to 0 in some induced CP-net. \Box

THEOREM 4. L_N -OPTDECISION can be computed in polynomial time for tree structured PCP-nets.

Let Q be a tree structured PCP-net with dependency graph G. We propose an algorithm that visits each variable in G in a bottom-up, post order manner. Let X be visited in the current iteration, and let W denote the only parent of X. Suppose we have computed the quantity l_x^w for every $x \in D(X)$, which stores the minimum possible contribution to the neighborhood loss from X and its descendants when W = w and X = x. Then, for every $w \in D(W)$ we determine the assignment $x \in D(X)$ to X that minimizes the contribution to the neighborhood loss from X and its descendants and store it in $val_X^w = \arg\min_x l_x^w$ by minimizing over $x \in D(X)$. Intuitively, val_X^w stores the value of X that can ensure the lowest contribution to the neighborhood loss from assignments X and its descendants. We now revisit the computation of l_x^w . Let Y be the descendants of X. l_x^w is computed as $l_x^w = l_{val_Y}^x + f_{Q,w}^X(\bar{x} \succ x)$.

When the algorithm computes the value of the root variable that minimizes the l value, we can retrieve the solution \vec{d} by backtracking in a top down manner: At each iteration, let the current vertex be X with the assignment x, and its descendants be the set of variables W. Set each W to the value val_W^x .

EXAMPLE 3. Consider the example PCP-net in Figure 2. We trace the steps performed by the algorithm in Theorem 4.

At iteration 1, we start at **W** and compute the distribution $l_{\mathbf{W}}^{\mathbf{M}} = (l_{W_r}^{M_b} = 0.4, l_{W_w}^{M_b} = 0.6, l_{W_r}^{M_f} = 0.7, l_{W_w}^{M_f} = 0.3)$. We can now compute $val_{\mathbf{W}}^{M_b} = W_r, val_{\mathbf{W}}^{M_f} = W_w$. Then we move up one level.

At iteration 2, we are currently at **M** and compute $l_{\mathbf{M}}^{\emptyset} = (l_{M_b}^{\emptyset} = 0.3 + l_{W_r}^{M_b}, l_{M_f}^{\emptyset} = 0.7 + l_{W_w}^{M_f}) = (l_{M_b}^{\emptyset} = 0.3 + 0.4, l_{M_f}^{\emptyset} = 0.7 + 0.3).$

The choice of M_b guarantees the lowest possible neighborhood loss from **M** and its descendants. We have that $val_{\mathbf{W}}^{M_b} = W_r$. Indeed, serving beef with red wine guarantees the lowest possible neighborhood loss.

THEOREM 5. L_G -OPTDECISION is coNP-hard for acyclic PCP-nets.

PROOF. We show a reduction from 3-SAT to the complement of L_G -OPTDECISION, $\overline{L_G}$ -OPTDECISION defined as: given a PCPnet Q, a parameter k, is it true that $\forall \vec{d}, L_G(Q, \vec{d}) > k$. It is easy to verify that the problem is in PSPACE. The construction is a slight modification of the construction used in the proof of Theorem 2. The PCP-net Q (See Figure 5) is different from the CP-net in the proof of Theorem 2 in the following ways. We note that $k = 2^{2m+n}$ remains the same.

• The number of D variables is 4m + n + 1 now (vs. 2m + n + 1 in the proof of Theorem 2).

• For all V_i , \overline{V}_i , we now have $1 \succ 0$ with probability 0.5.



Figure 5: Construction of PCP-net from 3-SAT instance for Theorem 5.

Let ϕ satisfy \mathcal{F} . Consider the CP-net instance C where for every i such that $\phi_i = 1$, C has CP-table entries $1 \succ 0$ for V_i , and $0 \succ 1$ for \bar{V}_i . Similarly for every i such that $\phi = 0$, let $0 \succ 1$ be the entry for V_i , and $1 \succ 0$ be the entry for \bar{V}_i . This CP-net is induced with probability 0.5^{2m} . Let \vec{d} have $d_{V_i}, d_{\bar{V}_i}$ set according to ϕ , all $d_{C_j} = 1$, and have all $d_{D_i} = 0$. It is clear that $L_G(C, \vec{d}) = 2^{4m+n}$. Now, consider the set of assignments $\vec{d'}$ that do not match \vec{d} in the values of any or all of the variables V_i, \bar{V}_i or C_j . By construction of C, there is always a sequence of improving flips from such $\vec{d'}$ to \vec{d} as follows: If $\vec{d'}$ differs in the value of V_i or \bar{V}_i : then either $V_i \neq \bar{V}_i$ (then there is an improving flip to $V_i = \bar{V}_i$), or $V_i = \bar{V}_i$ already. In either case, there is an improving sequence to an assignment where $C_n = 0$, and subsequently to one where all $D_i = 0$. Then, there is always an improving sequence to $\vec{d'}$. Every such assignment $\vec{d'}$ has loss of at least 2^{4m+n} in C.

Consider the remaining assignments $\vec{d'}$ that match \vec{d} in values of V_i, \bar{V}_i , and C_j , but some $k \ge 1$ among $D_0, ..., D_{4m+n}$ are set to 1. Consider the case where $D_0 = 0$, then there is an improving sequence from $\vec{d'}$ to \vec{d} . Now, consider the case where $D_0 = 1$ in $\vec{d'}$. Then, consider the CP-net C' induced with probability 0.5^{2m} where variable of type V_i, \bar{V}_i has preference $1 \succ 0$ over it. There is an improving sequence from $\vec{d'}$ to a $\vec{d''}$ where all D_i are set to 1.

By construction of C', there is an improving sequence to an assignment where all variables V_i, \bar{V}_i are set to 1, and all C_j are set to 0. Subsequently, there is a flip to an assignment where $D_0 = 0$ and then $D_i, 1 \le i \le 4m + n$ can flip independently to 0. The loss of $\vec{d'}$ in C' is at least 2^{4m+n} . We have shown that when \mathcal{F} is satisfiable, every assignment has a loss at least 2^{4m+n} w.r.t. some CP-net which occurs with probability 0.5^{2m} . Therefore, every assignment has expected global loss of at least 2^{2m+n} .

Let \mathcal{F} be unsatisfiable. Consider the assignment $\vec{0}$. By construction there does not exist any assignment to V_i , \bar{V}_i that causes improving flips from $\vec{0}$ to an assignment where $C_n = 1$. For sake of contradiction, consider an assignment $\vec{d'}$ where $C_n = 1$ obtained by an improving sequence from $\vec{0}$ w.r.t. some CP-net C. Consider the sequence S used to obtain $\vec{d'}$. By construction every $C_{i < n}$ must be flipped to 1 before C_n , and every such flip happens in a setting of V_i , \bar{V}_i that is consistent with an assignment to the Boolean variables x_i that satisfies the clause c_i . Note that once either V_i , \bar{V}_i is flipped to 1, it cannot be flipped back. Together, this implies that there is an assignment of the Boolean variables which satisfies \mathcal{F} , a contradiction.

Therefore, for any CP-net C that is induced with non-zero probability according to Q, the global loss of $\vec{0}$ is at most $2^{2m+n} - 1$, and involves improving flips in the values of 2m variables V_i, \bar{V}_i , and n variables C_i . Therefore, when \mathcal{F} is unsatisfiable, the assignment $\vec{0}$ has loss less than 2^{2m+n} . \Box

5. COMPUTING OPTIMAL DECISIONS FOR CP-NET PROFILES

Given a profile $P = (P_1, ..., P_n)$, a collection of *n* CP-nets, we define the loss of a decision \vec{d} w.r.t. *P* and a loss function *L* as $L(P, \vec{d}) = \sum_{i=1}^{n} L(P_i, \vec{d})$. An optimum decision is one that minimizes the loss. This leads to a new class of voting rules characterized by a loss function. Given a loss function *L*, the voting rule r_L takes as input a profile *P* of CP-nets and outputs a set of outcomes that minimize the loss w.r.t. the preferences in *P* and the loss function *L*. Formally, $r_L(P) = \arg \min_{\vec{d}} L(P, \vec{d})$. We define the decision problem of computing optimal joint decisions under this setting for a profile of CP-net preferences, *L*-OPTJOINTDECISION, as follows.

DEFINITION 9 (*L*-OPTJOINTDECISION). Given a profile *P*, a collection of *CP*-net preferences, a loss function *L*, and a number $k \in \mathbb{R}$, does there exist an assignment \vec{d} such that $L(P, \vec{d}) \leq k$?

PROPOSITION 6. L_{0-1} -OPTJOINTDECISION is in P for a profile with acyclic CP-nets and NP-complete for cyclic CP-nets.

PROOF. For every CP-net $P_i \in P$, there exists a unique decision with loss 0 which corresponds to the unique undominated outcome, and every other decision has loss 1. This outcome can be computed in polynomial time. It is easy to check that the set of decisions that have $0 L_{0-1}$ loss in a majority of the CP-nets in P minimize the loss w.r.t. L_{0-1} and that this set can be computed in polynomial time by computing the unique, undominated outcome for each CP-net in the profile.

The NP-completeness for the case of cyclic CP-nets follows from Proposition 3. \Box

THEOREM 6. L_N -OPTJOINTDECISION is NP-complete for an O-legal profile of acyclic CP-nets.

PROOF. We give a reduction from 3-SAT. Given a 3-SAT instance $\mathcal{F} = C_1 \wedge ... \wedge C_n$, we consider the following construction of an instance of L_N -OPTJOINTDECISION on an O-legal profile P with two votes P_1 and P_2 with the same dependency graph:

• $I = \{V_i, \bar{V}_i\}_{1 \le i \le m} \cup \{C_i\}_{1 \le i \le n} \cup \{D\}$ is a set of binary variables. Each V_i, \bar{V}_i corresponds to a Boolean variable x_i involved in the 3-SAT instance. Each C_i corresponds to the clause C_i in \mathcal{F} . • For all $C_i \in I$, let x_{i1}, x_{i2}, x_{i3} be the variables involved in clause C_i . Then, (a) for all $V_i, \bar{V}_i \in I$, we let $Pa(V_i) = Pa(\bar{V}_i) = \emptyset$, (b) $Pa(C_i) = \{V_{i1}, ..., \bar{V}_{i3}\}$, and importantly, (c) for all $2 \le i \le n$, we let $Pa(C_i) = Pa(C_i) \cup \{C_{i-1}\}$.

•
$$Pa(D) = C_n$$
.

We now define the CP-tables. The CP-net P_1 has CP-tables as follows:

• For all $V_i, \overline{V}_i, 1 \succ 0$.

• For all C_i , we add the entry $1 \succ 0$ for every assignment to $Pa(C_i)$ where there exists a $k \leq 3$ such that all the following conditions are satisfied: (1) $V_{i_k \neq} \overline{V}_{i_k}$, (2) $V_{i_k} = 1$ if x_{i_k} is in clause j, OR $V_{i_k} = 0$ if $\neg x_{i_k}$ is in C_j , and (3) $C_{i-1} = 1$ if i > 1. Add entry $0 \succ 1$ for all remaining assignments.

• For D: if $C_n = 1, 1 \succ 0$. Otherwise, $0 \succ 1$.

The CP-net P_2 has CP-tables as follows:

• For all $V_i, \overline{V}_i, 0 \succ 1$.

• For all C_i , we add the entry $1 \succ 0$ for every assignment to $Pa(C_i)$ where there exists a $k \leq 3$ such that all the following conditions are satisfied: (1) $V_{i_k} \neq \overline{V}_{i_k}$, (2) $V_{i_k} = 1$ if x_{i_k} is in clause j, OR $V_{i_k} = 0$ if $\neg x_{i_k}$ is in C_j , and (3) $C_{i-1} = 1$ if i > 1. Add entry $0 \succ 1$ for all remaining assignments.

• For $D, 1 \succ 0$.

We show that \mathcal{F} is satisfiable if and only if there exists an assignment \vec{d} such that $L_N(P, \vec{d}) \leq 2n$.

Note that the only outcomes that contribute to the neighborhood loss of a given outcome are those that can obtained using a single improving flip i.e. in the change in the value of a single variable that is locally improving. Note also that for any assignment \vec{d} , the total contribution from improving flips involving the variables V_i, \bar{V}_i from both the CP-nets together is exactly 2n.

⇒ Let ϕ be an assignment to the Boolean variables that satisfies \mathcal{F} . Let \vec{d} be the assignment where (i) whenever $\phi_i = 1$, $d_{V_i} = 1, d_{\bar{V}_i} = 0$, and whenever $\phi_i = 0, d_{V_i} = 0, d_{\bar{V}_i} = 1$, (ii) all $d_{C_i} = 1$, and (iii) $d_D = 1$. By construction, in either of the CP-nets P_1, P_2 , the only variables that can change value in a single improving flip are the variables V_i, \bar{V}_i . Thus, the total neighborhood loss of \vec{d} w.r.t. the profile P is exactly 2n.

 $\Leftarrow \text{Let } \mathcal{F} \text{ be unsatisfiable, and for the sake of contradiction, let } \vec{d}$ be an assignment with loss $L_N(P, \vec{d}) \leq 2n$. Every assignment has neighborhood loss of exactly 2n contributed by the variables V_i, \bar{V}_i from both the CP-nets P_1, P_2 together. Now, if $d_{C_n} = 0$, then by construction, for any value of d_D , there is an improving flip in the value of D w.r.t. the preferences in one of the CP-nets P_1, P_2 . If $d_{C_n} = 1$, and there is some i < n such that $d_{C_i} = 0$, then there must exist a pair $C_j, C_{j+1}, j < n$ such that $d_{C_j} = 0, d_{C_{j+1}} = 1$. Then, there is an improving flip to 0 involving C_{j+1} in at least one of the CP-nets. If $d_{C_n} = 1$, and $d_{C_i} = 1$ for all i < n, then, by construction, either there is an improving flip in the value of some C_i or \mathcal{F} is satisfiable, a contradiction. \Box

THEOREM 7. L_N -OPTJOINTDECISION is in P for a profile of acyclic, tree structured CP-nets with a common dependency graph G.

PROOF. Let $P = (P_1, ..., P_n)$ be a profile of tree structured CP-net preferences over a set of issues I, that share the same dependency graph G. We propose a small modification to the algorithm in Theorem 4 that iteratively visits each variable in G in a bottom-up, post order manner. We will describe the algorithm for

the case of binary valued variables for the sake of presentation, but we note that it is easy to extend to multi-valued variables.

Let X be the variable that is being visited in the current iteration, and let W be the parent of X in G. For every CP-net P_i , every $x \in D(X)$, and every $w \in D(W)$, we store a value $l_{i,x}^w$ that tracks the minimum contribution to the neighborhood loss from X and its descendants in G when W = w, and X = x. For every $x \in D(X)$, and every $w \in D(W)$, we store a value $l_x^w = \sum_{1 \le i \le n} l_{i,x}^w$ which tracks the contribution for the entire profile. Note that for a given value w of the variable W, and quantities l_x^w for every $x \in D(X)$, $val_x^w = \arg\min_x l_x^w$ determines the value of x that ensures the lowest contribution to the neighborhood loss from improving flips in the values of X and its descendants in G from the entire profile.

Let us revisit the computation of l_x^w . Let Y be the descendants of X. The quantity l_x^w is computed as: $l_x^w = \sum_{1 \le i \le n} l_{i,val_Y}^x + \{1, if \bar{x} \succ_{P_i,w}^X x; 0, otherwise\}.$

When the algorithm computes the value of the root variable that minimizes the l value, we can retrieve the solution \vec{d} by backtracking in a top down manner: At each iteration, let the current vertex be X with the assignment x, and its descendants be the set of variables W. Set each W to the value val_W^x . \Box

THEOREM 8. L_G -OPTJOINTDECISION is coNP-hard for an O-legal profile of acyclic CP-nets.

PROOF. We give a reduction from 3-SAT. Given a 3-SAT instance $\mathcal{F} = C_1 \wedge \ldots \wedge C_n$, we give a polynomial time reduction to the complement of L_G -OPTJOINTDECISION, $\overline{L_G}$ -OPTJOINTDECISION which we define as: given a profile of CP-net preferences P, a parameter k, is it true that $\forall \vec{d}, L_G(P, \vec{d}) > k$. Consider the following construction of an instance of $\overline{L_G}$ -OPTJOINTDECISION on an O-legal profile P. We will show that \mathcal{F} is satisfiable if and only if $\forall \vec{d}, L_G(P, \vec{d}) > 2^{2m+n} - 1$.

All the CP-nets in P are defined over the following set of variables:

• $I = \{V_i, \overline{V}_i\}_{1 \leq i \leq m} \cup \{C_i\}_{1 \leq i \leq n} \cup \{D_i\}_{0 \leq i \leq 2m+n}$ is a set of binary variables. Each V_i, \overline{V}_i corresponds to a Boolean variable x_i involved in the 3-SAT instance. Each C_i corresponds to the clause C_i in \mathcal{F} .

The constructed profile $P = (P_0, P_1, \overline{P}_1, ..., P_m, \overline{P}_m)$ of 2m+1votes is *O*-legal w.r.t. $O = V_1 \succ \overline{V}_1 \succ ... \succ \overline{V}_m \succ C_1 \succ ... \succ C_n, D_0 \succ ... \succ D_{2m+n}$.

► The CP-net P_0 has the following dependency graph (See Figure 3):

• For all $C_i \in I$, let x_{i1}, x_{i2}, x_{i3} be the variables involved in clause C_i . Then, (a) for all $V_i, \overline{V_i} \in I$, we let $Pa(V_i) = Pa(\overline{V_i}) = \emptyset$, (b) $Pa(C_i) = \{V_{i1}, ..., \overline{V_{i3}}\}$, and importantly, (c) for all $2 \le i \le n$, we let $Pa(C_i) = Pa(C_i) \cup \{C_{i-1}\}$.

•
$$Pa(D_0) = \{C_n\}$$

• For all i = 1, ..., 2m + n, $Pa(D_i) = \{D_0\}$

- \triangleright We populate the CP-tables of P_0 as follows:
- For all $V_i, \overline{V}_i, 1 \succ 0$.

• For all C_i , we add the entry $1 \succ 0$ for every assignment to $Pa(C_i)$ where there exists a $k \leq 3$ such that all the following conditions are satisfied: (1) $V_{i_k} \neq \overline{V}_{i_k}$, (2) $V_{i_k} = 1$ if x_{i_k} is in clause j, OR $V_{i_k} = 0$ if $\neg x_{i_k}$ is in C_j , and (3) $C_{i-1} = 1$ if i > 1. Add entry $0 \succ 1$ for all remaining assignments.

• For D_0 : if $C_n = 1$, we add the entry $1 \succ 0$. Otherwise, $0 \succ 1$.

• For all i = 1, ..., 2m + n, D_i : if $D_0 = 1, 1 \succ 0$. Otherwise, $0 \succ 1$.

▶ For every j = 1, ..., m, we construct CP-nets P_j and \overline{P}_j . We describe the construction of P_j below. The CP-net P_j has the fol-

lowing dependency graph (See Figure 6):

• For all $1 \leq i \leq m$, $Pa(V_i) = Pa(\overline{V}_i) = \emptyset$. For all $1 \leq i \leq n$, $Pa(C_i) = \emptyset.$ • $Pa(D_0) = \{V_i\}.$

- For all $i = 1, ..., n + m, Pa(D_i) = \{D_0\}$
- \triangleright We populate the CP-tables of P_i , $1 \le j \le m$ as follows:
- For all $V_i, V_i, 0 \succ 1$.
- For all $C_i, 0 \succ 1$.
- For D_0 : if $V_i = 1, 1 \succ 0$. Otherwise, $0 \succ 1$.
- For all i = 1, ..., n + m, D_i : if $D_0 = 1, 1 \succ 0$. Otherwise, $0 \succ 1$.

The construction of \bar{P}_j differs only in \bar{V}_j taking the place of V_j in the above description.



Figure 6: Construction of CP-nets $P_i, 1 \le j \le m$ in the proof of Theorem 8. CP-nets \bar{P}_i are constructed in a similar manner.

 \Rightarrow Let \mathcal{F} be a satisfiable instance of 3-SAT and ϕ be an assignment to the Boolean variables that satisfies \mathcal{F} . We start by showing that when \mathcal{F} is satisfiable, for every assignment d, $L_G(P, d) >$ $2^{2m+n} - 1$. First, consider any decision \vec{d} such that $d_{D_0} = 1$. By construction of the CP-net P_0 , there is a sequence of improving flips from \vec{d} to the assignment $\vec{1}$. By construction of P_0 , there exists a sequence of improving flips from $\vec{1}$ to every $\vec{d'}$ where one or more of $d_{D_{1 \le i \le 2m+n}} = 0$. Therefore, by construction of P_0 , any such \vec{d} has loss $L_G(P_i, \vec{d}) \ge 2^{2m+n}$. Now, consider any decision \vec{d} where for some $1 \le i \le m, d_{V_i} = 1$. If $d_{D_0} = 0$, then by the construction of CP-net P_i , $L_G(P_i, \vec{d}) \geq 2^{2m+n}$. Lastly, consider the decision $\vec{0}$. By construction of P_0 , there is an improving sequence to an assignment $\vec{d'}$ such that if $\phi_i = 1, d'_{V_i} = 1, d'_{V_i} = 0$, and if $\phi_i = 0, d'_{V_i} = 0, d'_{V_i} = 1$. Again, by construction there is an improving sequence $\vec{d^1}, ..., \vec{d^n}$ where each of $C_1, ..., C_n$ are flipped to 1 in turn. Finally, there is an improving sequence to every $\vec{d''}$ where any or all of $d''_{D_0 \le i \le 2m+n} = 1$. Therefore, $L_G(P_0, \vec{0}) \ge 2^{2m+n}$. This completes the proof that if \mathcal{F} is satisfiable, then for every decision \vec{d} , $L_G(P, \vec{d}) > 2^{2m+n} - 1$.

 \Leftarrow Suppose for the sake of contradiction that \mathcal{F} is unsatisfiable and $L_G(\hat{P}, \vec{0}) > 2^{2m+n} - 1$. Note that by construction, for ever $1 \leq i \leq m, L_G(P_i, \vec{0}) = 0$ and $L_G(\bar{P}_i, \vec{0}) = 0$. Then, it must be that $L_G(P_0, \vec{0}) > 2^{2m+n} - 1$ i.e. that all the loss is contributed by the CP-net P_0 . However, the loss contributed by improving flips in variables V_i, \bar{V}_i, C_i is exactly $2^{2m+n} - 1$. Therefore, there must be a sequence of improving flips involving an flip in the value of one of the variables $D_0, ..., D_{2m+n}$. Consider any such sequence S. There must be an assignment in S where C_n is first flipped to 1, which must be preceded by assignments where $C_1, ..., C_{n-1}$ are flipped to 1 in turn. As argued in the proof of Theorem 2, this implies that \mathcal{F} is satisfiable, a contradiction.

While the exact complexity remains open, it is easy to see that the problem is in PSPACE, by the result in Theorem 2. \Box

5.1 Axiomatic Properties

Let P be any profile. A voting rule r satisfies (i) anonymity, if for every profile P' obtained by permuting the names of the voters, r(P') = r(P), (ii) category-wise neutrality [16], if for every profile P' obtained by applying a set of permutations that each permutes the elements in the domain of the same variable, the result r(P') is the set of outcomes in r(P) permuted in the same way, (iii) consistency, if for every pair of profiles P^1, P^2 , where $r(P^1) \cap r(P^2) \neq \emptyset, r(P^1) = r(P^2) = r(P^1 \cup P^2)$, (iv) weak *monotonicity*, if for every $\vec{d} \in r(P)$, and for every P' obtained by replacing a CP-net $C \in P$ by a CP-net C' where for some X_i , the rank of d_i is raised in the CP-table entry corresponding to the valuation $d_{Pa(X_i)}$ of variables $Pa(X_i)$, it holds that $\vec{d} \in r(P')$.

THEOREM 9. For every loss function L in our framework, the voting rule r_L satisfies anonymity, category-wise neutrality, consistency and weak monotonicity.

PROOF. (Sketch) Let $N = \{1, ..., n\}$ be a set of agents. Let $P = (P_1, ..., P_n)$ be a profile of CP-nets over $I = \{X_1, ..., X_p\},\$ where P_i represents the vote of agent $i \in N$.

Anonymity. The set of CP-nets remains unchanged in the profile obtained by permuting the names of agents.

Consistency. For any two profiles P^1 , P^2 , if \vec{d} minimizes the loss for P^1 , P^2 individually, \vec{d} minimizes the loss for $P^1 \cup P^2$.

Category-wise neutrality. Let $M = (M_1, ..., M_p)$ be a collection of permutations where each M_i only permutes $D(X_i)$. Let P' be the profile obtained by applying M to the CP-nets in P. Let C'be a CP-net obtained by applying M to C. Let \vec{e} be an assignment obtained by performing an improving flip in, say, the value of X_i , from an assignment \vec{d} according to C. Let $\vec{d'}, \vec{e'}$ be assignments obtained by applying M to \vec{d} , \vec{e} respectively. It is easy to check that $\vec{e'}$ can be obtained by an improving flip in X_i from $\vec{d'}$ according to C'. Therefore, $L(C', \vec{d'}) = L(C, \vec{d})$, and if an assignment \vec{d} minimizes the loss w.r.t. loss function L for profile P, $\vec{d'}$ minimizes the loss w.r.t. P'.

Weak monotonicity. Let $\vec{d} \in r_L(P)$, and C be a CP-net in P. Let C' be obtained from C by increasing the rank of d_i in the CP-table entry of X_i corresponding to the valuation $Pa(X_i) = d_{Pa(X_i)}$. Let P' be obtained from P by replacing C with C'. It is easy to check that for any $\vec{d'}$ where $\vec{d'}_{Pa(X_i)} \neq \vec{d}_{Pa(X_i)}, L(C', \vec{d'}) =$ $L(C, \vec{d'})$. For any $\vec{d'}$ where $\vec{d'}_{Pa(X_i)} = \vec{d}_{Pa(X_i)}$, and $d'_i = \vec{d}_i$, $L(C', \vec{d'}) > L(C, \vec{d'})$. For any $\vec{d'}$ where $\vec{d'}_{Pa(X_i)} = \vec{d}_{Pa(X_i)}$, and $d'_i = d_i, L(C', \vec{d'}) < L(C, \vec{d})$, and among these \vec{d} minimizes the loss w.r.t. C'. The contribution to the loss of \vec{d} from every other CP-net in P remains unchanged. Therefore, if $\vec{d} \in r_L(P)$, then $\vec{d} \in r_L(P')$.

6. SUMMARY AND FUTURE WORK

In this paper, we introduced the notion of loss functions to make optimal decisions for PCP-nets and collections of CP-nets with acyclic and possibly cyclic dependencies. The results for PCP-nets are, to the best of our knowledge, the first of their kind. We also introduced a new class of voting rules characterized by a loss function that computes the set of optimal loss minimizing decisions for a profile of CP-nets. We characterized the computational complexity of specific loss functions and showed that every loss function in our framework satisfies desirable axiomatic properties. The full space of reasonable restrictions and assumptions under which it is possible to efficiently find optimal solutions remains to be explored. We also intend to study social choice normative properties of mechanisms under our framework.

REFERENCES

- D. Bigot, B. Zanuttini, H. Fargier, and J. Mengin. Probabilistic Conditional Preference Networks. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, 2013.
- [2] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.
- [3] R. I. Brafman and Y. Dimopoulos. Extended Semantics and Optimization Algorithms for CP-Networks. *Computational Intelligence*, 20(2):218—245, 2004.
- [4] F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. In G. Weiss, editor, *Multiagent Systems*. MIT Press, 2013.
- [5] V. Conitzer, J. Lang, and L. Xia. Hypercubewise preference aggregation in multi-issue domains. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Catalonia, Spain, 2011.
- [6] V. Conitzer and L. Xia. Paradoxes of Multiple Elections: An Approximation Approach. In Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR), pages 179–187, Rome, Italy, 2012.
- [7] C. Cornelio, J. Goldsmith, N. Mattei, F. Rossi, and K. B. Venable. Updates and uncertainty in CP-nets. In *Advances in Artificial Intelligence*, pages 301–312. Springer, 2013.
- [8] C. Cornelio, U. Grandi, J. Goldsmith, N. Mattei, F. Rossi, and K. B. Venable. Reasoning with pcp-nets in a multi-agent context. In *Proceedings of the 2015 International Conference* on Autonomous Agents and Multiagent Systems, pages 969–977. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
- [9] J. Goldsmith, J. Lang, M. Truszczynski, and N. Wilson. The computational complexity of dominance and consistency in Cp-nets. *Journal of Artificial Intelligence Research*, 33:403–432, 2008.
- [10] D. Koller and N. Friedman. Probabilistic Graphical Models: Principles and Techniques. MIT press, 2009.

- [11] J. Lang. Vote and aggregation in combinatorial domains with structured preferences. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence* (*IJCAI*), pages 1366–1371, Hyderabad, India, 2007.
- [12] J. Lang, J. Mengin, and L. Xia. Aggregating conditionally lexicographic preferences on multi-issue domains. In *Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming (CP)*, pages 973–987, 2012.
- [13] J. Lang and L. Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences*, 57(3):304–324, 2009.
- [14] M. Li, Q. B. Vo, and R. Kowalczyk. An Efficient Procedure for Collective Decision-making with CP-nets. In *Proceedings of the 19th European Conference on Artificial Intelligence*, pages 375–380, 2010.
- [15] M. Li, Q. B. Vo, and R. Kowalczyk. Majority-rule-based preference aggregation on multi-attribute domains with structured preferences. In *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2011.
- [16] E. Mackin and L. Xia. Allocating indivisible items in categorized domains. In *Proceedings of IJCAI-2016*, 2016.
- [17] F. Rossi, K. B. Venable, and T. Walsh. mCP nets: Representing and reasoning with preferences of multiple agents. In *Proceedings of AAAI-04*, pages 729–734, 2004.
- [18] L. Xia, V. Conitzer, and J. Lang. Voting on multiattribute domains with cyclic preferential dependencies. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 202–207, Chicago, IL, USA, 2008.
- [19] L. Xia, V. Conitzer, and J. Lang. Aggregating preferences in multi-issue domains by using maximum likelihood estimators. In *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS), pages 399–406, Toronto, Canada, 2010.
- [20] L. Xia, J. Lang, and M. Ying. Strongly decomposable voting rules on multiattribute domains. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 776–781, Vancouver, BC, Canada, 2007.

Natural Interviewing Equilibria in Matching Settings

Joanna Drummond University of Toronto Canada jdrummond@cs.toronto.edu omerl@cs.toronto.edu

Omer Lev University of Toronto Canada

Allan Borodin University of Toronto Canada bor@cs.toronto.edu

Kate Larson University of Waterloo Canada kate.larson@uwaterloo.ca

ABSTRACT

While matching markets are ubiquitous, much of the work on stable matching assumes that both sides of the market are able to fully specify their preferences. However, as the size of matching markets grows, this assumption is unrealistic, and so there is interest in understanding how agents may use interviews to refine their preferences over alternatives. In this paper we study a market where one side (e.g.hospital residency programs) maintains a common preference master list, while the other side (e.g residents) have individual preferences which they can refine by conducting a limited number of interviews. The question we study is How should residents choose their interview sets, given the choices of others? We describe a payoff function for this imperfect information game, and show that this game always has a pure strategy equilibrium. Moreover, for certain structures of residents' utility there is a unique Bayesian equilibrium in which residents interview assortatively: with k interviews, each resident group $r_{kj+1}, ..., r_{kj+k}$ interviews with hospitals $h_{kj+1}, ..., h_{kj+k}$. For Borda-based linear utility functions, this equilibrium only exists when two interviews are allowed. We show this equilibrium varies for other utility functions, including exponential, and show general results regarding when this equilibrium does and does not exist.

1. INTRODUCTION

Real world matching problems are ubiquitous and cover many domains. One of the most widely studied matching problems is the canonical stable matching problem (SMP) [11]. Finding a stable matching is key in many real-world matching markets including college admissions, school choice, reviewer-paper matching, various labor-market matching problems [21], and, famously, the residency matching problem. where residents are matched to hospital programs via a centralized matching program (such as the National Residency Matching Program, NRMP, in the United States) [24].

This notion of stability, where no one in the market has both the incentive and ability to change their partner, has been empirically shown to be very valuable for real-world markets. For example, centralized mechanisms that produced a stable match tended to halt unraveling in residency matching programs, while unstable mechanisms tended to be abandoned [24]. Many matching markets that produce stable matches implement the Deferred Acceptance (DA) mechanism, introduced in Gale and Shapley's seminal paper [11].

However, to guarantee stability, stable matching mechanisms assume that participants are able to rank all their options. Assuming that participants do not have any information burden or interviewing budget is simply not the case in real-world markets: for example, in the NRMP in 2015, 27,293 positions were offered by 4,012 hospital programs [22], however residents tend to apply to an average of only 11 programs, spending between \$1,000 to \$5,000 [2]. This implies that, even if resident-proposing Deferred Acceptance (RP-DA) is the mechanism used, residents must be strategic about what hospital programs they choose to interview with, as they cannot be matched to a program with which they do not interview. Furthermore, by not carefully choosing with whom to interview, residents face the possibility of not being matched at all. There is significant evidence of this happening, as an aftermarket (SOAP) exists for the NRMP; with SOAP having matched 1,129 positions to residents in 2015, or 4.14% of the initial available positions [22]. We thus wish to study *interviewing equilibria*, not stability, for matching markets.

In spite of there being many examples where it is not feasible for participants to specify full preferences over all alternatives, there has been only limited work which has addressed participants' strategic considerations (notable exceptions include [6, 5, 16]). There similarly is little work investigating how people people choose their interviews in practice, though there is some work that suggests people tend to interview assortatively (i.e., in tiers): the best candidates apply to the best schools/hospitals, and the worst candidates apply to the worst schools/hospitals (e.g., [1]).

In this paper, using the residency matching problem as a motivating example, we initiate a study of the equilibrium behavior of participants who must decide with whom to interview, knowing they are participating in a centralized matching market running the resident-proposing deferred acceptance algorithm. In particular, under the assumption that hospitals maintain a *master list*, a commonly known fixed ranking over all residents, and that residents

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

can interview with at most k hospitals, we study which subset of hospitals residents will choose to interview and then rank. Many real-world matching markets use master lists; for example, university entrances in Turkey and China are determined by test scores [12, 25], as is high-school choice in Mexico City and Ghana [7, 1]. We further note that stating our problem using master lists also provides results for other problems: this problem can be re-contextualized as a serial dictatorship mechanism with known picking order [3].

We first formalize a payoff function for any resident in this game and show that a pure strategy equilibrium always exists under general conditions on the distributions and valuation functions from which residents' underlying preferences are drawn. We then turn to investigating when assortative interviewing forms an equilibrium, under various assumptions regarding residents' preferences. We instantiate residents' preferences as drawn from a ϕ -Mallows model (i.e., resident's idiosyncratic preferences are described by a noisy universal ranking). Under this setting, we provide a condition that is sufficient (though not necessary) to guarantee assortative interviewing. We further instantiate agents' valuation functions using classes of scoring rules from the social choice literature [4], for which there exists some evidence suggesting they may approximate the structure of participants preferences [17, 19]. We study the interplay between valuation-function structure, interview-budget size and assortative interviewing. For small interviewing budgets (of size 2 or 3), assortative interviewing may be an equilibrium depending on the valuation functions of residents and if the dispersion is not too large. However, for larger interviewing budgets our results indicate that for a large segment of resident preference structures, assortative interviewing is not an equilibrium.

2. RELATED RESEARCH

While there is a large body of research on the problem of finding stable matchings for various markets and market conditions (including when master lists are present, e.g. [13]), there has been significantly less work on the interviewing problem in which we are interested. Interviews are information-gathering activities and one research direction has looked at interviewing policies which attempt to minimize the number of interviews conducted while ensuring that a stable matching is found. Rastegari et al. showed that while finding the minimal interviewing policy is NPhard in general, there are special cases where a polynomialtime algorithm exists [23]. Drummond and Boutilier looked at a similar problem, using minimax regret and heuristic approaches for interviewing policies [10]. Neither of these papers study strategic issues arising when agents get to choose with whom they wish to interview.

Motivated by the college admissions problem, Chade and co-authors have looked at how students may strategically apply to colleges, where they assume that there is an agreedupon ranking of the colleges, but that students' quality or caliber is determined by a noisy signal [6, 5]. This work investigates how students decide where to apply in a decentralized market. We instead focus on centralized matching markets which result in stable matchings. Coles *et al.* [8] discuss signalling in matching markets. They assume that agents' preferences are distributed according to some (restricted) distributions, known *a priori*, and each agent knows their own preferences. Firms can make at most one job offer, and workers can send one *signal* to a firm indicating their interest, paralleling, in some sense, a very restricted interviewing problem. Under this setting, firms can often do better than simply offering their top candidate a job, though there are also examples where signalling may be harmful [14]. Again, the market structure in these works is quite different than the centralized matching markets we are interested in.

The work most closely related to the problem in this paper is by Lee and Schwarz [16]. They studied an interviewing game where firms and workers (or hospitals and residents) interview with each other in order to be matched. They formulate a two-stage game where firms were required to first choose workers to interview for some fixed cost. The interview action reveals both workers' and firms' preferences, which are then revealed to a market mechanism running (firm-proposing) DA. They showed that if there is no coordination then firms' best response is picking k workers at random to interview. However, if firms can coordinate then it is best for them to each select k workers so that there is perfect overlap (forming a set of disconnected complete bipartite interviewing subgraphs). This result relies heavily on the assumption that all firms and workers are *ex-ante* homogeneous, with agents' revealed preferences being idiosyncratic and independent. This assumption is very strong; for the results to hold either agents have effectively no information about their preferences before they interview, or the market must be perfectly decomposable into homogeneous sub-markets that are known before the interviewing process starts. In this paper we study a similar interviewing game, but use a different (and we believe, more realistic) set of assumptions on the structure and knowledge of preferences.

3. MODEL

There are *n* residents and *n* hospital programs. The set of residents is denoted by $R = \{r_1, ..., r_n\}$; the set of hospital programs is denoted by $H = \{h_1, ..., h_n\}$. We are interested in *one-to-one matchings* which means that residents can only do their residency at a single hospital, and that hospitals can accept at most one resident. A *matching* is a function $\mu : R \cup H \to R \cup H$, such that $\forall r \in R$, $\mu(r) \in H \cup \{r\}$, and $\forall h \in H, \mu(h) \in R \cup \{h\}$. If $\mu(r) = r$ or $\mu(h) = h$ then we say that *r* or *h* is unmatched. A matching μ is *stable* if there does not exist some $(r, h) \in R \times H$, such that $h \succ_r \mu(r)$ and $r \succ_h \mu(h)$.

Both hospitals and residents have (strict) preferences over each other, and we let H_{\succ} and R_{\succ} denote the sets of all possible preference rankings over H and R respectively. We assume that hospitals have identical preferences over all residents, which we call the master list, \succ_H . Without loss of generality, let $\succ_H = r_1 \succ r_2 \succ \ldots \succ r_n$ where $r_i \succ_H r_j$ means that r_i is preferred to r_j , according to \succ_H . We further assume that the master list is common knowledge to all members of H and R. That is, all hospitals agree on the preference ranking over residents and each resident knows where they, and all others, rank in the list. While each resident, r, has idiosyncratic preferences over the hospitals, we assume that these are drawn i.i.d. from some common distribution D, and that this is common knowledge. If resident r draws preference ranking η from D, then $h_i \succ_{\eta} h_j$ means that h_i is preferred to h_i by r under η . We assume there is some common scoring function $v: H \times H_{\succ} \mapsto \mathbb{R}$, applied to rankings η drawn from D such that, given any $\eta \in H_{\succ}$ with $h_i \succ_{\eta} h_j, v(h_i, \eta) > v(h_j, \eta).$

Critical to our model is the assumption that residents do not initially know their true preferences, but can refine their knowledge by conducting a number of *interviews*, not exceeding their interviewing budget k. We let $I(r_i) \subset H$ denote the interview set of resident r_j , and $|I(r_j)| \leq k$ for some fixed $k \leq n$. Once r_j has finished interviewing, r_i knows her preference ranking over $I(r_i)$. She then submits this information to the matching algorithm, residentproposing deferred acceptance (RP-DA). The matching proceeds in rounds, where in each round unmatched residents propose to their next favorite hospital from their interview set to whom they have not yet proposed. Each hospital chooses its favorite resident from amongst the set of residents who have just proposed and its current match, and the hospital and its choice are then tentatively matched. This process continues until everyone is matched. The resulting matching, μ , is guaranteed to be stable, resident-optimal, and hospital-pessimal [11]. This matching is also guaranteed to be unique, as stable matching problems with master lists have unique stable solutions [13]. Thus our results directly hold for any mechanism that returns a stable match, including hospital-proposing deferred acceptance, and the greedy linear-time algorithm [13].

3.1 Description of the Game

We now describe the *Interviewing with a Limited Budget* game:

- 1. Each resident $r \in R$ simultaneously selects an interviewing set $I(r) \subset H$, based on their knowledge of D and the hospitals' master list \succ_H , where $|I(r)| \leq k$.
- 2. Each resident, r, interviews with hospitals in I(r) and discovers their preferences over members of I(r).
- 3. Each resident reports their learned preferences over I(r) and reports all other hospitals as unacceptable. Each hospital reports the master list to a centralized clearinghouse, which runs resident-proposing deferred acceptance (RP-DA), resulting in the matching μ .

3.2 Payoff function for Interviewing with a Limited Budget

Let M be the set of all matchings, and let μ denote the ex-post matching resulting from all agents playing the *Inter*viewing with a Limited Budget game. In order for resident r_j to choose their interview set $I(r_j) \subset H$, she has to be able to evaluate the payoff she expects to receive from that choice, where the payoff depends on both the actual preference ranking she expects to draw from D, the interview sets of the other residents, and the expected matching achieved from the mechanism as described. Crucially, we observe that r_j need only be concerned about the interview set of resident r_i when $r_i \succ_H r_j$. If $r_j \succ_H r_i$ then, because we run RP-DA, r_j would always be matched before r_i with respect to any hospital they both had in their interview set. Thus, we can denote r_j 's expected payoff for choosing interview set S by: $u_{r_i}(S) = u_{r_i}(S|D, I(r_1), ..., I(r_{j-1}))$.

Given fixed interviewing sets $I(r_1), ..., I(r_{j-1})$, and some partial match $m = \mu_{|r_1,...,r_{j-1}}$, we must compute the probability that m happened via RP-DA. Let $m(r_i)$ denote who resident r_i is matched to under m. For any r_i , there is a set of rankings consistent with r_i being matched with $m(r_i)$ under RP-DA (and the hospitals' master list \succ_H). Denote this set as $T(r_i, m)$. Formally, $T(r_i, m) \subseteq H_{\succ}$ is:

$$T(r_i, m) = \{ \xi \in H_{\succ} | \forall h' \in H \text{ s.t. } h' \in I(r_i) \land h' \succ_{\xi} m(r_i), \\ \exists r_a \text{ s.t. } r_a \succ_H r_i \land m(r_a) = h' \}$$

Given the interviewing sets of residents r_1, \ldots, r_{j-1} , the probability of partial match m is

$$P(m|I(r_1), ..., I(r_j)) = \prod_{r_i \in \{r_1, ..., r_{j-1}\}} \sum_{\xi \in T(r_i, m)} P(\xi|D).$$
(1)

where $P(\xi|D)$ is the probability that some resident drew ranking $\xi \in H_{\succ}$ from D.

Using Eq. 1, we can now determine the probability that some hospital h is matched to r_j using RP-DA, when r_j has interviewed with set S, and has preference list η . We simply sum over all possible matches in which this could happen. Because RP-DA is resident optimal, and all hospitals have a master list, any hospital that r_j both interviews with and prefers to h must already be matched. We formally define the set of such matchings, $M^*(S, \eta, I(r_1), ..., I(r_{j-1}))$:

$$\begin{split} M^*(S, \eta, I(r_1), ..., I(r_{j-1}), h) &= \\ \{m \in M | m(r_j) = h; \forall r_i \in \{r_1, ..., r_{j-1}\} m(r_i) \in I(r_i); \\ \text{and } \forall x \in S, \text{ if } x \succ_{\eta} h, \exists r_i \in \{r_1, ..., r_{j-1}\} \text{ s.t. } x \in I(r_i) \text{ and } m(r_i) = x \rbrace \end{split}$$

Thus, the probability that h is matched to r_j using RP-DA given η , S, and the interviewing sets for all residents preferred to r_j on the hospitals' master list is

$$P(\mu(h) = r_j | \eta, S, I(r_1), ..., I(r_{j-1})) = \sum_{m \in M^*(S, \eta, I(r_1), ..., I(r_{j-1}), h)} P(m | I(r_1), ..., I(r_{j-1})).$$
(2)

For readability, we will frequently refer to $P(\mu(h) = r_j | \eta, S, I(r_1, ..., I(r_{j-1}) \text{ as } P(\mu(h) = r_j | \eta, S)$. Finally, we have all of the building blocks to formally define the payoff function. Recall that $v(h, \eta)$ is the imposed utility function, dependent on η : for any given η , $v(h, \eta)$ is fixed. Then, our payoff function is:

$$u_{r_j}(S) = \sum_{h \in S} \sum_{\eta \in H_{\succ}} v(h, \eta) P(\eta | D) P(\mu(h) = r_j | \eta, S, I(r_1), ..., I(r_{j-1}))$$
(3)

Intuitively, what the payoff function in Eq. 3 does is weight the value for some given alternative by how likely r_j is to be matched to that item, given the interview sets of the "more desirable" residents, r_1, \ldots, r_{j-1} .

As an illustrative example, imagine there are two residents, r_1 and r_2 , each of whom have interviewed with hospitals h_1 and h_2 . Resident r_1 will be matched with whomever she most prefers, while r_2 will be assigned the other. The probability that r_2 will be assigned h_1 is simply the probability that r_1 drew ranking $h_2 \succ h_1$, while the probability that r_2 is matched to h_2 is the probability that r_1 drew ranking $h_1 \succ h_2$.

3.3 Probabilistic Preference Models

While our payoff-function formulation, described in the previous section, is general in that we do not instantiate it with a particular distribution function, we do assume that some distribution is used over the space of possible rankings of hospitals. The Mallows model is characterized by a reference ranking σ , and a dispersion parameter $\phi \in (0, 1]$,¹

¹A ϕ -Mallows model is not well defined for $\phi = 0$, but if all residents are guaranteed to draw the reference ranking, the equilibrium is trivial.

which we denote as $D^{\phi,\sigma}$. Let A denote the set of alternatives that we are ranking, and let A_{\succ} denote the set of all permutations of A (the index $i \in [1, n]$ in $a_i \in A$ indicates rank in σ). The probability of any given ranking r is:

$$P(r|D^{\phi,\sigma}) = \frac{\phi^{d(r,\sigma)}}{Z}$$

Here d is Kendall's τ distance metric, and Z is a normalizing factor; $Z = \sum_{r' \in P(A)} \phi^{d(r,\sigma)} = (1)(1+\phi)(1+\phi+\phi^2)...(1+...+\phi^{|A|-1})$ [18].

As $\phi \to 0$, the distribution approaches drawing the reference ranking σ with probability 1; when $\phi = 1$, this is equivalent to drawing from the uniform distribution. The Mallows model (and mixtures of Mallows) have plausible psychometric motivations and are commonly used in machine learning [20, 15, 18]. Mallows models have also been used in previous investigations of preference elicitation schemes for stable matching problems (e.g., [9, 10]).

To prove our equilibria results, we will need additional results regarding properties of Mallows models. To the best of our knowledge, the following have not been stated previously, and may be of more general interest. Proofs omitted due to space constraints

LEMMA 1. Given some Mallows model $D^{\phi,\sigma}$ with fixed dispersion parameter ϕ and reference ranking $\sigma = a_i \succ a_j$, then the probability that a ranking η is drawn from $D^{\phi,\sigma}$ such that $a_i \succ_{\eta} a_j$ is equal to drawing from some distribution $D^{\phi,\sigma'}$ where σ is a prefix of σ' . By symmetry, this proof also holds when σ is a suffix of σ' .

COROLLARY 2. Given any reference ranking σ and two alternatives $a_i, a_{i+1}, P(a_i \succ a_{i+1} | D^{\phi,\sigma}) = \frac{1}{1+\phi}$.

COROLLARY 3. Given any reference ranking σ and three alternatives a_i, a_{i+1}, a_{i+2} and some $\eta \in \{a_i, a_{i+1}, a_{i+2}\}_{\succ}$, then the probability that we draw some ranking β consistent with η is: $P(\beta|D^{\phi,\sigma}) = \frac{\phi^{d(\eta,a_i \succ a_{i+1} \succ a_{i+2})}}{(1+\phi)(1+\phi+\phi^2)}$.

Lemma 4. The probability a_i will be ranked in place j is $\frac{\phi^{|j-i|}}{1+\phi+\ldots+\phi^{n-1}}.$

LEMMA 5. Let $\eta \in D^{\phi,\sigma}$ in which $a_j \succ_{\eta} a_i$ for i < j, then $P(\eta) < \frac{\phi^{j-i}}{\pi}$.

4. GENERAL EQUILIBRIA FOR INTERVIEW-ING MARKETS WITH MASTER LISTS

We provide an equilibrium analysis for the game presented in Section 3. We first show that a pure equilibria for this game always exists, even under arbitrary distributions and scoring functions, but may be computationally infeasible to directly calculate. We then instantiate this model for various distributions and scoring functions, focusing on one family of distributions: the ϕ -Mallows model. We provide a necessary and sufficient condition for *assortative* interviewing under a Mallows model and then investigate what values of ϕ and k will result in assortative interviewing for various scoring functions.

4.1 General Equilibria for Interviewing Markets with Master Lists

We start our analysis by studying the most general form of the *Interviewing with a Limited Budget game*, and show that a pure strategy equilibrium always exists. THEOREM 6. A pure strategy always exists for the Interviewing with a Limited Budget game.

PROOF. We wish to show that if every resident chooses their expected utility maximizing interviewing set, this forms a pure strategy. Given any resident r_j who is *j*th in the hospitals' rank ordered list, r_j 's expected payoff function only depends on residents r_1, \ldots, r_{j-1} . As r_j knows that each other resident r_i is drawing from distribution D *i.i.d.*, she can calculate r_1, \ldots, r_{j-1} 's expected utility maximizing interview set, using Eq. 3. Her payoff function depends only on D and $I(r_1), \ldots, I(r_{j-1})$, both of which she now has. She then calculates the expected payoff for each $\binom{n}{k}$ potential interviewing sets, and interviews with the one that maximizes her expected utility. \Box

We note that Theorem 6 is an existence theorem and does not provide any additional insight into the equilibrium behavior, nor does it provide guidance as to how such an equilibrium might be computed. Our next result begins to provide some intuition as to equilibrium behavior. In particular it shows that if residents have interviewing budgets of size k and the equilibrium behavior for resident r_k is to interview assortatively (*i.e.* it chooses to interview with hospitals h_1, \ldots, h_k), then assortative interviewing is the equilibrium strategy for all residents.

PROPOSITION 7. Given an interviewing budget of k interviews, some known distribution from which residents draw their preferences D and a scoring function v, if resident r_k 's best response to all others interview assortatively is to interview assortatively, then assortative interviewing is an equilibrium for all residents.

(Proof omitted due to space constraints)

4.2 Interviewing Equilibria Under Mallows Models with Master Lists

In this section we instantiate the distribution from which residents are drawing their preferences with a Mallows model in order to gain a deeper understanding of the results from the previous section. In particular, we provide a characterization of when assortative interviewing will form an equilibrium for this class of resident-preferences. Before proving our main result, we require some additional lemmas addressing characteristics of assortative interviewing in Mallows models.

All proofs are omitted due to space constraints.

LEMMA 8. Given an interviewing budget of k interviews, a dispersion parameter ϕ , and a scoring function v, if resident r_k prefers interviewing with hospitals $\{h_1, ..., h_k\}$ to $\{h_1, ..., h_{k+1} \setminus \{h_j\}$ for all $h_j \in \{h_1, ..., h_k\}$, then for resident r_k , interviewing with $\{h_1, ..., h_k\}$ dominates interviewing with any other set of size k.

We now provide a necessary and sufficient condition for assortative interviewing to hold when residents draw their preference from a Mallows model with dispersion ϕ . Let $P(h_i \text{ avail})$ denote the probability that hospital h_i is available for resident r_k (i.e., residents $r_1, ..., r_{k-1}$ are all matched to different alternatives). As we assume residents $r_1, ..., r_{k-1}$ interview assortatively, only one of $\{h_1, ..., h_k\}$ will be available.

LEMMA 9. Given an interviewing budget of k interviews, a dispersion parameter ϕ , and a scoring function v, if residents $r_1, ..., r_{k-1}$ all interview assortatively (i.e., with hospital set $S = \{h_1, ..., h_k\}$), satisfying the following inequality for all $h_j \in \{h_1, ..., h_k\}$ when $S' = S \setminus \{h_j\} \cup \{h_{k+1}\}$ is both sufficient and necessary to show that assortative interviewing is an equilibrium for resident r_k :

$$P(h_j \text{ avail})\mathbb{E}(v(h_j)|D^{\phi,\sigma}) \ge$$

$$P(h_j \ avail)\mathbb{E}(v(h_{k+1})|D^{\phi,\sigma}) + \sum_{j=0}^{\infty} P(n_j|D^{\phi,\sigma}) \left[\sum_{j=0}^{\infty} P(h_j|D^{\phi,\sigma}) - h_j(h_j|D^{\phi,\sigma})\right]$$

$$\sum_{\eta \in H_{\succ}} P(\eta | D^{\phi,\sigma}) \cdot \left[\sum_{h_i \in S'} P(h_i \text{ avail}) \chi(h_{k+1} \succ_{\eta} h_i) v(h_{k+1}, \eta) \right]$$

(4)

Where $\chi(h_i \succ_{\eta} h_j)$ is an indicator function that is 1 iff $h_i \succ_{\eta} h_j$, and 0 otherwise.

THEOREM 10. Given an interviewing budget of k interviews, a dispersion parameter ϕ , and a scoring function v, satisfying the inequality found in Lemma 9 for all $h_j \in \{h_1, ..., h_k\}$ is both sufficient and necessary to show that assortative interviewing is an equilibrium for all residents.

PROOF. This follows directly from combining Proposition 7 and Lemma 9. $\hfill \square$

We now provide a more simplified condition for assortative interviewing, that is sufficient, though not necessary (and leave the proof to the appendix):

LEMMA 11. Given an interviewing budget of k interviews, a dispersion parameter ϕ , and a scoring function v, if residents $r_1, ..., r_{k-1}$ all interview assortatively (i.e., with hospital set $S = \{h_1, ..., h_k\}$), satisfying the following inequality for all $h_j \in \{h_1, ..., h_k\}$ when $S' = S \setminus \{h_j\} \cup \{h_{k+1}\}$ is sufficient to show that assortative interviewing is an equilibrium for resident r_k :

$$P(h_j \ avail)\mathbb{E}(v(h_j)|D^{\phi,\sigma}) \ge P(h_j \ avail)\mathbb{E}(v(h_{k+1})|D^{\phi,\sigma}) + \sum_{h_i \in S'} P(h_i \ avail)\mathbb{E}(v(h_k')|D^{\phi,\sigma'}) \frac{\phi}{Z(1-\phi)}$$
(5)

(Where σ' is equivalent to the reference ranking σ with one element h_i s.t. $h_j \succ_{\sigma} h_i$ removed, and h'_k is the kth item in σ' .)

Though we primarily discuss assortative interviewing as it is a technique commonly used in real-world interviewing markets, we note that the n/k complete disjoint bipartite subgraph equilibrium shown in Lee and Schwarz for uniform distributions on both sides of the market also holds when one side is drawing uniform iid (equivalently, a Mallows model with $\phi = 1.0$), and the other side has a master list.

OBSERVATION 12. When residents draw iid from uniform, and hospitals have a master list, an equilibrium exists such that the interviewing graph forms n/k complete disjoint bipartite subgraphs. Moreover, any resident r_{ik+j} interviews with hospitals $\{h_{(j-1)k+1}, \ldots, h_{jk}\}$.

5. ASSORTATIVE EQUILIBRIA FOR SMALL BUDGETS

We now discuss assortative equilibria when participants' interviewing budget is $k \leq 3$. We do so by instantiating specific scoring rules, and investigating under what circumstances assortative interviewing forms an equilibrium. We

now formally define Borda, plurality, and exponential scoring rules, following definitions typically used in voting. We define all scoring rules with a multiplicative factor of 1, and an additive factor of 0, as these terms do not affect the analysis. For any slot s_i , $v(s_i) = n - i + 1$ in Borda, where n is the number of alternatives in the market. Under plurality, $v(s_1) = 1$, $v(s_i) = 0$ for all i > 1. We investigate a class of exponential functions that are dominated by the function $v(s_i) = (\frac{\varepsilon}{2})^{i-1}$, $1 > \varepsilon > 0$.

The proofs for the following two lemmas are omitted due to space constraints

LEMMA 13. If for a particular interviewer budget k, a dispersion parameter ϕ , the condition of Lemma 14 is satisfied for a plurality valuation function with a strict inequality, then there are exponential valuations which form an assortative equilibrium.

In particular, any exponential valuation dominated by $(\frac{\varepsilon}{2})^{(i-1)}$ satisfies this condition, with $\varepsilon > 0$ determined by k.

LEMMA 14. A necessary and sufficient condition for assortative interviewing under plurality is:

$$P(h_j \ avail) \ge \phi^{k-j+1} \tag{6}$$

This follows from instantiating plurality into Eq. 6, applying Lemmas 7 and 4, and simplifying.

5.1 Assortative Interviewing with Two Interviews

We provide direct proofs showing that assortative interviewing is an equilibrium for Borda and plurality. Exponential follows directly from Lemma 13.

THEOREM 15. Given plurality as residents' scoring function and a budget of k = 2 interviews, for a Mallows model with dispersion parameter ϕ such that $0 < \phi \leq 0.6180$, assortative interviewing forms an equilibrium.

PROOF. We begin by using the condition from Lemma 14. We provide the calculation for h_1 ; h_2 follows analogously (providing a bound of $0 < \phi \leq 0.7549$). We thus wish to show conditions on ϕ s.t. $P(h_1 \text{ avail}) \geq \phi^2$, when resident r_2 is choosing their interview set. For r_2 , h_1 is available iff r_1 happened to draw a ranking over her preferences s.t. $h_2 \succ h_1$. Then, by Corollary 2, $P(h_1 \text{ avail}) = \frac{\phi}{1+\phi}$, implying we need to satisfy the equation $\frac{\phi}{1+\phi} \geq \phi^2$, which is true whenever $0 < \phi \leq 0.6180$. \Box

THEOREM 16. Given Borda as residents' scoring function and a budget of k = 2 interviews, for a Mallows model dispersion parameter ϕ such that $0 < \phi \le 0.2650$, assortative interviewing forms an equilibrium.

PROOF. Because of Lemma 7, we only need to show that assortative interviewing is an equilibrium when $0 < \phi \leq$ 0.265074 for resident r_2 , and it will hold for all r_i . Furthermore, by Lemma 8, we only need to prove that $\{h_1, h_2\}$ dominates both $\{h_1, h_3\}$ and $\{h_2, h_3\}$ to show that it dominates all other possible interviewing sets of size 2.

We prove that choosing $\{h_1, h_2\}$ is better than choosing $\{h_2, h_3\}$, for all values of ϕ such that $0 < \phi \leq 0.265074$. We prove this by summing over all possible preference rankings that induce a specific permutation of the alternatives h_1, h_2, h_3 . We then pair these summed permutations in such a manner that makes it easy to find a lower bound for $u_{r_2}(\{h_1, h_2\}) - u_{r_2}(\{h_2, h_3\})$. This lower bound is entirely in terms of ϕ , meaning that for any ϕ such that this bound is above 0, it will be above 0 for any market size n.

We look at three cases, pairing all possible permutations of h_1, h_2, h_3 as follows:

Case 1: all rankings η consistent with $h_2 \succ h_1 \succ h_3$ or η' consistent with $h_2 \succ h_3 \succ h_1$;

Case 2: all rankings η consistent with $h_1 \succ h_2 \succ h_3$ or η' consistent with $h_3 \succ h_2 \succ h_1$;

Case 3: all rankings η consistent with $h_1 \succ h_3 \succ h_2$ or η' consistent with $h_3 \succ h_1 \succ h_2$.

Note that as we have enumerated all possible permutations of h_1, h_2, h_3 , these three cases generate every ranking in H_{\succ} . Furthermore, for any one of the three cases, we can iterate over only all possible rankings η that are consistent with the first member of the pair, and generate the ranking η' consistent with the second member of the pair by simply swapping two alternatives in the rank. Moreover, given some η , the number of discordant pairs in η' is simply the number in η , plus the number of additional discordant pairs between h_1, h_2, h_3 caused by swapping the two alternatives.

For clarity, let $u_{r_2}(\{h_1, h_2\}) - u_{r_2}(\{h_2, h_3\}) = U_1 + U_2 + U_3$, where U_1, U_2, U_3 correspond to our three cases. We also introduce the notation $P_{\mu(r_i)}(h)$ to denote the probability that r_i is matched to hospital h under matching μ . That is, $P_{\mu(r_i)}(h) = P(\mu(r_i) = h)$. The case proofs proofs are omitted due to space constraints

Once considering all cases, we combine them together:

$$u_{r_{2}}(\{h_{1},h_{2}\}) - u_{r_{2}}(\{h_{2},h_{3}\}) \geq \frac{\phi^{2}}{(1+\phi)(1+\phi)(1+\phi+\phi^{2})}(1-\phi) + \frac{2(\phi-\phi^{3}-\phi^{4})}{(1+\phi)(1+\phi)(1+\phi+\phi^{2})} - \frac{\phi}{(1+\phi)(1+\phi)(1+\phi+\phi^{2})}\left(\frac{\phi}{(1-\phi)^{4}} + \frac{1}{3(1-\phi)^{3}} + \frac{2}{3}\right)(1+\phi) + \frac{\phi^{2}}{(1+\phi)(1+\phi)(1+\phi+\phi^{2})}(1-\phi)$$

$$(7)$$

Thus, Eq. 7 gives us a lower bound for the difference in expected utility between $\{h_1, h_2\}$ and $\{h_2, h_3\}$ for resident r_2 , for all n. Using numerical methods to approximate the roots of Eq. 7, we get that there is a root at 0, and a root at $\phi \approx 0.265074$.

As the calculations are analogous, we omit the discussion of their derivation, but it can be shown that:

$$u_{r_{2}}(\{h_{1},h_{2}\}) - u_{r_{2}}(\{h_{1},h_{3}\}) \geq \frac{1}{(1+\phi)(1+\phi+\phi^{2})} \left[1+\phi-2\phi^{2}-2\phi^{3}-2\phi^{3}\left(\frac{\phi}{(1-\phi)^{4}}+\frac{1}{3(1-\phi)^{3}}+\frac{2}{3}\right)\right]$$

$$(8)$$

Using numerical methods, it can be shown that this is positive for $0 < \phi < 0.413633$.

Thus, for the interval $0 < \phi \le 0.265074$, we have shown that r_2 's best move in this interval is to interview with $\{h_1, h_2\}$. Then, by Lemma 7, this is an equilibrium for all r_i as required. \Box

5.2 Assortative Interviewing with Three Interviews

Unlike when only two interviews are present, assortative interviewing is *not* an equilibrium under Borda when participants have a budget of 3 interviews. Under plurality (and exponential), assortative interviewing is still an equilibrium.

THEOREM 17. Assortative interviewing is not guaranteed to be an equilibrium under the Borda valuation function, even for any ϕ .

(Proof omitted due to space constraints)

Under Borda, an assortative interviewing equilibrium is not guaranteed to exist, even for any $1 \ge \phi > 0$. However, we now show that assortative interviewing is an equilibrium for plurality (and thus exponential) for k = 3:

THEOREM 18. Given an interviewing budget of k = 3 interviews, and the plurality scoring function, assortative interviewing is an equilibrium for $0 < \phi \le 0.4655$.

PROOF. For k = 3, we simply check Eq. 6 from Lemma 14 with $h_j = h_1, h_2, h_3$. We find that the marginal contribution from h_1 is less than the marginal contribution of h_2 or h_3 , and thus only present the calculation for h_1 . We directly compute $P(h_1 \text{ avail})$, by multiplying the probability that r_1 did not take h_1 , and multiplying it by the probability that r_2 did not take h_1 , given that r_1 also did not take h_1 . To calculate this we enumerate the probabilities of any possible rankings:

$$P(h_1 \text{ avail}) = P(\mu(r_1) \neq h_1)P(\mu(r_2) \neq h_1|\mu(r_1) \neq h_1)$$
$$P(h_1 \text{ avail}) = \left(\frac{\phi + 2\phi^2 + \phi^3}{(1+\phi)(1+\phi+\phi^2)}\right) \left(\frac{\phi^2 + 2\phi^3}{(1+\phi+\phi^2)}\right)$$

Using numerical methods to find the roots of $P(h_1 \text{ avail}) - \phi^3$, we can show that Eq. 6 holds when $0 < \phi \le 0.4655$. \Box

6. ASSORTATIVE EQUILIBRIA FOR LARGE BUDGETS

We begin by providing a few final results regarding properties of interviewing under Mallows models, including that when there is a setting for which there is no assortative equilibria for plurality, then there is no valuation function with assortative equilibria. We use this result to show that, for sufficiently small ϕ and a large enough budget of interviews (k > 3), assortative interviewing cannot be an equilibrium under any valuation function. We then provide a specific counterexample for all ϕ when k = 4 for plurality, implying there is no assortative equilibrium for any valuation function. This suggests that, for a wide category of resident valuation functions under Mallows, contrary to real-world behavior, assortative interviewing is not an equilibrium.

LEMMA 19. Given a Mallows model with dispersion parameter ϕ , assortative interviewing for residents r_1, \ldots, r_{k-1} , and a hospital $h_i \in \{h_1, \ldots, h_k\}$ (i.e., the residents' interview set), then any profile $\eta_1, \ldots, \eta_{n-1} \in D^{\phi,\sigma}$ of k-1 preferences (for r_1, \ldots, r_{k-1}) such that h_i is available for r_k has a probability of: $P(r_1 = \eta_1, r_2 = \eta_2, \ldots, r_{k-1} = \eta_{k-1} | h_i \text{ avail}) < \frac{\phi^{\gamma}}{Z^{k-1}}$, where $\gamma = \sum_{j=1}^{k-i} j$.

PROOF. In order for h_i to be available, there need to be r'_{i+1}, \ldots, r'_k with preference orders $\eta_{i+1}, \ldots, \eta_k \in D^{\phi,\sigma}$ such that they were assigned hospitals h_{i+1}, \ldots, h_k . Hence, $h_{i+1} \succ_{\eta_{i+1}} h_i, \ldots, h_k \succ_{\eta_k} h_i$. According to Lemma 5, the probability for each of these events is at least $\frac{\phi}{Z}, \ldots, \frac{\phi^{k-i}}{Z}$ (respectively). Since they are independent of each other, and since the maximal probability for any particular $\eta \in D^{\phi,\sigma}$ is $\frac{1}{Z}$, the probability of a particular preference set occurring in which h_i is available is at least $\frac{\phi}{Z^{k-1}}$. \Box THEOREM 20. If for a particular interviewer budget k, a dispersion parameter ϕ , when using plurality valuation there are no assortative equilibria due to h_1 violating Lemma 9's condition, then for that k and ϕ there is no assortative equilibria for any valuation function.

PROOF. Looking at the condition of Lemma 9

$$\begin{split} & P(h_j \text{ avail}) \mathbb{E}(v(h_j) | D^{\phi, \sigma}) \geq \\ & P(h_j \text{ avail}) \mathbb{E}(v(h_{k+1}) | D^{\phi, \sigma}) + \\ & \sum_{\eta \in H_{\succ}} P(\eta | D^{\phi, \sigma}) \Big[\sum_{h_i \in S'} P(h_i \text{ avail}) \chi(h_{k+1} \succ_{\eta} h_i) v(h_{k+1}, \eta) \Big] \end{split}$$

We again begin by expanding the value expectation (\mathbb{E})This can be divided to n different inequalities:

$$P(h_j \text{ avail})P(h_j \text{ in } s_1)v(s_1) \ge v(s_1)[P(h_j \text{ avail})P(h_{k+1} \text{ in } s_1)+$$

$$\sum_{\substack{\eta \in H_{\succ} \mid \\ h_{k+1} \text{ in } s_1}} P(\eta | D^{\phi,\sigma}) \sum_{h_i \in S'} P(h_i \text{ avail}) \chi(h_{k+1} \succ_{\eta} h_i)]$$

:

ŀ

h

$$\begin{split} & P(h_j \text{ avail})P(h_j \text{ in } s_{n-1})v(s_{n-1}) \geq \\ & v(s_{n-1})[P(h_j \text{ avail})P(h_{k+1} \text{ in } s_{n-1}) + \\ & \sum_{\substack{\eta \in H_\succ | \\ h_{k+1} \text{ in } s_{n-1}}} P(\eta | D^{\phi,\sigma}) \sum_{h_i \in S'} P(h_i \text{ avail})\chi(h_{k+1} \succ_{\eta} h_i)] \end{split}$$

 $P(h_j \text{ avail})P(h_j \text{ in } s_n)v(s_n) \ge v(s_n)P(h_j \text{ avail})P(h_{k+1} \text{ in } s_n)$

We shall show that under the theorem's assumptions, none of these inequalities hold for h_1 , and therefore the general inequality (Lemma 9) does not hold.

Note that for each inequality we can simply ignore $v(s_{\ell})$ $(1 \leq \ell \leq n)$, since they appear on both sides of the inequality. The assumption of theorem is that first inequality does not hold, i.e.,

$$\begin{split} P(h_1 \text{ avail})P(h_1 \text{ in } s_1) < P(h_1 \text{ avail})P(h_{k+1} \text{ in } s_1) + \\ \sum_{\substack{\eta \in H_{\succ} \mid \\ h_{k+1} \text{ in } s_1}} P(\eta | D^{\phi,\sigma}) \sum_{h_i \in S'} P(h_i \text{ avail})\chi(h_{k+1} \succ_{\eta} h_i) \end{split}$$

As shown in Lemma 4, for any $1 < \ell \leq k$ the probability of h_1 being in any spot s_ℓ is monotonically decreasing with ℓ , while the probability of h_{k+1} being in spot s_ℓ is monotonically increasing with ℓ . Hence, $P(h_1 \text{ avail})P(h_1 \text{ in } s_1) > P(h_1 \text{ avail})P(h_1 \text{ in } s_\ell)$. Similarly, $P(h_1 \text{ avail})P(h_{k+1} \text{ in } s_1) < P(h_1 \text{ avail})P(h_{k+1} \text{ in } s_\ell)$. We analogously see that:

$$\sum_{\substack{\eta \in H_{\succ} \mid \\ h_{k+1} \text{ in } s_1}} P(\eta | D^{\phi,\sigma}) \sum_{h_i \in S'} P(h_i \text{ avail}) \chi(h_{k+1} \succ_{\eta} h_i) < \sum_{\substack{\eta \in H_{\succ} \mid \\ h_{k+1} \text{ in } s_\ell}} P(\eta | D^{\phi,\sigma}) \sum_{h_i \in S'} P(h_i \text{ avail}) \chi(h_{k+1} \succ_{\eta} h_i)$$

Simply put, the LHS gets smaller, while the RHS increases. Hence, for $1 \le \ell \le k$:

$$\begin{split} P(h_1 \text{ avail})P(h_1 \text{ in } s_\ell) &< P(h_1 \text{ avail})P(h_{k+1} \text{ in } s_\ell) + \\ \sum_{\substack{\eta \in H_\succ | \\ h_{k+1} \text{ in } s_\ell}} P(\eta | D^{\phi,\sigma}) \sum_{h_i \in S'} P(h_i \text{ avail})\chi(h_{k+1} \succ_\eta h_i) \end{split}$$

By Lemma 4, for any $\ell > k$, $P(h_1 \text{ in } s_\ell) < P(h_{k+1} \text{ in } s_\ell)$ which gives us:

 $P(h_1 \text{ avail})P(h_1 \text{ in } s_{\ell}) < P(h_1 \text{ avail})P(h_{k+1} \text{ in } s_{\ell}) \Longrightarrow$ $P(h_1 \text{ avail})P(h_1 \text{ in } s_{\ell}) < P(h_1 \text{ avail})P(h_{k+1} \text{ in } s_{\ell}) +$

$$\sum_{k=1}^{n} \frac{D(k-1)}{2} \sum_{k=1}^{n} \frac{D(k-1$$

$$\sum_{\substack{\eta \in H_{\succ} \mid \\ k+1 \text{ in } s_{\ell}}} P(\eta | D^{\phi,\sigma}) \sum_{h_i \in S'} P(h_i \text{ avail}) \chi(h_{k+1} \succ_{\eta} h_i)$$

Starting with the assumption that assortative interviewing does not hold for plurality, we show that none of the inequalities above hold for any slot s_{ℓ} , and therefore that the condition in Lemma 9 does not hold for $j = h_1$ for any valuation function. \Box

THEOREM 21. Given an interviewing budget of k > 3 interviews, there exists $0 < \varepsilon < 1$ s.t. for any scoring function v no assortative interviewing forms an equilibrium for dispersion parameter $\phi < \varepsilon$.

PROOF. Thanks to Theorem 20, it is enough for us to shown there is no assortative equilibrium under plurality (and that h_1 violates Lemma 9's condition). We again begin with the simplification from Lemma 14: $P(h_j \text{ avail}) \geq \phi^{k-j+1}$. Thanks to Lemma 19, we know $P(h_j \text{ avail})$ is of the form:

$$P(h_j \text{ avail}) = \frac{X(k)}{Z^{k-1}} \phi^{\sum_{i=1}^{k-j} i} + \frac{X^1(k)}{Z^{k-1}} \phi^{1+\sum_{i=1}^{k-j} i} + \dots + \frac{X^{\ell}(k)}{Z^{k-1}} \phi^{(k\sum_{i=1}^{k-j} i)-1} + \frac{1}{Z^{k-1}} \phi^{k\sum_{i=1}^{k-j} i}$$
(9)

 $(X(k), X^1(k), \ldots, X^{\ell}(k))$ are functions that calculate the number of different sets of possible preference orders for r_1, \ldots, r_k , with each set being of probability $\phi^{\sum_{i=1}^{k-j} i}$ for $X(k), \phi^{1+\sum_{i=1}^{k-j} i}$ for $X^1(k)$, etc.)

When $\phi \to 0$, $Z^{k-1} \to 1$, and Equations 9 becomes $P(h_j \text{ avail}) \to X(k)\phi^{\sum_{i=1}^{k-j}i}$. In particular, there is ε' , such that $P(h_1 \text{ avail}) < X(k)\phi^{(\sum_{i=1}^{k-j}i)-1}$, and there is $\varepsilon = \min(\varepsilon', \frac{1}{X(k)})$ such that for $\phi < \varepsilon$, for k > 3:

$$\phi^k \ge \phi^{(\sum_{i=1}^{k-j}i)-2} > X(k)\phi^{(\sum_{i=1}^{k-j}i)-1} > P(h_1 \text{ avail})$$

Contradicting our condition (Equation 6). \Box

Moreover, we show that for k = 4, assortative interviewing is *never* an equilibrium.

THEOREM 22. Given an interviewing budget of k = 4 interviews and any scoring function, assortative interviewing is not an equilibrium for any dispersion parameter ϕ .

PROOF. We begin by instantiating the plurality valuation function. By Theorem 20, if assortative interviewing is not an equilibrium for plurality, it is never an equilibrium for any scoring rule. As noted before Eq. 6 is tight, so if we compute the marginal contribution from some $h^* \in \{h_1, h_2, h_3, h_4\}$, and the contribution from h^* is strictly less than the contribution from h_5 for any ϕ , assortative interviewing is not an equilibrium for k = 4 and plurality. We find that the contribution from h_1 is less than the marginal contribution from h_4 .

To calculate $P(h_1 \text{ avail})$, we simply iterate over all 6 possible allocations for r_1, r_2, r_3 such that h_1 is not taken, and directly calculate the probabilities of each ranking profile for r_1, r_2, r_3 that allows that to happen. In the interest of clarity, we only provide a symbolic representation. Let A be the set of all permutations of h_2, h_3, h_4 , so that $(a_1, a_2, a_3) \in A$.

$$P(h_1 \text{ avail}) = \sum_{(a_1, a_2, a_3) \in A} P(\mu(r_1) = a_1) P(\mu(r_2) = a_2 | \mu(r_1) = a_1) P(\mu(r_3) = a_3 | \mu(r_1) = a_1, \mu(r_2) = a_2)$$

We instantiate the above equation using the probabilities of each potential match, and use numerical methods to show the function $P(h_1 \text{ avail}) - \phi^4$ is negative for any ϕ in $0 < \phi \leq 1$. \Box

7. DISCUSSION

We investigate equilibria for interviewing (for example, between residents and hospitals) with a limited budget when a master ranked list (say, of residents) is known. We provide a generic payoff function, that is indifferent to participants' interviewing budgets, preference distributions, and scoring functions. We show that a pure strategy interviewing equilibrium always exists.

We then focus on this game for different scoring rules (Borda, plurality and exponential scoring rules), when residents' preferences are independently drawn from the same Mallows distribution. We find evidence that, for all scoring rules investigated, interviewing budgets typically seen in real-world markets do not admit assortative interviewing equilibria, even though this is a strategy frequently played in these markets. We do find that this is an equilibrium strategy for small interviewing budgets, when residents' preferences are sufficiently "similar" (i.e., low dispersion). Moreover, this assortative equilibrium strategy is a naturally arising equilibrium in which the maximum number of residents are matched; namely, the residents interview assortatively in tiers, forming a bipartite graph interviewing graph structure with n/k disconnected complete components. A similar bipartite graph interviewing structure is present in the work of Lee and Schwarz [16]. However, this structure naturally arises in our model, and we characterize a very different preference space than the Lee and Schwarz paper, which investigates the impartial culture model (i.e., a Mallows model with $\phi = 1$, or uniform distribution). We also provide an equilibrium for impartial culture in markets with master lists.

We hypothesize the difference in behavior seen in realworld markets and the equilibria shown here could result from a variety of factors. First, we only investigate assortative interviewing under a ϕ -Mallows model. As discussed in Section 3.3, while under some circumstances the Mallows model is viewed as a realistic model, it is possible participants' preferences in these markets are not sufficiently described by such a model. Another critical modeling assumption in this work is that of master lists, though assortative interviewing behavior is seen both in matching markets with and without master lists.

We also assume perfectly rational actors. Both prospect theory and quantal response equilibria could explain the difference in real-world behavior and the equilibria shown here. Individuals tend to misjudge probabilities, overestimating small probabilities, and underestimating near-certainties. (Perhaps leading them to believe assortative interviewing is a best response).

We hypothesize that, like in decentralized matching markets, the structure of the interviewing equilibria will contain both "reach" and "safety" schools, where participants diversify their interviewing portfolio to get both the benefit of a desirable, unlikely option, and a likely, but less desirable option. We find some evidence of this equilibrium in small markets with Borda valuations. Figure 1, depicts a market with 4 hospitals, 4 residents, and 2 interviews (n = 4, k = 2) and shows the explicit trade-off between high-value unlikely alternatives, and more choice over alternatives. The figure shows the exact payoff for each interviewing set, for given dispersion ϕ . As ϕ increases, we explicitly see the tradeoff between more choice, and a better expected payoff value for individual alternatives. For sufficiently large ϕ , choice dominates individual expectations so that for r_2 , interview-



Figure 1: r_2 's expected payoff for interviewing with various interviewing sets, as ϕ goes from 0 to 1, n = 4.

ing $\{h_3, h_4\}$ dominates interviewing with any other set. For small ϕ , interviewing with $\{h_1, h_2\}$ dominates interviewing with any other set.Interestingly, for $\phi \in [0.5, 0.6]$, r_2 's best option is to split the difference, and interview with one hospital (h_3) he is guaranteed to get and one hospital (h_2) that will be available with sufficiently high probability, and has a higher expected value. This choice available to r_2 results in some of the "reach" behavior we see in college admissions markets; r_3 's best response now is to interview with h_1, h_4 (i.e., a "reach" choice, and a "safe" bet). We hypothesize that this "reach" and "safety" behavior is not only present for larger ϕ in markets with small interviewing budgets, but also in markets with large interviewing budgets.

We hypothesize that results similar to the ones presented in this paper hold for different scoring functions and preference distributions (e.g., Plackett-Luce). Furthermore, the results presented here only investigate one-to-one matching markets. We believe that most of our results will directly hold for many-to-one markets where each hospital hhas known capacity q_h . Another interesting future direction would be to relax the assumption that interviewing with any hospital has identical cost. In this regard, we wish to investigate equilibria when each resident has a known budget k. and each resident r has some known cost $c_r(h)$ for interviewing with hospital h; residents must then choose an interviewing set S s.t. $\sum_{h \in S} c_r(h) \leq k$. Perhaps the most important direction for future work is relaxing the master list assumption; we hypothesize that similar equilibria arise if preferences on both sides of the market are distributed according to a Mallows model with low dispersion. We also believe this work could lead to interesting questions in mechanism design, where the mechanism is a joint interviewing/matching mechanism, with a limited budget for interviews explicitly incorporated into the mechanism.

Acknowledgements

This work was supported in part by NSERC Discovery Grant RGPIN 7631.

REFERENCES

- K. F. Ajayi. School choice and educational mobility: Lessons from secondary school applications in Ghana. University of California-Berkeley Working Paper, 2011.
- [2] K. D. Anderson, M. M. Dorough, C. R. Stein, S. A. Optenberg, I. M. Thompson, and others. The urology residency matching program in practice. *The Journal* of Urology, 163(6):1878–1887, 2000.
- [3] S. Bade. Serial dictatorship: The unique optimal allocation rule when information is endogenous. *Theoretical Economics*, 10(2):385–410, May 2015.
- [4] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.
- [5] H. Chade, G. Lewis, and L. Smith. Student portfolios and the college admissions problem. *The Review of Economic Studies*, 81(3):971–1002, 2014.
- [6] H. Chade and L. Smith. Simultaneous search. Econometrica, 74(5):1293–1307, 2006.
- [7] L. Chen and J. S. Pereyra. Self-selection in school choice: Theory and evidence from Mexico City high school match, 2015. Working paper.
- [8] P. Coles, A. Kushnir, and M. Niederle. Preference signaling in matching markets. Technical report, National Bureau of Economic Research, 2010.
- [9] J. Drummond and C. Boutilier. Elicitation and approximately stable matching with partial preferences. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI-13), Beijing, 2013.
- [10] J. Drummond and C. Boutilier. Preference elicitation and interview minimization in stable matchings. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-14), pages 645–653, 2014.
- [11] D. Gale and L. S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.
- [12] I. E. Hafalir. Stability of marriage with externalities. International Journal of Game Theory, 37(3):353–369, 2008.
- [13] R. W. Irving, D. F. Manlove, and S. Scott. The stable matching problem with master preference lists. *Discrete Applied Mathematics*, 156:2959–2977, 2008.
- [14] A. Kushnir. Harmful signaling in matching markets. Games and Economic Behavior, 80:209–218, July 2013.
- [15] G. Lebanon and Y. Mao. Non-parametric modeling of partially ranked data. *Journal of Machine Learning Research*, 9:2401–2429, 2008.
- [16] R. S. Lee and M. Schwarz. Interviewing in two-sided matching markets. Working Paper 14922, National Bureau of Economic Research, Apr. 2009.
- [17] G. F. Loewenstein, L. Thompson, and M. H. Bazerman. Social utility and decision making in interpersonal contexts. *Journal of Personality and Social Psychology*, 57(3):426–441, 1989.
- [18] T. Lu and C. Boutilier. Learning Mallows models with pairwise preferences. In *Proceedings ICML 2011*, pages 145–152, Bellevue, WA, 2011.

- [19] D. M. Messick and K. P. Sentis. Estimating social and nonsocial utility functions from ordinal data. *European Journal of Social Psychology*, 15:389–399, 1985.
- [20] T. B. Murphy and D. Martin. Mixtures of distance-based models for ranking data. *Computational Statistics and Data Analysis*, 41:645–655, January 2003.
- [21] M. Niederle, A. E. Roth, and T. Sonmez. Matching and market design. In S. N. Durlauf and L. E. Blume, editors, *The New Palgrave Dictionary of Economics* (2nd Ed.), volume 5, pages 436–445. Palgrave Macmillan, Cambridge, 2008.
- [22] N. R. M. Program. National resident matching program, results and data: 2015 main residency match[®], 2015.
- [23] B. Rastegari, A. Condon, N. Immorlica, and K. Leyton-Brown. Two-sided matching with partial information. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, pages 733–750. ACM, 2013.
- [24] A. E. Roth. The economist as engineer: Game theory, experimentation, and computation as tools for design economics. *Econometrica*, 70(4):1341–1378, July 2002.
- [25] M. Zhu. College admissions in China: A mechanism design perspective. *China Economic Review*, 30:618–631, Sept. 2014.

Budgeted Online Assignment in Crowdsourcing Markets: Theory and Practice

Pan Xu, Aravind Srinivasan Dept. of Computer Science University of Maryland, College Park, USA {panxu, srin}@cs.umd.edu

ABSTRACT

We consider the following budgeted online assignment problem motivated by crowdsourcing. We are given a set of offline tasks that need to be assigned to workers who come from the pool of types $\{1, 2, \dots, n\}$. For a given time horizon $\{1, 2, \dots, T\}$, at each instant of time t, a worker j arrives from the pool in accordance with a known probability distribution $[p_{jt}]$ such that $\sum_j p_{jt} \le 1$; j has a known subset N(j) of the tasks that it can complete, and an assignment of one task i to j (if we choose to do so) should be done before task *i*'s deadline. The assignment e = (i, j) (of task $i \in N(j)$) to worker j) yields a profit w_e to the crowdsourcing provider and requires different quantities of K distinct resources, as specified by a cost vector $\mathbf{a}_e \in [0, 1]^K$; these resources could be client-centric (such as their budget) or worker-centric (e.g., a driver's limitation on the total distance traveled or number of hours worked in a period). The goal is to design an online-assignment policy such that the total expected profit is maximized subject to the budget and deadline constraints.

We propose and analyze two simple linear programming (LP)based algorithms and achieve a competitive ratio of nearly $1/(\ell + 1)$, where ℓ is an upper bound on the number of non-zero elements in any \mathbf{a}_e . This is nearly optimal among all LP-based approaches. We also propose several heuristics adapted from our algorithms and compare them to other non-LP-based algorithms over a large set of random instances. Experimental results show that our LPbased heuristics significantly outperform the non-LP-based ones, sometimes by nearly 90%.

CCS Concepts

•Computing methodologies → Multi-agent systems;

Keywords

approximation algorithms; online algorithms; crowdsourcing market

1. INTRODUCTION

Crowdsourcing markets (e.g., Amazon Mechanical Turk or Crowdflower) have evolved to be powerful platforms that bring together task performers (or workers) and task requesters (or consumers). In recent years, problems arising from online decision making in such Kanthi K. Sarpatwar, Kun-Lung Wu IBM Thomas J. Watson Research Center Yorktown Heights, NY, USA {sarpatwa, klwu}@us.ibm.com

settings have been attracting tremendous attention (see the survey [37]). A typical problem arising in such settings, considered by [6], is to schedule a batch of consumer tasks using a pool of workers who become available in an online fashion (i.e., in real time). More specifically, we are given a set *I* of *offline* tasks, where each task $i \in I$ has a deadline d_i after which it cannot be scheduled. Workers arrive in an online fashion (according to an adversarial or random permutation order) and submit bids on a subset of tasks that interest them. When a worker *j* arrives, a decision must be made immediately and irrevocably - either assign it an available task or reject its service. If the worker *j* is allocated a task *i*, we must pay the worker their bid amount b_{ij} . The goal is to maximize the number of tasks assigned while constrained by a given bid budget of *B*. Our work deals with a natural variant of this problem.

As per standard notation, we use [n] to denote the set of integers $\{1, 2, ..., n\}$. Further let us assume a time horizon [T]. In this work, we model the arrival of workers as follows. At any given instant of time (referred to as *round*) $t \in [T]$, a single worker is chosen from a known pool of worker types [n] in accordance with a *known* probability distribution $[p_{jt}]$ such that $\sum_j p_{jt} \leq 1^1$ (noting that such a choice is made independently for each round t). Current related works in the domain of *mechanism design* for crowdsourcing markets mainly model the arrival pattern of online workers as either random arrival order (e.g., [7]) or *known independent identical distributions (i.i.d)* (e.g., [35, 36]). Our arrival setting can be viewed a natural generalization in the way that we allow the arrival distributions change over time. Notice that we do not consider if each worker will submit her bid truthfully when designing the allocating policy, which is one of the major concerns of mechanism design.

Another key distinction from the previous models is that we consider multiple budget constraints. That is, we assume that there are *K* distinct resources and that each assignment e = (i, j) has a bid cost vector $\mathbf{a}_e \in [0, 1]^K$, where the k^{th} component of the vector corresponds to the amount of resource type *k* needed by the assignment. These resources could be task-requester-centric (such as their budget) or worker-centric (e.g., a driver's limitation on the total distance traveled or number of hours worked in a period). Resource *k* is called integral if $a_{e,k} \in \{0,1\}$ for all *e*, otherwise we refer to it as *non-integral*. We note that [6] set a constraint that each task be assigned at most once, while [25] generalize it to the setting where each task may be assigned at most $b_i \in \mathbb{Z}_+$ times. Putting this in context with our work, these settings can be viewed as special cases of our setting as follows: each task itself is an integral resource with budget either 1 or b_i respectively.

Finally, instead of maximizing the throughput (i.e., number of tasks completed), each assignment e is associated with a *known*

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

¹Here we allow that with probability $1 - \sum_{j} p_{jt}$, none of the workers is chosen at *t*.

weight or utility w_e and we aim to maximize the expected utility collected from those successful assignments. [25] consider the case where the weights are unknown and must be learned, while workers arrive online in a random permutation order.

Our Contributions. We deal with several theoretical and practical aspects of the above budgeted online assignment problem (BOA), under the assumption that the arrival distribution is known in advance. Before discussing our contributions, we define a couple of useful parameters that can help appreciate our results better. Let ℓ_1 (resp. ℓ_2) denote the maximum number of integral (resp. non-integral) resources requested (in a non-zero amount) in any assignment cost vector \mathbf{a}_e .

First, we consider the simple and natural case where all the resources are integral and each assignment requests at most $\ell_1 = \ell$ resources. We present two simple LP-based algorithms, ALG₁ and ALG₂, that are non-adaptive and adaptive respectively. Here we say an online algorithm is adaptive if it somehow incorporates all information observed so far including online arrivals and outcomes of previous strategies to design the current strategy. In Section 5, we prove the following theorems.

Theorem 1.1. There exists an online non-adaptive algorithm ALG_1 for the BOA problem with a competitive ratio of $\frac{1}{\ell+1}(1-\frac{1}{\ell+1})^{\ell} \ge \frac{1}{e(\ell+1)}$, assuming all the resources are integral.

Theorem 1.2. There exists an online adaptive algorithm ALG_2 for the BOA problem with a competitive ratio of $(1 - \varepsilon)/(\ell + 1)$, for any given $\varepsilon > 0$, assuming all the resources are integral.

Our competitive ratio analysis is tight for the non-adaptive algorithm ALG₁ (as shown in Example 5.2). In other words, Theorem 1.1 states the best possible ratio that ALG₁ could get. Another notable point is that ALG₂ is nearly optimal among all LP (4.1)-based approaches, i.e., all possible algorithms using the LP (4.1) as a benchmark, since it has an integrality gap at least $\ell - 1 + 1/\ell$ ([21]). The main technical challenge facing us is to lower bound Pr[$\wedge_k S_k$] for a certain family of *negatively correlated* events { S_k }. We develop two different useful techniques (see Section 5.2 and 5.3) to tackle this challenge and use them to prove the optimality of our analyses.

Subsequently, we consider the general case of resources being both integral and non-integral (see Section 6) and show that the above theorems (i.e., Theorem 1.1 and 1.2) can be readily applied assuming that the budget of any non-integral resource is at least *moderately large* enough. More precisely, we prove these two theorems. Let *B* be the minimum budget for any non-integral resource.

Theorem 1.3. For the BOA problem, ALG₁ yields a competitive ratio of $\frac{1}{\ell_1+1}\left((1-\frac{1}{\ell_1+1})^{\ell_1}-\varepsilon\right)$, for any $\varepsilon > 0$, assuming $B \ge 2\ln(\frac{\ell_2}{\varepsilon})\left(1+\frac{3\ell_1+2}{\ell_1^2}\right)+2$.

Theorem 1.4. For the BOA problem, ALG₂ yields a competitive ratio of $\frac{1-2\varepsilon}{\ell_1+1}$ for any given $\varepsilon > 0$, assuming $B \ge 3 \ln(\frac{\ell_2}{\varepsilon})(1+\frac{1}{\ell_1})+2$.

In the proof of Theorem 1.4, we apply the technique of *vir*tual algorithms to tackle the technical challenge of upper bounding $\Pr[\sum_i X_i \ge (1 + \delta) \mathbb{E}[\sum_i X_i]]$ for a family of *positively correlated* random variables $\{X_i \in [0, 1]\}$. Our results show that the knowledge about arrival distributions holds a significant edge over the adversarial model or the random permutation model. Let us compare our results with those of [6]. As discussed before, their setting fits our model when $\ell_1 = \ell_2 = 1$. From Theorem 1.4, we obtain a $(\frac{1}{2} - \epsilon)$ competitive ratio assuming $B \ge 12 \ln(1/\epsilon)$ while [6] obtain a ratio of $O(\frac{1}{R^{\epsilon} \ln R})$, assuming $B \ge \frac{R}{\epsilon}$ and $R \doteq \frac{\max b_{i,j}}{\min b_{i,j}}$ (i.e., the ratio of the largest bid to the smallest bid over all possible assignments). In fact, we completely remove the dependency on R and obtain a constant ratio while relaxing the lower bound assumption on B significantly. Our result may be seen as theoretical evidence to advocate the use of historical data to learn arrival distributions.

Finally, we propose various LP-based heuristics and evaluate them against certain non-LP-based approaches. Our experiments show that the LP-based approaches yield a far superior competitive ratio compared to LP-blind approaches, sometimes by nearly 90%.

2. RELATED WORK

As an offline version of our model, the classic column-sparse packing (CSP) problem has been well studied in the theoretical computer science community. The basic setting is as follows: we are given *n* items and *K* resources; each item $i \in [n]$ has a size vector $\mathbf{a}_i \in [0, 1]^K$ and a profit $w_i > 0$; given a budget $\mathbf{B} \in \mathbb{R}_+^K$, the goal is to choose a subset of items such that the total profit is maximized without violating the budget constraints. More generally, our offline model is reduced to the Multidimensional Knapsack problem (MKP) when there is no restriction on the sparsity of each size vector \mathbf{a}_i . We see that the offline model such as MKP can fit our online model as a special case when T = n and $p_{jt} = 1$ iff j = t and 0 otherwise for all $j \in [n], t \in [T]$. Note that we have more restrictions here: we are not allowed to look at all the items before making decisions; instead we have to make an instant irrevocable decision whenever an item comes. Many common techniques such as *permutation of* all items [8] and alteration [9] shown useful in the offline setting, are not applicable to our online problems. Notice that any hardness result from CSP problem will also apply here though.

We now briefly describe several recent results for the CSP problem. Let k be the column sparsity, i.e., the number of non-zero entries in any column vector (this is equivalent to the parameter ℓ in our BOA problem). For the general case when each $\mathbf{a}_i \in [0, 1]^K$, [9] gave a randomized algorithm with the approximation ratio of 1/(ek + o(k)) and constructed an instance showing integrality gap of even a strengthened LP to be at least 2k - 1. As a special case of CSP when all \mathbf{a}_i are binary and **B** is integral, the k-set packing problem is extensively studied before [11, 24, 26, 5]. Note that [21] showed that the natural LP relaxation for k-set packing has an integrality gap at least k - 1 + 1/k. [8] considered a stochastic version where each \mathbf{a}_i is a binary-vector valued random variable with each outcome having at most k non-zero elements. They obtained a 2k-approximation algorithm. Further, they presented a (k + 1)-approximation algorithm as well when each \mathbf{a}_i has monotone outcomes. An important variant of the stochastic CSP problem, known as the stochastic matching problem, arises with k = 2 and has received considerable attention recently [14, 10, 1, 22].

As for our online model, there is a long line of research related to our problem: online bipartite matching and its variants, which are motivated by applications to online advertisement business. Two notable special cases, Adwords and Display Ads, have been studied extensively in recent years: [13, 15, 16, 17, 18, 33]. Both these models can be modeled as each assignment consumes only a single integral (Display Ads) or non-integral (Adwords) resource even though the potential number of distinct resources can be huge. More recently, [28] considered a natural generalization of Adwords, where there are multi-tier budgets forming a laminar structure. Regarding the online arrival assumption, there are three main categories: adversarial, random arrival order, and known distributions (see the book [32] for more details). A majority of the recent work under known distributions, focuses on the case when when the distributions in each round are independent and identical (referred to as *known i.i.d*) [19, 23, 31, 27, 12]. We refer to another variant of known distributions where the distributions can change over rounds as the *known adversarial*. Note that this is the setting we consider here and is more general compared to the *known i.i.d* as descried before. For this setting, [4] considers the online stochastic generalized assignment problem while [3] considers the online prophet-inequality matching problem. Note that most of the current online-matching models under known distributions can fit into our model as a special case except the fact that some assume the cost or bid is a random variable while we model it as deterministic here. There are several papers considering online packing LP problem as well under a random permutation order [29, 2]. To be specific, [29] presented an algorithm achieving a $(1 - \epsilon)$ -competitive ratio, provided $B = \Omega(\ln(\ell)/\epsilon^2)$, where *B* is the largest capacity ratio and ℓ is the cost-vector sparsity.

3. PROBLEM STATEMENT

In this section, we present a formal statement of our problem. Let $I = \{i \in [m]\}$ be the set of offline tasks and $J = \{j \in [n]\}$ be the set of online workers. On a finite time horizon T, each task *i* has a deadline $d_i \in [T]$ after which it will become unavailable. Let G = (I, J, E) be the bipartite graph that models the relation between the tasks and workers: there is an edge e = (i, j) iff worker *j* is interested in the task *i*. Let $N(j) = \{i : (i, j) \in E\}$ be the set of tasks that interest worker j and $N(i) = \{j : (i, j) \in E\}$ be the set of workers who are interested in task *i*. Each edge e = (i, j)has a weight w_e denoting the profit obtained by assigning task *i* to worker j. Each assignment e = (i, j) has a requirement for one or more of a given set of K types of resources. The requirement of an assignment e is given by a K-dimensional vector $\mathbf{a}_e \in [0, 1]^K$, where the k^{th} dimension $a_{e,k}$ represents the amount of resource k needed. Each resource type k has a budget $B_k \in \mathbb{R}_+$ that must not be violated. For each e, let $S_e = \{k \in [K] : a_{e,k} > 0\}$, i.e., the set of resources it requests.

At any instant $t \in [T]$, a worker j arrives with a probability p_{jt} such that $\sum_j p_{jt} \le 1$ (thus, with probability $1 - \sum_j p_{j,t}$, no worker arrives at time t). Let $E_{j,t} = \{e = (i, j), i \in N(j) : d_i \ge t\}$ denote the set of *available* assignments for the worker j at time t. In this paper, we assume without loss of generality that each task can be assigned for an arbitrary number of times before its deadline. Any potential restriction on the number of assignments can easily be modeled by an additional budget constraint: the task itself is an integral resource and the corresponding budget is the upper bound on the number of assignments. For each $e \in E_{i,t}$, we say e is safe or valid iff for each $k \in S_e$, resource k has remaining budget larger or equal to $a_{e,k}$. When a worker j arrives at t, we have to make an immediate and irrevocable decision: either reject it or choose a safe option $e \in E_{j,t}^2$ and get a resultant profit w_e . Once a safe assignment *e* is scheduled, the budget of each resource $k \in S_e$ will be reduced by $a_{e,k}$. Our goal is to design an online assignment policy such that the expected profit is maximized.

In most applications, we need to deal with two kinds of resources, namely integral and non-integral. A resource *k* is integral if $a_{e,k} \in \{0, 1\}$ for all $e \in E$ and $B_k \in \mathbb{Z}_+$. On the other hand a resource *k* is non-integral if $a_{e,k} \in [0, 1]$ and $B_k \in \mathbb{R}_+$. This captures resources such as money and time that cannot be quantified as integral. Let $\mathcal{K}_1 = \{1, 2, \dots, K_1\}$ and $\mathcal{K}_2 = \{K_1 + 1, \dots, K_1 + K_2\}$ denote the set of integral and non-integral resources respectively. As defined in the introduction, for each assignment e, $|S_e \cap \mathcal{K}_1| \leq \ell_1$ and $|S_e \cap \mathcal{K}_2| \leq \ell_2$.

4. BENCHMARK LP

For an online algorithm ALG, the competitive ratio is defined as the ratio of the expected performance of ALG to the expected offline optimal over all possible realizations. A common technique is to use an LP (we called benchmark LP) to upper bound the latter value, thereby obtaining a lower bound on the competitive ratio.

Recall that $E_{j,t}$ is the set of available assignments for a worker j arriving at t. For any t, let $E_t = \bigcup_j E_{j,t}$ be the set of all available assignments at t. Further, for each t and $e \in E_t$, let $x_{e,t}$ be the probability that we make the assignment e at t in the offline optimal solution. Our benchmark LP can now be described as follows:

maximize
$$\sum_{t} \sum_{e \in E_t} w_e x_{e,t}$$
 (4.1)

subject to
$$\sum_{e \in E_{j,t}} x_{e,t} \le p_{jt}$$
 $\forall j \in J, t \in [T]$ (4.2)

$$\sum_{t} \sum_{e \in E_{t}} x_{e,t} a_{e,k} \le B_{k} \quad \forall k \in [K]$$
(4.3)

$$0 \le x_{e,t} \le 1 \qquad \qquad \forall e \in E, t \in [T] \qquad (4.4)$$

Lemma 4.1. The optimal value to LP(4.1) is a valid upper bound for the offline optimal.

Our benchmark LP is essentially the same as that used in [3] and [4]. The detailed proof can be found there. We provide a rough proof here.

Proof. The simple idea is to show that all the constraints in the above LP are valid for the offline optimal. For each given *t* and worker *j*, $\sum_{e \in E_{j,t}} x_{e,t}$ can be interpreted as the sum of the expected number of assignments related to *j* we could make in the offline optimal, which is surely no larger than the probability that *j* comes at *t*. This justifies constraints (4.2). Any offline algorithm should satisfy the budget constraints as well and by linearity of expectation, we see constraints (4.3) are valid.

5. THE CASE OF INTEGRAL RESOURCES

In this section, we consider the case when $K_2 = 0$, i.e., all resources are integral with $a_{e,k} \in \{0,1\}$ and $B_k \in \mathbb{Z}_+$ for all $e \in E$ and $k \in [K]$. Let $\ell_1 = \ell$, i.e., each assignment requests at most ℓ (integral) resources.

As shown in Section 2, the *k*-set packing problem can be reformulated as a special case here. Thus from [21], it follows that even for the special case of unit budget, i.e., $B_k = 1$ for all $k \in [K]$, LP (4.1) has an integrality gap at least $\ell - 1 + 1/\ell$. That implies by using the LP (4.1) as the benchmark, we cannot get an online algorithm achieving a ratio beating $1/(\ell - 1 + 1/\ell)$.

5.1 A simple non-adaptive algorithm

In this section, we present a simple LP-based non-adaptive algorithm. Suppose $\{x_{e,t}^* | t \in [T], e \in E_t\}$ is an optimal solution for the LP (4.1). The main idea behind our algorithm (described in Algorithm 1) is as follows: at each time t when some worker j arrives, if safe make the assignment $e \in E_{j,t}$ with probability $\alpha x_{e,t}^*/p_{j,t}$, where $\alpha \in (0, 1]$ is a parameter that will be optimized later.

We note that the last step of Algorithm 1 is well defined because $\sum_{e \in \hat{E}_{j,t}} \alpha x_{e,t}^* / p_{jt} \leq \sum_{e \in E_{j,t}} x_{e,t}^* / p_{jt}$, which is at most 1.

Theorem 5.1. By choosing $\alpha = \frac{1}{2\ell}$, ALG₁ achieves an online competitive ratio of at least $\frac{1}{4\ell}$.

Proof. WLOG assume that t = T and fix an assignment $e \in E_T$. Recall that S_e is the set of resources requested by e. For each $k \in S_e$,

²In the case when some worker j can accept multiple assignments each time, say L, we can simply add L copies of j to our graph G.

Algorithm 1: A simple non-adaptive algorithm (ALG₁)

1 For each time *t*, assume some worker *j* arrives.

- 2 Let $\hat{E}_{j,t} \subseteq E_{j,t}$ be the set of *safe* available assignments we can make for *j*.
- 3 If $\hat{E}_{j,t} = \emptyset$, then reject *j*; otherwise sample at most one assignment $e \in \hat{E}_{j,t}$ with probability $\alpha x_{e,t}^*/p_{jt}$.

let S_k be the event that e is safe at T with respect to a resource k. We now lower bound the value $\Pr[\wedge_{k \in S_e} S_k]$. Fix one such $k \in S_e$. Let U_k be the usage of resource k at the beginning of t = T and $X_{e',t'}$ be the indicator random variable for assignment $e' \in E_{t'}$ chosen at $t' \in [T-1]$. We have $U_k = \sum_{t' < T} \sum_{e' \in E_{t'}} X_{e',t'}a_{e',k}$. By definition, e is safe with respect to resource k iff $U_k \leq B_k - 1$. Observe that $\mathbb{E}[X_{e',t'}] \leq \alpha x_{e',t'}^*$. By Markov inequality we see

$$\Pr[U_k \le B_k - 1] = 1 - \Pr[U_k \ge B_k] \ge 1 - \alpha$$
 (5.1)

Thus we get

$$\Pr[\wedge_{k \in S_e} S_k] = \Pr\left[\bigwedge_{k \in S_e} \left(U_k \le B_k - 1\right)\right] \ge 1 - \ell\alpha \qquad (5.2)$$

So we get that for the given (e, t), e will be made with probability at least $\alpha x_{e,t}^*(1-\ell\alpha)$. By setting $\alpha = \frac{1}{2\ell}$, we get that each assignment e is made with probability at least $x_{e,t}^*/(4\ell)$.

5.2 A tight analysis for ALG₁ with unit budget

In this section, we consider a special case when $B_k = 1$ for all $k \in K$ and show a *tight* analysis for ALG₁. Consider the following example.

Example 5.1. Consider an unweighted star graph G = (I, J, E)where $|I| = 1, |J| = 3, E = (e_1, e_2, e_3)$ with T = 2 and $d_1 = T$ (no deadline constraints). Suppose at t = 1, j = 1, 2 arrives with equal probability 1/2 and at t = 2, j = 3 will arrive with probability 1. Let e_1, e_2, e_3 denote respectively the assignment we consider when j = 1 comes at t = 1, j = 2 comes at t = 1 and j = 3 comes at t = 2. Let K = 2 with $\mathbf{B} = (1, 1)$ and $\mathbf{a}_{e_1} = (1, 0), \mathbf{a}_{e_1} = (0, 1)$ and $\mathbf{a}_{e_3} = (1, 1)$. Suppose LP (4.1) offers us such an optimal solution: $x_{e_1}^* = x_{e_2}^* = 1/2$ and $x_{e_3}^* = 1/2$ (notice that unweighted). Let us analysis the assignment e_3 when j = 3 comes at t = 2 by running ALG₁.

According to ALG_1 , at t = 1 we will choose e_j with probability α whenever j = 1 or j = 2 comes. Notice that at t = 2, the first and the second resource are each safe with respective probability $1 - \alpha/2$ and both of the two are safe with probability $1 - \alpha$. \Box

The above example suggests us two things: (1) the events that two different resources are safe can be *negatively* correlated. This means we can not apply the FKG inequality which is widely used in the offline version [9, 8, 10] to replace the union bound in inequality (5.2); (2) we could potentially strengthen the lower bound that each resource is safe, which is currently obtained by Markov inequality (5.1). Now we follow these ideas to present a tight analysis for ALG₁ for the case of unit budget.

Theorem 5.2. By choosing $\alpha = \frac{1}{\ell+1}$, ALG₁ has an online competitive ratio of $\frac{1}{\ell+1}(1-\frac{1}{\ell+1})^{\ell}$ with unit budget.

Proof. As before, we consider the case that t = T and an assignment $e \in E_T$. For each t' < T and $k \in S_e$, let $E_{k,t'} = \{e' | e' \in E_{t'}, S_{e'} \in E_{t'}\}$

k} be the set of assignments which are available at *t'* and participate in the budget constraint of *k*. Let $B_{k,t'}$ be the (random) budget of *k* at the beginning of *t'*. Define $A_{k,t'} = (B_{k,t'+1} = 1|B_{k,t'} = 1)$ and

$$A_{t'} = \wedge_{k \in S_e} A_{k,t'} = \left(\wedge_{k \in S_e} B_{k,t'+1} = 1 | \wedge_{k \in S_e} B_{k,t'} = 1 \right)$$

We see that

$$\Pr[A_{k,t'}] = 1 - \sum_{e' \in E_{k,t'}} \alpha x_{e',t'}^*, \Pr[A_{t'}] \ge 1 - \sum_{e' \in \bigcup_{k \in S_e} E_{k,t'}} \alpha x_{e',t'}^*$$

It follows that

$$\Pr[\wedge_{k \in S_e} \mathcal{S}_k] = \prod_{t' < t} \Pr[A_{t'}] \ge \prod_{t' < t} \left(1 - \sum_{e' \in \bigcup_{k \in S_e} E_{k,t'}} \alpha x_{e',t'}^* \right)$$
(5.3)

The above inequality can be made tight when $\{E_{k,t'} | k \in S_e\}$ is disjoint for each t'. Here are two useful observations. The first one is $\sum_{e' \in \cup_k \in S_e} E_{k,t'} \alpha x_{e',t'}^* \leq \sum_{e' \in E_{t'}} \alpha x_{e',t'}^* \leq \alpha$. The second one is

$$\sum_{t' < T} \sum_{e' \in \cup_{k \in S_e} E_{k,t'}} \alpha x^*_{e',t'} \leq \sum_{k \in S_e} \sum_{t' < T} \sum_{e' \in E_{k,t'}} \alpha x^*_{e',t'} \leq \alpha \ell$$

These two observations lead to the fact that the rightmost expression of inequality (5.3) has a minimum value of $(1-\alpha)^{\ell}$. Therefore *e* will be made at *t* with overall probability $x_{e,t}^* \alpha (1-\alpha)^{\ell}$. By choosing $\alpha = 1/(\ell + 1)$, we prove our claim.

The example below shows the above analysis is tight.

Example 5.2. Consider a star graph G = (I, J, E) where $|I| = 1, |J| = \ell + 1, E = \{e_j | j \in [J]\}$ with T = J. Let $d_1 = T$, i.e., no deadline constraints. For each $t \in [T]$, $p_j = 1$ if j = t and 0 otherwise. In other words, at each time $t \in [T]$, only worker j = t will come surely and no one else. Suppose we use \mathbf{a}_j and x_j^* to denote the terms \mathbf{a}_{e_j} and $x_{e_j,t=j}^*$ before. Let $K = \ell$ with $\mathbf{B} = \mathbf{1}$ (dimension of K) and $\mathbf{a}_j = \mathbf{e}_j$ for each $j \leq \ell$, where \mathbf{e}_j is the *j*th standard-basis unit vector, and $\mathbf{a}_j = \mathbf{1}$ for $j = \ell + 1$. Suppose LP (4.1) offers us such an optimal solution: $x_j^* = 1 - \epsilon$ for each $j \leq \ell$ and $x_{\ell+1}^* = \epsilon$.

Now focus on the assignment $e = e_J$ when j = J comes at t = T. Let us analyze the probability that e is safe at T, denoted by $\Pr[S_{e,T}]$, in ALG_1 with some parameter $\alpha \in (0, 1)$. Notice that e will be safe at t = T iff none of $e_j, j \leq l$ is made before. According to ALG_1 , each time $t, e_{j=t}$ will be made with probability equal to $\frac{\alpha x_j^*}{p_j} = \alpha(1 - \epsilon)$. That implies $\Pr[S_{e,T}] = (1 - \alpha(1 - \epsilon))^{\ell}$, which matches our lower bound as shown in the proof of Theorem 5.2. \Box

5.3 A tight analysis for ALG1 with general integral budget

In Section 5.2, we give a tight analysis for ALG_1 for the case of unit budget. Intuitively, we should be in a better situation when each B_k is larger than 1. For example, by the Chernoff bound, we see that the probability that the usage of resource k at T overflows B_k should decrease exponentially as B_k gets larger. In this section, we give a tight analysis for ALG_1 by extending the result in Theorem 5.2 to the case of general integral budget.

Let us present an equivalent but simpler model of our problem. Suppose we have *K* types of balls and for each type $k \in [K]$, the number of balls is $B_k \in \mathbb{Z}_+$. We have a set of choices $E = \{e|e \in E\}$ and each choice is associated with a binary vector $\mathbf{a}_e \in \{0, 1\}^K$, which has at most ℓ non-zero elements. Once we make the choice *e*, we will take one ball of type *k* whenever $a_{e,k} = 1$. For each time $t \in [T]$, one choice e will arrive with probability $x_{e,t}^*$ such that $\sum_{e \in E} x_{e,t}^* \leq 1$ for each t. Each time t, for whatever choice comes, we will accept it non-adaptively with some probability $\alpha \in (0, 1)$. Consider a fixed choice e and t = T and let $S_e \subseteq K$ be the set of types of balls choice e will take. For each $k \in S_e$, let S_k be the event that at t = T, we still have at least one ball of type k left. Our question is that how the adversary minimize $\Pr[\wedge_{k \in S_e} S_k]$ subject to the constraints $(1) \sum_{t \in [T-1]} x_{e,t}^* a_{e,k} \leq B_k$ for each $k \in S_e$ and $(2) \sum_{e \in E} x_{e,t}^* \leq 1$ for each t. The equivalence between this new model and our original problem can be seen as follows: (1) each assignment corresponds a choice here; (2) for some assignment e with deadline t, we set $x_{e,t'}^* = 0$ for all t' > t. Thus we can safely ignore the deadline issue as far as ALG₁ is considered.

Consider a given $k \in S_e$. Let $E_k = \{e \in E | a_{e,k} = 1\}$ be the set of choices *e* that participate in the resource constraint of *k*. Let $x_{k,t}^* = \sum_{e \in E_k} x_{e,t}^*$. Notice that $x_{k,t}^* \leq 1$ and at time *t*, one of the choices in E_k arrives with probability $x_{k,t}^*$. Let $A_{k,t}$ be the indicator random variable that one of the choices in E_k arrives at *t* and $A_k = \sum_{t \leq T-1} A_{k,t}$, which denotes the random number of arrivals of choices in E_k over T - 1 rounds. For an integral *A* and *B*, let $p(A, \alpha, B) \doteq \Pr[Z \leq B - 1]$ where $Z \sim \operatorname{Bi}(A, \alpha)$ (binomial distribution) and we assume $p(A, \alpha, B) = 1$ for any $0 \leq A \leq B - 1$. Now consider a given set $A = \{A_k | k \in S_e\}$.

Lemma 5.1.

$$\Pr[\mathcal{S}_k|A_k] \ge p(A_k, \alpha, B_k), \quad \Pr[\wedge_{k \in S_e} \mathcal{S}_k|A] \ge \prod_{k \in S_e} p(A_k, \alpha, B_k)$$

Proof. Consider a given *k* and A_k . Given A_k trials and each time we take one ball independently with probability at most α . Thus we end at at least $B_k - 1$ balls with probability at least $p(A_k, \alpha, B_k)$. Notice that the events $\{(S_k|A_k)|k \in S_e\}$ are positively correlated by the FKG inequality [20], which yields the second inequality. \Box

Lemma 5.2.

$$\Pr[\wedge_{k \in S_e} S_k] \ge \prod_{k \in S_e} \exp\left(\mathbb{E}\left[\ln(p(A_k, \alpha, B_k))\right]\right)$$

Proof. First notice that $\Pr[\wedge_{k \in S_e} S_k] = \mathbb{E}_A \left[\Pr[\wedge_{k \in S_e} S_k | A] \right]$ by conditioning on the event *A*. From Lemma 5.1, we see the latter should be at least $\mathbb{E}_A \left[\prod_{k \in S_e} p(A_k, \alpha, B_k) \right]$. Thus

$$\Pr[\wedge_{k \in S_{e}} S_{k}] = \mathbb{E}_{A} \left[\Pr[\wedge_{k \in S_{e}} S_{k} | A] \right] \ge \mathbb{E}_{A} \left[\prod_{k \in S_{e}} p(A_{k}, \alpha, B_{k}) \right]$$
$$= \mathbb{E}_{A} \left[\exp \left(\sum_{k} \ln(p(A_{k}, \alpha, B_{k})) \right) \right] \ge \exp \left(\sum_{k} \mathbb{E} \left[\ln(p(A_{k}, \alpha, B_{k})) \right]$$
$$= \prod_{k \in S_{e}} \exp \left(\mathbb{E} \left[\ln(p(A_{k}, \alpha, B_{k})) \right] \right)$$

The inequality in the second line to the third line is due to Jensen's inequality. Recall that $A_k = \sum_{t \leq T-1} A_{k,t}$ where $A_{k,t}$ is a Bernoulli random variable indicating if a choice $e \in E_k$ arrives at *t*. Notice that $\mathbb{E}[A_k] = \sum_{t \leq T-1} x_{k,t}^* \leq B_k$.

Lemma 5.3. For any $\alpha \in [0, \frac{1}{2}]$ and integer $B_k \ge 1$,

$$\mathbb{E}_{A_k}[\ln(p(A_k, \alpha, B_k))] \ge \ln(1 - \alpha)$$

We can show that in the worst scenario, the adversary will designate each A_k as a Poisson random variable with mean B_k such that $\mathbb{E}_{A_k}[\ln(p(A_k, \alpha, B_k))]$ gets minimized. The full proof of Lemma

5.3 can be seen in the full version. Now we have all ingredients to prove Theorem 1.1.

Proof. The proof is very similar to that of Theorem 5.2. Consider a given assignment *e* and t = T - 1 w.l.o.g. Notice that $\alpha = \frac{1}{\ell+1} \leq \frac{1}{2}$. From Lemma 5.2 and 5.3, we see that $\Pr[\wedge_{k \in S_e} S_k] \geq (1 - \alpha)^{\ell}$. Thus by plugging in $\alpha = \frac{1}{\ell+1}$, we prove our claim.

5.4 Simulation-based adaptive algorithm

In this section, we present a simulation-based algorithm. The main idea is as follows. Suppose we aim to develop an online algorithm achieving a ratio of $\gamma \in [0, 1]$. Consider an assignment $e = (i, j) \in E_t$ when worker *j* arrived at some time *t*. Let $S_{e,t}$ be the event that *e* is safe conditioning on the arrival of *e* at *t*. By simulating the current strategy up to *t*, we can get an estimation of $\Pr[S_{e,t}]$, say $\beta_{e,t}$, within an arbitrary small error. Therefore in the case *e* is safe at *t*, we can sample it with probability $\frac{x_{e,t}}{p_{j,t}} \frac{\gamma}{\beta_{e,t}}$, which leads to the fact that *e* is sampled with probability $\gamma x_{e,t}$ unconditionally.

The simulation-based attenuation technique has been used to attack other stochastic optimization problems as well such as stochastic knapsack [30] and stochastic matching [1]. Assume for now we can always get an accurate estimation $\beta_{e,t}$ of $\Pr[S_{e,t}]$ for all tand e (It is easy to see that the sampling error can be folded into a multiplicative factor of $(1 - \epsilon)$ in the competitive ratio by standard Chernoff bounds). The formal statement of our algorithm, denoted by ALG₂, is as follows.

Algorithm 2: Simulation-based adaptive algorithm (ALG ₂)	
--	--

- 1 For each time t, assume some worker j arrives.
- 2 Let $\hat{E}_{j,t} \subseteq E_{j,t}$ be the set of *safe* available assignments we can make for *j*.
- 3 If $\hat{E}_{j,t} = \emptyset$, then reject *j*; otherwise sample an assignment

 $e \in \hat{E}_{j,t}$ with probability $\frac{x_{e,t}^*}{p_{j,t}} \frac{\gamma}{\beta_{e,t}}$.

Note that $\beta_{e,t}$ is the value of $\Pr[S_{e,t}]$, which assumes to be known exactly through simulation. To ensure the above algorithm works with parameter γ , it suffices to show that $\beta_{e,t} \ge \gamma$ for all possible *t* and *e*.

Lemma 5.4. By choosing $\gamma = 1/(\ell + 1)$, we have $\beta_{e,t} \ge \gamma$ for all $t \in [T]$ and $e \in E_t$.

Proof. The proof is similar to that of Theorem 5.1. Consider a given t and $e \in E_t$. Focus on a given $k \in S_e$ and let $U_{k,t}$ be the usage of resource k at the beginning of t. For each t' < t and $e' \in E_{t'}$, let $X_{e',t'}$ be the indicator random variable that e' is chosen at t'. Notice that $U_{k,t} = \sum_{t' < t} X_{e',t'} a_{e',k}$.

Now we prove by induction on *t*. For the base case t = 1, we see $\beta_{e,t} = 1$ for all $e \in E_t$. Thus we claim is valid. Assume our claim works for all t' < t, which leads to the fact that for all $e' \in E_{t'}$ with t' < t, e' will be made at t' with probability *exactly* equal to $x_{e',t'}^* \gamma$. In other words, $\mathbb{E}[X_{e',t'}] = x_{e',t'}^* \gamma$. Consider the event that e is safe at t with respect to resource k. By Markov's inequality, we have

$$\Pr[U_{k,t} \le B_k - 1] = 1 - \Pr[U_{k,t} \ge B_k] \ge 1 - \gamma$$

Thus we have

$$\Pr[S_{e,t}] = \Pr\left[\bigwedge_{k \in S_e} \left(U_{k,t} \le B_k - 1\right)\right] \ge 1 - \ell \gamma \ge \gamma$$

The last inequality is valid since $\gamma \leq 1/(\ell + 1)$.

The above Lemma validates ALG₂. By manipulating the simulation error in a proper way as shown in [1, 30], we can make sure that final ratio will have a relative error at most ε for any given $\varepsilon > 0$. Thus we prove our claim for Theorem 1.2. Note that the running time will depend on $1/\varepsilon$ polynomially.

6. EXTENSION TO COMBINED INTEGRAL AND NON-INTEGRAL RESOURCES

Recall that \mathcal{K}_2 is the set of non-integral resources and for each $k \in \mathcal{K}_2$, all $a_{e,k} \in [0, 1]$. Let $B = \min_{k \in [K_2]} B_k$ and we assume *B* is large. In this section, we discuss how to extend the results in Section 5 here when non-integral resources are added with the large *B* assumption. In particular, we are interested in how large *B* should be such that we lose at most ϵ in the competitive ratio. By default we assume $\mathcal{K}_1 \neq \emptyset$ and $\ell_1 \ge 1$.

6.1 Extension of ALG₁

In this section, we analyze the performance of ALG₁ with parameter $\alpha = 1/(\ell_1 + 1) \le 1/2$ when non-integral resources are added. Recall that in ALG₁, each assignment *e* is made at *t* non-adaptively with probability at most $\alpha x_{e,t}^*$. Let $X_{e,t}$, $Y_{e,t}$ indicate if *e* is made at *t* and if *e* is safe at *t* respectively. Let $Z_{e,t}$ indicate if *e* comes *and* gets sampled at *t* when *e* is safe at *t*. Here we treat $Z_{e,t}$ is Bernoulli random variable with mean $\alpha x_{e,t}^*$ and *independent* from $Y_{e,t}$ in the following way: when *e* comes at *t* while *e* is not safe, we continue to set $Z_{e,t} = 1$ with probability $\alpha x_{e,t}^*/p_{j,t}$ and 0 otherwise, i.e., pretending *e* is safe. Observe that (1) $X_{e,t} = Y_{e,t}Z_{e,t} \le Z_{e,t}$; (2) For any two random variables $Z_{e,t}$ and $Z_{e',t'}$, the two will be independent if $t \neq t'$ and negatively correlated if t = t'. Now we start to prove Theorem 1.3.

Proof. Focus on a given t and an assignment $e \in E_t$. Let $S_1 = S_e \cap \mathcal{K}_1$ and $S_2 = S_e \cap \mathcal{K}_2$. Let $S_{k,t}$ be the event that e is safe with respect to resource k at t. From the analysis of Theorem 1.1, we see that $\Pr[\wedge_{k \in S_1} S_{k,t}] \ge (1 - \alpha)^{\ell_1}$. Now we focus on analyzing the value $\Pr[\wedge_{k \in S_2} S_{k,t}]$. Let $U_{k,t}$ be the usage of resource k at the beginning of t, i.e., $U_{k,t} = \sum_{t' < t} \sum_{e' \in E_t} X_{e',t'} a_{e',k}$.

Notice that for each $k \in S_2$,

$$\Pr[S_{k,t}] \ge 1 - \Pr[U_k \ge B_k - 1] \\ = 1 - \Pr[\sum_{t' < t} \sum_{e' \in E_{t'}} X_{e',t'}a_{e',t'} \ge B_k - 1] \\ \ge 1 - \Pr[\sum_{t' < t} \sum_{e' \in E_{t'}} Z_{e',t'}a_{e',t'} \ge B_k - 1]$$

Let $H_{k,t} = \sum_{t' < t} \sum_{e' \in E_{t'}} Z_{e',t'} a_{e',t'}$. Notice that (1) $\mathbb{E}[H_{k,t}] \le \alpha B_k$ and (2) former discussion shows that $\{Z_{e',t'}|e' \in E_{t'}, t' < t\}$ are 1-correlated as defined in [34]. Thus from there, we can effectively view them as "independent" and apply the Chernoff bound to upper bound the value $\Pr[Z_{k,t} \ge B_k - 1]$. WLOG assume $B_k = B$ and we have

$$\begin{split} \Pr[H_{k,t} \geq B - 1] &\leq & \exp\left(\frac{-\alpha B(\frac{B-1}{\alpha B} - 1)^2}{\frac{B-1}{\alpha B} + 1}\right) \\ &= & \exp\left(-\frac{1 - \alpha - 1/B}{1 + \alpha - 1/B}(B(1 - \alpha) - 1)\right) \\ &\leq & \exp\left(-\frac{1}{2}\frac{1 - \alpha}{1 + \alpha}(B(1 - \alpha) - 1)\right) \end{split}$$

To get the last inequality we assume $B \ge 4$. Thus

$$\Pr[\wedge_{k \in S_2} \mathcal{S}_{k,t}] \ge 1 - \ell_2 \exp\left(-\frac{1}{2}\frac{1-\alpha}{1+\alpha}(B(1-\alpha)-1)\right) \doteq 1 - \epsilon$$

We solve that it will suffice $B \ge 2 \ln(\frac{\ell_2}{\epsilon}) \left(1 + \frac{3\ell_1 + 2}{\ell_1^2}\right) + 2$. In this case, we get a competitive ratio of $\frac{1}{\ell_1 + 1} \left((1 - \frac{1}{\ell_1 + 1})^{\ell_1} - \epsilon\right)$.

6.2 Extension of ALG₂

П

Suppose we aim for a competitive ratio of $\gamma = \frac{1-\epsilon}{\ell_1+1}$ for ALG₂ where the multiplicative loss ϵ is due to the adding of non-integral resources (we ignore all simulation errors first and handle them later). This implies, for each time *t* and assignment *e*, we try to maintain that *e* is made at *t* with probability equal to $\frac{1-\epsilon}{\ell_1+1}$. From the analysis in Section 5.4, it would suffice to show at each time *t*, *e* is safe with probability $\beta_{e,t} \ge \gamma$. Focus on a given assignment *e* and let $S_{k,t}$ be the event that *e* is safe at *t* with respect to the resource *k*. Let $S_1 = S_e \cap \mathcal{K}_1$ and $S_2 = S_e \cap \mathcal{K}_2$. From the proof of Lemma 5.4, we see that all integral resources are safe at *t* with probability $\Pr[\wedge_{k \in S_1} S_{k,t}] \ge 1 - \frac{(1-\epsilon)\ell_1}{\ell_1+1}$. Thus the remaining issue is to show that $\Pr[\wedge_{k \in S_2} S_{k,t}] \ge 1 - \epsilon$, which by union bound leads to the fact that $\beta_{e,t} = \Pr[\wedge_{k \in S_e} \mathcal{S}_{k,t}] \ge \gamma = \frac{1-\epsilon}{\ell_1+1}$.

Section 6.1 shows that when *B* is large, all non-integral resources are almost safe throughout *T* in ALG₁ by applying Chernoff bound and union bound. As for ALG₂, the same analysis failed due to the following challenges: (1) we cannot upper bound $X_{e,t}$ by some *independent or negatively correlated* $Z_{e,t}$ as before; (2) { $X_{e,t}$ } itself can be positively correlated as shown in the following example.

Example 6.1. Consider an unweighted star graph G = (I, J, E)where $|I| = 1, |J| = 3, E = (e_1, e_2, e_3)$ with T = 2. Suppose at t = 1, j = 1, 2 arrives with equal probability $p_j = 1/2$ and at t = 2, j = 3 will arrive with probability $p_j = 1$. Let e_1, e_2, e_3 denote respectively the assignment we consider when j = 1, 2 comes at t = 1 and j = 3 comes at t = 2. Let K = 2 with $\mathbf{B} = (1, 1)$ and $\mathbf{a}_{e_1} = (1, 0), \mathbf{a}_{e_1} = (0, 1)$ and $\mathbf{a}_{e_3} = (0, 1)$. Suppose LP (4.1) offers us the following optimal solution: $x_{e_1}^* = x_{e_2}^* = 1/2$ and $x_{e_3}^* = 1/2$. In our context, $\ell_1 = 1, \gamma = 1/2$ and ALG₂ goes as follows: at $t = 1, e_1$ and e_2 will be made with probability 1/2 when each comes; at the beginning of $t = 2, e_3$ is safe with probability $\beta = 3/4$ and accordingly, it will be made with probability $\frac{x_3^* \gamma}{p_3 \beta} = \frac{1}{3}$ when it comes.

Recall that $X_{e,t}$ indicates if the assignment e is made at t. We can verify that $\Pr[X_{e_1,t=1} = 1] = x_1^*/2 = 1/4$ and $\Pr[X_{e_3,t=2} = 1] = x_3^*/2 = 1/4$. $\Pr[X_{e_3,t=2} = 1|X_{e_1,t=1} = 1] = 1/3$, that is because e_3 is safe with probability 1 at t = 2 conditioning on $X_{e_3,t=2} = 1$. \Box

We use the technique of *virtual algorithms* to attack the potential positive correlation among $\{X_{e,t}\}$. Suppose we run ALG₂ with some parameter γ up to the time *t* such that for each e' and t' < t, $\Pr[X_{e',t'} = 1] = \gamma x_{e',t'}^*$. Now we try to lower bound the value $\beta_{k,t} \doteq \Pr[S_{k,t}]$ for a given *e* and $k \in S_2$ with $S_2 = S_e \cap \mathcal{K}_2$.

Consider the simple setting where only one non-integral resource k is involved. Suppose we run ALG₁ with parameter $\alpha = \frac{\gamma}{1-\delta}$ as a virtual algorithm up to time t and let $\beta'_{k,t}(\delta) = \Pr[S'_{k,t}]$ be the probability that e is safe at time t with respect to resource k in the virtual algorithm. Here $\delta = o(1)$ when $B \to \infty$.

Lemma 6.1. For any δ with $\beta'_{k,t}(\delta) \ge 1 - \delta$, we have $\beta_{k,t} \ge 1 - \delta$.

Proof. Consider a feasible δ with $\beta'_{k,t}(\delta) \ge 1 - \delta$. For each e' and t' < t, let $X'_{e',t'}$ indicate that e' is made at t' in the virtual algorithm.

We see $\Pr[X'_{e',t'}] = \frac{\gamma x^*_{e',t'}}{1-\delta} \Pr[e' \text{ is safe at } t'] \ge \gamma x^*_{e',t'}$. Notice that in our algorithm ALG₂ with parameter γ , each assignment e' will be made with probability equal to $\gamma x^*_{e',t'}$. Therefore we claim that in ALG₂, $\beta_{k,t} = \Pr[S_{k,t}] \ge \Pr[S'_{k,t}] = \beta'_{k,t} \ge 1 - \delta$.

Now we have all ingredients to prove Theorem 1.4.

Proof. Focus on an assignment *e* and *t*. Ignore the simulation error first and we try to show that when $B \ge 3 \ln(\frac{\ell_2}{\epsilon})(1 + \frac{1}{\ell_1}) + 2$, $\Pr[S_{k,t}] \ge 1 - \frac{\epsilon}{\ell_2}$ for each $k \in S_2$.

Lemma 6.1 tells us that we just need to find a feasible δ such that $\beta'_{k,t} \geq 1 - \delta$. In this case, we have $\Pr[S_{k,t}] \geq 1 - \delta$ and setting $\epsilon = \ell_2 \delta$ will complete the proof. Consider the virtual algorithm ALG₁ with parameter $\alpha = \frac{(1-\epsilon)/(\ell_1+1)}{1-\delta}$ and let $H_{k,t} = \sum_{t' < t} \sum_{e' \in E_t} Z_{e',t'} a_{e',t'}$ where $\Pr[Z_{e',t'} = 1] = \alpha$ for each e' and t' < t. Notice that $\Pr[S'_{k,t}] \geq 1 - \Pr[H_{k,t} \geq B - 1]$ and $\mathbb{E}[H_{k,t}] \leq \alpha B$. WLOG assume $\mathbb{E}[H_{k,t}] = \alpha B$ and $\ell_1 \geq 2$. Let $\Delta = \frac{B-1}{\alpha B} - 1$. We have

$$\begin{split} \Delta &= (1 - \frac{1}{B}) \frac{(1 - \delta)(1 + \ell_1)}{1 - \epsilon} - 1 \\ &= \frac{\ell_1 + \epsilon - \delta - \delta\ell_1}{1 - \epsilon} - \frac{1}{B} \frac{(1 - \delta)(1 + \ell_1)}{1 - \epsilon} \ge \frac{1 - \delta}{1 - \epsilon} \left(\ell_1 - \frac{\ell_1 + 1}{B}\right) \ge 1 \end{split}$$

The last inequality assumes that $B \ge 3 \ge 1 + \frac{2}{\ell_1 - 1}$. Therefore by the Chernoff Bound, we have

$$\begin{aligned} \Pr[H_{k,t} \geq B - 1] &= \Pr\left[H_{k,t} \geq \mathbb{E}[H_{k,t}](1+\Delta)\right] \\ &\leq \exp\left(-\frac{1}{3}\frac{B(1-\epsilon)}{(\ell_1+1)(1-\delta)}\frac{1-\delta}{1-\epsilon}\left(\ell_1 - \frac{\ell_1+1}{B}\right)\right) \\ &= \exp\left(-\frac{1}{3}B\left(1 - \frac{1}{B} - \frac{1}{\ell_1+1}\right)\right) \end{aligned}$$

which implies that

$$\Pr[S_{k,t}] \ge \Pr[S'_{k,t}] \ge 1 - \exp\left(-\frac{1}{3}B\left(1 - \frac{1}{B} - \frac{1}{\ell_1 + 1}\right)\right)$$

When $B \ge 3 \ln(\frac{\ell_2}{\epsilon})(1 + \frac{1}{\ell_1}) + 2$, we can verify that the right-hand side value at least $1 - \delta = 1 - \frac{\epsilon}{\ell_2}$. Thus we prove our claim that for each $k \in S_2$, $\Pr[S_{k,t}] \ge 1 - \frac{\epsilon}{\ell_2}$, which yields that ALG₂ achieves a ratio of $(1 - \epsilon)/(\ell_1 + 1)$. After incorporating the simulation error, we will have an additional multiplicative factor $(1 - \epsilon)$ in the competitive ratio. Thus we prove Theorem 1.4.

7. EXPERIMENTAL EVALUATION

In this section, we propose and evaluate a number of heuristic algorithms for the BOA problem. We start with the case when only integral resources are involved. Section 5 shows that non-adaptive ALG₁ and adaptive ALG₂ can achieve a ratio of at least $\frac{1}{\ell+1}\frac{1}{e}$ and $\frac{1}{\ell+1}$ respectively, where ℓ is the upper bound of integral resources requested by each assignment. In our experiments, we show that the performance is far better than these theoretical worst case bounds (such bounds hold only for some extremely specialized cases such as the one shown in Example 5.2).

Our experimental setup is as follows.

1. For each *j*, recall that N(j) is the set of tasks that interest *j*. We generate N(j) by sampling each $i \in [m]$ independently with some probability, say 0.3. We propose to study the sensitivity to this parameter further in the future.

- 2. Let P_1 be the arrival probability matrix of size $n \times T$ such that $P_1(i, j) = p_{i,j}$. We first generate a random "seed" matrix P_0 of size $n \times T_1$ such that for each $t \in [T_1]$, the values in the t^{th} column of P_0 are uniformly distributed over [0, 1] *conditioning on the column sum is* 1, i.e., $\sum_t P_0(i, t) = 1$. We achieve this by running the file "randfixedsum.m" due to Roger Stafford ³. Once we have a fixed P_0 , we generate P_1 by sampling one column from P_0 uniformly for T times. Notice that if we generate P_1 in the direct way as P_0 , then each j will have almost the same arrivals over T rounds since T assumes to be very large. In our case we set $T_1 = m \ll T$ and we hope we can create some potential bias of the arrivals over all $j \in [n]$ and that can pass to P_1 .
- 3. Let *E* be the set of assignments generated as shown in the first point. For each assignment $e \in E$, we independently choose a uniform value $w_e \in [0, 1]$.
- 4. Recall that \mathcal{K}_1 and \mathcal{K}_2 are the set of integral and non-integral resources respectively. We generate a budget B_k by uniformly sampling an integer from $[UB] = \{1, 2, 3, \dots, UB\}$ for each $k \in \mathcal{K}_1$ and from [LB, 5 * LB] for each $k \in \mathcal{K}_2$ respectively. Here UB and LB are parameters specified in advance.
- 5. Recall S_e is the set of resources requested by e. For each e, we first generate a random permutation π_1 over \mathcal{K}_1 and then set $S_e \cap \mathcal{K}_1$ as the first $\lceil \rho_0 * K_1 \rceil$ elements of π_1 . Set $a_{e,k} = 1$ for each $k \in S_e \cap \mathcal{K}_1$. We then generate another random permutation π_2 over \mathcal{K}_2 and set $S_e \cap \mathcal{K}_2$ as the first $\lceil \rho_0 * K_2 \rceil$ elements of π_2 . Sample a uniform value from [0, 1] for $a_{e,k}$ for each $k \in S_e \cap \mathcal{K}_2$. Here $\rho_0 \in [0, 1]$ is a parameter given in advance.
- 6. For each *e*, let d_e be the deadline of *e*. We sample a random integer from [T/2, T] uniformly as d_e for each $e \in E$. In this experiment we consider a relative more flexible setting: allow assignments with respect to a single task to have potentially distinctive deadlines.

Let $ALG_1(\alpha)$ denote the algorithm shown in Section 5.1 with parameter α . Theorem 1.1 shows that $ALG_1(\frac{1}{\ell+1})$ can achieve a ratio at least $\frac{1}{\ell+1}\frac{1}{e}$. Our experimental results suggest that it will be too conservative for the choice of $\alpha = \frac{1}{\ell+1}$. This inspires us to propose the following four heuristics. All these four algorithms are non-adaptive essentially except the last one. Consider some time *t* when *j* comes and let $E_{j,t} = \{e = (i, j) | i \in N(j), d_e \ge t\}$ be the set of available (not necessarily safe) assignments related to *j*.

- 1. NAdap: sample an assignment $e \in E_{j,t}$ with probability $\frac{x_{e,t}^*}{\sum_{e \in E_{j,t}} x_{e,t}^*}$. Make it iff *e* is safe.
- 2. ALG₁(1): sample an assignment $e \in E_{j,t}$ with probability $\frac{x_{e,t}^*}{P_{i,t}}$. Make it iff *e* is safe.
- USamp: sample an assignment e ∈ E_{j,t} uniformly from E_{j,t}. Make it iff e is safe.
- 4. Greedy: choose the assignment $e \in E_{j,t}$, which has the largest weight w_e among all safe options in $E_{j,t}$.

Remark: (1) the first two are both LP-based non-adaptive algorithms; the third is non-adaptive but blind to the LP solution; the last one is adaptive and blind to the LP solution as well, the strategy

³ https://www.mathworks.com/matlabcentral/fileexchange/9700random-vectors-with-fixed-sum/content/randfixedsum.m

gets updated as the set of safe options shrinks in later rounds; (2) the second can be viewed as the first one plugged with an attenuation factor $\frac{\sum_{e \in E_{j,t}} x_{e,t}^*}{p_{j,t}} \leq 1$. (3) we did not test ALG₂ since the implementation is really time-consuming even on moderate problem size.

For each set of parameters $\mathcal{P} = (m, n, K_1, K_2, T, UB, LB, \rho_0)$, we generate a set $\mathcal{I}(\mathcal{P})$ of 5 random instances as described before. For each instance $I \in \mathcal{I}(\mathcal{P})$, we run the above five algorithms each on I for 100 times and take the mean as the final performance. For each given instance I, let OPT(I) be the LP optimal value and ALG(I) be the final performance on I. We define $\rho(ALG, I) = ALG(I)/OPT(I)$, which is the ratio of performance of ALG to the LP value on I. For each set of parameters $\mathcal{P} = (m, n, K_1, K_2, T, UB, LB, \rho_0)$, we generate 5 random instances as described before and set the mean ratio as $\rho(ALG, \mathcal{P})$ for each ALG. The results can be seen in Figures 1, 2 and 3. The detailed discussion can be found in the full version.



Figure 1: Performance of the four algorithms as *UB* increases where: m = 10, n = 50, $K_1 = 90$, $K_2 = 0$, T = 3000, $\rho_0 = 0.1$. The best LP-based heuristic ALG₁(1) (red-colored) strictly beats the best LP-blind strategy Greedy (blue-colored).

Acknowledgments

Part of this work is done when Pan Xu interned at the IBM T. J. Watson Research Center during the summer of 2016. Aravind Srinivasan's research was supported in part by NSF Awards CNS-1010789 and CCF-1422569, and by a research award from Adobe, Inc. Pan Xu's research was supported in part by NSF Awards CNS-1010789 and CCF-1422569.

REFERENCES

- M. Adamczyk, F. Grandoni, and J. Mukherjee. Improved approximation algorithms for stochastic matching. In *Algorithms-ESA 2015*, pages 1–12. Springer, 2015.
- [2] S. Agrawal, Z. Wang, and Y. Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.
- [3] S. Alaei, M. Hajiaghayi, and V. Liaghat. Online prophet-inequality matching with applications to ad allocation. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 18–35. ACM, 2012.
- [4] S. Alaei, M. Hajiaghayi, and V. Liaghat. The online stochastic generalized assignment problem. In *Approximation*,



Figure 2: Performance of the four algorithms as *UB* increases where m = 10, n = 50, $K_1 = 90$, $K_2 = 0$, T = 3000, $\rho_0 = 0.5$. The best LP-based heuristic ALG₁(1) (red-colored) strictly beats the best LP-blind strategy Greedy (blue-colored).



Figure 3: Performance of the four algorithms as (UB, LB) increases where: $m = 10, n = 50, K_1 = 50, K_2 = 40, T = 2000, \rho_0 = 0.5$. The best LP-based heuristic ALG₁(1) (red-colored) strictly beats the best LP-blind strategy Greedy (blue-colored).

Randomization, and Combinatorial Optimization. Algorithms and Techniques, pages 11–25. Springer, 2013.

- [5] E. M. Arkin and R. Hassin. On local search for weighted k-set packing. *Mathematics of Operations Research*, 23(3):640–648, 1998.
- [6] S. Assadi, J. Hsu, and S. Jabbari. Online assignment of heterogeneous tasks in crowdsourcing markets. In *Third* AAAI Conference on Human Computation and Crowdsourcing, 2015.
- [7] A. Badanidiyuru, R. Kleinberg, and Y. Singer. Learning on a budget: posted price mechanisms for online procurement. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 128–145. ACM, 2012.

- [8] N. Bansal, A. Gupta, J. Li, J. Mestre, V. Nagarajan, and A. Rudra. When LP is the cure for your matching woes: Improved bounds for stochastic matchings. *Algorithmica*, 63(4):733–762, 2012.
- [9] N. Bansal, N. Korula, V. Nagarajan, and A. Srinivasan. Solving packing integer programs via randomized rounding with alterations. *Theory of Computing*, 8(1):533–565, 2012.
- [10] A. Baveja, A. Chavan, A. Nikiforov, A. Srinivasan, and P. Xu. Improved bounds in stochastic matching and optimization. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 40. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.
- [11] P. Berman. A d/2 approximation for maximum weight independent set in d-claw free graphs. In *Scandinavian Workshop on Algorithm Theory*, pages 214–219. Springer, 2000.
- [12] B. Brubach, K. A. Sankararaman, A. Srinivasan, and P. Xu. New algorithms, better bounds, and a novel model for online stochastic matching. *arXiv preprint arXiv:1606.06395*, 2016.
- [13] N. Buchbinder, K. Jain, and J. S. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *European Symposium on Algorithms*, pages 253–264. Springer, 2007.
- [14] N. Chen, N. Immorlica, A. R. Karlin, M. Mahdian, and A. Rudra. Approximating matches made in heaven. *Automata, Languages and Programming*, pages 266–278, 2009.
- [15] N. R. Devanur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM conference on Electronic commerce*, pages 71–78. ACM, 2009.
- [16] N. R. Devanur, K. Jain, B. Sivan, and C. A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 29–38. ACM, 2011.
- [17] N. R. Devanur, B. Sivan, and Y. Azar. Asymptotically optimal algorithm for stochastic adwords. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 388–404. ACM, 2012.
- [18] J. Feldman, N. Korula, V. Mirrokni, S. Muthukrishnan, and M. Pál. Online ad assignment with free disposal. In *International Workshop on Internet and Network Economics*, pages 374–385. Springer, 2009.
- [19] J. Feldman, A. Mehta, V. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *Foundations of Computer Science*, 2009. FOCS'09. 50th Annual IEEE Symposium on, pages 117–126. IEEE, 2009.
- [20] C. M. Fortuin, P. W. Kasteleyn, and J. Ginibre. Correlation inequalities on some partially ordered sets. *Comm. Math. Phys.*, 22(2):89–103, 1971.
- [21] Z. Füredi, J. Kahn, and P. D. Seymour. On the fractional matching polytope of a hypergraph. *Combinatorica*, 13(2):167–180, 1993.
- [22] A. Gupta and V. Nagarajan. A stochastic probing problem with applications. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 205–216. Springer, 2013.
- [23] B. Haeupler, V. S. Mirrokni, and M. Zadimoghaddam. Online stochastic weighted matching: Improved approximation algorithms. In *International Workshop on Internet and Network Economics*, pages 170–181. Springer, 2011.

- [24] E. Hazan, S. Safra, and O. Schwartz. On the complexity of approximating k-set packing. *computational complexity*, 15(1):20–39, 2006.
- [25] C.-J. Ho and J. W. Vaughan. Online task assignment in crowdsourcing markets. In AAAI, volume 12, pages 45–51, 2012.
- [26] C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an sdr, with an application to the worst-case ratio of heuristics for packing problems. *SIAM Journal on Discrete Mathematics*, 2(1):68–72, 1989.
- [27] P. Jaillet and X. Lu. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research*, 39(3):624–646, 2013.
- [28] N. Kell and D. Panigrahi. Online budgeted allocation with general budgets. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pages 419–436. ACM, 2016.
- [29] T. Kesselheim, A. Tönnis, K. Radke, and B. Vöcking. Primal beats dual on online packing lps in the random-order model. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 303–312. ACM, 2014.
- [30] W. Ma. Improvements and generalizations of stochastic knapsack and multi-armed bandit approximation algorithms. In Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1154–1163. Society for Industrial and Applied Mathematics, 2014.
- [31] V. H. Manshadi, S. O. Gharan, and A. Saberi. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research*, 37(4):559–573, 2012.
- [32] A. Mehta. Online matching and ad allocation. *Theoretical Computer Science*, 8(4):265–368, 2012.
- [33] A. Mehta, A. Saberi, U. Vazirani, and V. Vazirani. Adwords and generalized online matching. *Journal of the ACM* (*JACM*), 54(5):22, 2007.
- [34] A. Panconesi and A. Srinivasan. Randomized distributed edge coloring via an extension of the chernoff–hoeffding bounds. *SIAM Journal on Computing*, 26(2):350–368, 1997.
- [35] Y. Singer and M. Mittal. Pricing mechanisms for crowdsourcing markets. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1157–1166. ACM, 2013.
- [36] A. Singla and A. Krause. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In Proceedings of the 22nd international conference on World Wide Web, pages 1167–1178. ACM, 2013.
- [37] A. Slivkins and J. W. Vaughan. Online decision making in crowdsourcing markets: Theoretical challenges. ACM SIGecom Exchanges, 12(2):4–23, 2014.
- [38] A. Subramanian, G. S. Kanth, S. Moharir, and R. Vaze. Online incentive mechanism design for smartphone crowd-sourcing. In *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on*, pages 403–410. IEEE, 2015.

If You Like It, then You Shoulda Put a Sticker on It

A Model for Strategic Timing in Voting

Alan Tsang, Kate Larson Cheriton School of Computer Science University of Waterloo, Canada {akhtsang,klarson}@uwaterloo.ca

ABSTRACT

Sticker Voting is a voting method where ballots are cast by placing stickers on favored candidates. It differs from many other voting methods because the act of voting reveals information to other players, which induces an asymmetry of information available to subsequent voters. Voters may strategize through both the choice of the submitted ballot and the timing of its submission. In this paper, we introduce and analyze a model for strategic voter behavior in Sticker Voting. We find its equilibrium behavior and discuss how it reflects human voting behavior.

1. INTRODUCTION

Because voting is a process that takes place over time, there is an asymmetry of information that is available to earlier and later voters. The ballots cast by earlier voters inform subsequent voters. The latter may use this information to vote strategically, maximizing their chances of casting a pivotal ballot; The former may gain a first-mover advantage, establishing their favorite candidate as a lead runner by shaping what information is available to later voters. Strategic voters must decide not only which ballot to cast, but also *when* to cast their ballot.

The U.S. presidential primaries is an example of such a sequential procedure. The primaries determine each parties' presidential nominee, and are conducted as a series of elections in each state. Each state-level election determines how many delegates are sent in support of each nominee by that state. States schedule their own primary dates. The resulting elections are spread over several months. In 2016, primaries began in February and ended in June, in preparation for the November election [1]. Both parties and individual states recognize the importance of strategic timing. Certain time slots are highly prized by both the Republican and Democratic Parties. Both parties award bonus delegates to states holding their elections later in the primary season [4, 14].

Online polls are another domain which allows for strategic timing. These polls are used as a social choice mechanism for selecting anything from the cutest animal, to artistic direction for crowdfunded projects, to the winner of the Webby People's Voice award. A popular implementation of online polls is the popular group scheduling platform Doodle. Doodle allows participants to approve or decline proposed time slots. Importantly, Doodle supports *open* polls, which allow voters to view the ballots cast by previous voters before committing their own, or waiting and revisiting the poll at a later time.

In this paper, we propose the Sticker Voting framework, where voters are invited to place a sticker (their ballot) on their chosen candidate (which becomes common knowledge). In addition to the potential for casting a strategic ballot, this process also invites voters to be strategic in timing their vote. We propose a model for strategic voter behavior that incorporates strategic timing, and we analyze the strategic equilibrium in a simple Sticker Plurality Voting game. Finally, we discuss how we may use our model to capture voting behavior in the real world.

2. RELATED WORK

In the social choice literature, Sequential Voting describes a voting process (frequently based on plurality) where voters cast their ballots in a particular order, and preceding ballots become common knowledge. This is in contrast with Sticker Voting, where the agent has a choice over the order of votes and strategic timing is possible.

Callander examines "herding" and "bandwagon" effects in Sequential Voting as an information aggregation process [5]. Voters are asked to pick from two candidates, one of which is objectively better than the other. Voters each receive a private noisy signal, and cast their ballots sequentially, based on their signal and preceding ballots. Callander shows that the resulting election begins in an informative phase, where ballots aggregate private signals of voters, but can enter a cascade phase where all subsequent voters agree on the current leading candidate. Alon et al. [2] further extend this model by giving voters intrinsic utility for having voted for the winner (in addition to selecting the better candidate). Battaglini, Morton and Palfrey [3] show that in such a game with costly votes and the possibility for abstention, early voters bear a larger cost when they choose to contribute to the information aggregation process.

Sandholm and Vulkan [15] examine bargaining games in distributed systems where agents have externally imposed deadlines. Prior to their deadline, agents may negotiate by making offers in continuous time. Interestingly, they find that the sequential equilibrium behavior for the agents is to wait until the deadline, at which point they will concede fully. This is due to the informational effect that accompanies making an early offer, which signals a weakness in

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

bargaining position. Moreover, an accepted offer shows that the offerer has already conceded too much, and would have been better off by waiting.

Tal, Meir and Gal [16] study online human voting behavior in response to poll information. They conduct experiments on Amazon Mechanical Turk where participants are given preferences (in the form of small monetary rewards) for playing in a plurality voting game. The game may be one-shot, where poll information is fictitious; or it may be a game of Iterative Voting with other participants. Aside from a small number of erratic voters (who act randomly), most voters exercise either the "default" option (a truthful ballot in the one-shot game, or maintaining the same ballot in an iterated game), or utilized a myopic beset response.

Desmedt and Elkind [7] explore strategic behavior in Sequential Voting where voters may choose to abstain. They show how the subgame perfect Nash equilibrium may be computed, and show that when there are more than 3 candidates, the equilibrium behavior of voters are complex and sometimes counterintuitive. The outcome of the election is sensitive to the risk adversity of the voters, and the voter order.

Gaspers, Naroditskiy, Narodytska, and Walsh [8] examine the possible and necessary winner problem in Sequential Voting (which they term "social polls") when conducted in a social network setting. They find that the possible winner problem is NP-hard to compute, but propose an efficient algorithm for finding necessary winners.

Xia and Conitzer [17] study strategic behavior of agents in Sequential Voting (which they term "Stackelberg Voting"), where voter preference and voting order are public knowledge. The resulting voting game can be solved via backward induction, and may result in highly suboptimal candidates being selected.

Most relevant to our investigation of Sticker Voting is by Dekel and Piccione [6], who examine Sequential Voting where voting occurs in 2 periods, and voters are allowed to choose the period in which they wish to vote. Their model differs from ours in that, this choice must be made *prior* to the election, and prior to the realization of the voters' own preferences. Under their model, Dekel and Piccione find that all voters prefer to vote in the second period, making the sequential outcome equivalent to the simultaneous outcome.

Doodle recently emerged as a popular online poll platform for group scheduling, allowing groups to perform approval voting with open (public) or closed (private) ballots in real time. Zou, Meir and Parkes [18] examine voting behavior in over 340,000 polls. They find marked difference in voting behavior between open and closed polls. Moreover, they find that in open polls, early voters behave differently from later voters, showing evidence of strategic reasoning based on the additional information. Obraztsova, Polukarov, Rabinovich and Elkind [12] propose the Doodle Poll Game capturing this behavior, where users derive additional utility from appearing to be available. Reinecke et al. [13] have also examined how Doodle voting behavior may be affected by national culture and social norms.

3. STICKER VOTING MODEL

We consider a non-sequential voting game \mathcal{G} with n voters and m candidates \mathcal{M} . Let \mathcal{B} be the set of admissible ballots a voter may cast, and \mathcal{B}^n be the set of possible ballots cast by the population of voters. Let \mathcal{F} be a social choice function mapping \mathcal{B}^n to the set of winners, a non-empty subset of \mathcal{M} . Each voter v has a private utility function $u_v : 2^{\mathcal{M}} \to \mathbb{R}$ mapping each outcome to a utility value.

We define a Sticker Voting game based on \mathcal{G} by specifying a number of voting rounds $T \geq 1$. In each round, voters may cast a ballot or choose to "Wait"; this choice is made simultaneously within each round. Once a voter casts a ballot, it is committed and irreversible. Formally, in each round, each voter plays an action from the action set $\mathcal{B} \cup \{\emptyset\}$, where \emptyset corresponds to "Wait" action. Once a voter casts a ballot $b \in \mathcal{B}$, their action space for subsequent rounds is reduced to the set $\{b\}$; we refer to this as moving from the **controlled game** to the **uncontrolled game**. Let $H_t \in \mathcal{B}^n$ denote the set of actions played by agents in round t. The history of play prior to current round t, $\mathcal{H}_t = (H_1, H_2, \dots, H_{t-1})$ is common knowledge. The winner set is $\mathcal{F}(H_T)$, where \emptyset actions are interpreted as "Abstain".

In round t, a voter may act according to a pure strategy function S, which maps \mathcal{H}_t to an action $a^t \in \mathcal{B} \cup \{\emptyset\}$. Smaps to the action b if the agent entered the uncontrolled game by casting ballot $b \in \mathcal{B}$ in a prior round. We also allow voters to play mixed strategies, which map \mathcal{H}_t to a mixed strategy, i.e. a distribution over $\mathcal{B} \cup \{\emptyset\}$.

We focus on *Markovian* strategies, where the voters do not care about the history of ballots prior to the previous round t-1. A Markovian strategy S maps t and H_{t-1} to a mixed strategy.

3.1 Plurality Sticker Voting

In this paper, we focus on the Resolute Plurality Voting Rule. Admissible ballots \mathcal{B} are the candidates \mathcal{M} . For round t, denote the standing \mathbf{s}_t as a vector whose *i*-th element corresponds to the number of ballots supporting candidate i in H_{t-1} , or the zero vector if t = 1. The social choice function \mathcal{F} maps the final votes H_T to the unique candidate i with the highest s_T^i , breaking ties uniformly at random.

We consider Markovian strategies that are also anonymous to other voters. In round t, while in the controlled game, an agent's strategy simply maps t and s_t to a mixed strategy.

3.2 Solution Concept

The Sticker Voting Game uses the solution concept of the Perfect Bayesian Equilibrium (PBE). PBE is a refinement of Subgame Perfect Equilibrium (SPE) for sequential games. In a SPE, players act according to strategies that form a Nash equilibrium in every subgame of the original game. PBE additionally allows players to have incomplete information, where certain nodes of the game tree are indistinguishable from each other to particular players; these are called Information Sets. Players maintain beliefs corresponding to the probability that they are in a particular node in the current Information Set; their strategies are defined according to these beliefs (and may depend on the history of play). In the Sticker Voting Game, Information Sets correspond to voters not knowing the types of the other voters.

In the Plurality Sticker Voting Game, the current round and tally forms a tuple (t, \mathbf{s}_t) that uniquely identifies the information set for the player in the controlled game. Each information set consists of nodes representing the possible types that the remaining uncommitted voters may have. The voter has a belief over the distribution of types of the uncommitted voters.

A second set of nodes capture the uncontrolled games, with a unique node for each round t and uncontrolled tally $\mathbf{s_t}$.

4. COMPLETE INFORMATION GAME

We first consider a simplified scenario with n = 3 voters, {1,2,3}, with complete information, and m = 3 candidates, {A, B, C}, in a T = 2 round game. Player 1 has preference $A \succ B \succ C$; player 2, $B \succ C \succ A$; player 3, $C \succ A \succ$ B, forming a Condorcet cycle. Each player gains utility u_1 if their favorite candidate wins, u_2 utility for their second choice, and 0 for their third choice, with $u_1 > u_2 > 0$. We also require that $2u_2 > u_1$ so that conceding to one's second place alternative is better than a three-way tie. The types of all agents are public knowledge. The following table summarizes the utilities:

Voter	A	B	C
1	u_1	u_2	0
2	0	u_1	u_2
3	u_2	0	u_1

For simplicity of notation, we denote voter v's favorite candidate as $b_{v,1}$, the second choice as $b_{v,2}$, and so on. When the v is clear from context, we omit v from the subscript. We also use $b_{v,i}$ to denote the action where v votes for i. We will actualize the utility values as $u_1 = 3$ and $u_2 = 2$.

Analysis: Final Round

Since the types are common knowledge, we use the more general solution concept of the Subgame Perfect Equilibrium, and use backward induction to solve the game. Without lost of generality, we take the perspective of Agent 1.

We begin with the final round T. If the agent is still in control, she may find the game in a number of different states:

Case 1: 2 ballots for the same candidate.

Agent 1's vote is irrelevant, and that candidate is selected **Case 2**: 2 ballots for different candidates.

Agent 1 breaks ties in favor of the better option.

Case 3: 1 ballot for A

Agent 1 also votes A and gets A as the outcome.

Case 4: 1 ballot for B

Note that this ballot must be cast by Agent 2, since Agent 3 would never vote for B. In this scenario, we can break down the utilities for the remaining players in the following table. Entries indicate the winning candidate, with the payoff for the row and column players in parentheses.

		Agent 3		
		C	A	
Acont 1	A	tie(5/3, 5/3)	A(3,2)	
Agent I	B	B(2,0)	B(2,0)	

It is clear both agents will **coordinate on action** $b_{1,1} = b_{3,2} = A$ as other actions are strictly dominated, and we may iteratively remove dominated strategies.

Case 5: Only Agent 3 has voted, for $C = b_{1,3}$

We also break down utilities here:

		Agent 2			
		B C			
Acout 1	A	tie(5/3, 5/3)	C(0,2)		
Agent I	B	B(2, 3)	C(0,2)		

By the same argument before, the two agents will coordinate on selecting B.

Case 6: Only Agent 2 has voted, for $C = b_{1,3}$

Since Agent 3 has not voted, this is actually **Case 1** from the perspective of Agent 3. That is, since C is Agent 3's top choice, Agent 3 will also vote C, secure it as the outcome. Agent 1's vote is irrelevant.

Case 7: No votes observed

Assuming each agent plays symmetric strategies, each outcome is equally likely, giving an expected utility of 5/3.

Interestingly, **Case 4** and **Case 5** clearly show that there is no straight forward first-mover advantage in this scenario. Any agent that is the sole voter in the initial round, and votes for b_1 , will force the remaining agents to coordinate in the next round, and produce b_3 as the outcome.

Analysis: Initial Round

Using backward induction, we determine the course of play in the initial round. We assume symmetric play; that is, each player v plays action $b_{v,i}$ with probability p_i , $i = \emptyset, 1, 2$, $0 \le p_{\emptyset}, p_1, p_2 \le 1$ and $p_{\emptyset} + p_1 + p_2 = 1$. We analyze the expected utility for Agent 1 for each action.

Case 1: Agent 1 plays A

As we have established, if Agent 1 plays A and the other agents plays \emptyset , the other agents will coordinate to select C, yielding 0 utility for Agent 1. However, Agent 1 may potentially gain an advantage if the other agents choose not to wait. The following table shows the outcomes and their payoffs for Agent 1, based on the actions of Agents 2 and 3.

		Agent 3		
		C	A	Ø
	B	tie(5/3)	A(3)	A(3)
Agent 2	C	C(0)	A(3)	C(0)
	Ø	C(0)	A(3)	C(0)

The expected utility for voting b_1 in the first round is

$$E(u|b_1) = \frac{5}{3}p_1^2 + 3p_2 + 3p_{\emptyset}p_1 \tag{1}$$

Case 2: Agent 1 plays B

		Agent 3		
		C A \emptyset		
	B	B(2)	B(2)	B(2)
Agent 2	C	C(0)	tie(5/3)	C(0)
	Ø	B(2)	B(2)	B(2)

The expected utility for voting b_2 in the first round is

$$E(u|b_2) = 2p_1 + 2p_{\emptyset} + \frac{5}{3}p_2^2 \tag{2}$$

Case 3: Agent 1 plays \emptyset

		Agent 3		
		C	A	Ø
	B	B(2)	A(3)	A(3)
Agent 2	C	C(0)	A(3)	C(0)
	Ø	B(2)	A(3)	*(5/3)

The expected utility for Waiting in the first round is

$$E(u|b_{\emptyset}) = 3p_2 + 2p_1^2 + 5p_{\emptyset}p_1 + \frac{5}{3}p_{\emptyset}^2$$
(3)

Notice immediately that even when factoring in the possibility of multiple agents voting in the initial round, Waiting dominates voting A. So we conclude that $p_1 = 0$.

Suppose we are at a symmetric mixed Nash Equilibrium, then Agent 1 must be ambivalent over the actions in its support (i.e. b_2 and \emptyset). So we may set equations (2) and (3) equal, and solve.

Surprisingly, the symmetric mixed Nash Equilibrium strategy for the initial round is for each agent to play b_2 with probability 0.2, and Wait with probability 0.8.

4.1 Rational Voter Behavior

In this simple, complete information game, rational voters will never vote for their top choice in the first round. Instead, they will vote b_2 with probability 0.2, or otherwise Wait in the first round. In the latter case, Agent 1 will vote for her favorite candidate in the second round, unless both other voters have committed their ballots and she must break a tie in her favor; or Agent 3 casts the only ballot and has voted for C, in which case Agent 1 votes for B.

5. INCOMPLETE INFORMATION GAME

Next, we consider an incomplete information scenario based on the simple game above. As before, we have n = 3 voters $\{1, 2, 3\}$ and m = 3 alternatives $\{A, B, C\}$. Players may be one of three types: Type A players have preference $A \succ B \succ C$; Type $B, B \succ C \succ A$; and Type $C, C \succ A \succ B$. The possible types form a Condorcet cycle, but there is no guarantee that such a cycle will exist in a particular realization of types. Nature assigns a type to each player with equal probability. Players know their own types, but do not know the types of other players. The game will be played over $T \ge 2$ rounds. We impose the same utility structure as before.

Analysis: Final Round T

WLOG, we consider the game from the perspective of Agent 1, who is Type A. If we are in the final round of the controlled game, with tally $\mathbf{s_t}$, let the voters' strategy $\mathcal{S}(t, \mathbf{s_t})$ be a mixed strategy playing b_i with probability $p_i^{t,\mathbf{s_t}}$, where $i \in \{1, 2, 3, \emptyset\}$. We will omit the t and/or $\mathbf{s_t}$ from the superscript where it is clear from context. Additionally, because voter strategies are symmetric with respect to type, we adopt the notational convenience of permuting the voter's *i*-th favorite candidate.

Playing b_3 is strictly dominated, so by the iterated removal of dominated strategies, $p_3 = 0$ in all situations. Moreover, since this is the final round, Waiting is strictly dominated by voting b_1 , so $p_{\emptyset} = 0$. Therefore, for any particular s, $p_1^{\text{s}} + p_2^{\text{s}} = 1$. All probability values are bounded within [0, 1]. Case 1: 2 ballots for the same alternative.

Agent 1's vote is irrelevant, at that alternative is selected. There are three outcomes, with utilities for Agent 1 being 3, 2 or 0.

Case 2: 2 ballots for different alternatives.

Agent 1 breaks ties in favor of the better option. There are 6 outcomes here. Agent 1 may break the tie to gain her top choice in 4 cases, and get her second choice in 2 cases. **Case 3**: Agent 3 (WLOG) casts the only vote, for $A = b_{1,1}$

Agent 1 also votes A and gets A as the outcome.

Case 4: Agent 3 casts the only vote, for $B = b_{1,2}$ Agent 2 may be of one of three types. If Agent 2 is Type

B, then they will also vote for B. Agent 1's vote is irrelevant, and gets a payoff of 2. The following tables break down the utility of Agent 1's actions for the other two cases:

Utility breakdown if Agent 2 is Type A

		Agent 2		
		A	В	
Amont 1	A	A(3,3)	B(2,2)	
Agent 1	B	B(2,2)	B(2,2)	

Utility breakdown if Agent 2 is Type C

		Agent 2	
		C A	
Acont 1	A	tie(5/3, 5/3)	A(3,2)
Agent 1	B	B(2,0)	B(2, 0)

Since Agent 2's type is not known to Agent 1, neither action is dominant. But we can calculate the expected utility for each action.

$$E(u|b_1^{(0,1,0)}) = \frac{1}{3}(3p_1^{(0,1,0)} + 2p_2^{(0,1,0)}) + \frac{1}{3}(\frac{5}{3}p_1^{(0,0,1)} + 3p_2^{(0,0,1)}) + \frac{1}{3}(2)$$
(4)

$$E(u|b_2^{(0,1,0)}) = 2 \tag{5}$$

If there is a mixed equilibrium, then Agent 1 will be ambivalent over the two choices. We set equations (4) = (5), and solve to obtain

$$p_1^{(0,1,0)} = \frac{4}{3}p_1^{(0,0,1)} - 1 \tag{6}$$

We set aside this equation, and carry it forward to Case 5.

Case 5: Agent 3 casts the only vote, for $C = b_{1,3}$

Agent 2 may be of one of three types. If Agent 2 is Type C, then they will vote for C and Agent 1's action is irrelevant, and they get utility 0. The following tables break down the utility of Agent 1's actions for the other two cases:

Utility breakdown if Agent 2 is Type A

		Agent 2		
		A	В	
Agont 1	A	A(3,3)	tie(5/3, 5/3)	
Agent 1	B	tie(5/3, 5/3)	B(2,2)	

Utility breakdown if Agent 2 is Type B

		Agent 2		
		B	C	
Accept 1	A	tie(5/3, 5/3)	C(0,2)	
Agent 1	B	B(2,3)	C(0,2)	

Since Agent 2's type is not known to Agent 1, neither action is dominant. But we can calculate the expected utility for each action. As before, if we are at a mixed equilibrium, we set the two expected utilities and solve to obtain

$$5p_1^{(0,0,1)} - 1 - p_1^{(0,1,0)} = 0 (7)$$

More over, we can substitute equation (6) into (7) to obtain $p_1^{(0,0,1)} = 0$. But substituting this result back into Equation 6, we get $p_1^{(0,1,0)} = -1$. A contradiction. So we are not at a mixed equilibrium.

We then consider the pure strategy outcomes based on the actions in Case 4 and Case 5. An agent who observes (0,1,0) may play $p_1^{(0,1,0)} = 1$ or $p_2^{(0,1,0)} = 1$. In addition, an agent who observes (0,0,1) has options $p_1^{(0,0,1)} = 1$ or $p_2^{(0,0,1)} = 1$. There are four possible pure strategy combinations, and we may calculate the expected payoff for each player, in each scenario. For example, consider $p_1^{(0,1,0)} = 1$ and $p_1^{(0,0,1)} = 1$, where both players will play b_1 regardless of their observation. That means, if Agent 1 observed (0,1,0), we will reach one of three possible outcomes: we elect A, Bor reach a Tie. Thus, the expected utility will be 20/9. We repeat these calculations to formulate the outcomes in the matrix below:¹

		Observes $(0,0,1)$		
		$p_1^{(0,0,1)} = 1$	$p_2^{(0,0,1)} = 1$	
Observes	$p_1^{(0,1,0)} = 1$	$(\frac{20}{9}, \frac{14}{9})$	$(\frac{8}{3}, \frac{4}{3})$	
(0,1,0)	$p_2^{(0,1,0)} = 1$	(2, 1)	$(2, \frac{2}{3})$	

Notice three of the pure strategies are dominated, leaving only the top left cell as the unique symmetric Nash Equilibrium for the final round. This corresponds to the actions of **voting for the top choice regardless of the nature of the single ballot observed**. This nets an expected utility of $\frac{20}{9}$ if Agent 1 observed a ballot for her second choice, and $\frac{14}{9}$, for her third choice.

Čase 6: No agent has cast any ballots, in which case Agent 1's best response is to vote honestly and hope for the best:

 $p_1^{(0,0,0)} = 1$, with probability $\frac{5}{9}$ of electing A, $\frac{2}{9}$ of getting a tie, $\frac{1}{9}$ of getting B, and $\frac{1}{9}$ of getting C. This results in an expected utility of $\frac{61}{27}$.

Analysis: Preceding Round t

Now that we have an equilibrium analysis of the last round, we extend our analysis to preceding rounds via backward induction. Here, each agent has three actions, and Waiting is not a clearly dominated action: $p_1^{(0,0,1)} + p_2^{(0,0,1)} + p_{\emptyset}^{(0,0,1)} = 1$, and $p_1^{(0,1,0)} + p_2^{(0,1,0)} + p_{\emptyset}^{(0,1,0)} = 1$. If Agent 1 takes the Wait action \emptyset , she proceeds into the

If Agent 1 takes the Wait action \emptyset , she proceeds into the information state $(t+1, \mathbf{s}^+)$ of the controlled game, where \mathbf{s}^+ is obtained from \mathbf{s}_t by adding a number of ballots up to an including the number of uncommitted voters, representing new ballots cast this turn by the other voters. If Agent 1 casts a ballot *b*, then she enters into the uncontrolled game $(t+1, \mathbf{s}^+)$ (see Appendix B).

Case 1: 2 ballots for the same alternative.

Agent 1's vote is irrelevant.

Case 2: 2 ballots for different alternatives.

Agent 1 breaks ties in favor of the better option. **Case 3**: Agent 3 (WLOG) casts the only vote, for $A = b_{1,1}$

Agent 1 also votes A and gets A as the outcome. Case 4: Agent 3 casts the only vote, for $B = b_{1,2}$

As before, we may lay out the possible actions of each agent, based on the possible types of Agent 2 (recall if Agent 2 is type B, the outcome is decided regardless of the actions Agent 1):

Utility breakdown if Agent 2 is Type A

		Agent 2		
		A	B	Ø
	A	A(3,3)	B(2,2)	A(3,3)
Agent 1	B	B(2, 2)	B(2, 2)	B(2,2)
	Ø	A(3,3)	B(2,2)	*(H,H)

Utility breakdown if Agent 2 is Type C

		Agent 2		
		C	A	Ø
	A	tie(5/3, 5/3)	A(3, 2)	A(3,2)
Agent 1	B	B(2, 0)	B(2,0)	B(2,0)
	Ø	B(2, 0)	A(3, 2)	*(H,L)

Importantly, the outcome designated as * represents the outcome computed in the inductive step for the next round, where the expected utility for a player who observes a ballot for her second choice is H, or is L if a ballot for her last choice is observed (H > L, and H > 2). If the current round is t = T - 1, then $H = \frac{20}{9}$ and $L = \frac{14}{9}$.

round is t = T - 1, then $H = \frac{20}{9}$ and $L = \frac{14}{9}$. As before, we can write equations for expected utilities and solve to show that $b_2^{(0,1,0)}$ is dominated by $b_{\emptyset}^{(0,1,0)}$, if $H \ge 2$. We solve the remaining equalities in conjunction with Case 5 below.

Case 5: Agent 3 casts the only vote, for $C = b_{1,3}$

Agent 2 may be of one of three types. If Agent 2 is Type C, then they will vote for C and Agent 1's action is irrelevant, and they get utility 0. The following tables break down the utility of Agent 1's actions for the other two cases:

¹While this matrix resembles a normal form game, it is only analogous to one. The rows and columns represent information states that the players find themselves in, and the actions they may take. The cell represents the payoff to the player for a particular pure strategy the agents symmetrically pursue.

Utility breakdown if Agent 2 is Type A

		Agent 2		
		A	В	Ø
	A	A(3,3)	tie(5/3, 5/3)	A(3,3)
Agent 1	B	tie(5/3, 5/3)	B(2, 2)	B(2,2)
	Ø	A(3,3)	B(2,2)	*(L,L)

		Agent 2		
		B	C	Ø
	A	tie(5/3, 5/3)	C(0,2)	C(0,2)
Agent 1	B	B(2, 3)	C(0, 2)	B(2,3)
	Ø	B(2,3)	C(0,2)	*(L,H)

We formulate expected utilities as before. We utilize Gambit [9] to solve this subgame for the t = T - 1 case, and find that $p_2^{(0,0,1)} = 0$. Using this information (see Appendix A for details), we may solve the system of equations exactly to obtain

$$p_{\emptyset}^{(0,1,0)} = -\frac{3H - 3L - 1}{24H - 3L - 71} \tag{8}$$

$$p_{\emptyset}^{(0,0,1)} = \frac{3H - 3L - 8}{24H - 3L - 71} \tag{9}$$

and expected utilities

$$E(u|b_{\emptyset}^{(0,1,0)}) = \frac{4(41H - 6L - 121)}{3(24H - 3L - 71)}$$
(10)

$$E(u|b_{\emptyset}^{(0,0,1)}) = \frac{117H - 19L - 333}{3(24H - 3L - 71)}$$
(11)

In particular, for t = T - 1 of the controlled game, when observing (0, 1, 0), Agent 1 should vote b_1 with probability $p_1^{(0,1,0)} = 64/67$ (and Wait otherwise) for an expected utility of 2.34. When observing (0, 0, 1), she should vote b_1 with probability $p_1^{(0,0,1)} = 49/67$ for an expected utility of 1.53.

Case 6: No ballots observed.

If no ballots are observed, all agents are in the same information set, and we may assume they act symmetrically. We denote the probability that they play their top choice, second choice and Wait as p_1 , p_2 , and p_{\emptyset} , respectively.

If Agent 1 Waits, then with probability p_{\emptyset}^2 , we enter the next round with the tally (0, 0, 0), which gives an expected utility of N $(N = \frac{61}{27}$ in round T - 1). With probability $2p_{\emptyset}(1 - p_{\emptyset})$, we enter the next round with one other ballot cast (uniformly randomly selected between the candidates); each of these outcomes gives an expected utility of 3, H, and L. Finally, with probability $(1 - p_{\emptyset})^2$, both other agents cast their ballots. There are 9 possible outcomes (all equally likely); Agent 1 gains her top choice in 5 cases, her second choice in 3 cases, and her last choice in 1 case. This gives an expected utility of $\frac{7}{3}$. Therefore, the expected utility of waiting is

$$E(u|b_{\emptyset}^{(0,0,0)}) = Np_{\emptyset}^{2} + 2p_{\emptyset}(1-p_{\emptyset})\frac{3+H+L}{3} + (1-p_{\emptyset})^{2}\frac{7}{3}$$
(12)

If Agent 1 votes for b_1 , then with probability p_{\emptyset}^2 , we enter the uncontrolled game (t + 1, (1, 0, 0)), with expected utility U_1 (see Appendix B). With probability $2p_{\emptyset}(1 - p_{\emptyset})$, one other agent has blindly voted, resulting in the vote vector (2, 0, 0) (utility = 3), (1, 1, 0) (utility = $\frac{8}{3})^2$, or (1, 0, 1)(utility = 1). Finally, with probability $(1 - p_{\emptyset})^2$, both other agents have blindly voted, giving a utility of $\frac{61}{27}$.

Thus, the expected utility for this action is

$$E(u|b_1^{(0,0,0)}) = U_1 p_{\emptyset}^2 + \frac{40}{9} p_{\emptyset}(1-p_{\emptyset}) + (1-p_{\emptyset})^2 \frac{61}{27} \quad (13)$$

By a similar set of calculations, we get the expected utility for casting a b_2 ballot is

$$E(u|b_2^{(0,0,0)}) = U_2 p_{\emptyset}^2 + 4p_{\emptyset}(1-p_{\emptyset}) + (1-p_{\emptyset})^2 \frac{49}{27}$$
(14)

where U_2 is the expected utility from the uncontrolled game (t + 1, (0, 1, 0)), and $U_2 < U_1$. Notice $E(u|b_2^{(0,0,0)})$ is smaller than $E(u|b_1^{(0,0,0)})$ for all values of p_0 . Therefore, we may assume $p_2^{(0,0,0)} = 0$, and $p_1^{(0,0,0)} + p_{\emptyset}^{(0,0,0)} = 1$.

Let us consider the difference of expected utility from the remaining two options:

$$E(u|b_1^{(0,0,0)}) - E(u|b_{\emptyset}^{(0,0,0)})$$

= $(U_1 - N + \frac{2}{3}(H + L) - \frac{68}{27})p_{\emptyset}^2 + (\frac{70}{27} - \frac{2}{3}(H + L))p_{\emptyset} - \frac{2}{27}$ (15)

Clearly, if $p_{\emptyset} = 0$, this would result in a negative value and $E(u|b_1^{(0,0,0)}) < E(u|b_{\emptyset}^{(0,0,0)})$, which is a contradiction. So we know that regardless of the values of H and L, there is a non-zero probability that an agent Waits.

If t = T - 1, then $N = U_1 = \frac{61}{27}$ and $H + L = \frac{34}{9}$, which zeroes out the p_0^2 term, and (15) becomes $\frac{2}{27}(p_{\emptyset} - 1)$. Therefore, $p_{\emptyset} = 1$ and Agent 1 waits.

We carry forward the induction to t = T - 2. N = 61/27 $U_1 = 2.1739$ and H + L = 3.8723. Equation 15 becomes $1/27(-2+0.2986p_{\emptyset} - 0.6033p_{\emptyset}^2)$, which is negative for all values of p_{\emptyset} . Thus, $E(u|b_1^{(0,0,0)}) < E(u|b_{\emptyset}^{(0,0,0)})$, and so Agent 1 waits as well. Trend continues in further rounds of induction.

Therefore, regardless of the number of rounds in the election, the rational voter always **Waits until the last round in the process before casting a sincere ballot for their top choice**. For this arrangement of candidates and voter preferences, Sticker Voting is equivalent to a simultaneous vote.

6. DISCUSSION & CONCLUSION

In our two simple instances of Sticker Voting, we observe that rational voter behavior differs dramatically. In the complete information game, voters will play a mixed strategy in

²Note that the Condorcet cycle is important here: if the remaining voter is Type C, she would strategically vote for A.

the first round, playing either their *second* choice or Waiting; if they chose to Wait, they will break any ties in their favor in the final round, or otherwise vote sincerely. In the incomplete information game, voters will always exercise the Wait option until they reach the final round, during which they vote sincerely.

It is interesting to contrast the two behaviors. The voters in the complete information game know that the other players are rivals, and therefore understand that there is a first-mover disadvantage if they are greedy. Yet there is also an incentive to concede early to secure acceptable compromise. In the incomplete information game, the voter is unsure as to the nature of the other players. However, more likely than not, one of the other players has the same type as her, so there is an opportunity to signal cooperation. But any incentive to do this is outweighed by the shrewdness of Waiting until the final round, where any other players with the same type as her will naturally coordinate their votes out of self interest. Additionally, in sharp contrast with the complete information game, voting second choice is never exercised as an option.

The result of our incomplete information game is in line with the results of Dekel and Piccione [6]. In their model, voters must commit to voting in one of two rounds. This decision is made prior to the election, and prior to realizing their own preferences. They find that rational voters will always vote in the second and final round. Battaglini, Morton and Palfrey [3] also remark in their work that latter voters benefit from informational effects revealed by earlier voters; while their model is fundamentally different from ours, a similar observation can be made. Finally, in Sandholm and Vulkan's bargaining game with deadlines [15], rational agents will wait until the final moment before their deadline before acting. Yet, these results appear to be at odds with the incentives offered by the Republican and Democratic Parties in the U.S., who award bonus delegates to states voting later in the primary season.

Moreover, our solution for the rational voter seem unintuitive when applied to human voters. In real world Sticker Voting venues and in online polls, we do not expect to see all (or even, a majority of) voters deliberating until the last minute to cast their ballots. We know that humans are impatient and place diminishing value on future payoffs; Are these important qualities to model in Sticker Voting? Human voters also place importance on the expressiveness of voting – they gain satisfaction from having expressed their opinion through voting sincerely. It would be interesting to conduct experiments similar to Battaglini, Morton and Palfrey [3] to elicit data on human voting behavior when using the Sticker Voting mechanism.

Additionally, we have made several assumptions about the preference structure and voter behavior for tractability of analysis. What happens when we relax these assumptions? The Condorcet cycle in the preference structure is an important element in at least one of the calculations in the model (see Footnote 2). Do the results hold if such cycle are rare in practice?

One possible model of bounded rationality that may applied to Sticker Voting is the Quantal Response Equilibrium (QRE) model [10], where players have a nonzero probability of playing each action, defined as a function of the expected payoff of that action. For instance, in the logit equilibrium (LQRE), the probability of playing an action a with ex-

pected utility $u(a|\mathbf{a_{-i}})$ where other players are using strategies $\mathbf{a_{-i}}$ is defined as

$$Pr(a|\mathbf{a}_{-\mathbf{i}}) = \frac{e^{\lambda u(a|\mathbf{a}_{-\mathbf{i}})}}{Z}$$

with sharpness parameter λ and normalization constant Z. QRE has also been extended to extensive form games, where the agents' future actions are treated as mixed strategies defined inductively [11].

Alternatively, it may be interesting to consider a setting where some proportion of voters are impulsive, and will commit to a ballot early in the voting process. How will the presence of such voters affect the behavior of the strategic voters? Will their actions cause a collapse in the "Waiting" equilibrium?

Finally, it would be interesting future work to investigate other models of deliberative agents in Sticker Voting setting. For instance, agents may also make use of history to infer the types of other agents, allowing them to update their beliefs of the distribution of types in population of uncommitted voters, and therefore strategize accordingly.

APPENDIX

A. UTILITIES FOR ROUND T

The expected utilities for playing b_1 , b_2 or b_{\emptyset} in round t, upon observing a single ballot for C can be calculated as follows:

$$E(u|b_1^{(0,0,1)*}) = \frac{1}{3} \left(\frac{5}{3} p_1^{(0,1,0)*} + 0 p_2^{(0,1,0)*} + 0 p_{\emptyset}^{(0,1,0)*}\right) + \frac{1}{3} \left(3 p_1^{(0,0,1)*} + \frac{5}{3} p_2^{(0,0,1)*} + 3 p_{\emptyset}^{(0,0,1)*}\right)$$
(16)

$$E(u|b_2^{(0,0,1)*}) = \frac{1}{3}(2p_1^{(0,1,0)*} + 0p_2^{(0,1,0)*} + 2p_{\emptyset}^{(0,1,0)*}) + \frac{1}{2}(\frac{5}{2}p_1^{(0,0,1)*} + 2p_2^{(0,0,1)*} + 2p_{\emptyset}^{(0,0,1)*})$$
(17)

$$E(u|b_{\emptyset}^{(0,0,1)*}) = \frac{1}{3}(2p_{1}^{(0,1,0)*} + 0p_{2}^{(0,1,0)*} + \frac{14}{9}p_{\emptyset}^{(0,1,0)*}) + \frac{1}{3}(3p_{1}^{(0,0,1)*} + 2p_{2}^{(0,0,1)*} + \frac{14}{9}p_{\emptyset}^{(0,0,1)*})$$
(18)

At this point, we may use Gambit to solve the game for the T-1 round numerically. We get the following mixed Nash equilibrium: $p_1^{(0,1,0)} = 0.96$, $p_2^{(0,1,0)} = 0$, $p_{\emptyset}^{(0,1,0)} = 0.045$, and $p_1^{(0,0,1)} = 0.73$, $p_2^{(0,0,1)} = 0$, $p_{\emptyset}^{(0,0,1)} = 0.27$. This leads to an expected utility of 2.31 for a player who observes a ballot for her second choice, or of 1.53 for a player who observes a ballot for her last choice.

In other words, in the second-to-last round, an agent plays a mixed strategy between playing her top choice and waiting. The probability of waiting is higher if she observes a ballot supporting her last choice.

More importantly, this informs us that playing b_2 is always dominated by another strategy, when observing both (0, 1, 0)and (0, 0, 1). This allows us to calculate the exact solution. If we assume that $p_2^{(0,0,1)*} = 0$, we may substitute

$$p_1^{(0,0,1)} + p_{\emptyset}^{(0,0,1)} = 1p_1^{(0,1,0)} + p_{\emptyset}^{(0,1,0)} = 1$$
(19)

into the previous expected utilities:

$$\begin{split} E(u|b_1^{(0,1,0)*}) &= \frac{1}{3}(3) + \frac{1}{3}(\frac{5}{3}p_1^{(0,0,1)*} + 3p_{\emptyset}^{(0,0,1)*}) + \frac{1}{3}(2) \\ &= \frac{4}{9}p_{\emptyset}^{(0,0,1)*} + \frac{20}{9} \\ E(u|b_{\emptyset}^{(0,1,0)*}) &= \frac{1}{3}(3p_1^{(0,1,0)*} + \frac{20}{9}p_{\emptyset}^{(0,1,0)*}) \\ &+ \frac{1}{3}(2p_1^{(0,0,1)*} + \frac{20}{9}p_{\emptyset}^{(0,0,1)*}) + \frac{1}{3}(2) \\ &= -\frac{7}{27}p_{\emptyset}^{(0,1,0)*}\frac{2}{27}p_{\emptyset}^{(0,0,1)*}) + \frac{7}{3} \\ E(u|b_1^{(0,0,1)*}) &= \frac{1}{3}(\frac{5}{3}p_1^{(0,1,0)*} + 0p_{\emptyset}^{(0,1,0)*}) + \frac{1}{3}(3p_1^{(0,0,1)*} + 3p_{\emptyset}^{(0,0,1)*}) \\ &= \frac{5}{9}p_1^{(0,1,0)*} + 1 \\ E(u|b_{\emptyset}^{(0,0,1)*}) &= \frac{1}{3}(2p_1^{(0,1,0)*} + \frac{14}{9}p_{\emptyset}^{(0,1,0)*}) \\ &+ \frac{1}{3}(3p_1^{(0,0,1)*} + \frac{14}{9}p_{\emptyset}^{(0,0,1)*}) \\ &= -\frac{4}{27}p_{\emptyset}^{(0,1,0)*} - \frac{13}{27}p_{\emptyset}^{(0,0,1)*} + \frac{5}{3} \end{split}$$

If we assume the equilibrium strategy is a mixed strategy comprised of the remaining actions, then we may also set $E(u|b_1^{(0,1,0)*}) = E(u|b_{\emptyset}^{(0,1,0)*})$, and $E(u|b_1^{(0,0,1)*}) = E(u|b_{\emptyset}^{(0,0,1)*})$, and solving gives us the system of equations:

$$7p_{\emptyset}^{(0,1,0)*} + 10p_{\emptyset}^{(0,0,1)*} = 3$$
$$-11p_{\emptyset}^{(0,1,0)*} + 13p_{\emptyset}^{(0,0,1)*} = 3$$

This solves to give us the exact solution that verifies with the empirical solution provided by Gambit, $p_{\emptyset}^{(0,1,0)*} = 3/67$ and $p_{\emptyset}^{(0,0,1)*} = 18/67$.

Using this same method allows us to compute the exact solution for any values for expected utility obtained for taking the Wait action for any given round. Let H(L) be the expected utility gained by waiting when observing (0, 1, 0) ((0, 0, 1)), respectively. The only changes are to the utility calculations for $E(u|b_{\emptyset}^{(0,1,0)*})$ and $E(u|b_{\emptyset}^{(0,0,1)*})$ (Equation (18)), as follows:

$$\begin{split} E(u|b_{\emptyset}^{(0,1,0)*}) &= \frac{1}{3} (3p_{1}^{(0,1,0)*} + 2p_{2}^{(0,1,0)*} + Hp_{\emptyset}^{(0,1,0)*}) \\ &+ \frac{1}{3} (2p_{1}^{(0,0,1)*} + 3p_{2}^{(0,0,1)*} + Hp_{\emptyset}^{(0,0,1)*}) + \frac{1}{3} (2) \\ &= \frac{1}{3} (3p_{1}^{(0,1,0)*} + Hp_{\emptyset}^{(0,1,0)*}) \\ &+ \frac{1}{3} (2p_{1}^{(0,0,1)*} + Hp_{\emptyset}^{(0,0,1)*}) + \frac{1}{3} (2) \\ &= \frac{1}{3} (3 + (H - 3)p_{\emptyset}^{(0,1,0)*}) \\ &+ \frac{1}{3} (2 + (H - 2)p_{\emptyset}^{(0,0,1)*}) + \frac{1}{3} (2) \\ &= \frac{(H - 3)}{3} p_{\emptyset}^{(0,1,0)*} \\ &+ \frac{(H - 2)}{3} p_{\emptyset}^{(0,0,1)*} + \frac{7}{3} \end{split}$$

$$E(u|b_{\emptyset}^{(0,0,1)*}) = \frac{1}{3}(2p_{1}^{(0,1,0)*} + 0p_{2}^{(0,1,0)*} + Lp_{\emptyset}^{(0,1,0)*})$$

$$+ \frac{1}{3}(3p_1^{(0,0,1)*} + 2p_2^{(0,0,1)*} + Lp_{\emptyset}^{(0,0,1)*})$$

$$= \frac{1}{3}(2p_1^{(0,1,0)*} + Lp_{\emptyset}^{(0,1,0)*})$$

$$+ \frac{1}{3}(3p_1^{(0,0,1)*} + Lp_{\emptyset}^{(0,0,1)*})$$

$$= \frac{1}{3}(2 + (L - 2)p_{\emptyset}^{(0,1,0)*})$$

$$+ \frac{1}{3}(3 + (L - 3)p_{\emptyset}^{(0,0,1)*})$$

$$= \frac{(L - 2)}{3}p_{\emptyset}^{(0,1,0)*} + \frac{(L - 3)}{3}p_{\emptyset}^{(0,0,1)*} + \frac{5}{3}$$

$$(0,1,0)$$

) We set $E(u|b_1^{(0,1,0)}) = E(u|b_{\emptyset}^{(0,1,0)*})$, and $E(u|b_1^{(0,0,1)*}) = E(u|b_{\emptyset}^{(0,0,1)*})$, and solve:

$$(H-3)p_{\emptyset}^{(0,1,0)*} + (H-\frac{10}{3})p_{\emptyset}^{(0,0,1)*} = -\frac{1}{3}$$
$$(L-\frac{1}{3})p_{\emptyset}^{(0,1,0)*} + (L-3)p_{\emptyset}^{(0,0,1)*} = -\frac{1}{3}$$

which gives the solution

$$p_{\emptyset}^{(0,1,0)} = -\frac{3H - 3L - 1}{24H - 3L - 71}$$
$$p_{\emptyset}^{(0,0,1)} = \frac{3H - 3L - 8}{24H - 3L - 71}$$

B. THE UNCONTROLLED GAME

We say Agent 1 enters the uncontrolled game node $(t+1, \mathbf{s})$ when she has chosen to cast a ballot in round t, resulting in the tally \mathbf{s} (which includes her ballot and other ballots submitted simultaneously in round t).

In particular, we are interested in the uncontrolled game (t+1, (1, 0, 0)). If t+1 = T, then we know (due to symmetry) both remaining agents will vote for their top preferences. This gives an expected utility $\frac{61}{27}$ as may be expected. However, in prior rounds t+1 < T, the remaining agents

However, in prior rounds t + 1 < T, the remaining agents may be able to coordinate if they happen to vote sequentially. This only matters if the remaining agents have types B and C (a 2 in 9 chance), and depends on the probability of them waiting upon observing the controlled information state t + 1, **s**. As a result, the expected utility of entering this uncontrolled game is

$$E(u|(t+1,(1,0,0)))$$
(20)
= $\frac{1}{27}(2p_{\emptyset}^{t+1,(0,1,0)}p_{\emptyset}^{t+1,(0,0,1)} + 8p_{\emptyset}^{t+1,(0,1,0)} - 10p_{\emptyset}^{t+1,(0,0,1)} + 61)$ (21)

where $p_{\emptyset}^{t+1,(0,1,0)}$ and $p_{\emptyset}^{t+1,(0,0,1)}$ are inductively calculated for round t+1 by Equations (8) and (9). The following table shows the expected utility of entering the uncontrolled game (t+1,(1,0,0)), i.e. by casting a sincere ballot in round tafter observing no ballots. Notice all are strictly less than $\frac{61}{27}$.

Round $t+1$	Т	T-1	T-2
Utility	2.26	2.17	2.18

REFERENCES

- 2016 presidential primary election schedule. https://www.washingtonpost.com/graphics/politics/2016election/primaries/schedule/. Accessed: 2016-02-01.
- [2] N. Alon, M. Babaioff, R. Karidi, R. Lavi, and M. Tennenholtz. Sequential voting with externalities: herding in social networks. In ACM Conference on Electronic Commerce (EC'12), page 36, 2012.
- [3] M. Battaglini, R. Morton, and T. Palfrey. Efficiency, equity, and timing of voting mechanisms. *American* political science Review, 101(03):409–424, 2007.
- [4] R. E. Berg-Andersson. The green papers: Democratic detailed delegate allocation - 2012. http://www.thegreenpapers.com/P12/D-Alloc.phtml. Accessed: 2016-02-05.
- [5] S. Callander. Bandwagons and momentum in sequential voting. *The Review of Economic Studies*, 74(3):653–684, 2007.
- [6] E. Dekel and M. Piccione. The strategic dis/advantage of voting early. American Economic Journal: Microeconomics, 6(4):162–179, 2014.
- [7] Y. Desmedt and E. Elkind. Equilibria of plurality voting with abstentions. In *Proceedings of the 11th* ACM conference on Electronic commerce, pages 347–356. ACM, 2010.
- [8] S. Gaspers, V. Naroditskiy, N. Narodytska, and T. Walsh. Possible and necessary winner problem in social polls. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multiagent Systems*, pages 613–620. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [9] R. McKelvey, A. McLennan, and T. Turocy. Gambit: Software tools for game theory. 2002.
- [10] R. D. McKelvey and T. R. Palfrey. Quantal response equilibria for normal form games. 1993.
- [11] R. D. McKelvey and T. R. Palfrey. Quantal response equilibria for extensive form games. *Experimental* economics, 1(1):9–41, 1998.
- [12] S. Obraztsova, E. Elkind, M. Polukarov, and Z. Rabinovich. Doodle poll games. AGT@ IJCAI, 2015.
- [13] K. Reinecke, M. K. Nguyen, A. Bernstein, M. Näf, and K. Z. Gajos. Doodle around the world: online scheduling behavior reflects cultural differences in time perception and group decision-making. In *Proceedings* of the 2013 Conference on Computer Supported Cooperative Work, pages 45–54. ACM, 2013.
- [14] Republican National Committee. 2/27/96 Nicholson to Chair National GOP Task Force on Presidential Primary Schedule. Press release, 1996.
- [15] T. Sandholm, N. Vulkan, et al. Bargaining with deadlines. In AAAI/IAAI, pages 44–51, 1999.
- [16] M. Tal, R. Meir, and Y. K. Gal. A study of human behavior in online voting. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 665–673. International Foundation for Autonomous Agents and Multiagent Systems, 2015.
- [17] L. Xia and V. Conitzer. Stackelberg voting games: Computational aspects and paradoxes. In AAAI, 2010.

[18] J. Zou, R. Meir, and D. Parkes. Strategic voting behavior in doodle polls. In Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, pages 464–472. ACM, 2015.

Approximation and Parameterized Complexity of Minimax Approval Voting

Marek Cygan Department of Mathematics, Informatics and Mechanics University of Warsaw, Poland cygan@mimuw.edu.pl Łukasz Kowalik Department of Mathematics, Informatics and Mechanics University of Warsaw, Poland kowalik@mimuw.edu.pl

Krzysztof Sornat Department of Mathematics and Computer Science University of Wrocław, Poland krzysztof.sornat@cs.uni.wroc.pl

Arkadiusz Socała Department of Mathematics, Informatics and Mechanics University of Warsaw, Poland arkadiusz.socala@mimuw.edu.pl

ABSTRACT

We present three results on the complexity of MINIMAX APPROVAL VOTING. First, we study MINIMAX APPROVAL VOTING parameterized by the Hamming distance d from the solution to the votes. We show MINIMAX APPROVAL VOTING admits no algorithm running in time $\mathcal{O}^{\star}(2^{o(d \log d)})$, unless the Exponential Time Hypothesis (ETH) fails. This means that the $\mathcal{O}^{\star}(d^{2d})$ algorithm of Misra et al. [AAMAS 2015] is essentially optimal. Motivated by this, we then show a parameterized approximation scheme, running in time $\mathcal{O}^{\star}((3/\epsilon)^{2d})$, which is essentially tight assuming ETH. Finally, we get a new polynomial-time randomized approximation scheme for MINIMAX APPROVAL VOTING, which runs in time $n^{\mathcal{O}(1/\epsilon^2 \cdot \log(1/\epsilon))} \cdot \operatorname{poly}(m)$, where n is a number of voters and m is a number of alternatives. It almost matches the running time of the fastest known PTAS for CLOSEST STRING due to Ma and Sun [SIAM J. Comp. 2009].

CCS Concepts

•Theory of computation \rightarrow Rounding techniques; Fixed parameter tractability; *Problems, reductions and completeness*; Linear programming; •Computing methodologies \rightarrow Multi-agent systems;

Keywords

minimax approval voting, computational social choice, lower bound, parameterized complexity, ptas

1. INTRODUCTION

One of the central problems in artificial intelligence and computational social choice is aggregating preferences of individual agents (see the overview of Conitzer [8]). Here we focus on *multi-winner choice*, where the goal is to select a k-element subset of a set of candidates. Given preferences of

Will be published in: Proceedings of The Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17). February 4th-9th, 2017. San Francisco, California, USA.

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil. the agents over the candidates, a multi-winner voting rule can be used to select a subset of candidates that in some sense are preferred by the agents. This scenario covers a variety of settings: nations elect members of parliament or societies elect committees [7], web search engines choose pages to display in response to a query [11], airlines select movies available on board [31, 12], companies select a group of products to promote [25], etc.

In this work we restrict our attention to approval-based multi-winner rules, i.e., rules where each voter expresses his or her preferences by providing a subset of the candidates which he or she approves. Various voting rules are studied in the literature. In the simplest one, Approval Voting (AV), occurrences of each candidate are counted and k most often approved candidates are selected. While this rule has many desirable properties in the single winner case [13], in the multi-winner scenario its merits are often considered less clear [18], e.g., because it fails to reflect the diversity of interests in the electorate [17]. Therefore, numerous alternative rules have been proposed, including Satisfaction Approval Voting, Proportional Approval Voting, and Reweighted Approval Voting (see [17] for details). In this paper we study a rule called Minimax Approval Voting (MAV), introduced by Brams et al. [3]. Here, we see the votes and the choice as 0-1 strings of length m (characteristic vectors of the subsets, i.e., the candidate i is approved if the string contains 1 at position i). For two strings x and y of the same length the Hamming distance $\mathcal{H}(x, y)$ is the number of positions where x and y differ, e.g., $\mathcal{H}(011, 101) = 2$. In MAV, we look for a 0-1 string with k ones that minimizes the maximum Hamming distance to a vote. In other words, MAV minimizes the disagreement with the least satisfied voter and thus it is highly egalitarian: no voter is ignored and a majority of voters cannot guarantee a specific outcome [3, 19].

Our focus is on the computational complexity of computing the choice based on the MAV rule. In the MINIMAX APPROVAL VOTING decision problem, we are given a multiset $S = \{s_1, \ldots, s_n\}$ of 0-1 strings of length m (also called votes), and two integers k and d. The question is whether there exists a string $s \in \{0, 1\}^m$ with exactly k ones such that for every $i = 1, \ldots, n$ we have $\mathcal{H}(s, s_i) \leq d$. In the optimization version of MINIMAX APPROVAL VOTING we minimize d, i.e., given a multiset S and an integer k as before, the goal is to
find a string $s \in \{0, 1\}^m$ with exactly k ones which minimizes $\max_{i=1,...,n} \mathcal{H}(s, s_i)$.

A reader familiar with string problems might recognize that MINIMAX APPROVAL VOTING is tightly connected with the classical NP-complete problem called CLOSEST STRING, where we are given n strings over an alphabet Σ and the goal is to find a string that minimizes the maximum Hamming distance to the given strings. Indeed, LeGrand et al. [20] showed that MINIMAX APPROVAL VOTING is NP-complete as well by reduction from CLOSEST STRING with binary alphabet. First proof of NP-completeness of MINIMAX APPROVAL VOTING was shown using reduction from VERTEX COVER [19]. This motivated the study on MINIMAX APPROVAL VOTING in terms of approximability and fixed-parameter tractability.

Previous results on MINIMAX APPROVAL VOTING.

First approximation result was a simple 3-approximation algorithm due to LeGrand et al. [20], obtained by choosing an arbitrary vote and taking any k approved candidates from the vote (extending it arbitrarily to k candidates if needed). Next, a 2-approximation was shown by Caragiannis et al. [6] using an LP-rounding procedure. Finally, Byrka et al. [5] presented a polynomial time approximation scheme (PTAS), i.e., an algorithm that for any fixed $\epsilon > 0$ gives a $(1 + \epsilon)$ -approximate solution in polynomial time. More precisely, their algorithm runs in time $m^{\mathcal{O}(1/\epsilon^4)} + n^{\mathcal{O}(1/\epsilon^3)}$ which is polynomial in the number of voters n and the number of alternatives m. The PTAS uses information extraction techniques from fixed size ($\mathcal{O}(1/\epsilon)$) subsets of voters and random rounding of the optimal solution of a linear program.

In the area of fixed parameter tractability (FPT) every instance I of a problem P contains additionally an integer r, called a *parameter*. The goal is to find a *fixed parameter* algorithm (also called FPT algorithm), i.e., an algorithm with running time of the form f(r) poly(|I|), where f is a function, which is typically at least exponential for NPcomplete problems. If such an algorithm exists, we say that the problem P parameterized by r is fixed parameter tractable (FPT). For more details about FPT algorithms see the textbook of Cygan et al. [9] or the survey Bredereck et al. [4] (in the context of computational social choice). The study of FPT algorithms for MINIMAX APPROVAL VOTING was initiated by Misra et al. [28]. They show for example that MINIMAX APPROVAL VOTING parameterized by k (the number of ones in the solution) is W[2]-hard, which implies that there is no FPT algorithm, unless there is a highly unexpected collapse in parameterized complexity classes. From a positive perspective, they show that the problem is FPT when parameterized by the maximum allowed distance d or by the number of votes n. Their algorithm runs in time¹ $\mathcal{O}^{\star}(d^{2d})$ ² For a study on FPT complexity of generalizations of MINIMAX APPROVAL VOTING see Baumeister et al. [2].

Previous results on CLOSEST STRING.

It is interesting to compare the known results on MINIMAX APPROVAL VOTING with the corresponding ones on the better researched CLOSEST STRING. The first PTAS for CLOSEST STRING was given by Li et al. [21] with running time bounded by $n^{\mathcal{O}(1/\epsilon^4)}$ where *n* is the number of the input strings. This was later improved by Andoni et al. [1] to $n^{\mathcal{O}(\frac{\log 1/\epsilon}{\epsilon^2})}$, and then by Ma et al. [26] to $n^{\mathcal{O}(1/\epsilon^2)}$.

The first FPT algorithm for CLOSEST STRING, running in time $\mathcal{O}^*(d^d)$ was given by Gramm et al. [14]. This was later improved by Ma et al. [26], who gave an algorithm with running time $\mathcal{O}^*(2^{\mathcal{O}(d)} \cdot |\Sigma|^d)$, which is more efficient for constant-size alphabets. Further substantial progress is unlikely, since Lokshtanov et al. [24] have shown that CLOSEST STRING admits no algorithms running in time $\mathcal{O}^*(2^{o(d \log d)})$ or $\mathcal{O}^*(2^{o(d \log |\Sigma|)})$, unless the Exponential Time Hypothesis (ETH) [15] fails.

The discrepancy between the state of the art for CLOSEST STRING and MINIMAX APPROVAL VOTING raises interesting questions. First, does the additional constraint on the number of ones in MINIMAX APPROVAL VOTING really make the problem harder and the PTAS has to be significantly slower? Similarly, although in MINIMAX APPROVAL VOTING the alphabet is binary, no $\mathcal{O}^*(2^{\mathcal{O}(d)})$ -time algorithm is known, in contrast to CLOSEST STRING. Can we find such an algorithm? The goal of this work is to answer these questions.

Our results.

We present three results on the complexity of MINIMAX APPROVAL VOTING. Let us recall that the Exponential Time Hypothesis (ETH) of Impagliazzo et al. [15] states that there exists a constant c > 0, such that there is no algorithm solving 3-SAT in time $\mathcal{O}^{\star}(2^{cn})$. In recent years, ETH became the central conjecture used for proving tight bounds on the complexity of various problems, see Lokshtanov et al. [23] for a survey. Nevertheless, ETH-based lower bounds seem largely unexplored in the area of computational social choice [30]. We begin with showing that, unless the ETH fails, there is no algorithm for MINIMAX APPROVAL VOTING running in time $\mathcal{O}^*(2^{o(d \log d)})$. In other words, the algorithm of Misra et.al [28] is essentially optimal, and indeed, in this sense MINIMAX APPROVAL VOTING is harder than CLOSEST STRING. Motivated by this, we then show a parameterized approximation scheme, i.e., a randomized Monte-Carlo algorithm which, given an instance (S, k, d) and a number $\epsilon > 0$, finds a solution at distance at most $(1 + \epsilon)d$ in time $\mathcal{O}^{\star}((3/\epsilon)^{2d})$ or reports that there is no solution at distance at most d (with arbitrarily small positive constant probability of error). Note that our lower bound implies that, under (randomized version of) ETH, this is essentially optimal, i.e., there is no parameterized approximation scheme running in time $\mathcal{O}^{\star}(2^{o(d \log(1/\epsilon))})$. Indeed, if such an algorithm existed, by picking $\epsilon = 1/(d+1)$ we would get an exact algorithm which contradicts our lower bound. Finally, we get a new polynomial-time randomized approximation scheme for MINIMAX APPROVAL VOTING, which runs in time $n^{\mathcal{O}(1/\epsilon^2 \cdot \log(1/\epsilon))} \cdot \operatorname{poly}(m)$ (with arbitrarily small positive constant probability of error). Thus the running time almost matches the one of the fastest known PTAS for CLOSEST STRING (up to a $\log(1/\epsilon)$ factor in the exponent).

 $^{^1\}mathrm{The}\ \mathcal{O}^\star$ notation suppresses factors polynomial in the input size.

²Actually, in the article [28] the authors claim the slightly better running time of $\mathcal{O}^*(d^d)$. However, there is a flaw in the analysis [22, 27]: it states that the initial solution v is at distance at most d from the solution, while it can be at distance 2d because of what we call here the k-completion operation. This increases the maximum depth of the recursion to d (instead of the claimed d/2).

Organization of the paper.

In Section 2 we introduce some notation and we recall standard probability bounds that are used later in the paper. In Section 3 we present our lower bound for MINIMAX APPROVAL VOTING parameterized by *d*. Next, in Section 4 we show a parameterized approximation scheme. Finally, in Section 5 we show a new randomized PTAS. The paper concludes with Section 6, where we discuss directions for future work.

2. DEFINITIONS AND PRELIMINARIES

For every integer n we denote $[n] = \{1, 2, ..., n\}$. For a set of words $S \subseteq \{0, 1\}^m$ and a word $x \in \{0, 1\}^m$ we denote $\mathcal{H}(x, S) = \max_{s \in S} \mathcal{H}(x, s)$. For a string $s \in \{0, 1\}^m$, the number of 1's in s is denoted as $n_1(s)$ and it is also called the Hamming weight of s; similarly $n_0(s) = m - n_1(s)$ denotes the number of zeroes. Moreover, the set of all strings of length m with k ones is denoted by $S_{k,m}$, i.e., $S_{k,m} = \{s \in \{0, 1\}^m : n_1(s) = k\}$. s[j] means the j-th letter of a string s. For a subset of positions $P \subseteq [m]$ we define a subsequence $s|_P$ by removing the letters at positions $[m] \setminus P$ from s.

For a string $s \in \{0, 1\}^m$, any string $s' \in S_{k,m}$ at distance $|n_1(s) - k|$ from s is called a k-completion of s. Note that it is easy to find such a k-completion s': when $n_1(s) \ge k$ we obtain s' by replacing arbitrary $n_1(s) - k$ ones in s by zeroes; similarly when $n_1(s) < k$ we obtain s' by replacing arbitrary $k - n_1(s)$ zeroes in s by ones.

3. A LOWER BOUND

In this section we show a lower bound for MINIMAX AP-PROVAL VOTING parameterized by d. To this end, we use a reduction from a problem called $k \times k$ -CLIQUE. In $k \times k$ -CLIQUE we are given a graph G over the vertex set $V = [k] \times [k]$, i.e., V forms a grid (as a vertex set; the edge set of G is a part of the input and it can be arbitrary) with k rows and kcolumns, and the question is whether in G there is a clique containing exactly one vertex in each row.

LEMMA 3.1. Given an instance I = (G, k) of $k \times k$ -CLIQUE with $k \ge 2$, one can construct an instance I' = (S, k, d) of MINIMAX APPROVAL VOTING, such that I' is a yes-instance iff I is a yes-instance, d = 3k - 3 and the set S contains $\mathcal{O}(k \binom{2k-2}{k-2})$ strings of length $k^2 + 2k - 2$ each. The construction takes time polynomial in the size of the output.

PROOF. Each string in the set S will be of size $m = k^2 + 2k - 2$. Let us split the set of positions [m] into k + 1 blocks, where the first k blocks contain exactly k positions each, and the last (k + 1)-th block contains the remaining 2k - 2 positions. Our construction will enforce that if a solution exists, it will have the following structure: there will be a single 1 in each of the first k blocks and only zeroes in the last block. Intuitively the position of the 1 in the first block encodes the clique vertex of the first row of G, the position of the 1 in the second block encodes the clique vertex of the second row of G, etc.

We construct the set S as follows.

(nonedge strings) For each pair of nonadjacent vertices v, v' ∈ V(G) of G belonging to different rows, i.e., v = (a,b), v' = (a',b'), a ≠ a', we add to S a string s_{vv'}, where all the blocks except a-th and a'-th are filled with zeroes, while the blocks a, a' are filled







with ones, except the *b*-th position in block *a* and the *b'*-th position in block *a'* which are zeroes (see Fig. 1). Formally, $s_{vv'}$ contains ones at positions $\{(a-1)k+j: j \in [k], j \neq b\} \cup \{(a'-1)k+j: j \in [k], j \neq b'\}$. Note that the Hamming weight of $s_{vv'}$ equals 2k-2.

• (row strings) For each row $i \in [k]$ we create exactly $\binom{2k-2}{k-2}$ strings, i.e., for $i \in [k]$ and for each set X of exactly k-2 positions in the (k+1)-th block we add to S a string $s_{i,X}$ having ones at all positions of the *i*-th block and at X, all the remaining positions are filled with zeroes (see Fig. 2). Note that similarly as for the nonedge strings the Hamming weight of each row string equals 2k-2, and to achieve this property we use the (k+1)-th block.

To finish the description of the created instance I' = (S, k, d) we need to define the target distance d, which we set as d = 3k - 3. Observe that as the Hamming weight of each string $s' \in S$ equals 2k - 2, for $s \in \{0, 1\}^m$ with exactly k ones we have $\mathcal{H}(s, s') \leq d$ if and only if the positions of ones in s and s' have a non-empty intersection.

Let us assume that there is a clique K in G of size k containing exactly one vertex from each row. For $i \in [k]$ let $j_i \in [k]$ be the column number of the vertex of K from row i. Define s as a string containing ones exactly at positions $\{(i-1)k + j_i : i \in [k]\}$, i.e., the (k + 1)-th block contains only zeroes and for $i \in [k]$ the *i*-th block contains a single 1 at position j_i . Obviously s contains exactly k ones, hence it suffices to show that s has at least one common one with each of the strings in S. This is clear for the row strings, as each row string contains a block full of ones. For a nonedge string $s_{vv'}$, where v = (a, b) and v' = (a', b') note that K does not contain v and v' at the same time. Consequently s has a common one with $s_{vv'}$ in at least one of the blocks a, a'.

In the other direction, assume that s is a string of length m with exactly k ones such that the Hamming distance between s and each of the strings in S is at most d, which by construction implies that s has a common one with each of the strings in S. First, we are going to prove that s contains a 1 in each of the first k blocks (and consequently has only zeroes in block k + 1). For the sake of contradiction assume that this is not the case. Consider a block $i \in [k]$ containing only zeroes. Let X be any set of k - 2 positions in block k + 1 holding only zeroes in s (such a set exists as block k + 1 has 2k - 2 positions). But the row string $s_{i,X}$ has 2k - 2 ones at positions where s has zeroes, and consequently $\mathcal{H}(s, s_{i,X}) = k + (2k - 2) = 3k - 2 > d = 3k - 3$, a contradiction.

As we know that s contains exactly one one in each of the first k blocks let $j_i \in [k]$ be such a position of block $i \in [k]$. Create $X \subseteq V(G)$ by taking the vertex from column j_i for each row $i \in [k]$. Clearly X is of size k and it contains exactly one vertex from each row, hence it remains to prove that Xis a clique in G. Assume the contrary and let $v, v' \in X$ be two distinct nonadjacent vertices of X, where $v = (i, j_i)$ and $v' = (i', j_{i'})$. Observe that the nonedge string $s_{vv'}$ contains zeroes at the j_i -th position of the *i*-th block and at the $j_{i'}$ -th position of the i'-th block. Since for $i'' \in [k], i'' \neq i, i'' \neq i'$ block i'' of $s_{vv'}$ contains only zeroes, we infer that the sets of positions of ones of s and $s_{vv'}$ are disjoint leading to $\mathcal{H}(s, s_{vv'}) = k + (2k - 2) = 3k - 2 > d, \text{ a contradiction.}$

As we have proved that I is a yes-instance of $k \times k$ -CLIQUE iff I' is a yes-instance of MINIMAX APPROVAL VOTING, the lemma follows. \Box

In order to derive an ETH-based lower bound we need the following theorem of Lokshtanov et al. [24].

THEOREM 3.2. (Lokshtanov et al. [24]) Assuming ETH, there is no $2^{o(k \log k)}$ -time algorithm for $k \times k$ -CLIQUE.

We are ready to prove the main result of this section.

THEOREM 3.3. There is no $2^{o(d \log d)} \operatorname{poly}(n, m)$ -time algorithm for MINIMAX APPROVAL VOTING unless ETH fails.

PROOF. Using Lemma 3.1, the input instance G of $k \times k$ -CLIQUE is transformed into an equivalent instance I' =(S, k, d) of MINIMAX APPROVAL VOTING, where n = |S| = $\mathcal{O}(k\binom{2k-2}{k-2}) = 2^{\mathcal{O}(k)}$, each string of S has length $m = \mathcal{O}(k^2)$ and $d = \Theta(k)$. Using a $2^{o(d \log d)} \operatorname{poly}(n, m)$ -time algorithm for MINIMAX APPROVAL VOTING we can solve $k \times k$ -CLIQUE in time $2^{o(k \log k)} 2^{\mathcal{O}(k)} = 2^{o(k \log k)}$, which contradicts ETH by Theorem 3.2. \Box

PARAMETERIZED APPROXIMATION 4. SCHEME

In this section we show the following theorem.

THEOREM 4.1. There exists a randomized algorithm which, given an instance $(\{s_i\}_{i=1,...,n}, k, d)$ of MINIMAX APPROVAL VOTING and any $\epsilon \in (0,3)$, runs in time $\mathcal{O}\left(\left(\frac{3}{\epsilon}\right)^{2d}mn\right)$ and either

- (i) reports a solution at distance at most $(1 + \epsilon)d$ from S, or
- (ii) reports that there is no solution at distance at most d from S.

In the latter case, the answer is correct with probability at least 1 - p, for arbitrarily small fixed p > 0.

Let us proceed with the proof. In what follows we assume p = 1/2, since then we can get the claim even if p < 1/2by repeating the whole algorithm $\lceil \log_2(1/p) \rceil$ times. Indeed, then the algorithm returns an incorrect answer only if each of the $\lceil \log_2(1/p) \rceil$ repetitions returned an incorrect answer, which happens with probability at most $(1/2)^{\log_2(1/p)} = p$.

Assume we are given a yes-instance and let us fix a solution $s^* \in S_{k,m}$, i.e., a string at distance at most d from all the input strings. Our approach is to begin with a string Pseudocode 1: Parameterized approximation scheme for MINIMAX APPROVAL VOTING.

- 1 if $|n_1(s_1) k| > d$ then return NO;
- **2** $x_0 \leftarrow$ any k-completion of s_1 ;
- **3** for $j \in \{1, 2, ..., d\}$ do
- 4
- 5
- 6
- 7
- $\begin{array}{l} j \in \{1, 2, ..., a\} \text{ do} \\ \text{if } \mathcal{H}(x_{j-1}, S) \leq (1+\epsilon)d \text{ then return } x_{j-1}; \\ \text{otherwise there exists } s_i \text{ s.t. } \mathcal{H}(x_{j-1}, s_i) > (1+\epsilon)d; \\ P_{j,0} \leftarrow \{a \in [m] : 0 = x_{j-1}[a] \neq s_i[a] = 1\}; \\ P_{j,1} \leftarrow \{a \in [m] : 1 = x_{j-1}[a] \neq s_i[a] = 0\}; \\ \text{if } \min(|P_{j,0}|, |P_{j,1}|) = 0 \text{ then return NO}; \\ \text{Cet } x_i \text{ from } x_i \text{ by suppring } 0 \text{ and } 1 \text{ on pair of ran} \end{array}$ 8
- Get x_j from x_{j-1} by swapping 0 and 1 on pair of random 9 positions from $P_{j,0}$ and $P_{j,1}$;
- 10 if $\mathcal{H}(x_d, S) \leq (1+\epsilon)d$ then return x_d ;

11 else return NO;

 $x_0 \in S_{k,m}$ not very far from s^* , and next perform a number of steps. In the *j*-th step we either conclude that x_{j-1} is already a $(1 + \epsilon)$ -approximate solution, or with some probability we find another string x_i which is closer to s^* .

First observe that if $|n_1(s_1) - k| > d$, then clearly there is no solution and our algorithm reports NO. Hence in what follows we assume

$$|n_1(s_1) - k| \le d.$$
 (1)

We set x_0 to be any k-completion of s_1 . By (1) we get $\mathcal{H}(x_0, s_1) \leq d$. Since $\mathcal{H}(s_1, s^*) \leq d$, by the triangle inequality we get the following bound.

$$\mathcal{H}(x_0, s^*) \le \mathcal{H}(x_0, s_1) + \mathcal{H}(s_1, s^*) \le 2d.$$
(2)

Now we are ready to describe our algorithm precisely (see also Pseudocode 1). We begin with x_0 defined as above. We are going to create a sequence of strings x_0, x_1, \ldots satisfying $n_1(x_j) = k$ for every j. For $j = 1, \ldots, d$ we do the following. If for every $i = 1, \ldots, n$ we have $\mathcal{H}(x_{i-1}, s_i) \leq (1 + \epsilon)d$ the algorithm terminates and returns x_{j-1} . Otherwise, fix any $i = 1, \ldots, n$ such that $\mathcal{H}(x_{j-1}, s_i) > (1 + \epsilon)d$. Let $P_{j,0} = \{a \in [m] : 0 = x_{j-1}[a] \neq s_i[a] = 1\}$ and $P_{j,1} = \{a \in a \in a\}$ $[m]: 1 = x_{j-1}[a] \neq s_i[a] = 0$. The algorithm samples a position $a_0 \in P_{j,0}$ and a position $a_1 \in P_{j,1}$. In case $P_{j,0} = \emptyset$ or $P_{j,1} = \emptyset$ we return NO because it means that $\mathcal{H}(s_i, S_{k,m}) =$ $\mathcal{H}(s_i, x_{j-1}) > d$. Then, x_j is obtained from x_{j-1} by swapping the 0 at position a_0 with the 1 at position a_1 . If the algorithm finishes without finding a solution, it reports NO.

The following lemma is the key to get a lower bound on the probability that the x_j 's get close to s^* .

LEMMA 4.2. Let x be a string in $S_{k,m}$ such that $\mathcal{H}(x, s_i) \geq$ $(1+\epsilon)d$ for some $i = 1, \ldots, n$. Let $s^* \in S_{k,m}$ be any solution, *i.e.*, a string at distance at most d from all the strings s_i , $i = 1, \ldots, n$. Denote

$$P_0^* = \{a \in [m] : 0 = x[a] \neq s_i[a] = s^*[a] = 1\},\$$
$$P_1^* = \{a \in [m] : 1 = x[a] \neq s_i[a] = s^*[a] = 0\}.$$

Then,

$$\min(|P_0^*|, |P_1^*|) \ge \frac{\epsilon d}{2}.$$

PROOF. Let P be the set of positions on which x and s_i differ, i.e., $P = \{a \in [m] : x[a] \neq s_i[a]\}$ (see Fig. 3). Note that $P_0^* \cup P_1^* \subseteq P$. Let $Q = [m] \setminus P$.

The intuition behind the proof is that if $\min(|P_0^*|, |P_1^*|)$ is small, then s^* differs too much from s_i , either because



Figure 3: Strings x, s_i and s^* after permuting the positions.

 $s^*|_P$ is similar to $x|_P$ (when $|P_0^*| \approx |P_1^*|$) or because $s^*|_Q$ has much more 1's than $s_i|_Q$ (when $|P_0^*|$ differs much from $|P_1^*|$).

We begin with a couple of useful observations on the number of ones in different parts of x, s_i and s^* . Since xand s_i are the same on Q, we get

$$n_1(x|_Q) = n_1(s_i|_Q).$$
(3)

Since $n_1(x) = n_1(s^*)$, we get $n_1(x|_P) + n_1(x|_Q) = n_1(s^*|_P) + n_1(s^*|_Q)$, and further

$$n_1(s^*|_Q) - n_1(x|_Q) = n_1(x|_P) - n_1(s^*|_P).$$
(4)

Finally note that

$$n_1(s^*|_P) = |P_0^*| + n_1(x|_P) - |P_1^*|.$$
(5)

We are going to derive a lower bound on $\mathcal{H}(s_i, s^*)$. First,

$$\begin{aligned} \mathcal{H}(s_i|_P, s^*|_P) &= |P| - (|P_0^*| + |P_1^*|) = \\ &= \mathcal{H}(x, s_i) - (|P_0^*| + |P_1^*|) \ge (1 + \epsilon)d - (|P_0^*| + |P_1^*|). \end{aligned}$$

On the other hand,

$$\mathcal{H}(s_i|Q, s^*|Q) \ge |n_1(s^*|Q) - n_1(s_i|Q)| =
\stackrel{(3)}{=} |n_1(s^*|Q) - n_1(x|Q)| =
\stackrel{(4)}{=} |n_1(x|P) - n_1(s^*|P)| =
\stackrel{(5)}{=} ||P_1^*| - |P_0^*||.$$

It follows that

(

$$d \ge \mathcal{H}(s_i, s^*) = \mathcal{H}(s_i|_P, s^*|_P) + \mathcal{H}(s_i|_Q, s^*|_Q) \ge \\ \ge (1+\epsilon)d - (|P_0^*| + |P_1^*|) + ||P_1^*| - |P_0^*|| = \\ = (1+\epsilon)d - 2\min(|P_0^*|, |P_1^*|).$$

Hence, $\min(|P_0^*|, |P_1^*|) \geq \frac{\epsilon d}{2}$ as required. \Box

COROLLARY 4.3. Assume that there is a solution $s^* \in S_{k,m}$ and that the algorithm created a string x_j , for some $j = 0, \ldots, d$. Then,

$$\Pr[\mathcal{H}(x_j, s^*) \le 2d - 2j] \ge \left(\frac{\epsilon}{3}\right)^{2j}$$

PROOF. We use induction on j. For j = 0 the claim follows from (2). Consider j > 0. By the induction hypothesis,

$$\Pr[\mathcal{H}(x_{j-1}, s^*) \le 2d - 2j + 2] \ge \left(\frac{\epsilon}{3}\right)^{2j-2}.$$
 (6)

Assume that $\mathcal{H}(x_{j-1}, s^*) \leq 2d - 2j + 2$. Since x_j was created, $\mathcal{H}(x_{j-1}, s_i) > (1+\epsilon)d$ for some $i = 1, \ldots, n$. Since $\mathcal{H}(s^*, s_i) \leq d$, by the triangle inequality we get the following.

$$|P_{j,0}| + |P_{j,1}| = \mathcal{H}(x_{j-1}, s_i) \le \le \mathcal{H}(x_{j-1}, s^*) + \mathcal{H}(s^*, s_i) \le 3d - 2j + 2 \le 3d.$$
(7)

Then, by Lemma 4.2

$$\Pr[\mathcal{H}(x_j, s^*) \le 2d - 2j \mid \mathcal{H}(x_{j-1}, s^*) \le 2d - 2j + 2] \ge \\ \ge \frac{|P_0^*| \cdot |P_1^*|}{|P_{j,0}| \cdot |P_{j,1}|} \ge \frac{\left(\frac{\epsilon d}{2}\right)^2}{\left(\frac{3d}{2}\right)^2} = \left(\frac{\epsilon}{3}\right)^2.$$
(8)

The claim follows by combining (6) and (8). \Box

In order to increase the success probability, we repeat the algorithm until a solution is found or the number of repetitions is at least $(3/\epsilon)^{2d}$. By Corollary 4.3 the probability that there is a solution but it was not found is bounded by

$$\left(1 - \left(\frac{\epsilon}{3}\right)^{2d}\right)^{(3/\epsilon)^{2d}} = \left(1 - \frac{1}{(3/\epsilon)^{2d}}\right)^{(3/\epsilon)^{2d}} \le \frac{1}{e} < \frac{1}{2}.$$

This finishes the proof of Theorem 4.1.

5. A FASTER POLYNOMIAL TIME APPROXIMATION SCHEME

The goal of this section is to present a PTAS for the optimization version of MINIMAX APPROVAL VOTING running in time $n^{\mathcal{O}(1/\epsilon^2 \cdot \log(1/\epsilon))} \cdot \operatorname{poly}(m)$. It is achieved by combining the parameterized approximation scheme from Theorem 4.1 with the following result, which might be of independent interest. Throughout this section OPT denotes the value of an optimum solution s for the given instance $(\{s_i\}_{i=1,...,n}, k)$ of MINIMAX APPROVAL VOTING, i.e., OPT = $\max_{i=1,...,n} \mathcal{H}(s, s_i)$,

THEOREM 5.1. There exists a randomized polynomial time algorithm which, for arbitrarily small fixed p > 0, given an instance $(\{s_i\}_{i=1,...,n}, k)$ of MINIMAX APPROVAL VOTING and any $\epsilon > 0$ such that OPT $\geq \frac{122 \ln n}{\epsilon^2}$, reports a solution, which with probability at least 1 - p is at distance at most $(1 + \epsilon) \cdot \text{OPT}$ from S.

In what follows, we prove Theorem 5.1. As in the proof of Theorem 4.1 we assume w.l.o.g. p = 1/2. Note that we can assume $\epsilon < 1$, for otherwise it suffices to use the 2-approximation of Caragiannis et al. [6]. We also assume $n \ge 3$, for otherwise it is a straightforward exercise to find an optimal solution in linear time. Let us define a linear program (9–12):

minimize
$$d$$
 (9)

$$\sum_{j=1}^{m} x_j = k \tag{10}$$

$$\sum_{\substack{j=1,\dots,m\\s_i[j]=1}} (1-x_j) + \sum_{\substack{j=1,\dots,m\\s_i[j]=0}} x_j \le d \quad \forall i \in \{1,\dots,n\}$$
(11)

$$x_j \in [0,1] \quad \forall j \in \{1,\ldots,m\}$$
(12)

The linear program (9–12) is a relaxation of the natural integer program for MINIMAX APPROVAL VOTING, obtained by replacing (12) by the discrete constraint $x_j \in \{0, 1\}$. Indeed, observe that x_j corresponds to the *j*-th letter of the solution $x = x_1 \cdots x_m$, (10) states that $n_1(x) = k$, and (11) states that $\mathcal{H}(x, S) \leq d$.

Our algorithm is as follows (see Pseudocode 2). First we solve the linear program in time poly(n, m) using the interior

Pseudocode 2: The algorithm from Theorem 5.1

 $\begin{array}{c|c} \mathbf{1} & \text{Solve the LP } (9{-}12) \text{ obtaining an optimal solution} \\ & (x_1^*, \dots, x_m^*, d^*); \\ \mathbf{2} & \text{for } j \in \{1, 2, \dots, m\} \text{ do} \\ \mathbf{3} & & \text{Set } x[j] \leftarrow 1 \text{ with probability } x_j^* \text{ and } x[j] \leftarrow 0 \text{ with} \\ & & \text{ probability } 1 - x_j^* \end{array}$

- 4 $y \leftarrow$ any k-completion of x;
- 5 return y

point method [16]. Let $(x_1^*, \ldots, x_m^*, d^*)$ be the obtained optimal solution. Clearly, $d^* \leq \text{OPT}$. We randomly construct a string $x \in \{0, 1\}^m$, guided by the values x_j^* . More precisely, for every $j = 1, \ldots, m$ independently, we set x[j] = 1 with probability x_j^* . Note that x needs not contain k ones. Let y by any k-completion of x. The algorithm returns y.

Clearly, the above algorithm runs in polynomial time. In what follows we bound the probability of error. To this end we prove upper bounds on the probability that x is far from S and the probability that the number of ones in x is far from k. This is done in Lemmas 5.3 and 5.4, which can be shown using standard Chernoff bounds (see e.g. Chapter 4.1 in [29]).

THEOREM 5.2. (Motwani et al. [29]) Let X_1, X_2, \ldots, X_n be n independent random 0-1 variables such that for every $i = 1, \ldots, n$ we have $\Pr[X_i = 1] = p_i$, for $p_i \in [0, 1]$. Let $X = \sum_{i=1}^n X_i$. Then,

• for any $0 < \epsilon \leq 1$ we have:

$$\Pr\left[X > (1+\epsilon) \cdot \mathbb{E}\left[X\right]\right] \le \exp\left(-\frac{1}{3}\epsilon^2 \cdot \mathbb{E}\left[X\right]\right) \quad (13)$$

$$\Pr\left[X < (1-\epsilon) \cdot \mathbb{E}\left[X\right]\right] \le \exp\left(-\frac{1}{2}\epsilon^2 \cdot \mathbb{E}\left[X\right]\right) \quad (14)$$

• for any $1 < \epsilon$ we have:

Г

$$\Pr\left[X > (1+\epsilon) \cdot \mathbb{E}\left[X\right]\right] \le \exp\left(-\frac{1}{3}\epsilon \cdot \mathbb{E}\left[X\right]\right) \quad (15)$$

$$\Pr\left[X < (1 - \epsilon) \cdot \mathbb{E}\left[X\right]\right] = 0 \tag{16}$$

Lemma 5.3.

$$\Pr\left[\mathcal{H}(x,S) > (1 + \frac{\epsilon}{2}) \cdot \operatorname{OPT}\right] \le \frac{1}{4}.$$

PROOF. For every $i = 1, \ldots, n$ we define a random variable D_i that measures the distance between x^* and s_i

$$D_i = \sum_{\substack{j \in [m] \\ s_i[j]=1}} (1 - x[j]) + \sum_{\substack{j \in [m] \\ s_i[j]=0}} x[j].$$

Note that x[i] are independent 0-1 random variables. Using linearity of the expectation we obtain

$$\mathbb{E}[D_i] = \mathbb{E}\left[\sum_{j \in [m], s_i[j]=1} (1 - x[j]) + \sum_{j \in [m], s_i[j]=0} x[j]\right] = \\ = \sum_{j \in [m], s_i[j]=1} (1 - \mathbb{E}[x[j]]) + \sum_{j \in [m], s_i[j]=0} \mathbb{E}[x[j]] = \\ = \sum_{j \in [m], s_i[j]=1} (1 - x_j^*) + \sum_{j \in [m], s_i[j]=0} x_j^* \le \\ \le d^* \le \text{OPT.}$$
(17)

Note that D_i is a sum of m independent 0-1 random variables $X_j = 1-x[j]$ when $s_i[j] = 1$ and $X_j = x[j]$ otherwise. Denote

 $\delta = \epsilon \cdot \frac{\text{OPT}}{2\mathbb{E}[D_i]}$. We apply Chernoff bounds. For $\delta < 1$ we have

$$\Pr[D_i > \left(1 + \frac{\epsilon}{2}\right) \cdot \operatorname{OPT}] \stackrel{(17)}{\leq} \\ \leq \Pr\left[D_i > \mathbb{E}[D_i] + \frac{\epsilon}{2} \cdot \operatorname{OPT}\right] = \\ = \Pr\left[D_i > (1 + \delta) \cdot \mathbb{E}[D_i]\right] \stackrel{(13)}{\leq} \\ \leq \exp\left(-\frac{1}{3}\left(\epsilon \cdot \frac{\operatorname{OPT}}{2\mathbb{E}[D_i]}\right)^2 \mathbb{E}[D_i]\right) \stackrel{(17)}{\leq} \\ \leq \exp\left(-\frac{\epsilon^2 \cdot \operatorname{OPT}}{12}\right).$$

In case $\delta \geq 1$ we proceed analogously, using the Chernoff bound (15)

$$\Pr[D_i > \left(1 + \frac{\epsilon}{2}\right) \cdot \text{OPT}] \stackrel{(15)}{\leq} \\ \leq \exp\left(-\frac{\epsilon \cdot \text{OPT}}{6}\right) \stackrel{1>\epsilon}{\leq} \exp\left(-\frac{\epsilon^2 \cdot \text{OPT}}{12}\right).$$

Now we use the union bound to get the claim.

$$\Pr\left[\mathcal{H}(x,S) > (1+\frac{\epsilon}{2}) \cdot \operatorname{OPT}\right] =$$

$$= \Pr\left[\exists i \in [n] \quad D_i > (1+\frac{\epsilon}{2}) \cdot \operatorname{OPT}\right] \leq$$

$$\leq n \cdot \exp\left(-\frac{\epsilon^2 \cdot \operatorname{OPT}}{12}\right) \leq$$

$$\leq n \cdot \exp\left(-\frac{\frac{122 \ln n}{\operatorname{OPT}} \cdot \operatorname{OPT}}{12}\right) < n^{-9} \stackrel{n \geq 3}{\leq} \frac{1}{4}.$$
(18)

Lemma 5.4.

$$\Pr\left[|n_1(x) - k| > \frac{\epsilon}{2} \cdot \operatorname{OPT}\right] \le \frac{1}{4}.$$

PROOF. First we note that

$$\mathbb{E}[n_1(x)] = \mathbb{E}\Big[\sum_{j \in [m]} x[j]\Big] = \sum_{j \in [m]} \mathbb{E}[x[j]] = \sum_{j \in [m]} x_j^* = k.$$
(19)

Pick an $i = 1, \ldots, n$. Define the random variables

$$E_i = \sum_{j \in [m], s_i[j]=1} (1 - x[j]), \quad F_i = \sum_{j \in [m], s_i[j]=0} x[j].$$

Let $D_i = E_i + F_i$, as in the proof of Lemma 5.3. By (17) we have

$$\mathbb{E}[E_i] \le \mathbb{E}[E_i] + \mathbb{E}[F_i] = \mathbb{E}[D_i] \le \text{OPT}$$
(20)

$$\mathbb{E}[F_i] \le \mathbb{E}[E_i] + \mathbb{E}[F_i] = \mathbb{E}[D_i] \le \text{OPT}$$
(21)

Both E_i and F_i are sums of independent 0-1 random variables and we apply Chernoff bounds as follows. When $\frac{1}{4} \epsilon \cdot \frac{OPT}{\mathbb{E}[E_i]} \leq 1$ then using (13) and (14) we obtain

$$\Pr\left[\left|E_{i} - \mathbb{E}[E_{i}]\right| > \frac{1}{4}\epsilon \cdot \operatorname{OPT}\right] \stackrel{(13),(14)}{\leq} \\ \leq \exp\left(-\frac{1}{3} \cdot \frac{1}{16}\epsilon^{2} \cdot \frac{(\operatorname{OPT})^{2}}{\mathbb{E}^{2}[E_{i}]} \cdot \mathbb{E}[E_{i}]\right) + \\ + \exp\left(-\frac{1}{2} \cdot \frac{1}{16}\epsilon^{2} \cdot \frac{(\operatorname{OPT})^{2}}{\mathbb{E}^{2}[E_{i}]} \cdot \mathbb{E}[E_{i}]\right) \stackrel{20}{\leq} \\ \leq 2 \cdot \exp\left(-\frac{1}{48}\epsilon^{2} \cdot \operatorname{OPT}\right),$$

otherwise $\left(\frac{1}{4}\epsilon \cdot \frac{\text{OPT}}{\mathbb{E}[E_i]} > 1\right)$, using (15) and (16), we have

$$\begin{split} &\Pr\left[\left|E_{i} - \mathbb{E}[E_{i}]\right| > \frac{1}{4}\epsilon \cdot \operatorname{OPT}\right] \stackrel{(15),(16)}{\leq} \\ &\leq \exp\left(-\frac{1}{3} \cdot \frac{1}{4}\epsilon \cdot \frac{\operatorname{OPT}}{\mathbb{E}[E_{i}]} \cdot \mathbb{E}[E_{i}]\right) + 0 \leq \\ &\leq \exp\left(-\frac{1}{12}\epsilon \cdot \operatorname{OPT}\right) \stackrel{1 > \epsilon}{\leq} 2 \cdot \exp\left(-\frac{1}{48}\epsilon^{2} \cdot \operatorname{OPT}\right). \end{split}$$

To sum up, in both cases we have shown that

$$\Pr\left[\left|E_{i} - \mathbb{E}[E_{i}]\right| > \frac{\epsilon}{4} \cdot \operatorname{OPT}\right] \le 2 \cdot \exp\left(-\frac{1}{48}\epsilon^{2} \cdot \operatorname{OPT}\right).$$
(22)

Similarly we show

$$\Pr\left[\left|F_{i} - \mathbb{E}[F_{i}]\right| > \frac{\epsilon}{4} \cdot \operatorname{OPT}\right] \leq 2 \cdot \exp\left(-\frac{1}{48}\epsilon^{2} \cdot \operatorname{OPT}\right).$$
(23)

We see that

1

$$n_1(x) = \sum_{j \in [m]} x[j] = n_1(s_i) - \sum_{j \in [m], s_i[j]=1} (1 - x[j]) + \sum_{j \in [m], s_i[j]=0} x[j] = n_1(s_i) - E_i + F_i \quad (24)$$

and hence

$$\mathbb{E}[n_1(x)] = n_1(s_i) - \mathbb{E}[E_i] + \mathbb{E}[F_i].$$
(25)

Additionally we will use

$$\forall x, y \in \mathbb{R} \quad |x - y| > a \implies |x| > a/2 \lor |y| > a/2.$$
 (26)

Now we can write

$$\begin{aligned} \Pr\left[\left|n_{1}(x)-k\right| &> \frac{1}{2}\epsilon \cdot \operatorname{OPT}\right] \stackrel{(19)}{=} \\ &= \Pr\left[\left|n_{1}(x)-\mathbb{E}[n_{1}(x)]\right| &> \frac{1}{2}\epsilon \cdot \operatorname{OPT}\right] \stackrel{(24),(25)}{=} \\ &= \Pr\left[\left|n_{1}(s_{i})-E_{i}+F_{i}+\right. \\ &\left.-n_{1}(s_{i})+\mathbb{E}[E_{i}]-\mathbb{E}[F_{i}]\right| &> \frac{1}{2}\epsilon \cdot \operatorname{OPT}\right] \stackrel{(26)}{\leq} \\ &\leq \Pr\left[\left|E_{i}-\mathbb{E}[E_{i}]\right| &> \frac{1}{4}\epsilon \cdot \operatorname{OPT} \quad \lor \right. \\ &\left. \lor \quad \left|F_{i}-\mathbb{E}[F_{i}]\right| &> \frac{1}{4}\epsilon \cdot \operatorname{OPT}\right] \leq \\ &\leq \Pr\left[\left|E_{i}-\mathbb{E}[E_{i}]\right| &> \frac{1}{4}\epsilon \cdot \operatorname{OPT}\right] + \\ &\left.+\Pr\left[\left|F_{i}-\mathbb{E}[F_{i}]\right| &> \frac{1}{4}\epsilon \cdot \operatorname{OPT}\right] \stackrel{(22),(23)}{\leq} \\ &\leq 4 \cdot \exp\left(-\frac{1}{48}\epsilon^{2} \cdot \operatorname{OPT}\right) \stackrel{\text{assum.}}{\leq} 4 \cdot \exp\left(-\frac{122\ln n}{48}\right) \stackrel{n \geq 3}{\leq} \frac{1}{4}. \end{aligned}$$

Now we can finish the proof of Theorem 5.1. By Lemmas 5.3 and 5.4 with probability at least 1/2 both $\mathcal{H}(x, S) \leq (1 + \frac{1}{2}\epsilon) \cdot \text{OPT}$ and $\mathcal{H}(y, x) = |n_1(x) - k| \leq \frac{1}{2}\epsilon \cdot \text{OPT}$. By the triangle inequality this implies that $\mathcal{H}(y, S) \leq (1 + \epsilon) \cdot \text{OPT}$, with probability at least 1/2 as required.

We conclude the section by combining Theorems 4.1 and 5.1 to get a faster PTAS.

THEOREM 5.5. For each $\epsilon > 0$ we can find $(1 + \epsilon)$ approximation solution for the MINIMAX APPROVAL VOTING

problem in time $n^{\mathcal{O}\left(\frac{\log 1/\epsilon}{\epsilon^2}\right)} \cdot \operatorname{poly}(m)$ with probability at least 1-r, for any fixed r > 0.

PROOF. First we run algorithm from Theorem 4.1 for $d = \lceil \frac{122 \ln n}{\epsilon^2} \rceil$ and p = r/2.

If it reports a solution, for every $d' \leq d$ we apply Theorem 4.1 with p = r/2 and we return the best solution. If $OPT \geq d$, even the initial solution is at distance at most $(1 + \epsilon)d \leq (1 + \epsilon)OPT$ from S. Otherwise, at some point d' = OPT and we get $(1 + \epsilon)$ -approximation with probability at least 1 - r/2 > 1 - r.

In the case when the initial run of the algorithm from Theorem 4.1 reports NO, we just apply the algorithm from Theorem 5.1, again with p = r/2. With probability at least 1 - r/2 the answer NO of the algorithm from Theorem 4.1 is correct. Conditioned on that, we know that $OPT > d \geq \frac{122 \ln n}{\epsilon^2}$ and then the algorithm from Theorem 5.1 returns a $(1 + \epsilon)$ -approximation with probability at least 1 - r/2. Thus, the answer is correct with probability at least $(1 - r/2)^2 > 1 - r$.

The total running time can be bounded as follows.

$$\mathcal{O}^*\left(\left(\frac{3}{\epsilon}\right)^{\frac{244\ln n}{\epsilon^2}}\right) \subseteq n^{\mathcal{O}\left(\frac{\log 1/\epsilon}{\epsilon^2}\right)} \cdot \operatorname{poly}(m)$$

6. FURTHER RESEARCH

We conclude the paper with some questions related to this work that are left unanswered. Our PTAS for MINIMAX APPROVAL VOTING is randomized, and it seems there is no direct way of derandomizing it. It might be interesting to find an equally fast deterministic PTAS. The second question is whether there are even faster PTASes for CLOSEST STRING or MINIMAX APPROVAL VOTING. Recently, Cygan et al. [10] showed that under ETH, there is no PTAS in time $f(\epsilon) \cdot n^{o(1/\epsilon)}$ for CLOSEST STRING. This extends to the same lower bound for MINIMAX APPROVAL VOTING, since we can try all values $k = 0, 1, \ldots, m$. It is a challenging open problem to close the gap in the running time of PTAS either for CLOSEST STRING or for MINIMAX APPROVAL VOTING.

Acknowledgments.

Marek Cygan would like to thank Daniel Lokshtanov for helpful conversations about existing algorithms for the Closest (Sub)String problem. The authors thank Piotr Skowron for helpful remarks concerning the introduction and they thank reviewers of AAAI-17 and EXPLORE-2017 for their insightful comments on the paper. The work of M. Cygan is a part of the project TOTAL that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 677651). L. Kowalik and A. Socała were supported by the National Science Centre, Poland, grant number 2013/09/B/ST6/03136. K. Sornat was supported by the National Science Centre, Poland, grant number 2015/17/N/ST6/03684.



REFERENCES

- A. Andoni, P. Indyk, and M. Patrascu. On the Optimality of the Dimensionality Reduction Method. In 47th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2006, pages 449–458, 2006.
- [2] D. Baumeister, T. Böhnlein, L. Rey, O. Schaudt, and A. Selker. Minisum and Minimax Committee Election Rules for General Preference Types. In ECAI 2016 -22nd European Conference on Artificial Intelligence, volume 285 of Frontiers in Artificial Intelligence and Applications, pages 1656–1657. IOS Press, 2016.
- [3] S. J. Brams, D. M. Kilgour, and M. R. Sanver. A Minimax Procedure for Electing Committees. *Public Choice*, 132(3-4):401–420, 2007.
- [4] R. Bredereck, J. Chen, P. Faliszewski, J. Guo, R. Niedermeier, and G. J. Woeginger. Parameterized Algorithmics for Computational Social Choice: Nine Research Challenges. *Tsinghua Science and Technology*, 19(4):358–373, Aug 2014.
- [5] J. Byrka and K. Sornat. PTAS for Minimax Approval Voting. In Proceedings of 10th International Conference Web and Internet Economics, WINE 2014, pages 203–217, 2014.
- [6] I. Caragiannis, D. Kalaitzis, and E. Markakis. Approximation Algorithms and Mechanism Design for Minimax Approval Voting. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI, 2010.
- J. R. Chamberlin and P. N. Courant. Representative Deliberations and Representative Decisions: Proportional Representation and the Borda Rule. *American Political Science Review*, 77:718–733, 9 1983.
- [8] V. Conitzer. Making Decisions Based on the Preferences of Multiple Agents. Commun. ACM, 53(3):84–94, 2010.
- [9] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.
- [10] M. Cygan, D. Lokshtanov, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. Lower Bounds for Approximation Schemes for Closest String. In 15th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2016, pages 12:1–12:10, 2016.
- [11] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank Aggregation Methods for the Web. In *Proceedings* of the Tenth International World Wide Web Conference, WWW 2001, pages 613–622, 2001.
- [12] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of Multiwinner Voting Rules. *Social Choice and Welfare*, 48(3):599–632, 2017.
- [13] P. C. Fishburn. Axioms for Approval Voting: Direct Proof. Journal of Economic Theory, 19(1):180–185, 1978.
- [14] J. Gramm, R. Niedermeier, and P. Rossmanith. Fixed-Parameter Algorithms for Closest String and Related Problems. *Algorithmica*, 37(1):25–42, 2003.
- [15] R. Impagliazzo and R. Paturi. On the Complexity of k-SAT. J. Comput. Syst. Sci., 62(2):367–375, 2001.
- [16] N. Karmarkar. A New Polynomial-time Algorithm for Linear Programming. *Combinatorica*, 4(4):373–396, 1984.

- [17] D. M. Kilgour. Approval Balloting for Multi-winner Elections. In J.-F. Laslier and R. M. Sanver, editors, *Handbook on Approval Voting*, pages 105–124. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [18] J. Laslier and M. Sanver. Handbook on Approval Voting. Studies in Choice and Welfare. Springer Berlin Heidelberg, 2010.
- [19] R. LeGrand. Analysis of the Minimax Procedure. Technical Report WUCSE-2004-67, Department of Computer Science and Engineering, Washington University, St. Louis, Missouri, 2004.
- [20] R. LeGrand, E. Markakis, and A. Mehta. Some Results on Approximating the Minimax Solution in Approval Voting. In 6th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2007, pages 1193–1195, 2007.
- [21] M. Li, B. Ma, and L. Wang. On the Closest String and Substring Problems. *Journal of the ACM*, 49(2):157–171, 2002.
- [22] H. Liu and J. Guo. Parameterized Complexity of Winner Determination in Minimax Committee Elections. In Proceedings of the 2016 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2016, pages 341–349, 2016.
- [23] D. Lokshtanov, D. Marx, and S. Saurabh. Lower Bounds Based on the Exponential Time Hypothesis. Bulletin of the EATCS, 105:41–72, 2011.
- [24] D. Lokshtanov, D. Marx, and S. Saurabh. Slightly Superexponential Parameterized Problems. In Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, pages 760-776, 2011.
- [25] T. Lu and C. Boutilier. Budgeted Social Choice: From Consensus to Personalized Decision Making. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011, pages 280–286, 2011.
- [26] B. Ma and X. Sun. More Efficient Algorithms for Closest String and Substring Problems. SIAM Journal of Computing, 39(4):1432–1443, 2009.
- [27] N. Misra. personal communication, 2016.
- [28] N. Misra, A. Nabeel, and H. Singh. On the Parameterized Complexity of Minimax Approval Voting. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, pages 97–105, 2015.
- [29] R. Motwani and P. Raghavan. Randomized Algorithms. Cambridge University Press, 1995.
- [30] R. Niedermeier. Lower Bound Issues in Computational Social Choice, 2015. A talk at the workshop Satisfiability Lower Bounds and Tight Results for Parameterized and Exponential-Time Algorithms, Simons Institute, Berkeley, November 10, 2015.
- [31] P. K. Skowron, P. Faliszewski, and J. Lang. Finding a Collective Set of Items: From Proportional Multirepresentation to Group Recommendation. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI 2015, pages 2131–2137, 2015.

Propositionwise Opinion Diffusion with Constraints

Sirin Botan ILLC, University of Amsterdam sbotan@uva.nl Umberto Grandi IRIT, University of Toulouse umberto.grandi@irit.fr Laurent Perrussel IRIT, University of Toulouse laurent.perrussel@irit.fr

ABSTRACT

We study the diffusion of opinions on a social network as an iterated process of aggregating neighbouring opinions. Individual views are modelled as vectors of yes/no answers to a number of propositions connected by an integrity constraint, and each individual updates her opinion by looking at the aggregated opinion of her influencers. We propose and compare two alternative methods for such a process. The first simply ignores the inconsistent aggregated opinion, while the second performs propositionwise revisions whilst maintaining consistency. We characterise the set of integrity constraints that allow individuals to reach the aggregated opinion of their influencers by means of propositionwise updates, and we study under what conditions the termination of the two proposed processes can be guaranteed.

CCS Concepts

•Computing methodologies \rightarrow Multi-agent systems;

Keywords

Social networks, judgment aggregation, opinion transformation, axiomatic method, strategy-proofness

1. INTRODUCTION

When faced with the opinions of others over multiple issues, people will often be influenced to change their own opinions in line with the opinions of their influencers. Social influence describes the effect one person's opinion can have on the opinions of those around her, and in formal models of social influence, is often represented by means of an influence or trust network. Taking inspiration from game-theoretic models of social influence on networks [15, 14], a number of diffusion methods have been proposed for complex representations of individual opinions, such as preferences [3], beliefs [29], or judgments [17]. A common characteristic of all these settings is that changes in agents opinions are driven by the aggregate opinions of their influencers in the trust network. Thus, each agent has some initial opinion that might change if the agent is connected to others in the network who disagree with her on one or more issues.

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

In complex settings, when individual opinions are formed on multiple interconnected issues, a diffusion process needs to carefully account for the constraints that relate the issues at stake. Recent work in the diffusion of beliefs [12] realised the importance of considering integrity constraints on the set of opinions held by individuals. One of the central problems in this setting is the design of individual updates when there are certain dependencies between issues, and aggregating opinions might not always result in a rational opinion. Consider a voter in the U.S. that will vote for Donald Trump unless at least one of the following three conditions are true: (a) Trump will actually build a wall separating the U.S. from Mexico, (b) Trump was supported by Russia in his campaign, and (c) Trump will continue to manage his company during his presidency. Such a voter may be faced with a third of her influencers believing only (a), a different third believing only (b), and the remaining third believing only (c). Hence, the voter can safely go ahead and vote for Trump, since for each condition there is a two-third majority of her influencers that is of the opinion that the statement is false. However, none of her influencers vote for Trump, since they all believe in at least one of the three conditions. How should such an agent update her opinion, and what kind of opinion updates might lead the agent to change opinion on her vote?

In this paper we represent opinions as binary vectors, building on work by Grandi et al. [17], generalizing this setting to take into account possible correlations among the issues at stake. We represent such correlations by means of an integrity constraint, which prevents agents from holding certain opinions over the set of issues. The addition of the constraint to the framework is problematic in some cases, as it prevents certain changes in opinion.

Related work.

Diffusion on networks has been extensively studied in the field of social network analysis, be it diffusion of diseases, information, or opinions [20, 8]. Some of these models are developed to deal with the diffusion of individual opinions, in which individual views are updated by averaging the views of neighboring individuals. Two classical such examples are threshold models [18], with its more recent generalisations [22, 23], and the De Groot or Lehrer-Wagner model [5, 25], which are however based on a simple representation of opinions as a binary view on a single issue, or a real-valued view in the interval [0, 1]. Of particular interest is the recent work of Friedkin et al.[12], which studies the propagation of real-valued beliefs over multiple issues interconnected by logical

integrity constraints.

Building on this literature, a recent stream of papers have adapted averaging models to more complex and realistic representations of opinions: knowledge bases [29, 30], preferences over alternatives [13, 3], and binary evaluations [17]. The latter paper used techniques from judgment aggregation (see, e.g., [19, 10]) and binary aggregation [16], representing opinions as vectors of binary views on *uncorrelated* issues, obtaining a diffusion model assimilable to that of threshold models. We build on this latter paper by adding a constraint connecting the multiple issues at stake, tackling the non-trivial problem of updating individual opinions towards one that does not satisfy the constraint.

We mention in passing that a logical perspective on diffusion in social networks has been explored in a number of papers (see, e.g., [2, 4, 31]). The strategic aspects of diffusion have been studied in the related setting of product adoption [32, 1], which however mostly focuses on threshold models on uncorrelated products. Finally, our axiomatic analysis draws inspiration from the work of Miller and Osherson [27], Knight and Johnson [24], and Dryzek and List [7], who use the notion of inter-agent communication as a mean of reconciling ideas from deliberative democracy with those from social choice.

Paper overview.

The paper is organised as follows. In Section 2 we define two mechanisms for opinion diffusion with constraints on a network. In Section 3 we examine the characteristics of the opinion profiles which result from each of the two mechanisms and in Section 4 we study the termination of the iterative diffusion processes we defined. In Section 5 we study strategic aspects of the diffusion processes, and in Section 6 we propose an axiomatic study of diffusion as a judgment transformation function. Section 7 concludes the paper and points at directions for future work.

2. THE GENERAL FRAMEWORK

In this section we present two models of opinion diffusion in the presence of an integrity constraint. The first is a straightforward generalization of the process of propositional opinion diffusion [17]. The second is instead based on updates to single issues. We represent agents' opinions as answers to a set of yes/no questions which are possibly connected by means of an integrity constraint. We model the social influence network as a directed graph.

2.1 Individual Opinions

Let $\mathcal{I} = \{p_1, \ldots, p_m\}$ be a finite set of *m* issues (or propositions), where each issue represents a binary choice. We call $\mathcal{D} = \{0, 1\}^{\mathcal{I}}$ the domain associated with this set of issues. For a finite set of agents, $\mathcal{N} = \{1, \ldots, n\}$, we say $B_i \in \mathcal{D}$ is the opinion of agent $i \in \mathcal{N}$ over all issues in \mathcal{I} . A vector $\mathbf{B} \in \mathcal{D}^{\mathcal{N}}$ of all opinions of agents in \mathcal{N} is a profile. Each opinion B represents an agent's acceptance/rejection of each of the issues in \mathcal{I} . For example, if our set of issues is $\mathcal{I} = \{p, q, r\}$, then B = (110) is the opinion which accepts the first two issues p and q and rejects the third issue r. We call flip(B, p) the opinion B. In our example above where B = (110), flip(B, p) = (010).

We write $B_i(p)$ to mean agent *i*'s judgment on $p \in \mathcal{I}$ in the profile **B**. Thus if B = (110), then B(p) = B(q) = 1

and B(r) = 0. We write $\mathbf{B} =_{-i} \mathbf{B'}$ to mean two profiles \mathbf{B} and $\mathbf{B'}$ are identical if we ignore agent *i*'s opinion.

An integrity constraint IC $\subseteq \mathcal{D}$ defines a domain of feasible opinions. For instance, if we have three issues, p, q and r, and each agent can only accept at most two of the three, then IC = {(110), (011), (101), (100), (010), (001), (000)}. Integrity constraints are often represented compactly by means of a formula of propositional logic, such as $(\neg p \lor \neg q \lor \neg r)$ for the previous example. Issues of succinctness and computational complexity are out of the scope of this paper, hence we assume a set-theoretic representation of feasible opinions. For each agent i, we assume that $B_i \in IC$, meaning each individual opinion must satisfy the given rationality constraint.

2.2 The Social Influence Process

We assume that agents are connected by a social influence network $G = (\mathcal{N}, E)$ where $(i, j) \in E$ means agent *i* influences agent *j* and $Inf(i)_G = \{j \in \mathcal{N} \mid (j, i) \in E\}$ is the set of influencers of agent *i* in the network *G*. Observe that we do not make any assumption on whether $i \in Inf(i)$, defining the framework in full generality.

The first update procedure we propose is a direct generalisation of the propositional opinion diffusion (F-POD) proposed by Grandi et al. [17], in which agents simply aggregate the opinions of their influencers using some judgment aggregation functions $F = (F_1, \ldots, F_n)$ —where $F_i : \mathrm{IC}^{Inf(i)} \to \mathcal{D}$ is the rule agent *i* uses—and copy this aggregate opinion only if it satisfies the integrity constraint IC. Each step of the process is done according to the function POD. This function takes as input a network *G*, a profile of opinions $\boldsymbol{B} \in \mathcal{D}^{\mathcal{N}}$ and an agent $i \in \mathcal{N}$. Note that we assume F_i is resolute, meaning the function outputs a single opinion, and we do not require the outcome of F_i to be a model of the integrity constraint. The POD function returns the updated opinion of *i* according to her aggregation rule:

$$POD(G, \boldsymbol{B}, i) = \begin{cases} F_i(\boldsymbol{B}_{Inf(i)}) & \text{if } F_i(\boldsymbol{B}_{Inf(i)}) \in IC \\ B_i & \text{otherwise.} \end{cases}$$

We propose an alternative update, propositionwise opinion diffusion (F-PWOD), in which each agent updates on one issue at the time, provided that the updated opinion is consistent with the constraint. The function takes as additional argument the issue $p \in \mathcal{I}$ which agent *i* updates.

$$PWOD(G, \boldsymbol{B}, i, p) = \begin{cases} flip(B_i, p) & \text{if } F_i(\boldsymbol{B}_{Inf(i)})(p) \neq B_i(p) \\ & \text{and } flip(B_i, p) \in IC \\ B_i & \text{otherwise.} \end{cases}$$

The two processes can result in two different updates, as the following example shows:

EXAMPLE 1. Three agents A, B, C, are voting on multiple refenda. They need to give opinions on three proposals; more parks in the city center, a homeless shelter and road repairs. Because of budget constraints, they can approve at most two of the proposals. Suppose the individuals are connected in the following social influence network, where the initial profile is $\mathbf{B} = (110, 011, 101)$, meaning A wants the parks and a homeless shelter, B wants the homeless shelter and road repairs.



Assume for each agent i, that F_i is the strict majority rule, accepting an issue only if a strict majority of the individuals accept it. If all agents update using the F-POD function, the resulting profile after one update will be $\mathbf{B'} =$ (110,011,011). If each agent uses the F-PWOD function instead, and we suppose they all update on the first issue, we get a different outcome after the first iteration— $\mathbf{B'} =$ (110,011,001).

2.3 The Iterative Process

A permissible transformation associates a profile of individual opinions with one of the possible outcomes of either a *F*-POD or an *F*-PWOD update:

DEFINITION 1. Let a network G, an integrity constraint IC, and a set of aggregators F_i for $i \in \mathcal{N}$ be given. We say there is a permissible F-POD transformation from profile **B** to profile **B'** if there exists $I \subseteq \mathcal{N}$ such that $B'_i = \text{POD}(G, \mathbf{B}, i)$ for all $i \in I$, and $B'_j = B_j$ for all $j \notin I$. Analogously, there is a permissible F-PWOD transformation from **B** to **B'** if there exists $I \subseteq \mathcal{N}$ and $p_i \in \mathcal{I}$ for each $i \in I$ such that $B'_i = \text{PWOD}(G, \mathbf{B}, i, p)$, and $B'_j = B_j$ for all $j \notin I$.

We say that a permissible transformation is *effective* if there is some $i \in \mathcal{N}$ such that $B_i \neq B'_i$. We further say that a profile **B** is a *termination profile* if no *effective* transformation exists. An agent's opinion B_i is called *stable* on a network G (wrt. F) in profile **B** if for any p, F-PWOD $(G, \mathbf{B}, i, p) = B_i$. Thus, a termination profile is a profile in which all individual opinions are stable.

Both *F*-POD and *F*-PWOD update functions can be used to define diffusion processes with discrete time. Let $turn : \mathbb{N} \to 2^{\mathcal{N}}$ be a turn function, indicating at each point in time *t* what are the agents that are updating their opinions. Let $\mathbf{B}^t = (B_1^t, \ldots, B_n^t)$ be the profile of opinions at time *t*. At time t + 1, all and only the agents in turn(t + 1)will perform an *F*-POD update aggregating the opinions of their influencers:

$$B_i^{t+1} = \text{POD}(G, \boldsymbol{B}^t, i).$$

An additional ingredient is required for F-PWOD. Let $prop_i : \mathbb{N} \to \mathcal{I}$ for each $i \in \mathcal{N}$ be a function which tells us which proposition agent i is allowed to update at any time t. At each time t all agents in turn(t) update their opinion according to the aggregated opinion of their influencers at time t - 1 on issues $prop_i(t)$:

$$B_i^{t+1} = \text{PWOD}(G, \boldsymbol{B}^t, i, prop_i(t+1))$$

for all $i \in turn(t + 1)$. Following Definition 1, there is a permissible transformation between each step of the diffusion process, i.e., each pair of profiles \boldsymbol{B}^t and \boldsymbol{B}^{t+1} . Observe that if \boldsymbol{B} is a termination profile and at time T we have that $\boldsymbol{B}^T = \boldsymbol{B}$ then it is the case that $\boldsymbol{B}^t = \boldsymbol{B}^T$ for all $t \geq T$.

3. TERMINATION PROFILES

In this section we focus on the properties of termination profiles and under which conditions F-POD and F-PWOD result in the same termination profiles. We characterise the set of integrity constraints for which F-PWOD termination profiles agree with the outcome of the respective aggregation functions, and we show that for both F-POD and F-PWOD, an agent's opinion at termination may be very distant from the opinions of her influencers.

3.1 Integrity Constraints with Open Structure

Given two opinions B and $B' \in \mathcal{D}$, recall that the Hamming distance between them is $H(B, B') = \sum_{p \in \mathcal{I}} |B(p) - B'(p)|$.

DEFINITION 2. An integrity constraint IC has an open structure if for any two opinions $B, B' \in IC$ where H(B, B') = k, there is some sequence of distinct opinions $B_1, \ldots, B_{k+1} - all$ in IC—such that $B_1 = B$, $B_{k+1} = B'$, and $H(B_i, B_{i+1}) = 1$ for all $1 \le i \le k$.

To visualise the idea underlining the above definition, we model the opinions on a hypercube. An edge between any two nodes means the Hamming distance between them is 1. An integrity constraint has an open structure if any two nodes at distance k are connected by a path of length exactly k.

EXAMPLE 2. We represent all opinions in the graph below, connecting only those that satisfy IC with a continuous edge. Let IC = {(000), (001), (010), (010), (011), (111)}. This integrity constraint does not have an open structure, and this can be visualised on the figure: the shortest path available between (100) and (111) is of length 4, which is strictly greater than the Hamming distance between the two models H(100, 111) = 2.



An important class of integrity constraints that has an open structure is the one used to represent preferences as linear orders over a set of alternatives (see, e.g., [28]). Let us see this example in details. Let A be a set of alternatives, a linear order is an irreflexive, transitive and complete binary relation over A. A linear order \succ can be represented as a binary evaluation over a set of issues $\mathcal{I}_A = \{p_{ab} \mid (a, b) \in$ $A \times A$ and $a \neq b\}$, such that $B(p_{ab}) = 1$ if and only if $a \succ b$. For each pair (a, b), we only include one of p_{ab} and p_{ba} in the issues as rejecting p_{ab} in a linear order is equivalent to accepting p_{ba} and vice versa. The integrity constraint IC \succ therefore contains all opinions over \mathcal{I}_A corresponding to linear orders over A.¹ We can now prove the following:

PROPOSITION 1. IC \succ has an open structure.

¹Representing preferences with binary evaluations is an idea that can be traced back to the work of Wilson [33].

PROOF. Let *B* and *B'* be two distinct opinions in IC_> such that H(B, B') = k, and let \prec and \prec' be the corresponding linear orders. Since the two orders \prec and \prec' are different, they also differ on pair which is adjacent in one of them, i.e., there exists a pair *ab* such that $B(p_{ab}) \neq B'(p_{ab})$ and there is no $c \in A$ such that $a \succ_i c \succ_i b$ or $b \succ_i c \succ_i a$.² Swapping an adjacent pair in a linear order results in a binary relation that still is a linear order, hence $flip(B, p_{ab}) \in IC_{\succ}$. By repeating updates on adjacent pairs we can therefore build a sequence of propositionwise updates of length *k* from *B* into *B'*.

3.2 *F*-Consistent Termination

We now give a formal definition that we will use to characterise integrity constraints on which the outcome of the propositionwise diffusion process matches the outcome of the rule F, if this outcome satisfies integrity constraint.

DEFINITION 3. An opinion diffusion process is said to be F-consistent on a network G if for all termination profiles **B** it is the case that for any $i \in \mathcal{N}$: if $F(\mathbf{B}_{Inf(i)}) \in IC$, then $B_i = F(\mathbf{B}_{Inf(i)})$.

Clearly F-POD is F-consistent. We show that the same holds for F-PWOD if and only if IC has an open structure.

PROPOSITION 2. F-PWOD is F-consistent if and only if IC has an open structure.

PROOF. For the right to left direction, we first assume that IC has an open structure. Suppose further F-PWOD terminates on a profile \boldsymbol{B} and $F(\boldsymbol{B}_{Inf(i)}) \in \text{IC}$, and suppose for contradiction that F-PWOD is not F-consistent. That is, there is some agent $i \in \mathcal{N}$ such that $B'_i \neq F_i(\boldsymbol{B}_{Inf(i)})$. If $F_i(\boldsymbol{B}_{Inf(i)}) \in \text{IC}$, then since IC has an open structure, there must be some $p \in \mathcal{I}$ s.t. $F_i(\boldsymbol{B}_{Inf(i)})(p) \neq B_i(p)$ and $flip(B_i, p) \in \text{IC}$. By Definition 1, this implies the existence of a permissible and effective transformation from \boldsymbol{B} to a second profile \boldsymbol{B}' by having *i* updating on *p*, against the assumption that \boldsymbol{B} is a termination profile.

For the left to right direction, suppose that IC does not have an open structure. Then it must be the case that there are two opinions B, B' such that H(B, B') = k and all paths of opinions in IC connecting them has length at least k + 1. By the pigeonhole principle, this implies the existence of two distinct opinions B'' and B''', possibly equal to B and B', such that there is no $p \in \mathcal{I}$ where $B''(p) \neq B'''(p)$ and $flip(B'', p) \in IC$. Let now $\mathcal{N} = \{1, 2\}, E = \{(1, 2)\}, \text{ and}$ B = (B''', B''). Observe that B is a termination profile, since $F(\mathbf{B}_{Inf(2)}) = B'''$ and by construction there is no p such that $B''(p) \neq B'''(p)$ and $flip(B'', p) \in IC$. But $B'' \neq B'''$ and therefore F-PWOD is not F-consistent. \Box

Proposition 2 shows that if aggregating an agent's influencers using F gives an opinion in the set IC, F-PWOD will eventually reach a state where each agent's opinion is equivalent to the outcome of F. We can in fact make a stronger claim if we know an agent's sources have stable opinions:

PROPOSITION 3. Let $i \in \mathcal{N}$ and \mathbf{B} be a profile on G. If all $j \in Inf(i)$ have stable opinions in \mathbf{B} and IC has an open structure, then for any F-PWOD termination profile $\mathbf{B''}$ and F-POD termination profile $\mathbf{B'}$, both resulting from \mathbf{B} , we have that $H(B''_i, F(\mathbf{B''}_{Inf(i)})) \leq H(B'_i, F(\mathbf{B'}_{Inf(i)}))$. PROOF. If all agents in Inf(i) have stable opinions on G, then $F(\mathbf{B''}_{Inf(i)}) = F(\mathbf{B'}_{Inf(i)}) = F(\mathbf{B}_{Inf(i)})$. Suppose $F(\mathbf{B}_{Inf(i)}) \in \text{IC}$. By Proposition 2, F-PWOD ensures that at termination $B''_i = F(\mathbf{B}_{Inf(i)})$, and the same will hold for F-POD. Suppose $F(\mathbf{B}_{Inf(i)}) \notin \text{IC}$. Then $B'_i = B_i$. If F-PWOD is not able to perform any updates, $B''_i = B_i$ as well, but if even one update is performed, $H(F(\mathbf{B''}_{Inf(i)}), B''_i) < H(F(\mathbf{B'}_{Inf(i)}), B'_i)$. \Box

Although the assumption of IC having an open structure guarantees that F-PWOD will be able to make at least as many updates as F-POD when faced with an outcome which does not satisfy the constraint, this might in some cases simply mean that neither F-PWOD nor F-POD will be able to update. In the worst case, this means that an agent will end up with an opinion that is very distant from the opinions of her influencers.

PROPOSITION 4. For any number of issues m, there is always some IC with open structure such that we can construct a network G and a F-PWOD termination profile **B** where an agent i is at distance m - 2 from $F(\mathbf{B}_{Inf(i)})$.

PROOF. Let $\mathcal{I} = \{p_1, \ldots, p_m\}$ and $\mathrm{IC} = (\neg p_1 \land \neg p_m) \rightarrow (p_2 \land, \ldots, \land p_{m-1})$, i.e. IC allows all opinions except those which reject the first and last issue and at least one other issue. We slightly abuse our notation and say that B_0 is the opinion which rejects all issues, B_k only accepts the kth issue, and B_* rejects the first and last issue, but accepts all others.

Let F_i be the strict majority rule. We first show that we can construct a network and termination profile such that there is an agent who is at distance m-2 from the outcome of F_i over her influencers. Take the following network G and profile **B**:



Here $F_i(\boldsymbol{B}_{Inf(i)}) = B_0 \not\models \text{IC}$. However, for any issue on which agent *i* does not agree with the majority, namely p_2 to p_{m-1} , she cannot update her opinion without ending up in one of the opinions prohibited by IC.

We now show such an IC must have an open structure. Let $B, B' \models$ IC. Suppose both B and B' accept the first (last) issue. Then since all opinions accepting the first (last) issue satisfy IC, we can freely move between the two by performing updates to single propositions. If both B and B' reject both the first and the last issue, then $B = B' = B_*$ as this is the only opinion which satisfies IC.

Thus, we only need to check if there is a required sequence of opinions between B and B' if they disagree on either the first issue or the last. W.l.o.g., suppose they disagree on the first issue and B rejects the first issue and B' accepts it. Then B can update on the first issue before performing any other updates, as $flip(B, p_1)$ satisfies IC. Now since, $flip(B, p_1)$ and B' both accept the first issue, there must be a sequence of opinions from $flip(B, p_1)$ to B' where the distance between any two successive opinions is 1 and each satisfies IC. Further, since $H(B, flip(B, p_1)) = 1$, we can conclude that the constructed sequence has length exactly H(B, B') + 1. \Box

²This result is folklore, a formal proof can be found in [9].

Note that in the construction in Proposition 4, *F*-POD would result in the same termination profile.

4. TERMINATION OF ITERATIVE OPINION DIFFUSION

In this section we compare the two proposed diffusion models with respect to the termination of the associated iterative process. We first need to introduce a number of definitions.

Recall that by fixing a *turn* function and functions $prop_i$ for $i \in \mathcal{N}$, deciding which agents are updating and on which issues, we can define iterative processes associated to F-POD and F-PWOD. The following definitions are straightforward adaptations of those proposed by Brill et al. [3]. We call an iterative process asynchronous if |turn(t)| = 1 for all $t \in \mathbb{N}$, and synchronous if $turn(t) = \mathcal{N}$ for all $t \in \mathbb{N}$. We say that the iterative process F-POD or F-PWOD universally terminate on a class of graphs \mathcal{E} if for all $G \in \mathcal{E}$ and each initial opinion profile \boldsymbol{B} there does not exists an infinite sequence of effective transformations starting from B. We say that F-POD or F-PWOD asymptotically terminate on a class of graphs \mathcal{E} if for all $E \in \mathcal{E}$ and profiles **B** the following condition holds: from all profiles B' reachable from \boldsymbol{B} there exists a path of permissible transformations leading to a termination profile. When both the turn and $prop_i$ functions select an agent and an issue uniformly at random, asymptotic termination implies that the probability of eventually reaching a termination state tends to 1 as t goes to infinity. Finally, a consensual termination profile is a profile **B** such that for all $i, j \in \mathcal{N}$ we have that $B_i = B_j$.

Aggregation functions F_i are typically classified by means of axiomatic properties. A full-blown analysis of the influence of these properties on termination is out of the scope of this paper, but we still need one such definition. We say that an aggregator F_i is *unanimous* for agent *i* if, whenever $B_j = B^*$ for all $j \neq i$ then $F_i(\mathbf{B}) = B^*$. In words, whenever all influences (excluding the updating agent) are unanimous, *F* updates according to the influencers.

4.1 Simple cycles

A simple cycle is a finite network E such that every agent has exactly one outgoing edge and exactly one incoming edge.

PROPOSITION 5. If G is a simple cycle and F_i are unanimous, then asynchronous F-POD terminates asymptotically to a consensual termination profile.

PROOF. Let \mathbf{B}^0 be a profile on the simple cycle G, where $E = \{(1, 2), \ldots, (i, i+1) \ldots, (n, 1)\}$. Let $i^* \in \mathcal{N}$ be such that $B_{i^*}^0 \neq B_{i^*+1}^0$. If such an agent does not exists then the profile \mathbf{B} is already a consensual termination profile. Let us now define the following turn function. Let $turn(t) = i^* + t + 1$, for $t = 0, \ldots, n-1$. Since $B_{i^*}^0$ satisfy the integrity constraint by assumption, and all F_i are unanimous aggregators, then at each iteration step t agent $i^* + t + 1$ will copy the opinion of agent $i^* + t$, obtaining a consensual termination profile at t = n - 1 in which all agents have the same opinion $B_{i^*}^0$.

The same result holds for F-PWOD, albeit under additional assumptions on the integrity constraint:

PROPOSITION 6. If G is a simple cycle, F is unanimous, and IC has an open structure, then asynchronous F-PWOD terminates asymptotically to a consensual termination profile. The same holds for synchronous F-PWOD if $|\mathcal{I}| \geq 2$. PROOF SKETCH. Let \mathbf{B}^0 be a profile on G, and let $i^* \in \mathcal{N}$ be such that $B_{i^*}^0 \neq B_{i^*+1}^0$. Since IC has an open structure, there is a sequence of propositionwise updates of length $k = H(B_{i^*}^0, B_{i^*+1}^0)$ that transforms the latter opinion into the former. By defining $turn(t) = i^* + 1$ for $t = 0, \ldots, k$, and prop_i according to the sequence above, we obtain a resulting profile \mathbf{B}^k such that $B_{i^*+1}^k = B_{i^*}^0$ and $B_j^k = B_j^0$ for all $j \neq i^* + 1$. The process can then be repeated for $i^* + 2$, and sequentially until reaching again agent i^* , to obtain a consensual termination profile in which all agents have the same opinion $B_{i^*}^0$.

The proof for synchronous F-PWOD uses the same construction as above, setting the $prop_i$ functions to update on irrelevant issues for the non updating agents. \Box

Observe that the set of termination profiles that can be reached starting from the same profile of initial opinions can be different depending on whether we are using F-POD or F-PWOD. In particular, while the former leads to profiles that are consensual on opinions that are already present in the initial profile, the second can result in consensual profiles on opinions that are a combination of the initial ones.

4.2 Directed acyclic graphs

A directed acyclic graph (DAG) is a directed graph that contains no cycle involving two or more vertices. A simple argument of propagation allows us to prove the following:

PROPOSITION 7. If G is a DAG, then both synchronous and asynchronous F-POD and F-PWOD converge universally.

PROOF SKETCH.. We define potential functions h_i for each node i, as follows: $h_i(t) = H(B_i^t, F_i(\boldsymbol{B}_{Inf(i)}))$, measuring the distance between an individual's opinion and the aggregated opinion of its influencers. Each effective transformation under both F-POD and F-PWOD decreases one such function, the one of the updating agent, and possibly increases others, those of the agents influenced by the one updating. By ordering such potential functions based on the distance from a node to a source, which is possible given the assumption that G is a DAG, we obtain a lexicographic ordering of all functions h_i that decreases strictly with each effective transformation. Therefore, for any set of aggregators F_i and any DAG it is impossible to build an infinite sequence of F-POD or F-PWOD effective transformations. \Box

4.3 Complete graphs

Let a complete graph be such that $E = \mathcal{N} \times \mathcal{N}$. Observe in particular that this means $i \in Inf(i)$ for each $i \in \mathcal{N}$. Using an idea from Farnoud et al. [11], we are able to show the following:

PROPOSITION 8. If G is the complete graph, then both synchronous and asynchronous F-POD and F-PWOD converge universally.

PROOF. On a complete graph the set of influencers $Inf(i) = \mathcal{N}$ for all *i*. Let therefore $h(t) = \sum_i H(B_i, F(\mathbf{B}))$ be a potential function that measures the overall distance of the individual opinions from the overall aggregated one. Every effective transformation for both *F*-POD and *F*-PWOD decreases the value of *h*, hence obtaining the desired result. \Box

A general result on the asymptotic convergence of F-POD or F-PWOD is an open problem. A proof similar to the

one used by [3] could be adapted to show that F-PWOD asymptotically converges on any graph, provided that at any point in time the aggregated opinion of any set of influencers satisfy the integrity constraint. This assumption seems however too restrictive for diffusion processes that are designed to deal with integrity constraints. Universal convergence cannot be guaranteed even on simple cycles, for both F-POD and F-PWOD, at least when more than two issues are present. To see this it is sufficient to consider a simple cycle with only one agent having opinion 11 and all others 00, and devise *turn* and *prop_i* functions that make the 11 opinion turn in the cycle whilst keeping all other opinions at 00.

4.4 Update Order Dependence

When updating on single propositions at a time, even with all agents updating synchronously, the order in which each agent updates their opinions matters in determining what the possible termination profiles look like. Consider for instance the following example:

EXAMPLE 3. Let a network and a profile of opinions be as in the figure below and let $IC = \mathcal{D} \setminus \{(111)\}.$



Two agents with the same initial opinion can have the same set of influencers yet end up with different opinions in a termination profile, depending on the order in which they update their opinions on the issues. We can see this with agents D and E who have the same initial opinion. In our case if D updates the issues in the order p, q, r, obtaining 110, and E in the order r, q, p, obtaining 011, these will be their opinions in the termination profile.

Similar situations occur when an integrity constraint blocks the update on a certain set of issues, even though the result of the majority rule is consistent. This does not happen with IC with open structure, as can be shown in the following proposition. Recall from Section 2 that by fixing a function $prop_i$ for each individual i we obtain an iterative diffusion process. We say that the $prop_i$ functions are *balanced* if all profiles at which the iterative process stabilizes, i.e., when there is a T such that $\mathbf{B}^t = \mathbf{B}^T$ for all $t \geq T$, then \mathbf{B}^T is a termination profile.

PROPOSITION 9. If IC has an open structure, that IC is guaranteed to be satisfied by the outcome of F, and that no ties will occur at any iteration step, then on any directed acyclic graph any choice of balanced prop_i functions results in the same termination profile as F-POD.

PROOF SKETCH. By Proposition 2, if IC has an open structure and the outcome of F is guaranteed to satisfy the integrity constraint, then the process will converge to the result of aggregating the influencers' opinions via F. By Proposition 7 we know that the iterative process on DAGs converge, and by a simple algorithm of propagation from the sources we can also show that the iterative F-PWOD process stabilises on a termination profile that is uniquely determined by the initial profile B^0 . \Box Proposition 9 does not generalize to arbitrary network containing cycles. Consider the following example:

EXAMPLE 4. Let G be as depicted in the figure below, and let there be no integrity constraint, i.e., IC = D.

$$A:11 \longleftrightarrow B:00$$

Suppose in the first round of updates, agent A updates on the first issue, and B on the second issue. In the second round, A updates on the second issue, and B on the first issue. PWOD will then terminate on the unanimous profile (01), (01). However if the agents switch the order of updates (A updates the second issue first, then the first issue, similarly for B), we arrive at the profile (10), (10).

5. STRATEGIC MANIPULATION

In this section we examine the possibility of a strategic agent guiding the outcome of a diffusion process by misreporting her initial opinion. We limit our attention to the source agents when considering possible cases of manipulation, as these are the only agents in the network who are in a sense, sure about their opinion and will only change it for strategic reasons.

We assume agents' preferences are defined by means of the Hamming distance wrt. their initial opinion (this is one of many possible choices, see, e.g., [6]). Each agent i with initial opinion B_i is associated with a weak ordering \succeq_i defined as follows: $B \succeq_i B'$ if and only if $H(B, B_i) \leq H(B', B_i)$, i.e., when the Hamming distance between her truthful opinion and B is less than or equal to the distance between her truthful opinion and the opinion B'. In what follows we provide two examples in which a source agent is able to guide the influence process to obtain a resulting opinion profile where agents influenced by her have opinions closer to hers if compared to the outcome of the diffusion process had she been honest about her opinion.

We begin by showing how *F*-POD can manipulated in presence of an integrity constraint.

EXAMPLE 5. Let there be four agents, and let $\mathcal{D} \setminus IC = \{111\}$, i.e. let (111) be the only forbidden opinion. Let the network and the profile be defined as below:



Suppose F_D is the strict majority rule, resulting in an aggregated result of (111), and no update for agent D. Then agent A will benefit by reporting (010) instead of her truthful opinion above: in the truthful profile D does not update, keeping her opinion which is at distance 3 from A's opinion, while in the second profile D updates to (110), which is at distance 2 from A's truthful opinion (011).

The situation is similar for F-PWOD, except that a potential manipulator needs to know the order of updates on the issues in advance to be sure of the effect of her manipulation.

EXAMPLE 6. Let $IC = \{111, 100, 010, 001, 011, 000\}$. Let the network and the profile be defined as follows:



If agent D updates first her opinion on p_1 then p_2 then p_3 , then her opinion at termination will be (100). If agent A knows this is the order in which D will perform the updates, it is in her interest to report the opinion (011) instead of her truthful opinion (111), as this results in the opinion (011) for agent D at termination. Note however that if agent D updates in a different order, say first p_3 then p_2 then p_3 , then D's opinion at termination is (111) and it is in agent A's interest to keep her initial opinion.

6. TRANSFORMATION FUNCTIONS

A transformation function is one way of representing opinion change among a group of agents. Following the definition of List [26], such a function takes as input a profile of opinions \boldsymbol{B} and outputs a second profile \boldsymbol{B}' , representing the influenced or updated opinions. One example of such a transformation function T is deference to unanimity, where $T_i(\boldsymbol{B})$ – the transformed opinion of agent i – accepts only those issues unanimously accepted by all agents in \boldsymbol{B} . In this section, we adapt this definition to take into account the network relating the individuals, and we adapt the axioms initially proposed by List [26], and subsequently formalized by Grossi and Pigozzi [19], to this setting.

6.1 **Opinion Transformation on a Network**

Given a social influence network and a profile corresponding to the opinions of the agents in the network, a *network-based opinion transformation function* returns a profile which comprises the updated opinions of each agent in the network. Formally (recall that \mathcal{D} is the set of all individual opinions):

$$T: \mathcal{D}^{\mathcal{N}} \times 2^{(\mathcal{N} \times \mathcal{N})} \to \mathcal{D}^{\mathcal{N}}$$

The propositionwise opinion diffusion mechanisms defined in Section 2 can be viewed as network-based opinion transformation functions. Given a set of issues \mathcal{I} , agents \mathcal{N} , an integrity constraints IC $\subseteq \mathcal{D}$, and an influence network $G = (\mathcal{N}, E)$, for any $p \in \mathcal{I}$ we can define a transformation function T where:

$$T_i(\boldsymbol{B}, G) = F$$
-PWOD $(G, \boldsymbol{B}, i, p)$.

With p corresponding to $prop_i(t)$, at any time t of the iterative diffusion process.

6.2 Axioms for Opinion Transformations

In this section we adapt some of the axioms proposed by List [26] to the current setting, and we propose novel network-specific properties. We begin with the following straightforward adaptation of some classical axioms. Note that by $T_{i,p}(\boldsymbol{B}, G)$ we mean the opinion of agent *i* on *p* in the transformed profile.

- **Rationality:** for all networks $G \in \mathcal{G}$, agents $i \in \mathcal{N}$, profiles $\boldsymbol{B} \in \mathrm{IC}^{\mathcal{N}}$ we have that $T_i(\boldsymbol{B}, G) \in \mathrm{IC}^{.3}$
- **Unanimity:** for all networks $G \in \mathcal{G}$ and opinions $B^* \in \mathcal{D}$, if it is the case that $B_i = B^*$ for all agents $i \in \mathcal{N}$, then $T_i(\mathbf{B}, G) = B^*$ for all $i \in \mathcal{N}$.

- **Responsiveness:** for all networks $G \in \mathcal{G}$ and agents $i \in \mathcal{N}$, there exist two profiles $\boldsymbol{B}, \boldsymbol{B'} \in \mathcal{D}^{\mathcal{N}}$ such that $\boldsymbol{B} = -i \boldsymbol{B'}, B_i \neq B'_i$ and $T_i(\boldsymbol{B}, G) \neq T_i(\boldsymbol{B'}, G)$.
- **Independence:** for all networks $G \in \mathcal{G}$, issues $p \in \mathcal{I}$, and pairs of profiles $\boldsymbol{B}, \boldsymbol{B}' \in \mathcal{D}^{\mathcal{N}}$, if it is the case that $B_i(p) = B'_i(p)$ for all $i \in \mathcal{N}$ then $T_{i,p}(\boldsymbol{B}, G) =$ $T_{i,p}(\boldsymbol{B}', G)$ for all $i \in \mathcal{N}$.
- **Monotonicity:** for all networks $G \in \mathcal{G}$, issues $p \in \mathcal{I}$, and pairs of profiles $\boldsymbol{B}, \boldsymbol{B'} \in \mathcal{D}^{\mathcal{N}}$, if for $i, j \in \mathcal{N}$, $\boldsymbol{B} =_{-j} \boldsymbol{B'}, B_j =_{-p} B'_j, B(p) = 0$ and B'(p) = 1, then $T_i(\boldsymbol{B}, G)(p) = 1 \Rightarrow T_i(\boldsymbol{B'}, G)(p) = 1$.

Rationality states that if the input to the transformation function is a profile of rational opinions, then the outcome of the transformation should be a profile of rational opinions. Unanimity states that if every opinion in the input profile is the same, then the function simply outputs this same profile. A transformation function is Responsive if there are two profiles in which only agent i changes her opinion, and her opinion in the outcome is different for the two profiles. Independence states that the opinion an agent has on a proposition p in the outcome of the transformation function depends only on agents' opinions on p in the input profile. Monotonicity requires that for any agent i, if they accepted a proposition p in the outcome of a transformation function T applied to a profile B, then added support to this proposition in a profile B' should imply that p remains accepted by agent i in the outcome of T.

Several of the axioms for transformation function have counterparts in judgment aggregation. For example, the Monotonicity Axiom for network-based transformation functions simply states that the aggregation function each agent uses must satisfy Monotonicity as defined for aggregation functions.

We now give three axioms that are specifically defined for transformations on a social network.

- **Influencer-Unanimity:** for all networks $G \in \mathcal{G}$, opinions $B^* \in \mathcal{D}$, and agents $i \in \mathcal{N}$, if for all agents $j \neq i \in Inf(i)$ we have that $B_i = B^*$ then $T_i(\mathbf{B}, G) = B^*$.
- **Influencer-Independence:** for all networks $G \in \mathcal{G}$, issues $p \in \mathcal{I}$, agents $i \in \mathcal{N}$, and profiles $\boldsymbol{B}, \boldsymbol{B'} \in \mathcal{D}^{\mathcal{N}}$, if it is the case that $B_j(p) = B'_j(p)$ for all $j \in Inf(i)$ then $T_{i,p}(\boldsymbol{B}, G) = T_{i,p}(\boldsymbol{B'}, G)$.
- **Exclusiveness:** for all networks $G \in \mathcal{G}$, agents $i \in \mathcal{N}$, and profiles $\boldsymbol{B}, \boldsymbol{B'} \in \mathcal{D}^{\mathcal{N}}$, if $[\forall j \in Inf(i) \cup \{i\} : B_j = B'_j]$, then $[T_i(\boldsymbol{B}, G) = T_i(\boldsymbol{B'}, G)]$.

Influencer-Unanimity states that if all influencers of an agent submit the same opinion in the input to the transformation function, then the agent submits that same opinion in the output profile. Influencer-Independence states that a transformation function is Independent with respect to the opinions of an agent's influencers. For the complete network where $Inf(i) = \mathcal{N}$ for all agents *i*, Influencer-Independence corresponds to Independence. Finally, a transformation function is Exclusive if an agent's opinion in the output of the transformation function depends only on her own opinion and the opinions of her influencers in the input profile. This means that someone who is not an influencer of an agent at any step in the diffusion.

³Observe that IC is a parameter of this axiom.

6.3 Majority PWOD

The aggregation rule F_{Maj} , which accepts only the issues accepted by a (strict) majority of agents, is the strict majority rule, and is defined such that for any proposition $p, F_{Maj}(\boldsymbol{B})(p) = 1$ if and only if $|N_p^{\boldsymbol{B}}| > \frac{n}{2}$, where $N_p^{\boldsymbol{B}}$ is the set of agents who accept p in the profile B.⁴ Let MAJ-PWOD be the propositionwise opinion diffusion model in which each agent uses the strict majority rule to update, i.e., where $F_i = F_{Maj}$ for all $i \in \mathcal{N}$. By definition, F-POD and F-PWOD for any F satisfy Rationality and Exclusiveness. The same holds for Responsiveness as agents must always take into account their own opinion to ensure that changes can be made on a subset of \mathcal{I} while still satisfying the constraint. Though the majority judgment aggregation rule satisfies Independence and Unanimity, the propositionwise updates lead to a violation of the corresponding axioms for transformation functions.

PROPOSITION 10. The MAJ-PWOD transformation function is rational, unanimous, responsive and monotonic. It does not satisfy independence, influencer-independence, or influencer-unanimity.

PROOF. As noted above, MAJ-PWOD satisfies Rationality and Exclusiveness by definition. Moreover, it is straightforward to observe that Unanimity is also satisfied, as if every agent submits the same ballot B, then any agent iwill agree with her influencers on any proposition p and will never change her opinion.

For Monotonicity, suppose for profiles \boldsymbol{B} and $\boldsymbol{B'}$ that for $i, j \in \mathcal{N}, \boldsymbol{B} =_{-j} \boldsymbol{B'}, B_j =_{-p} B'_j, B_j(p) = 0$ and $B'_j(p) = 1$, and further, that $T_{i,p}(\boldsymbol{B}, G) = 1$. If $j \neq i$ and $j \notin Inf(i)$ we know $T_{i,p}(\boldsymbol{B'}, G) = T_{i,p}(\boldsymbol{B}, G)$ because MAJ-PWOD is Exclusive. If $j \in Inf(i)$, then $T_{i,p}(\boldsymbol{B}, G) = 1$ means there was a majority of acceptances for p among agent i's influencers, or there was a majority of rejections but a change in opinion was blocked by IC. If it is the former, we know that an additional acceptance for p in $\boldsymbol{B'}$ means it remains the case that $T_{i,p}(\boldsymbol{B'}, G) = 1$. If it is the latter, then it must have been the case that $B_i(p) = 1$ and thus $B'_i(p) = T_{i,p}(\boldsymbol{B'}, G) = 1$. Now suppose i = j. If $j \in Inf(i)$, then the only way $T_{i,p}(\boldsymbol{B'}, G) = 0$ is if there is a majority of rejections for p among agent i's influencers, but since no agent but i changes her opinion, this cannot be the case. If $i \notin Inf(i)$ we fall back into the first case we analysed.

We provide a counterexample to show that Influencer-Unanimity fails. Let there be two issues $\mathcal{I} = \{p, q\}$ and suppose IC = $p \rightarrow q$. Let G be the following network and **B** the profile shown in the network below:



Let p be the issue agent i is updating. Then $T_i(\mathbf{B}, G) = B_i$ as an update to p would lead to an opinion which does not satisfy the constraint, falsifying Influencer-Unanimity.

Take now a second profile B' that coincides with the one described above, with the exception that $B'_i = (01)$,

hence such that $\boldsymbol{B} =_{-i} \boldsymbol{B}'$. We have that $T_{i,p}(\boldsymbol{B}, G) = 0$, since the update is blocked by the integrity constraint, while $T_{i,p}(\boldsymbol{B}', G) = 1$, falsifying Independence.

Influencer-Independence also fails, as can be seen in the following example. Take two profiles $\boldsymbol{B}, \boldsymbol{B'} \in \mathcal{D}^{\mathcal{N}}$, and let $B_j(p) = B'_j(p) = 1$ for all $j \in Inf(i)$ for some agent *i*. Suppose $i \notin Inf(i)$ and let $B_i = (10), B'_i = (01)$. Further, let IC = $\{(01), (10)\}$. Then, even if $F_{Maj}(\boldsymbol{B}_{Inf(i)})(p) = F_{Maj}(\boldsymbol{B'}_{Inf(i)})(p) = 1$, we still have that $T_i(\boldsymbol{B}, G) = (10)$ while $T_i(\boldsymbol{B'}, G) = (01)$, contradicting the axiom of Influencer-Independence. \Box

7. CONCLUSIONS AND FUTURE WORK

In this paper we have introduced and studied two models for opinion diffusion on multiple binary issues connected by an integrity constraint. Propositional opinion diffusion F-POD updates on all issues at the same time, provided that the aggregated opinion of one's influencers satisfy the integrity constraint. Propositionwise opinion diffusion F-PWOD, instead, updates on one issue at the time towards the aggregated opinion of the influencers, provided that this single change satisfies the integrity constraint. We have characterised the set of integrity constraints on which F-PWOD coincides with F-POD at termination of the diffusion process, and compared the two processes on the distance between an agent's opinion and the one of her influencers. We have given sufficient conditions for the termination of the iterated diffusion process, and provided initial results on the strategic abilities of source agents in the network. We also adapted axiomatic conditions for profile transformation functions, previously defined in judgment aggregation, to take into account a social network relating the individuals, and used these novel formulations to analyze the majoritarian propositionwise opinion diffusion method.

This paper poses a number of open questions, and suggests fascinating directions for future research. First, obtaining termination results for arbitrary integrity constraints, or characterising the set of constraints that guarantee termination on arbitrary networks, would be a major advancement. Techniques from finite Markov chains may be useful in such proofs (see, e.g., [21]). Second, the interplay between the properties of the aggregators, the structure of the integrity constraints, and the network, need to be investigated further. Third, issues of succinctness and computational complexity should be tackled. Once the integrity constraint is represented as a logical formula, a number of strategic questions related to influence maximisation may become intractable.

⁴Here we take n to be the number of opinions in the input to the aggregation rule and not the total number of agents in the network.

REFERENCES

- K. R. Apt and E. Markakis. Diffusion in social networks with competing products. In *Algorithmic Game Theory*. Springer, 2011.
- [2] A. Baltag, Z. Christoff, J. U. Hansen, and S. Smets. Logical models of informational cascades. *Studies in Logic*, 47:405–432, 2013.
- [3] M. Brill, E. Elkind, U. Endriss, and U. Grandi. Pairwise diffusion of preference rankings in social networks. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016.
- [4] Z. Christoff and J. U. Hansen. A logic for diffusion in social networks. *Journal of Applied Logic*, 13(1):48–77, 2015.
- [5] M. H. DeGroot. Reaching a consensus. Journal of the Americal Statistical Association, 69(345):118–121, 1974.
- [6] F. Dietrich and C. List. Strategy-proof judgment aggregation. *Economics and Philosophy*, 23(3):269–300, 2007.
- [7] J. Dryzek and C. List. Social choice theory and deliberative democracy: a reconciliation. British Journal of Political Science, 2003.
- [8] D. Easley and J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, New York, NY, USA, 2010.
- E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Algorithmic Game Theory*, pages 299–310. Springer, 2009.
- [10] U. Endriss. Judgment aggregation. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 17. Cambridge University Press, 2016.
- [11] F. Farnoud, E. Yaakobi, B. Touri, O. Milenkovic, and J. Bruck. Building consensus via iterative voting. In Proceedings of the 2013 IEEE International Symposium on Information Theory, 2013.
- [12] N. E. Friedkin, A. V. Proskurnikov, R. Tempo, and S. Parsegov. Network science on belief system dynamics under logical constraints. *Science*, 354(6310):321–326, 2016.
- [13] S. Ghosh and F. R. Velázquez-Quesada. Agreeing to agree: Reaching unanimity via preference dynamics based on reliable agents. In *Proceedings of the 14th International Joint Conference on Autonomous Agents* and Multiagent Systems (AAMAS), 2015.
- [14] M. Grabisch and A. Rusinowska. Different approaches to influence based on social networks and simple games. In A. Van Deemen and A. Rusinowska, editors, *Collective Decision Making*, volume 43 of *Theory and Decision Library C*, pages 185–209. Springer, 2010.
- [15] M. Grabisch and A. Rusinowska. A model of influence in a social network. *Theory and Decision*, 69(1):69–96, 2010.
- [16] U. Grandi and U. Endriss. Lifting integrity constraints in binary aggregation. Artificial Intelligence, 199-200:45–66, 2013.

- [17] U. Grandi, E. Lorini, and L. Perrussel. Propositional opinion diffusion. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2015.
- [18] M. Granovetter. Threshold models of collective behavior. American Journal of Sociology, 83(6):1420–1443, 1978.
- [19] D. Grossi and G. Pigozzi. Judgment Aggregation: A Primer. Morgan and Claypool, 2014.
- [20] M. O. Jackson and L. Yariv. Diffusion, strategic interaction, and social structure. In J. Benhabib, A. Bisin, and M. O. Jackson, editors, *Handbook of Social Economics*, volume 1, pages 645–678. North-Holland, 2011.
- [21] J. G. Kemeny and J. L. Snell. Finite Markov Chains. Springer-Verlag, 1976.
- [22] D. Kempe, J. M. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003.
- [23] D. Kempe, J. M. Kleinberg, and E. Tardos. Influential nodes in a diffusion model for social networks. In Proceedings of the 32nd International Colloquium on Automata, Languages and Programming (ICALP-2005), 2005.
- [24] J. Knight and J. Johnson. Aggregation and deliberation: On the possibility of democratic legitimacy. *Political Theory*, 22(2):277–296, 1994.
- [25] K. Lehrer and C. Wagner. Rational Consensus in Science and Society. Springer, 1981.
- [26] C. List. Group communication and the transformation of judgments: An impossibility result. *The Journal of Political Philosophy*, 19(1):1–27, 2011.
- [27] D. Miller. Deliberative democracy and social choice. *Political Studies*, 40(1):54–67, 1992.
- [28] F. Rossi, K. B. Venable, and T. Walsh. A Short Introduction to Preferences: Between Artificial Intelligence and Social Choice. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.
- [29] N. Schwind, K. Inoue, G. Bourgne, S. Konieczny, and P. Marquis. Belief revision games. In *Proceedings of* the 29th AAAI Conference on Artificial Intelligence (AAAI), 2015.
- [30] N. Schwind, K. Inoue, G. Bourgne, S. Konieczny, and P. Marquis. Is promoting beliefs useful to make them accepted in networks of agents? In Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI), 2016.
- [31] J. Seligman, F. Liu, and P. Girard. Facebook and the epistemic logic of friendship. In *Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2013.
- [32] S. Simon and K. R. Apt. Social network games. Journal of Logic and Computation, 2015.
- [33] R. B. Wilson. On the theory of aggregation. Journal
- of Economic Theory, 10(1):89–99, 1975.

Thwarting Vote Buying Through Decoy Ballots

David C. Parkes Paulson School of Engineering and Applied Sciences Harvard University Cambridge, MA 02138 USA parkes@eecs.harvard.edu

Paul Tylkin Paulson School of Engineering and Applied Sciences Harvard University Cambridge, MA 02138 USA ptylkin@g.harvard.edu

Lirong Xia Dept. of Computer Science Rensselaer Polytechnic Institute Troy, NY 12180 USA xial@cs.rpi.edu

ABSTRACT

There is increasing interest in promoting participatory democracy, in particular by allowing voting by mail or internet and through random-sample elections. A pernicious concern, though, is that of *vote buying*, which occurs when a bad actor seeks to buy ballots, paying someone to vote against their own intent. This becomes possible whenever a voter is able to sell evidence of which way she voted. We show how to thwart vote buying through *decoy ballots*, which are not counted but are indistinguishable from real ballots to a buyer. We show that an Election Authority can significantly reduce the power of vote buying through a small number of optimally distributed decoys, and model societal processes by which decoys could be distributed.

CCS Concepts

 $\bullet Computing \ methodologies \rightarrow Multi-agent \ systems;$

Keywords

vote buying, elections, opinion aggregation

1. INTRODUCTION

The goal of participatory democracy [9,11] is to engage citizens more frequently and with more granularity in the decision-making processes of government bodies. Technologies that can help with this transition are those that support voting from the home by mail or over the internet, and that make use of *random sample elections*, in which a representative subsample of the population is tasked with voting on a particular issue, allowing participatory democracy to function without everyone needing to be concerned with every issue.

A pernicious concern, though, is that of *vote buying*, where a bad actor attempts to gain improper influence in an election by purchasing ballots from voters and paying them to vote against their intent. The practical implications of this are manifold, since the social construct of elections relies on the perception of reliability and fairness. Vote buying has been an everlasting threat to democracy; for example, a survey shows that in the 1996 Thai general elections "*one third* of households were offered money to buy votes at the last general election" [13]. Schaffer [14] mentions that "/Vote buying]... is making an impressive comeback...it seems, a blossoming market for votes has emerged as an epiphenomenon of democratization". New technologies can make the situation worse. For example, web platforms can serve as middlemen, digital currency supports anonymous payments, and abundant data coupled with machine learning can help buyers discover entrapment schemes as well as identify voters to target with offers.

In this paper, we show that vote buying can be thwarted by distributing *decoy ballots*, which are not counted, in addition to real ballots. A vote buyer will not know whether a ballot is real or decoy, and thus, decoys (if sold) may deplete a buyer's budget. Voters who know that they have a decoy ballot are motivated to sell their ballots to a buyer, both for reasons of profit and out of civic duty, wanting to maintain the integrity of an election. Decoy ballots have been suggested by Chaum [4], but we are not aware of any analysis of how decoy ballots should be distributed, and how effective they are against vote buying.

We assume that real ballots impose a very high cost on society, for the reason that it takes effort for members of society to become informed about an issue and vote appropriately, thus representing their considered opinion on an issue.¹ Without the willingness to invest this effort, methods of participatory democracy may ultimately fail. For example, a simple calculation for the US shows that if we assume that 200M people will participate, and there are about 12,000 issues to decide per year,² then assuming that voters are willing to engage three times a year, we have a maximum of 50,000 voters per issue. At this scale, vote buying, especially on contentious issues, may pose a severe problem.

Turning to decoy ballots, we model these as costly but not so costly that the number of decoys to distribute cannot be considered as a design decision of the Election Authority. The cost of decoys comes about because, to be effective, voters need to be willing to go to the effort to sell the ballot (and thus, cast the ballot and prove which way it was cast) if approached by a buyer. But because any ballot cast is not ultimately counted, there is less emphasis on a voter needing to research an issue to form an opinion.

Although we situate our discussion in a societal context,

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

¹In some approaches, this cost comes about, in addition, as a result of needing to physically mail ballots [4].

 $^{^2{\}rm This}$ represents the approximate voter population and the number of issues before Congress per year, assuming 2 issues per bill.



Figure 1: Examples of type distribution $f(\theta)$, decoy distribution $\psi(\theta)$, and desirability to buyer $h(\theta)$ for (a) an optimal defense, (b) a civic duty defense with max type requesting a decoy $x_c = 0.5$ and 10% decoy ballots, (c) an auction-based defense with max type assigned a decoy $x_A = 0.5$ and 50% decoy ballots. Here f = Beta(1, 2).

similar themes can be easily imagined for economies of AIs [12], where it is desired to elicit and fairly aggregate multiple opinions, but would not be scalable to request input from every agent all the time.

Our Contributions

We provide a formal model of vote buying, including a characterization of the vote buyer's behavior and an optimal policy for distributing decoy ballots by the Election Authority (EA). In addition, we model two societal processes by which decoys could be distributed—these approaches freeing the EA of any concern that it could be seen to be biasing the outcome of an election when distributing decoys in any way other than reflecting a random sample of the population. In simulation, we show that the EA can make effective use of decoy ballots to maintain election integrity (e.g., reducing the probability that the buyer changes the outcome to less than 1%). For the optimal defense, we are able to achieve this by adding a small number of decoys that are proportional in quantity to the number of ballots the buyer can afford to buy. Interestingly, a "civic duty defense" that allocates decoys to a random subset of those who request one is almost as effective as the optimal defense in which the EA optimizes the distribution of voter types that receive decoys.

Related Work

There are numerous studies on vote buying, for example [8, 15,16,19]. These include game-theoretic models of vote buying, but none that consider the role of decoy ballots. In Dekel et al. [6], the game is played by the candidates themselves buying votes, Groseclose and Snyder [10] study vote buying in legislative bodies and analyze the optimal coalition size. Vicente [18] studies the incumbency advantage in a vote buying game. Within AI, the problem studied here related to studies of control (manipulation of the election structure, including changing the candidate slate) and bribery (voters are paid by an interested party to vote a certain way) as studied in computational social choice [2,7]. In particular, the *lobbying* problem considers an election with a binary outcome on a number of issues, and the vote buyer has a total budget that can be expended across all issues [1,3,5]. Ours is a special case with a single issue, but whereas previous research has focused on using computational complexity as a barrier against bribery and control, we adopt a game-theoretic model and study the power of decoy ballots. There is also a conceptual connection with work on security games [17], where the approach is to use game theory to design optimal strategies to prevent losses

from terrorist attacks.

2. THE MODEL

We assume that there is a large population of possible voters, and that this is a binary choice election with possible votes YES and NO. For expositional simplicity, we assume that all voters who receive a real ballot will place a vote. Similarly, we assume that every voter for whom it is profitable to sell a ballot (decoy or otherwise) will try to sell the ballot.³

The voters. Each voter *i* has an immutable, publiclyobservable voter type, θ_i , which indicates the probability that a random voter with this type will vote YES. We can think about θ_i as the prior that a voter will vote YES before she has carefully considered the merits of an issue. Voter types are drawn IID from a voter type distribution with probability density f, assumed to have full support on [0, 1]. We assume without loss of generality that $E_f[\theta] < 1/2$, i.e., that the outcome of the election without any interference by a buyer and with enough real ballots is NO.

The buyer. We model a single, budget-limited buyer. Given our assumption that $E_f[\theta] < 1/2$, we consider the interesting case of a YES-buyer, meaning that the buyer wants the election outcome to be YES. To keep things simple, we assume the buyer can find the voters with ballots, and will offer the same price p > 0 to some subset of these voters. The buyer has a budget B, representing the number of ballots that he can afford to purchase at price p, and has no utility for unspent budget. The buyer selects a random subset of voters if more respond to the offer than he can afford.

Conditioned on whether a voter's intent is to vote NO or YES, and whether they have a real or decoy ballot, all voters have the same utility function in regard to whether or not to sell. In particular, simple analysis yields that this ordering of the minimum price that a voter will require in order to agree to sell a ballot is real-NO > real-YES > decoy-YES > decoy-NO. For example, any price that is acceptable to a "real-YES" voter (real ballot, intent to vote YES) is also acceptable to "decoy-YES" and "decoy-NO" voters. Ballots from decoy-NO voters are the cheapest to buy.⁴

³It is simple to generalize the model so that the people who actually cast ballots are sampled uniformly from those who receive ballots, and similarly for those who try to sell ballots. ⁴To understand this ordering, consider that a voter with a real ballot has a cost for selling, representing the possibility of being caught. In addition, voters that intend to vote NO prefer not to change their vote and vote YES. Thus, these are the most expensive votes to buy. Analogously,



Figure 2: Comparing the power of different defenses, with f = Beta(2,4), 1000 ballots in total (some real, some decoy), and different buyer budgets *B*. (a) Optimal defense, varying the fraction of real ballots. (b) Civic duty defense, with the EA optimizing the number of decoy ballots to use for each value of parameter x_{c} (the 'max type requesting decoy'). (c) Auction-based defense, with the EA optimizing the number of decoy ballots to use for each value of parameter x_{A} (the 'max type assigned a decoy').

Based on this, the real-NO votes—and the only ones the buyer is interested in—are the most expensive ballots to buy. Because of this, we assume the buyer will set price p high enough for a real-NO voter to agree to sell if approached. This could be set based on market research, for example.

The game form. The voters who receive a real ballot are a random subset of the population, and thus with types that follow f. The choice of how to distribute decoy ballots is, in general, a design decision of the EA. Let ψ denote the density function for this decoy ballot distribution. Modeled as a sequential-move game, we view the election as proceeding in the three stages:

(1) the EA distributes some number of real and decoy ballots, with the number and type distribution of real ballots assumed fixed, but the number of decoy ballots, and perhaps type distribution ψ a design decision.

(2) the buyer learns who has received a ballot (possibly a decoy) and chooses to offer price p to some subset of voters who have (real or decoy) ballots. The voters who receive an offer decide whether or not to sell. The buyer breaks ties at random if multiple voters agree to sell.

(3) Both real and decoy ballots are cast, and the real ballots are tallied to determine the outcome. The buyer makes payments to voters who agreed to sell and provide a proof that they vote YES⁵.

We assume that f and the type of each voter is common knowledge. Our analysis will focus on the subgame perfect equilibrium of this game. Throughout, the voters have a simple equilibrium behavior—agree to sell if offered price p(which will, in equilibrium, be high enough to be acceptable.)

Proof of decoy. We assume the existence of a *proof-of-decoy*, which lets a voter with a decoy prove to anyone that she has a decoy. On the other hand, there is no way to prove the authenticity of a real ballot. This property is easy to support through standard cryptographic primitives; see, for example, Chaum [4].⁶

EA and Buyer objectives. We take as the objective of the EA that of maintaining election integrity, and thus *minimizing the probability that the buyer changes the election outcome.* In contrast, the interests of the buyer are diametrically opposed, and he wants to *maximize the probability that the outcome of the election is changed.*

3. BUYER ANALYSIS

Given the buyer's objective, the best response of the buyer to the EA is to maximize the expected number of real-NO ballots that he buys, given his budget B and knowledge about voters' types (probability of voting YES). Let $\mathcal{I} \subseteq$ [0, 1] denote the subset of voter types from which the buyer buys; in particular, the buyer will buy every ballot held (real or decoy) by voters of these types. Let n_r denote the number of real ballots and n_d the number of decoy ballots. The buyer wants to select the subset \mathcal{I} to solve:

$$\max_{\mathcal{I}} \int_{\mathcal{I}} \frac{n_r}{n_r + n_d} (1 - \theta) f(\theta) d\theta \text{ s.t.} \int_{\mathcal{I}} n_r f(\theta) + n_d \psi(\theta) d\theta \le B.$$

In this way, the buyer maximizes a quantity that is proportional to the expected number of real-NO ballots purchased, subject to the total budget. Let $h(\theta)$ denote the *probability* that a ballot is real-NO given type θ . By Bayes' rule, and recalling that the buyer has knowledge of f and ψ , this is

$$h(\theta) \stackrel{\text{def}}{=} P(\text{real} \wedge \text{NO}|\theta) = \frac{n_r(1-\theta)f(\theta)}{n_r f(\theta) + n_d \psi(\theta)}.$$
 (1)

Given a set $I \subseteq [0,1]$, let h(I) denote the set $\{h(\theta)\}$ for $\theta \in I$. Let $h(I_1) < h(I_2)$ mean that every value in I_1 is strictly less than every value in I_2 .

LEMMA 1 (BUYER OPTIMALITY). The optimal buyer strategy in the subgame perfect equilibrium is to buy in order of decreasing $h(\theta)$ until the budget is exhausted.

Where proofs are omitted, this is because of space. They will be provided in the long version of the paper. We assume

decoy-YES ballots are more expensive to buy than decoy-NO ballots because a voter who would vote NO (if she had a real ballot) has higher value for depleting the budget of a YES-buyer than a voter who would vote YES.

⁵Voters could provide proof of the way that they voted to the buyer by, for example, sending a video of themselves casting the vote or a photograph of their ballot.

⁶The asymmetry in proof-of-decoy but no proof-of-

authenticity is important in preventing a buyer from using coercion to buy only real ballots, while at the same time allowing a voter with a decoy ballot to sell with impunity to accusations of acting against the social good (since she can, if challenged to do so, prove that it is decoy, and thus that she is acting in good faith.) A voter will never choose to reveal that she holds a decoy to a buyer, since doing so would just cause the buyer to refuse to transact with her.

w.l.o.g. that if a YES-buyer has to choose between buying two subsets of [0, 1] for which $h(\theta)$ is equal, he will buy the subset with lower θ . Let $\mathfrak{M} \stackrel{\text{def}}{=} \int_{\mathcal{I}} f(\theta) d\theta$ denote the fraction of real ballots that the buyer buys. By 'election bought', we refer to the event that the buyer buys enough real ballots to change the outcome (with n_r real ballots); by 'correct outcome is NO', we refer to the event that the election outcome is NO (with $n_r + n_d$ real ballots).

LEMMA 2. The probability that the buyer changes the outcome in the subgame perfect equilibrium is given by

$$P(buyer changes outcome) = P([election bought] \land [correct outcome is NO]) \approx P\left(\frac{n_r(1-2\mathfrak{M}-2(1-\mathfrak{M})\mu_Y)}{2\sqrt{n_r(1-\mathfrak{M})\mu_Y(1-\mu_Y)}} < Z < \frac{(1-2\mu)\sqrt{n_r+n_d}}{2\sqrt{\mu(1-\mu)}}\right), \quad (2)$$

where $Z \sim \mathcal{N}(0, 1), \qquad \mu \stackrel{\text{def}}{=} E_f[\theta], \qquad and \qquad \mu_Y \stackrel{\text{def}}{=} \frac{1}{1-\mathfrak{M}} \int_{[0,1] \setminus \mathcal{I}} \theta f(\theta) d\theta.$

This allows us to compute the probability the buyer changes the election outcome, which is determined by the fraction of real ballots that he is able to buy given a defense.

4. OPTIMAL DECOY DISTRIBUTION

In this section, we assume that the EA can design defense distribution ψ , and study the equilibrium of the vote-buying game where the EA chooses an optimal defense given that the buyer will best respond.

DEFINITION 1 (CANONICAL DEFENSE). Defense ψ is canonical if there is some $x, 0 \leq x \leq 1$, s.t. $h(\theta) = \min(1-x, 1-\theta)$.

See Figure 1(a) for an illustration of a canonical defense, and Figure 3 for an illustration of the model. Let supp(g) denote the support of distribution g. Define the following two properties for ψ :

- (P1) $h(\theta)$ has the same value for all $\theta \in \operatorname{supp}(\psi)$.
- (P2) $\min_{\theta \in \operatorname{supp}(\psi)} h(\theta) \ge \max_{\theta \notin \operatorname{supp}(\psi)} h(\theta)$

LEMMA 3. Any defense ψ satisfying both P1 and P2 is canonical.

LEMMA 4. If the buyer buys all ballots in $\operatorname{supp}(\psi)$, then there is a canonical defense ψ' with the same value.

Lemma 3 characterizes canonical defenses in terms of the properties defined above. Lemma 4 shows that if the buyer can buy up all decoys, then how they are distributed no longer matters. Fixing the number of real ballots n_r , the EA's remaining choices are about n_d and ψ . We now state our main characterization result.

THEOREM 1. For a given n_r , n_d , and buyer budget B, the optimal strategy of the EA in the subgame perfect equilibrium is canonical.

PROOF. Assume for contradiction, that there is a noncanonical ψ that is better than any canonical defense. Let k be an index, and consider a sequence of defenses $\{\psi_k\} = \{\psi_0, \psi_1, ...\}$, where $\psi \stackrel{\text{def}}{=} \psi_0$. We will show that we can define a finite sequence that obtains a canonical defense at least as



Figure 3: Real ballots, distributed according to $f(\theta)$, and decoy ballots, distributed according to $\psi(\theta)$, are given to voters by the EA. In an optimal defense, decoys are assigned with higher propensity to the voter types that are a priori more attractive to the buyer, thus forcing the buyer to also deplete his budget on decoys.



Figure 4: Comparing the power of various defenses for f = Beta(2,4), x_{c} and $x_{\text{A}} = 0.5$, and 1000 total ballots.

good as ψ . Let $h_k(\theta)$ denote the function h that corresponds to ψ_k .

Let $\mathcal{I}_k \subseteq [0, 1]$ denote the set of intervals that are best for the buyer given ψ_k (solving for the buyer's objective subject to his budget). If the buyer buys all ballots in $\operatorname{supp}(\psi_k)$, then by Lemma 4, we can modify ψ_k to form a canonical ψ_{k+1} with the same value, and we are done.

Suppose otherwise, and that in addition ψ_k does not satisfy P1 and P2. That is, we have:

- (P0) the buyer does not buy all ballots in $\operatorname{supp}(\psi_k)$, and one or both of
- $(\neg P1) h_k(\theta)$ takes on multiple values for $\theta \in \operatorname{supp}(\psi_k)$
- $(\neg P2) \min_{\theta \in \operatorname{supp}(\psi_k)} h_k(\theta) < \max_{\theta \notin \operatorname{supp}(\psi_k)} h_k(\theta).$

By P0, we can construct some interval $S_k \subseteq \operatorname{supp}(\psi_k)$ (the source set), where the buyer is not buying all ballots, and an interval $T_k \subseteq \mathcal{I}_k$ (the target set), such that $h_k(S_k) < h_k(T_k)$

(and thus, $S_k \cap T_k = \emptyset$). Let $R_k = \operatorname{supp} \psi \setminus \mathcal{I}_k$ be the remaining subset of $\operatorname{supp}(\psi)$ that the buyer is not buying. We must have $\operatorname{arg\,min}_{\theta \in \operatorname{supp}(\psi_k)} h_k(\theta) \subseteq R_k$. The existence of T_k follows from $\neg P1$ because $\exists \theta \in \mathcal{I}_k$ for which $h_k(\theta) >$ $\min_{\theta \in \operatorname{supp}(\psi_k)} h_k(\theta)$ (the existence is guaranteed by values of $\theta \in \operatorname{supp}(\psi_k)$ that are greater than the minimum), and thus we have $\max_{\theta \in \mathcal{I}_k} h_k(\theta) > \min_{\theta \in \operatorname{supp}(\psi_k)} h_k(\theta)$. If $\neg P2$, then by buyer optimality (Lemma 1), $\operatorname{arg\,min}_{\theta \in \operatorname{supp}(\psi_k)} h_k(\theta) \subseteq R_k$. In both cases, $\operatorname{arg\,min}_{\theta \in \operatorname{supp}(\psi_k)} h_k(\theta) \subseteq S_k$.

We pick $\epsilon_S, \epsilon_T > 0$ to define a move of a uniform slice of ψ density from S_k to T_k such that,

- (i) $\int_{\theta \in S_k} \max(0, \psi_k(\theta) \epsilon_S) d\theta = \int_{\theta \in T_k} \epsilon_T d\theta$ [mass conservation]
- (ii) $h_{k+1}(S_k) < h_{k+1}(T_k)$ [target set still preferred by buyer to source set]

By continuity (except possibly on a set of measure 0) of $h(\theta)$, such an ϵ_S, ϵ_T pair that satisfies (ii) exists. We argue that $S_k \cap \mathcal{I}_{k+1} = \emptyset$. Before the ψ mass is moved, we have $\min h_k(\mathcal{I}_k) \geq h_k(T_k) > h_k(S_k)$. After the move, we have $\min h_{k+1}(\mathcal{I}_{k+1}) \geq h_{k+1}(T_k) > h_{k+1}(S_k)$. The inequality is because the buyer can always exhaust his budget by buying \mathcal{I}_k . Thus, we know that the buyer does not buy anything in S_k after the ψ mass has been moved. Let $Q_k \stackrel{\text{def}}{=} \int_{\mathcal{I}_k} (1-\theta)f(\theta)d\theta$. Thus, we have $Q_{k+1} \leq Q_k$ because the only set on which $h_{k+1}(\theta) > h_k(\theta)$ is S_k . In addition, $\min_{\theta \in \text{supp}(\psi_k)} h_k(\theta) < \min_{\theta \in \text{supp}(\psi_{k+1})} h_{k+1}(\theta)$. Because $\forall k \in \mathbb{Z}^+, \ \theta \in [0, 1], \ h_k(\theta) \geq 0$ the sequence must be finite. \Box

Theorem 1 says that for a given n_r and n_d , the optimal design of ψ by the EA is canonical. The next result shows that ψ (and its support, which is $[0, x_0]$, "o" for optimal) can be easily computed given any n_r and n_d .

THEOREM 2. For any given n_r and n_d , the optimal defense of the EA in the subgame perfect equilibrium is given by a decoy ballot distribution with density function

$$\psi(\theta) = \begin{cases} \frac{n_r}{n_d} \frac{(x_o - \theta)f(\theta)}{1 - x_o} & \text{for } \theta \in [0, x_o] \\ 0 & \text{for } \theta \in (x_o, 1] \end{cases},$$
(3)

where the threshold x_0 is determined by the following equation: $\frac{1}{1-x_0} \int_0^{x_0} F(\theta) d\theta = \frac{n_d}{n_r}$ and $F(\theta)$ is the CDF of f.

With this expression, we can determine the power of increasing the number of decoys, n_d , for any voter type distribution f, buyer budget B, and number of real ballots n_r .

5. NEUTRAL APPROACHES

In this section, we consider defenses where the EA does not design ψ , since doing so may be argued as the EA playing too active a role in running the election. Beyond neutrality, these new approaches have the additional advantage of not relying on the EA having knowledge of f.

5.1 A Constrained Defense

We first consider a *constrained defense*:

DEFINITION 2. Defense ψ is constrained if the EA distributes decoy ballots uniformly at random, i.e., $\psi = f$.

Having a constrained defense implies that $h(\theta) = \frac{n_r}{n_r + n_d} (1 - \theta)$ and $\mathcal{I} = [0, \tau_C]$ for some $\tau_C > 0$, such that the budget is spent, i.e., $F(\tau_C) = B/(n_r + n_d)$.

DEFINITION 3 (LOW BUDGET). A low budget is a budget where $\int_{\tau_C}^1 \theta f(\theta) d\theta < \frac{1}{2} - F(\tau_C)$.

DEFINITION 4 (HIGH BUDGET). A high budget is a budget where $\int_{\tau_C}^1 \theta f(\theta) d\theta > \frac{1}{2} - F(\tau_C)$.

In words, for a buyer with a low (high) budget, the expected number of real ballots the buyer buys is lower than (exceeds) the amount needed to change the election outcome.

One way to study the power of a constrained defense is to consider the following question: if the total number of ballots is fixed, what is the optimal mix of real and decoy ballots?

THEOREM 3. Fixing the total number of ballots, the best constrained defense for the EA in the subgame perfect equilibrium is all (one) real ballots for low (high) buyer budget under the Normal approximation (2).

With a low buyer budget, while a constrained defense makes the buyer buy some decoys, it also leaves unpurchased decoys and reduces the number of unpurchased real ballots, decreasing the accuracy of the result. Thus, decoys are not useful for the EA in this case. On the other hand, the best that the EA can do with a buyer with a high budget is to issue a single real ballot, with the hope that the buyer won't buy it, resulting in a high variance outcome based on the vote of a single voter. Decoys are used, but not to good effect.

5.2 Civic Duty Defense

In this model, the EA makes decoy ballots available to a random subset of those voters who make an explicit request for a decoy.⁷ The decision of the EA is thus the number of decoy ballots, but not how to distribute them. Rather, this decision arises through a simple model of a societal process.

In modeling this process, we assume that, for a YESbuyer, there is some distribution of civic-mindedness $\pi(\theta)$, with support on $[0, x_c]$, that determines the probability that a voter will request a decoy, where x_c is a fixed, publicly known quantity ("c" for civic). In particular, we assume for simplicity that $\pi(\theta) \propto x_c - \theta$. This captures the idea that the more extreme an agent's type, the more likely the agent is to request a decoy and thus help preserve the election's integrity.

Via Bayes' rule, the effect on the distribution on types ψ of those who get decoys is $\psi(\theta) = P(\theta|\text{request decoy}) \propto P(\text{request decoy}|\theta)f(\theta) = \pi(\theta) \cdot f(\theta) = (x_c - \theta)f(\theta).$

In fact, there will sometimes be a choice of n_d such that the civic duty defense is optimal. If the EA can choose a number of decoys n_d such that $\frac{n_d(1-x_c)}{n_r} = k$, where k is the normalization constant, then we see the canonical structure, with $h(\theta) = 1 - x_c$, $\forall \theta \in [0, x_c]$. We call the defense obtained via this model a *civic duty defense*. An example of this defense is illustrated in Figure 1(b).

⁷For the purpose of both this model and the next, we assume it is prohibitively costly for a buyer to acquire multiple, credible real-world identities in order to attack these distribution mechanisms.



Figure 5: Using decoys to thwart vote buying, for different buyer budgets (the number of ballots the buyer can buy). The number of real ballots is 750, the voter type distribution is f = Beta(2,4). (a) constrained defense, in which decoy ballots are distributed according to $f(\theta)$; (b) optimal defense; (c) auction-based defense with $x_A = 0.5$; (d) is civic duty defense with $x_C = 0.5$.

5.3 Auction-Based Defense

In this variation, the EA makes decoy ballots available to voters via an auction. We assume a simple n_d +1st price auction (when selling n_d decoy ballots), with the EA choosing n_d . The intent is not to model a sophisticated auction, but to adopt a strategyproof mechanism as a model for an idealized market-based approach for distributing decoy ballots to voters. The effect is that decoys go to voters with the highest value for decoys. As with the civic duty defense, the EA who makes use of an auction-based defense chooses the number of decoy ballots but not how to distribute them.

In modeling this societal process, we assume that the value to a voter for a decoy is monotonically increasing as the voter's type θ gets closer to zero.⁸ For this reason, we model the effect of the auction as being that there is some threshold $x_{\lambda} \in (0, 1)$, whereby the decoys are distributed according to voter type distribution f, conditioned on $\theta \leq x_{\lambda}$ ("A" for auction). In particular, for $\theta \in [0, x_{\lambda}]$, we have $\psi(\theta) \propto f(\theta)$.

6. SIMULATION RESULTS

We describe the results of an extensive simulation study to compare power of various defenses in preventing a buyer succeeding in changing the outcome of an election. We choose to present results for voter type distribution f =Beta(2, 4), but the analysis is qualitatively unchanged for other distributions, including those with mean voting types in [0.01,0.49] (e.g., voter type distribution Beta(9,11), which is quite concentrated around the mean of 0.45). Figure 5 fixes the number of real ballots, and shows that vote buying can be successfully thwarted by issuing sufficiently many decoy ballots. The optimal and civic duty defenses are most effective, but even issuing decoys according to the auction-based and constrained defenses substantially reduces the probability of a vote buyer's success. It is interesting that even a small number of decoys, relative to the number of real ballots, can be effective.

It also helps with understanding to compare the power of different defenses when fixing the total number of ballots, some of which will be real and some decoys, and varying the number of decoy ballots. Figure 2(a) shows the effect of varying the fraction of real ballots when using an optimal defense. Figure 2(b) shows that the effect of the civic duty defense for different values of model parameter $x_{\rm c}$ (the 'max type requesting a decoy'), and with the EA optimizing the number of decoys for each value of $x_{\rm c}$. Figure 2(c) shows the effect of the auction-based defense for different values of model parameter x_{A} (the 'max type winning a decoy'), also with the EA optimizing the number of decoys for each value of $x_{\rm A}$. The auction-based defense is the least effective, but even here there is a range of $x_{\rm A}$ for which the performance is better than without using any decoys. In Figures 2(b) and 2(c), a maximum type of 0 receiving a decoy corresponds to zero decoys. Also fixing the total number of ballots, we examine the relative power of the different defenses as a function of the buyer budget. In Figure 4 (with 1000 total ballots) we see that an optimal defense can use decoys to protect against buyers with around twice the budget of a 'no defense' approach that just uses all real ballots. For the civic-duty and auction-based defenses, we fix $x_{\rm c} = x_{\rm A} = 0.5$ and pick the best n_d at each point in the graph. The auctionbased defense is better than no defense or the constrained defense. The civic-duty defense has very good performance that is almost the same as that of the optimal defense for

⁸We insist, though, that the reasonable property holds that a voter's value for using a decoy is less than her value for a real ballot, and thus this auction-based societal process is consistent with our analysis in Section 2 in regard to the ordering of minimum acceptable offer price from a buyer across different kinds of voters.

many buyer budgets.

7. CONCLUSION

We have presented the first game-theoretic study of the power of decoy ballots in thwarting vote buyers. We have derived a characterization of the form of an optimal defense, and compared its power to those of neutral defenses that could be enabled through leveraging simple societal processes to distribute decoy ballots. Our results are positive: decoy ballots are effective in thwarting the power of a vote buyer. Amongst the neutral defenses, the civic duty defense, where decoys are given at random to a subset of those who request such a ballot, seems especially interesting for future study. Also of interest is to study defenses under the requirement that they must protect equally against a YESor NO-buyer, when there are more than two ballot choices, multiple buyers, simultaneous polls, and participants with value and cost heterogeneity.

REFERENCES

- D. Binkele-Raible, G. Erdélyi, H. Fernau, J. Goldsmith, N. Mattei, and J. Rothe. The Complexity of Probabilistic Lobbying. *Discret. Optim.*, 11:1–21, Feb. 2014.
- [2] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.
- [3] R. Bredereck, J. Chen, S. Hartung, S. Kratsch, R. Niedermeier, O. Suchý, and G. J. Woeginger. A Multivariate Complexity Analysis of Lobbying in Multiple Referenda. J. Artif. Intell. Res. (JAIR), 50:409–446, 2014.
- [4] D. Chaum. Random-Sample Voting. http://rsvoting.org/whitepaper/white_paper.pdf, 2016.
- [5] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, 2007.
- [6] E. Dekel, M. O. Jackson, and A. Wolinsky. Vote Buying: General Elections. *Journal of Political Economy*, 116(2):351–381, 2008.
- [7] P. Faliszewski and J. Rothe. Control and bribery in voting. In *Handbook of Computational Social Choice*, Chapter 7. Cambridge University Press, 2016.
- [8] F. Finan and L. Schechter. Vote-Buying and Reciprocity. *Econometrica*, 80(2):863–881, 2012.
- [9] A. Goel and D. T. Lee. Towards large-scale deliberative decision-making: Small groups and the importance of triads. In *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 287–303, 2016.
- [10] T. Groseclose and J. M. Snyder. Buying supermajorities. *The American Political Science Review*, 90(2):303–315, 1996.
- [11] D. T. Lee, A. Goel, T. Aitamurto, and H. Landemore. Crowdsourcing for participatory democracies: Efficient elicitation of social choice functions. In *Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing, HCOMP 2014, November 2-4, 2014, Pittsburgh, Pennsylvania, USA*, pages 133–142,

2014.

- [12] D. C. Parkes and M. P. Wellman. Economic Reasoning and Artificial Intelligence. *Science*, 349(6245):267–272, 2015.
- [13] P. Phongpaichit, N. Treerat, Y. Chaiyapong, and C. Baker. Corruption in the Public Sector in Thailand Perceptions and Experience of Households. Political Economy Center (Bangkok: Chulalongkorn University), 2000.
- [14] F. C. Schaffer, editor. Elections for Sale: The Causes And Consequences of Vote Buying. Lynne Rienner Pub, 2007.
- [15] F. C. Schaffer. Why Study Vote Buying? In Elections for Sale: The Causes And Consequences of Vote Buying, Chapter 1, pages 1–16. Lynne Rienner Pub, 2007.
- [16] S. C. Stokes, T. Dunning, M. Nazareno, and V. Brusco. Brokers, Voters, and Clientelism, Chapter 8. What Killed Vote Buying in Britain and the United States? Cambridge University Press, 2013.
- [17] M. Tambe. Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned. Cambridge University Press, 2011.
- [18] P. C. Vicente. A Model of Vote-buying with an Incumbency Advantage. http://www.pedrovicente.org/vb.pdf, 2013.
- [19] P. C. Vicente. Is Vote Buying Effective? Evidence from a Field Experiment in West Africa. *The Economic Journal*, 124(574):F356–F387, 2014.

Judgement Aggregation in Dynamic Logic of Propositional Assignments

Arianna Novaro IRIT, University of Toulouse arianna.novaro@irit.fr Umberto Grandi IRIT, University of Toulouse umberto.grandi@irit.fr Andreas Herzig CNRS-IRIT, University of Toulouse andreas.herzig@irit.fr

ABSTRACT

Judgment aggregation studies situations where groups of agents take a collective decision over a number of logically interconnected issues. A recent stream of papers is dedicated to modelling frameworks of social choice theory, including judgment aggregation, within logical calculi usually designed *ad hoc* for this purpose. In contrast, we propose the use of dynamic logic of propositional assignments (DL-PA), an instance of propositional dynamic logic based on atomic programs modifying propositional evaluations. We provide logical equivalents in DL-PA for the most known aggregation procedures from the literature, for axiomatic properties, and for properties of the constraints, thus showing the versatility of this language for dealing with judgment aggregation.

CCS Concepts

 $\bullet Computing \ methodologies \rightarrow Multi-agent \ systems;$

Keywords

Social Choice Theory; Dynamic logic; Modal logic; Computational Social Choice; Automated Reasoning

1. INTRODUCTION

Social choice theory gathers a number of mathematical models for the study of collective decisions, such as voting and elections, or the allocation of resources among a group of agents. Judgment aggregation is one such model, in which individuals express binary judgments over a set of interconnected issues, which are then aggregated into a collective choice by means of an aggregation rule. This model can be traced back to work by legal scholars [22] and it is now an established framework in artificial intelligence to study complex collective decisions [12, 18].

In judgment aggregation, the correlation among the issues is typically modelled by making use of simple propositional languages. This explicit link with logic inspired researchers to look for a full logical formalisation of the setting, developing logical formalisms that are able to express and reason about aggregation rules and their properties. These efforts are part of a fertile research agenda connecting logic with social choice theory (see, e.g., Endriss, [11]). To cite some examples, Arrow's Theorem [2], one of the cornerstones of social choice theory, has been formalised into higher-order logics [33, 27], first-order logic [17] and modal logic [7]. The ultimate goal of this program is to use automated reasoning techniques to discover new results, an objective that has been partially reached by combining the use of SAT solvers with mathematical lemmas, in preference aggregation [31], ranking sets of objects [15], and in classical social choice theory [5, 6].

Two full-fledged formalisation of judgment aggregation and preference aggregation made use of modal logic: namely, Judgment Aggregation Logic — of which both Hilbert-style [1] and natural deduction [29] axiomatisations have been provided — and the Logic for Social Choice Functions proposed by Troquard et al. [32]. In both cases, the authors develop their own modal languages to formalise judgment aggregation, making the application of automated reasoning techniques less immediate. In this paper, instead, we propose to use the existing language of Dynamic Logic of Propositional Assignments DL-PA [10, 3]. This logic is an instance of Propositional Dynamic Logic PDL (see, e.g., [4]), where atomic programs consist of assignments of truth values to propositional variables. DL-PA is also grounded on propositional logic: in other words, there exists a procedure to translate every modal formula in DL-PA as a propositional formula [10, 3], showing a direct connection with automated reasoning via the use of SAT solvers. Moreover, numerous knowledge representation problems have been expressed in DL-PA, such as belief change operations [19] and abstract argumentation problems [9], and it is arguably a natural choice for the setting of judgment aggregation, where individual opinions are represented as binary evaluations.

We translate most aggregation rules proposed in the literature on judgment aggregation as DL-PA programs, ensuring that the size of each program remains polynomial in the number of agents and issues. Consider for instance the classical majority rule, which collectively accepts a given issue if the number of agents accepting it is greater than the number of agents rejecting it. A straightforward translation of this rule would make use of the explicit description of all possible majorities (i.e., coalitions of more than half of the agents), which would take exponential space. The formalisation in DL-PA we propose solves this problem by a clever use of counters.

Aggregation rules are typically classified and justified by means of axiomatic properties, which are then used in the literature to obtain limitative results on the boundaries of

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

aggregation — the notorious impossibility theorems. We provide DL-PA formulas for the most used aggregation axioms, which can then be interpreted on the translation of a rule. As an aside, we obtain an interesting distinction between axioms that bound the result of the aggregation on one profile, for which we find a translation into propositional logic, and those that require reasoning about multiple profiles, for which DL-PA needs to be used to obtain a compact representation. The final part of the paper focuses on the problem of guaranteeing a safe aggregation, i.e., identifying those types of logical dependencies among the issues such that aggregating individual judgments yields a result consistent with them. In our framework, this problem boils down to checking the validity of a corresponding DL-PA formula.

The paper is organized as follows. In Section 2 we provide the basic definitions of judgment aggregation and of the DL-PA language, as well as setting the stage for a translation of the former into the latter. In Section 3 we propose DL-PA programs to compute the most known judgment aggregation procedures. Section 4 provides translations for the axiomatic properties of aggregation functions, and Section 5 focuses on characterising formulas for safe aggregation. Section 6 concludes the paper and points at a number of directions for future work. We omit most of the proofs in the interest of space: full proofs of the main results can be found in Novaro's Master Thesis [28].

2. PRELIMINARIES

In this section we introduce the formal framework of both binary aggregation with integrity constraints and *star-free* Dynamic Logic of Propositional Assignments. Moreover, we provide our first contribution by showing how to translate aggregation problems into the logic of our choice.

2.1 Binary Aggregation with Integrity Constraints

Two main frameworks can be considered for judgment aggregation: the classic *formula-based* model [24], in which individuals vote directly on complex logical formulas, and *binary aggregation with integrity constraints* [16] where agents have binary opinions on atomic issues linked by an integrity constraint. In this paper we choose the latter setting, and we present it briefly below.

Let $\mathcal{I} = \{1, \ldots, m\}$ be a finite non-empty set of *issues*, on which the *agents* in the finite non-empty set $\mathcal{N} = \{1, \ldots, n\}$, for odd n (as we shall see, this is just a technical assumption), express a binary opinion. Individual opinions form a *boolean combinatorial domain* $\mathcal{D} = \{0, 1\}^m$, where "1" denotes acceptance and "0" rejection. A simple propositional language \mathcal{L}_{PS} can be defined from the set of propositional symbols $PS = \{p_1, \ldots, p_m\}$, with one atom per issue in \mathcal{I} . Then, *integrity constraints* can be defined as formulas IC $\in \mathcal{L}_{PS}$, to express the existence of logical interdependencies among the issues. If there is none, we let IC $= \top$. Consider the following classical example of aggregation, known in the literature as the *discursive dilemma* [22]:

EXAMPLE 1. Three judges have to decide whether (1) a defendant is liable for breaching a contract, depending on whether (2) the contract forbade a particular action and (3) the defendant did it anyway. Let thus $IC = p_1 \leftrightarrow p_2 \wedge p_3$, and consider the profile below:

	p_1	p_2	p_3
Judge 1	1	1	1
$Judge \ 2$	0	0	1
$Judge \ 3$	0	1	0
Majority	0	1	1

As we can see, while the three judges all respect the integrity constraint, the majority outcome does not. Hence, it is not clear whether the judges should give their sentence based on the collective judgment on the conclusion (the defendant is not liable) or the premises (the defendant did an action that was forbidden by the contract).

A ballot $B = (b_1, \ldots, b_m) \in \mathcal{D}$ is a particular choice of zeroes and ones for the issues. The set of all ballots satisfying IC, written Mod(IC) = { $B \mid B \models \text{IC}$ }, is called the *models* of IC. We denote by B_i the *individual ballot* of agent *i*, and we assume $B_i \in \text{Mod}(\text{IC})$ for all $i \in \mathcal{N}$: the agents are *rational*. A profile $\mathbf{B} = (B_1, \ldots, B_n)$ collects all the individual ballots of the agents, such that b_{ij} indicates the *j*-th element of ballot B_i in \mathbf{B} . The set $N_{j:1}^{\mathbf{B}} = \{i \in \mathcal{N} \mid b_{ij} = 1\}$ is the coalition of supporters of issue *j* in \mathbf{B} .

An aggregation procedure (aggregation rule, aggregator) is a function F mapping a rational profile to a (possibly irrational) non-empty set of ballots.

DEFINITION 1. Given a set of agents \mathcal{N} , a set of issues \mathcal{I} and an integrity constraint IC, an aggregation procedure is a function $F : \operatorname{Mod}(\operatorname{IC})^{\mathcal{N}} \to 2^{\mathcal{D}} \setminus \emptyset$, for $2^{\mathcal{D}}$ the powerset of \mathcal{D} . A rule is called resolute if its outcome is a singleton for every profile, and irresolute otherwise. We denote by $F(\mathbf{B})_j$ the outcome of a resolute aggregator on issue j.

The Hamming distance measures how much two ballots disagree on the issues, and is defined as $H(B, B^*) = |\{j \in \mathcal{I} \mid b_j \neq b_j^*\}|$. For example, if $B_1 = (1, 0, 0)$ and $B_2 = (1, 1, 1)$, we have $H(B_1, B_2) = 2$, since they only differ on the last two issues.

2.2 Dynamic Logic of Propositional Assignments

To describe problems in judgment aggregation we choose the language of Dynamic Logic of Propositional Assignments DL-PA [10, 3], an instance of Propositional Dynamic Logic PDL, where atomic programs assign truth value true or false to propositional variables. This logic has already been used to model multi-agent scenarios, such as interactions of agents in normative systems [20] and social simulations [14]. More precisely, we focus on the *star-free* version of DL-PA, without unbounded iteration — which can be obtained from DL-PA via the elimination of the Kleene star [3].

The language of star-free DL-PA is given by the following Backus-Naur grammar:

$$\begin{array}{lll} \varphi & ::= & p \mid \top \mid \perp \mid \neg \varphi \mid \varphi \lor \varphi \mid \langle \pi \rangle \varphi \\ \pi & ::= & +p \mid -p \mid \pi \, ; \pi \mid \pi \cup \pi \mid \varphi ? \end{array}$$

where p ranges over $\mathbb{P} = \{p, q, ...\}$, a countable set of propositional variables.

Atomic formulas consist of variables and constants \top and \bot . Complex formulas are built via negation \neg , disjunction \lor , and a diamond modality for each program $\langle \pi \rangle$. Other Boolean connectives (e.g., conjunction \land , implication \rightarrow , biconditional \leftrightarrow , exclusive disjunction \oplus) and the dual operator $[\pi]\varphi$ are defined in the usual way. Atomic programs

+p and -p assign truth value true or false to variable p, respectively. Sequential composition π ; π' executes first π and then π' , nondeterministic union $\pi \cup \pi'$ nondeterministically chooses to execute either π or π' , and test φ ? checks that φ holds.

A valuation v is a subset of \mathbb{P} that specifies the truth value of every propositional variable, so that $\mathbb{V} = 2^{\mathbb{P}} = \{v_1, v_2, \ldots\}$ is the set of all valuations. When $p \in v$, we say that p is *true* in v (and we say that p is *false* in v otherwise). As illustrated in Table 1, DL-PA programs are interpreted through a unique relation between valuations

$$\begin{split} \|p\| &= \{v \in \mathbb{V} \mid p \in v\} \\ \|\top\| &= 2^{\mathbb{P}} \\ \|\bot\| &= \emptyset \\ \|\neg\varphi\| &= 2^{\mathbb{P}} \setminus \|\varphi\| \\ \|\varphi \lor \psi\| &= \|\varphi \| \cup \|\psi\| \\ \|\varphi \lor \psi\| &= \|\varphi \| \cup \|\psi\| \\ \|\langle\pi\rangle\varphi\| &= \{v \in \mathbb{V} \mid \exists v_1 \text{ s.t. } (v, v_1) \in \|\pi\| \text{ and } v_1 \in \|\varphi\| \} \\ \|+p\| &= \{(v_1, v_2) \mid v_2 = v_1 \cup \{p\} \} \\ \|-p\| &= \{(v_1, v_2) \mid v_2 = v_1 \setminus \{p\} \} \\ \|\pi; \pi'\| &= \|\pi\| \circ \|\pi'\| \\ \|\pi \cup \pi'\| &= \|\pi\| \cup \|\pi'\| \\ \|\varphi ?\| &= \{(v, v) \mid v \in \|\varphi\| \} \end{split}$$

Table 1: Interpretation of DL-PA expressions

Abbreviations have been introduced in the literature to make programs more readable [3, 4, 19]. As a convention, abbreviations for formulas will start with an uppercase letter, while those for programs and counters will start with a lowercase letter. We thus have skip := \top ?, if φ then π_1 else $\pi_2 := (\varphi?; \pi_1) \cup (\neg \varphi?; \pi_2), p \leftarrow q := \text{if } q \text{ then } + p \text{ else } - p$ and if φ do $\pi := \text{if } \varphi$ then π else skip, as well as repeated execution of program π for n times, or up to n times (where both programs execute flip for n = 0):

$$\pi^n := \pi; \pi^{n-1}$$

 $\pi^{\leq n} := (skip \cup \pi); \pi^{\leq n-1}$

We can write any number $s \in \mathbb{N}_0$ in DL-PA via its binary expression, thanks to a conjunction of $t = \lfloor \log s \rfloor + 1$ variables [3]. If x is the binary expression of s, we use a conjunction of q_i and $\neg q_i$ propositional variables, with $i \in \{0, \ldots, \lfloor \log s \rfloor\}$, such that a non-negated variable means that the corresponding binary digit in x is a 1, while a negated variable indicates a 0. For instance, if s = 11, we have that x = 1011 and the corresponding formula in DL-PA is $\mathbf{11} := q_3 \land \neg q_2 \land q_1 \land q_0$.

The following two programs increment or set to zero (i.e., assign truth value false to all the variables in P) a given counter [3]. Let $x^t := \{q_i^x \mid 0 \le i < t\}$ be a set of variables:

$$\begin{aligned} \mathsf{incr}(\mathsf{x}^t) &:= \neg \Big(\bigwedge_{\substack{0 \le i \le t-1 \\ q_i^x \ ; \ 0 \le i \le k-1 \\ q_i^x \ ; \ 0 \le i \le k-1 \\ q_i^x \ ; \ 0 \le i \le k-1 \\ \mathsf{zero}(P) &:= \ \mathbf{;} -p \end{aligned} \right) \\ \mathsf{zero}(P) &:= \ \mathbf{;} -p \end{aligned}$$

We can compare two numbers and check whether one of them is greater than the other, they are equal, or one of them is greater or equal to the other, via the following DL-PA formulas. The general idea is to compare the digits at the same position in the binary expressions of the two numbers.¹

$$\begin{aligned} \mathsf{x} > \mathsf{y} &:= \bigvee_{0 \le k < t} \left(\left(\bigwedge_{k < i < t} (q_i^x \leftrightarrow q_i^y) \right) \land q_k^x \land \neg q_k^y \right) \\ \mathsf{x} = \mathsf{y} &:= \bigwedge_{0 \le k < t} q_k^x \leftrightarrow q_k^y \\ \mathsf{x} \ge \mathsf{y} &:= \mathsf{x} > \mathsf{y} \lor \mathsf{x} = \mathsf{y} \end{aligned}$$

As a convention, we let $\bigwedge_{k < i < t} (q_i^x \leftrightarrow q_i^y) = \top$ for k = t - 1.

Additionally, we may want to flip the truth value of some variables in a set P. The first program below flips the truth value of a single, nondeterministically chosen, variable in P. The second resets the truth value of all variables in P to some new value: as a result, either their truth value has been flipped or not. Both programs execute skip for $P = \emptyset$.

$$\begin{aligned} \mathsf{flip}^1(P) &:= \bigcup_{p \in P} (p \leftarrow \neg p) \\ \mathsf{flip}^{\ge 0}(P) &:= \underset{p \in P}{:} (+p \cup -p) \end{aligned}$$

The next two formulas hold when different types of minimisation are achieved. The first is true if and only if $\neg \varphi$ holds whenever we *do not* change the truth value of some variable in the non-empty set *P*. The second holds if and only if we found the minimal Hamming distance *s* between the states of before and after flipping the variables in *P*, such that φ holds afterwards:

$$\begin{split} \mathsf{D}(\varphi, P) &:= \neg \langle \bigcup_{p \in P} \mathsf{flip}^{\geq 0}(P \setminus \{p\}) \rangle \varphi \\ \mathsf{H}(\varphi, P, \ge s) &:= \begin{cases} \top & \text{if } s = 0 \\ \neg \langle \mathsf{flip}^1(P)^{\le s-1} \rangle \varphi & \text{if } s > 0 \end{cases} \end{split}$$

Observe that $D(\varphi, P)$ does not imply that φ will hold if we flip the truth value of all the variables in P. In our setting this definition suffices, but such alternative formulation has been given as well [19].

2.3 Translating Aggregation Problems into DL-PA

We here show how to translate profiles and aggregation rules into DL-PA. The former is turned into a specific type of valuation, while the latter become programs. We also show how to check rationality in DL-PA and how to turn an arbitrary valuation into one corresponding to a profile.

As a first step, let $\mathbb{B} := \{p_{ij} \mid i, j \in \mathbb{N}\}$ be the subset of \mathbb{P} whose variables encode the opinion of any agent *i* on any issue *j*. Analogously, $\mathbb{O} := \{p_j \mid j \in \mathbb{N}\}$ is the subset of \mathbb{P} whose variables refer to the possible output for any issue *j*. From these two infinite sets, we derive two finite subsets for specific *n* agents and *m* issues. Namely, $\mathbb{B}^{n,m} := \{p_{ij} \mid i \in \mathcal{N}\}$ and $j \in \mathcal{I}\}$ is the set of propositional variables referring to the decision of the agents in \mathcal{N} on the issues in \mathcal{I} , and

¹Suppose two numbers can be expressed with a different amount of binary digits. In this case, if in some program we need to use many counters, we take the maximal value a counter could take as the upper bound for *all* counters in that program. Hence, if t is the maximal number of variables needed to express the maximal value a counter can take, and some other number is expressible by using only k variables (where k < t), it will nonetheless be expressed with t variables by imposing $\neg q_i$ for all $k < i \leq t$. We thus write x instead of x^t .

the variables in $\mathbb{O}^m := \{p_j \mid j \in \mathcal{I}\}$ refer to the collective decision on the issues in \mathcal{I} . Finally, $\mathbb{U} := \{q_i \mid i \in \mathbb{N}\}$ is the subset of \mathbb{P} whose variables are used for finitely many counters in our programs.

The following definition carves out the valuations that correspond to a profile in judgment aggregation.

DEFINITION 2. We say that valuation v_B translates profile $B = (B_1, \ldots, B_n)$ on m issues, in case:

(i)
$$v_{\mathbf{B}} \subseteq \mathbb{B}^{n,m}$$
, and

(*ii*) $p_{ij} \in v_{\mathbf{B}} \iff b_{ij} = 1.$

The first condition ensures that only variables corresponding to the decision of the agents on the issues could possibly be true in v_B . This means, in particular, that counters are initially set to zero. According to the second condition, a variable in v_B is true if and only if the corresponding entry in profile **B** has value 1. For example, if we have profile $\mathbf{B} = ((0,1), (0,0), (1,0))$ for 3 agents and 2 issues, the set $\mathbb{B}^{3,2} = \{p_{11}, p_{12}, p_{21}, p_{22}, p_{31}, p_{32}\}$ corresponds to the entries in the profile, the set $\mathbb{O}^2 = \{p_1, p_2\}$ handles the outcome of aggregation rules and valuation $v_B = \{p_{12}, p_{31}\} \subseteq \mathbb{B}^{3,2}$ encodes the values of the profile.

We now introduce the definition for translating aggregation rules as DL-PA programs.

DEFINITION 3. A program $f(\mathbb{B}^{n,m})$ translates aggregation rule F, if for all profiles **B** and valuations $v_{\mathbf{B}}$ translating **B** according to Definition 2, it is the case that:

• *F* is resolute and $(v_{\mathbf{B}}, v') \in || \mathbf{f}(\mathbb{B}^{n,m}) ||$, implies that for all $j \in \mathcal{I}$ and $p_j \in \mathbb{O}^m$:

$$p_j \in v' \iff F(\boldsymbol{B})_j = 1.$$

 F is irresolute and V^f_{vB} = {v' | (v_B, v') ∈ ||f(B^{n,m})||}, implies that there is a bijection g : F(B) → V^f_{vB} such that if g(B) = v' then for all j ∈ I and p_j ∈ O^m:

$$p_j \in v' \iff b_j = 1.$$

We write the integrity constraint as a formula IC over variables in \mathbb{O}^m . In order to check whether a particular choice of truth values over $\mathbb{B}^{n,m}$ corresponds to a profile, i.e., all the individual ballots satisfy the constraint, we check if the following formula holds.

$$\mathsf{Rational}_{\mathrm{IC}}(\mathbb{B}^{n,m}) := \bigwedge_{i \in \mathcal{N}} [; p_j \leftarrow p_{ij}] \mathrm{IC}$$

Namely, we check whether by copying into the outcome variables the truth values of the variables for each individual ballot, the constraint IC holds.

The following program leads from an arbitrary valuation to one that possibly corresponds to the encoding of a profile, by creating the "right" initial conditions:

$$\operatorname{prof}_{\operatorname{IC}}(\mathbb{B}^{n,m},\mathbb{O}^m) := \operatorname{zero}(\mathbb{O}^m); \operatorname{Rational}_{\operatorname{IC}}(\mathbb{B}^{n,m})?$$

Observe that after its execution all the outcome variables are false, but it is not enough to conclude that condition (i) of Definition 2 holds. Nonetheless, all programs encoding aggregation rules will just need to inspect variables in $\mathbb{B}^{n,m}$ and (possibly) change the truth values of variables in \mathbb{O}^m , and they will initialise at zero all counters as the first step. Therefore, we consider the valuation reached after the execution of $\mathsf{prof}_{\mathrm{IC}}(\mathbb{B}^{n,m},\mathbb{O}^m)$ as encoding a profile as well.

To conclude this section, we highlight an important remark. Since aggregation rules are defined over a specific number of issues, number of agents and integrity constraint, the programs we provide as their DL-PA translation are to be intended as general *program schemas*: a set of issues \mathcal{I} , set of agents \mathcal{N} and constraint IC need to be given to completely spell them out.

3. AGGREGATION RULES

Aggregation rules are the basic bricks of judgment aggregation, allowing to reach a group decision from individual choices. In this section we translate known aggregation rules as DL-PA programs, omitting the proof of correctness of our translations for space constraints.

3.1 Expressibility of Aggregation Rules

We begin by proving a general result that shows how any judgment aggregation rule, as introduced in Definition 1, can be expressed as a DL-PA program.

THEOREM 1. All aggregation rules $F : \text{Mod}(\text{IC})^{\mathcal{N}} \to 2^{\mathcal{D}} \setminus \emptyset$ for some \mathcal{N}, \mathcal{I} and IC are expressible as DL-PA programs.

PROOF. We first deal with the case of a resolute aggregation rule F. Consider the DL-PA program consisting of a sequential composition of sub-programs of the form if φ_{B} do $\pi_{F(B)}$ for each profile B, where $\varphi_{B} = (\bigwedge_{j \in \mathcal{I}} \bigwedge_{i \in N_{j:1}^{B}} p_{ij}) \land$ $(\bigwedge_{j \in \mathcal{I}} \bigwedge_{i \in (\mathcal{N} \setminus N_{j:1}^{B})} \neg p_{ij})$, i.e., φ_{B} completely identifies profile B, and $\pi_{F(B)} = ;_{\{j \in \mathcal{I} \mid F(B)_{j}=1\}} + p_{j}; ;_{\{j \in \mathcal{I} \mid F(B)_{j}=0\}} - p_{j}$, i.e., $\pi_{F(B)}$ modifies the outcome variables according to the result of F on profile B.

For irresolute F it sufficies to consider a sequential composition of sub-programs of the form if φ_B do $\bigcup_{B \in F(B)} \pi_B$, where π_B is defined as $\pi_B = ;_{\{j \in \mathcal{I} | b_j = 1\}} + p_j; ;_{\{j \in \mathcal{I} | b_j = 0\}} - p_j$, generating a non-deterministic program whose output consists of all outcomes of F. These two types of programs clearly translate resolute and irresolute aggregation rules.

While on the one hand the result above shows that DL-PA is fully expressive when it comes to translating judgment aggregation rules, on the other hand the formulas used in the proof are all of size exponential in the number of individuals and issues. More precisely, since all profiles are explicitly given in the specification of the programs, the size is in the order of $2^{|\mathcal{I}|} \times |\mathcal{N}|$. In the remainder of this section we thus present compact programs for a selection of well-known judgment aggregation rules.

3.2 Simple Aggregation Rules

We call the following rules *simple* because they are all resolute, they are easy to explain and understand, and they can also be found in real-world examples.

3.2.1 Dictatorship of Agent i

The dictatorial rule is perhaps the simplest and at the same time less attractive aggregation rule. For all profiles B, the outcome of the dictatorship of some fixed agent $i \in \mathcal{N}$ is her individual ballot. Namely, Dictatorship_i $(B)_j = 1 \iff b_{ij} = 1$ for all $j \in \mathcal{I}$. Its translation in DL-PA can easily be obtained as the following program:

PROPOSITION 1. Let \mathcal{I} and \mathcal{N} be given. Then, program $\mathsf{dict}_i(\mathbb{B}^{n,m}) := ;_{i \in \mathcal{I}}(p_j \leftarrow p_{ij})$ translates rule $\mathsf{Dictatorship}_i$.

3.2.2 Quota Rules

The majority rule is an instance of the more general class of *quota rules* [8]. A quota rule specifies for each issue a certain threshold of support that has to be reached in order for the issue to be accepted in the outcome. The quota qcan be any integer such that $0 \le q \le n + 1$, where n is the number of agents. In case all the issues have the same quota, we speak of *uniform* quota rules. If q_j is the quota for issue $j \in \mathcal{I}$ and $\vec{q} = (q_1, \ldots, q_m)$, we have:

$$\operatorname{Quota}_{\vec{a}}(\boldsymbol{B})_j = 1 \iff |N_{j:1}^{\boldsymbol{B}}| \ge q_j.$$

We now state a result that provides, for every choice of quotas q_j , a DL-PA program translating the corresponding quota rule (by using a counter quota_j for each issue j).

PROPOSITION 2. For \mathcal{I} a set of issues, \mathcal{N} a set of agents, and $0 \leq q_1, \ldots, q_m \leq |\mathcal{N}| + 1$, the Quota_{\vec{q}} rule is translated in the following DL-PA program:

$$\begin{split} \mathsf{quota}_{\vec{q}}(\mathbb{B}^{n,m}) &:= \begin{array}{c} \mbox{; } \mathsf{zero}(\mathsf{quota}_j)\, \mbox{; } \mbox{; } \mathsf{incr}(\mathsf{quota}_j)^{q_j}\, \mbox{; } \\ \mbox{; } \mbox{; } \mbox{(} \mathbb{zero}(\mathsf{supp})\, \mbox{; } \mbox{; } \mbox{if } p_{ij} \mbox{ do incr}(\mathsf{supp}))\, \mbox{; } \\ \mbox{; } \mbox{if } \mathbb{supp} \geq \mathsf{quota}_j \mbox{ do } + p_j \mbox{)}. \end{split}$$

We refer to the specific program for the majority rule as maj. Moreover, for the uniform quota rule with q = 1, called the *nomination rule*, an even more compact program is $\mathsf{nom}(\mathbb{B}^{n,m}) := ;_{j \in \mathcal{I}} (\text{if } \bigvee_{i \in \mathcal{N}} p_{ij} \text{ do } + p_j).$

3.3 Maximisation and Minimisation Rules

In this section we focus on two aggregation rules that are based on maximisation or minimisation processes and aim at amending the outcome of the majority rule, in case it does not satisfy the integrity constraint. The first one is the maximal subagenda rule, while the second one is the minimal number of atomic changes rule [23].

3.3.1 Maximal Subagenda Rule

The maximal subagenda rule returns ballots satisfying the integrity constraint and having maximal agreement (with respect to set inclusion) with the majority outcome:

$$MSA_{IC}(\boldsymbol{B}) = \underset{B \models IC}{\operatorname{argmax}} \{ j \in \mathcal{I} \mid b_j = Maj(\boldsymbol{B})_j \}.$$

Before presenting a DL-PA program translating this rule, we need some further notation. Consider the following programs, which all execute skip if $P = \emptyset$:

$$\begin{split} \mathsf{store}(P) \ &:= \ \mathop{;}_{p \in P} p' \leftarrow p \\ \mathsf{restore}^1(P) \ &:= \ \bigcup_{p \in P} (p \oplus p'?; p \leftarrow p') \\ \mathsf{restore}^{\geq 0}(P) \ &:= \ \mathop{;}_{p \in P} (\mathsf{skip} \cup p \leftarrow p') \end{split}$$

Program store stores the truth value of the variables in P in some fresh variables p', program restore¹(P) restores the truth value of just one variable p' in the corresponding variable in P, and program restore^{≥ 0}(P) restores the truth

value of none, some, or all variables p' in the corresponding variables in P.

We can now present the following program, inspired by analogous work in the literature on belief change [19]. Given that the MSA_{IC} is an irresolute rule we might need to handle multiple outcomes for the same profile: whence its (omitted) proof differs from that of Proposition 1.

PROPOSITION 3. Let \mathcal{I} be a set of issues, \mathcal{N} a set of agents and IC a propositional formula. The MSA_{IC} rule is translated in the following DL-PA program:

$$\begin{split} \mathsf{msa}_{\mathrm{IC}}(\mathbb{B}^{n,m}) :=& \mathsf{maj}(\mathbb{B}^{n,m}) \, ; \, \mathsf{store}(\mathbb{O}^m) \, ; \, \mathsf{flip}^{\geq 0}(\mathbb{O}^m) \, ; \, \mathrm{IC?} \, ; \\ & [\mathsf{restore}^1(\mathbb{O}^m) \, ; \, \mathsf{restore}^{\geq 0}(\mathbb{O}^m)] \neg \mathrm{IC?} \, . \end{split}$$

3.3.2 Minimal Number of Atomic Changes Rule

The minimal number of atomic changes rule returns the following outcome set:

$$MNAC_{IC}(\boldsymbol{B}) = \{B \mid Maj(\boldsymbol{B}^*) = B, B \models IC \text{ and for all } \boldsymbol{B}'$$
$$\sum_{i \in \mathcal{N}} H(B_i, B_i^*) \leq \sum_{i \in \mathcal{N}} H(B_i, B_i')\}.$$

Recall that the Hamming distance H(B, B') between two ballots is the number of issues on which they differ (cf. Section 2.1). This rule thus looks for profiles which are minimally different from the current one, such that the majority rule applied to them would return an outcome consistent with the constraint.

PROPOSITION 4. Let \mathcal{I} be a set of issues, \mathcal{N} a set of agents and IC a propositional formula. The MNAC_{IC} rule is translated in the following DL-PA program:

$$\begin{split} \mathsf{mnac}_{\mathrm{IC}}(\mathbb{B}^{n,m}) &:= \bigcup_{0 \leq d \leq m \cdot n} \left(\mathsf{H}(\langle \mathsf{prof}_{\mathrm{IC}}(\mathbb{B}^{n,m}, \mathbb{O}^m) \, ; \, \mathsf{maj}(\mathbb{B}^{n,m}) \rangle \mathrm{IC}, \\ & \mathbb{B}^{n,m}, \geq d)? \, ; \, \mathsf{flip}^1(\mathbb{B}^{n,m})^d \right); \\ & \mathsf{prof}_{\mathrm{IC}}(\mathbb{B}^{n,m}, \mathbb{O}^m) \, ; \, \mathsf{maj}(\mathbb{B}^{n,m}) \, ; \, \mathrm{IC}?. \end{split}$$

The program $\operatorname{mnac_{IC}}$ finds the minimal number d of variables in the set $\mathbb{B}^{n,m}$ whose truth values can be modified such that applying program maj to this new profile leads to a valuation where the outcome satisfies the constraint.

3.4 Preference Aggregation Rules

This section presents rules inspired by the literature on preference aggregation, and that have been formalised in judgment aggregation in a number of papers. The first is the Kemeny rule [21]. Then, we present the Slater rule, also known as the maxcard subagenda rule [23]. A program similar to the ones presented below can be designed to formalise the ranked pairs rule as well (see [28]).

3.4.1 Kemeny Rule

The outcome of the Kemeny rule consists of those ballots that satisfy the constraint and that minimise the sum of the Hamming distance to the individual ballots in the profile.

$$\operatorname{Kemeny}_{\operatorname{IC}}(\boldsymbol{B}) = \operatorname{argmin}_{B \models \operatorname{IC}} \sum_{i \in \mathcal{N}} H(B, B_i)$$

Let us first introduce the following program and formula:

$$\begin{split} \mathsf{sH}(\mathbb{O}^m, \mathbb{B}^{n,m}) &:= \mathsf{zero}(\mathsf{dis})\,;\,\, \begin{subarray}{l} \vdots &\in \mathcal{N} \\ &i \in \mathcal{N} \\ &i \in \mathcal{N} \\ &j \in \mathcal{I} \\ \end{bmatrix} if \, p_j \oplus p_{ij} \,\, \mathsf{do} \\ &i \mathsf{ncr}(\mathsf{dis})), \\ \mathsf{MD}(\mathbb{O}^m, \mathbb{B}^{n,m}, \mathrm{IC}) &:= [\mathsf{sH}(\mathbb{O}^m, \mathbb{B}^{n,m})\,; \mathsf{store}(\mathsf{dis})\,; \mathsf{flip}^{\geq 0}(\mathbb{O}^m)\,; \\ &\mathsf{sH}(\mathbb{O}^m, \mathbb{B}^{n,m})](\mathsf{dis}' > \mathsf{dis} \to \neg \mathrm{IC}). \end{split}$$

Program sH computes the sum of the Hamming distances between the outcome and the profile. Formula MD is true if and only if whenever some outcome is closer to the profile than the current one, with respect to the Hamming distance, then IC is not satisfied.

PROPOSITION 5. Let \mathcal{I} be a set of issues, \mathcal{N} a set of agents and IC a propositional formula. The Kemeny_{IC} rule is translated in the following DL-PA program:

$$\begin{aligned} & \operatorname{kem}_{\operatorname{IC}}(\mathbb{B}^{n,m}) := \bigcup_{0 \le d \le m} \left(\langle \operatorname{flip}^1(\mathbb{O}^m)^d \rangle (\operatorname{MD}(\mathbb{O}^m, \mathbb{B}^{n,m}, \operatorname{IC}) \wedge \\ & \operatorname{IC})? ; \operatorname{flip}^1(\mathbb{O}^m)^d \right); \operatorname{MD}(\mathbb{O}^m, \mathbb{B}^{n,m}, \operatorname{IC}) \wedge \operatorname{IC}?. \end{aligned}$$

The program kem_{IC} finds the right d such that by flipping the truth value of d outcome variables we get to a valuation that satisfies the constraint, and such that d is the minimal Hamming distance to the rest of the profile.

3.4.2 Slater Rule

The outcome of the Slater rule consists of those ballots satisfying the constraint and minimising the Hamming distance from the outcome of the majority rule for that profile.

$$\operatorname{Slater}_{\operatorname{IC}}(\boldsymbol{B}) = \operatorname*{argmin}_{B \models \operatorname{IC}} H(B, \operatorname{Maj}(\boldsymbol{B})).$$

PROPOSITION 6. Let \mathcal{I} be a set of issues, \mathcal{N} a set of agents and IC a propositional formula. The Slater_{IC} rule is translated in the following DL-PA program:

$$\begin{aligned} \mathsf{slater}_{\mathrm{IC}}(\mathbb{B}^{n,m}) :=& \mathsf{maj}(\mathbb{B}^{n,m}) \, ; \, \bigcup_{0 \leq d \leq m} \left(\mathsf{H}(\mathrm{IC}, \mathbb{O}^m, \geq d)? \, ; \\ & \mathsf{flip}^1(\mathbb{O}^m)^d \right) \, ; \mathrm{IC}?. \end{aligned}$$

The program slater_{IC} first computes the majority rule, and then it finds the minimal distance d such that by flipping the truth value of d variables in the outcome we reach a valuation where the constraint is satisfied. In case the majority outcome already satisfies IC, we have that d = 0.

4. AXIOMS

Aggregation rules can be characterised according to which general properties they satisfy. These properties are called *axioms* in the literature [8]. In line with similar work in preference aggregation, where properties are sometimes distinguished into *intra-profile* and *inter-profile* conditions [30], we here make a distinction between *single-profile* and *multiprofile* axioms. The former type relates the structure of a profile with the outcome of an aggregation rule applied on that profile. The latter type links the structure of two profiles with the outcomes of the same aggregation rule applied on them.

4.1 Single-profile Axioms

We present four classical single-profile axioms, for which we provide a translation in propositional logic. The full DL-PA machinery is thus not necessary in this case.

A rule F is *unanimous* if in case all agents agree on some issue j, the outcome of F for issue j agrees with them.

U: For all B, for all $j \in \mathcal{I}$ and for $x \in \{0, 1\}$, if $b_{ij} = x$ for all $i \in \mathcal{N}$ then $F(B)_j = x$.

A rule is *neutral with respect to the issues* if, when two issues are treated in the same way in the input, they are treated in the same way in the output.

 $N^{\mathcal{I}}$: For any two $j, k \in \mathcal{I}$ and any \boldsymbol{B} , if for all $i \in \mathcal{N}$ $b_{ij} = b_{ik}$ then $F(\boldsymbol{B})_j = F(\boldsymbol{B})_k$.

A rule is *neutral with respect to the domain* if, whenever two issues are treated in an opposite way in the input, their output should be opposite.

 $N^{\mathcal{D}}$: For all **B** and any $j, k \in \mathcal{I}$, if for all $i \in \mathcal{N}$ $b_{ij} = 1 - b_{ik}$ then $F(\mathbf{B})_j = 1 - F(\mathbf{B})_k$.

A rule is *neutral-monotonic* if the acceptance of an issue j in a given profile implies the acceptance of any other issue k which is accepted by a strict superset of individuals:

 M^{N} : For all **B** and any $j, k \in \mathcal{I}$, if $b_{ij} = 1$ implies $b_{ik} = 1$ for all $i \in \mathcal{N}$, and there is $s \in \mathcal{N}$ such that $b_{sj} = 0$ and $b_{sk} = 1$, then $F(\mathbf{B})_{j} = 1$ implies $F(\mathbf{B})_{k} = 1$.

We are now ready to present the following result:

THEOREM 2. Let $\mathbb{B}^{n,m}$ be the set of variables for agents in \mathcal{N} and issues in \mathcal{I} , let F be an aggregation rule for n and m, and let f be its DL-PA translation. Moreover, let:

$$\begin{split} \mathsf{U} &:= \bigwedge_{j \in \mathcal{I}} \left(((\bigwedge_{i \in \mathcal{N}} p_{ij}) \to p_j) \land ((\bigwedge_{i \in \mathcal{N}} \neg p_{ij}) \to \neg p_j) \right). \\ \mathsf{N}^{\mathcal{I}} &:= \bigwedge_{j \in \mathcal{I}} \bigwedge_{k \in \mathcal{I}} \left((\bigwedge_{i \in \mathcal{N}} (p_{ij} \leftrightarrow p_{ik})) \to (p_j \leftrightarrow p_k) \right). \\ \mathsf{N}^{\mathcal{D}} &:= \bigwedge_{j \in \mathcal{I}} \bigwedge_{k \in \mathcal{I}} \left((\bigwedge_{i \in \mathcal{N}} (p_{ij} \leftrightarrow \neg p_{ik})) \to (p_j \leftrightarrow \neg p_k) \right). \\ \mathsf{M}^{\mathsf{N}} &:= \bigwedge_{j \in \mathcal{I}} \bigwedge_{k \in \mathcal{I}} \left((\bigwedge_{i \in \mathcal{N}} (p_{ij} \to p_{ik}) \land \bigvee_{s \in \mathcal{N}} (\neg p_{sj} \land p_{sk})) \to (p_j \to p_k) \right). \end{split}$$

Then, the following equivalences hold:

(i) U holds $\iff \models [\operatorname{prof}_{\operatorname{IC}}(\mathbb{B}^{n,m},\mathbb{O}^m); f(\mathbb{B}^{n,m})] \mathsf{U}.$

(*ii*)
$$\mathbb{N}^{\mathcal{I}}$$
 holds $\iff \models [\operatorname{prof}_{\mathrm{IC}}(\mathbb{B}^{n,m},\mathbb{O}^m); f(\mathbb{B}^{n,m})]\mathbb{N}^{\mathcal{I}}.$

(iii) $N^{\mathcal{D}}$ holds $\iff \models [\operatorname{prof}_{\mathrm{IC}}(\mathbb{B}^{n,m},\mathbb{O}^m); f(\mathbb{B}^{n,m})] N^{\mathcal{D}}.$

(*iv*)
$$M^{N}$$
 holds $\iff \models [\operatorname{prof}_{IC}(\mathbb{B}^{n,m},\mathbb{O}^{m}); f(\mathbb{B}^{n,m})]M^{N}$.

4.2 Multi-profile Axioms

We now present three multi-profile axioms, which we translate as DL-PA formulas. In fact, to check whether an aggregation rules satisfies them, we need to compare the outcomes of the rule on different profiles. Dealing with multiple profiles means referring to more than one valuation, and applying the program expressing rule F more than once.

A rule is *independent* if, whenever an issue j is treated in the same way in two profiles, the outcome of the rule for j is identical in both of them. Formally:

I : For any $j \in \mathcal{I}$ and profiles \boldsymbol{B} and \boldsymbol{B}' , if $b_{ij} = b'_{ij}$ for all $i \in \mathcal{N}$, then $F(\boldsymbol{B})_j = F(\boldsymbol{B}')_j$.

A rule F is *independent-monotonic* if, whenever we consider two profiles such that the second one differs from the first in that some agent i first rejected issue j and then she accepts it, if j was accepted in the first outcome then it should still be accepted in the second. Let $(\mathbf{B}_{-i}, B'_i) = (B_1, \ldots, B'_i, \ldots, B_n)$ for some profile \mathbf{B} :

 M^{I} : For any issue $j \in \mathcal{I}$, agent $i \in \mathcal{N}$, profiles $\boldsymbol{B} = (B_{1}, \ldots, B_{n})$ and $\boldsymbol{B}' = (\boldsymbol{B}_{-i}, B'_{i})$, if $b_{ij} = 0$ and $b'_{ij} = 1$ then $F(\boldsymbol{B})_{j} = 1$ implies $F(\boldsymbol{B}')_{j} = 1$.

An *anonymous* rule treats each agent in the same way. That is, by permuting the order of the individual ballots in the input, the output for all the issues does not change.

A : For all **B** and any permutation $\sigma : \mathcal{N} \to \mathcal{N}$, $F(B_1, \ldots, B_n) = F(B_{\sigma(1)}, \ldots, B_{\sigma(n)}).$

We can now state the following result:

THEOREM 3. Let $\mathbb{B}^{n,m}$ be the set of variables for agents in \mathcal{N} and issues in \mathcal{I} , let F be an aggregation rule for nand m, and let f be its DL-PA translation. Moreover, for $\mathbb{B}_{i}^{n} := \{p_{ij} \mid i \in \mathcal{N}\}$ let:

$$\begin{split} \mathbf{I} &:= \bigwedge_{j \in \mathcal{I}} \left((p_j \to [\mathsf{flip}^{\geq 0}(\mathbb{B}^{n,m} \setminus \mathbb{B}_j^n) \, ; \operatorname{prof}_{\mathrm{IC}}(\mathbb{B}^{n,m}, \mathbb{O}^m) \, ; \\ & \mathbf{f}(\mathbb{B}^{n,m})]p_j \right) \wedge \, (\neg p_j \to [\mathsf{flip}^{\geq 0}(\mathbb{B}^{n,m} \setminus \mathbb{B}_j^n) \, ; \\ & \operatorname{prof}_{\mathrm{IC}}(\mathbb{B}^{n,m}, \mathbb{O}^m) \, ; \, \mathbf{f}(\mathbb{B}^{n,m})] \neg p_j) \Big) \end{split}$$

$$\mathsf{M}^{\mathsf{I}} := \bigwedge_{j \in \mathcal{I}} \left(p_j \to \bigwedge_{i \in \mathcal{N}} [+p_{ij}; \mathsf{prof}_{\mathrm{IC}}(\mathbb{B}^{n,m}, \mathbb{O}^m); \mathsf{f}(\mathbb{B}^{n,m})] p_j \right)$$

$$\begin{aligned} \mathsf{A} &:= [\mathsf{store}(\mathbb{O}^m) \, ; \, \left(\bigcup_{i,k \in \mathcal{N}} \, ;_{j \in \mathcal{I}} \left(\mathsf{if} \, p_{ij} \oplus p_{kj} \, \mathsf{do} \, (\mathsf{flip}^1(\{p_{ij}\}) \, ; \\ \mathsf{flip}^1(\{p_{kj}\})) \right)^{n-1} \, ; \mathsf{zero}(\mathbb{O}^m) \, ; \, \mathsf{f}(\mathbb{B}^{n,m})] \, \bigwedge_{i \in \mathcal{I}}(p_j \leftrightarrow p'_i) \end{aligned}$$

Then, the following is the case:

- (i) I holds $\iff \models [\operatorname{prof}_{\operatorname{IC}}(\mathbb{B}^{n,m},\mathbb{O}^m); f(\mathbb{B}^{n,m})]I.$
- (ii) M^{I} holds $\iff \models [prof_{IC}(\mathbb{B}^{n,m},\mathbb{O}^{m}); f(\mathbb{B}^{n,m})]M^{I}$.
- (*iii*) A holds $\iff \models [\operatorname{prof}_{\mathrm{IC}}(\mathbb{B}^{n,m},\mathbb{O}^m); f(\mathbb{B}^{n,m})]A.$

5. AGENDA SAFETY

A recurring problem in judgment aggregation is that the outcome of a rule might not respect the given logical dependencies among the issues, even though each agent satisfies the integrity constraint in her individual ballot. As a way out, it can be investigated whether we can ensure that the outcome of certain groups of aggregation rules will always satisfy a given constraint, provided that the constraint relates the issues to one another in a specific way. This approach was first studied in formula-based judgment aggregation under the name of the *safety of the agenda* [13].

5.1 Prime Implicants and Safety

In this preliminary section we make use of the area of logic studying prime implicants to redefine known concepts of the agenda safety problem. Given a set of axioms AX, we call the set $\mathcal{F}_{IC}[AX] := \{F \mid F \text{ satisfies all axioms in AX} and the domain of F is Mod(IC)^{\mathcal{N}} for some \mathcal{N}\}$ a *class of aggregation procedures.* The idea of safety for constraints is then defined as follows.

DEFINITION 4. An integrity constraint IC is safe for the class $\mathcal{F}_{IC}[AX]$ if and only if for all $F \in \mathcal{F}_{IC}[AX]$, we have $F(\boldsymbol{B}) \models IC$ for all inputs $\boldsymbol{B} \in Mod(IC)^{\mathcal{N}}$ for some \mathcal{N} .

Let a *literal* be either a variable p or its negation $\neg p$. A *term* D is a conjunction of distinct literals and D-D' is the subtraction operation over terms, resulting in all the literals of D that are not in D'. A term D is an *implicant* of φ if and only if $D \models \varphi$. We follow the presentation of Marchi et al. [25], and give the following definition:

DEFINITION 5. D is a prime implicant of φ if and only if

- (i) D is an implicant of φ ;
- (ii) for all literals L in D, $(D \{L\}) \not\models \varphi$.

Observe that any constraint IC can be rewritten as a conjunction of negations of prime implicants of \neg IC [26]: in the following we assume that constraints have this syntactical form. The following definitions reinterpret for integrity constraints some known *agenda properties* of formula based judgment aggregation, by making use of the concept of prime implicants. Let \mathbb{P}_{φ} be the set of variables used in φ .

DEFINITION 6. A constraint IC has the k-median property (kMP) if and only if any prime implicant D of \neg IC is such that $|\mathbb{P}_D| \leq k$.

A constraint IC has the simplified median property (SMP) if and only if any prime implicant D of $\neg IC$ is such that $|\mathbb{P}_D| = 2$ and for $p, q \in \mathbb{P}_D$ we have that $\neg L_p \land \neg L_q$ is also a prime implicant of $\neg IC$.

For k = 2 we speak of the *median-property* (MP). Observe that if $IC = \top$ we do not have any prime implicant of $\neg IC$, which means that the issues are all independent from one another — a condition known as *syntactic simplified median property* (SSMP) in the literature.

5.2 Safety in DL-PA

We start by proving a lemma which characterises by a DL-PA formula the valuations where some prime implicant of formula φ is true. Let thus φ be a formula and let $P \subseteq \mathbb{P}_{\varphi}$ be a subset of the variables of φ . Given a valuation v, let $P_v := \bigwedge_{1 \le k \le |P|} L_k$ be the term such that for all $p_k \in P$:

$$L_k := \begin{cases} p_k & \text{if } v \models p_k \\ \neg p_k & \text{otherwise} \end{cases}$$

LEMMA 1. Let v be a valuation, φ a formula and $P \subseteq \mathbb{P}_{\varphi}$ a subset of the variables in φ . Term P_v is a prime implicant of φ if and only if $v \models \mathsf{PI}(P, \varphi)$, where

$$\mathsf{PI}(P,\varphi) := [\mathsf{flip}^1(P)] \langle \mathsf{flip}^{\geq 0}(\mathbb{P}_{\varphi} \setminus P) \rangle \neg \varphi \wedge [\mathsf{flip}^{\geq 0}(\mathbb{P}_{\varphi} \setminus P)] \varphi.$$

PROOF. For the left-to-right direction, let P_v be a prime implicant of φ and suppose, for reductio, that $v \models \neg \mathsf{Pl}(P, \varphi)$. Observe that, if $\langle \mathsf{flip}^1(P) \rangle [\mathsf{flip}^{\geq 0}(\mathbb{P}_{\varphi} \setminus P)] \varphi$ is the case, we would have a contradiction with condition (*ii*) of Definition 5 $(P_v \text{ is not prime})$. In fact, we would have that some variable $p_k \in P_v$ corresponding to a literal L_k in P_v would make $(D - \{L_k\}) \models \varphi$ hold. On the other hand, if $\langle \mathsf{flip}^{\geq 0}(\mathbb{P}_{\varphi} \setminus P) \rangle \neg \varphi$ is the case, we would have a contradiction with condition (*i*) of Definition 5 $(P_v \text{ is not an implicant of } \varphi)$. In fact, there would be some valuation v' where the literals in P_v are true and yet $\neg \varphi$ holds. Therefore, we have $v \models \mathsf{Pl}(P, \varphi)$. We prove the right-to-left direction by contraposition. Suppose P_v is not a prime implicant of φ . By Definition 5 this means that either P_v is not an implicant of φ , which would imply that $v \not\models [\mathsf{flip}^{\geq 0}(\mathbb{P}_{\varphi} \setminus P)]\varphi$, or that P_v is not prime, which would imply that $v \not\models [\mathsf{flip}^1(P)]\langle \mathsf{flip}^{\geq 0}(\mathbb{P}_{\varphi} \setminus P) \rangle \neg \varphi$. Thus, in both cases we can conclude that $v \not\models \mathsf{Pl}(P, \varphi)$. \Box

PROPOSITION 7. Constraint IC has the kMP if and only if $\models \neg \text{IC} \rightarrow \bigvee_{\substack{P \subseteq \mathbb{P}_{\text{IC}} \\ |P| \leq k}} \mathsf{PI}(P, \neg \text{IC}).$

PROOF. For the left-to-right direction, assume that IC has the *k*MP and suppose, for reductio, that there is some v such that $v \models \neg IC$ and $v \models \bigwedge_{\substack{P \subseteq \mathbb{P}_{IC} \\ |P| \leq k}} \neg PI(P, \neg IC)$. Since $v \not\models IC$ and IC can be written as a conjunction of negations of prime implicants of $\neg IC$, we know that there must be some prime implicant D of $\neg IC$ such that $v \models D$ and that $|\mathbb{P}_D| \leq k$. By Lemma 1 we thus get that $v \models PI(\mathbb{P}_D, \neg IC)$, which contradicts $v \models \bigwedge_{\substack{P \subseteq \mathbb{P}_{IC} \\ |P| \leq k}} \neg PI(P, \neg IC)$. We prove the right-to-left direction by contraposition. Sup-

We prove the right-to-left direction by contraposition. Suppose IC does not have the kMP: hence, there is some prime implicant D of \neg IC such that $|\mathbb{P}_D| \ge k+1$. We now provide a valuation v such that $v \models \neg$ IC and $v \nvDash \bigvee_{\substack{P \subseteq \mathbb{P}_{\mathrm{IC}} \\ |P| \le k}} \mathsf{Pl}(P, \neg$ IC). Consider valuation v such that $v \models D$ and for all other prime implicants D' of \neg IC, we have $v \nvDash D'$ (such a valuation always exists). Since $v \models D$, we get by Definition 5 that $v \models \neg$ IC. Suppose there was some other term D' such that $v \models \mathsf{Pl}(\mathbb{P}_{D'}, \neg$ IC), $|D'| \le k$ and $v \models D'$: by Lemma 1 this would imply that D' is a prime implicant of \neg IC, contradicting our choice of valuation. \Box

PROPOSITION 8. Constraint IC has the SMP if and only if $\models \neg \text{IC} \rightarrow \bigvee_{p_i, p_k \in \mathbb{P}_{\text{IC}}} (\mathsf{PI}(\{p_i, p_k\}, \neg \text{IC}) \land [\mathsf{flip}(p_i); \mathsf{flip}(p_k)]$ $\mathsf{PI}(\{p_i, p_k\}, \neg \text{IC})).$

PROOF. For the left-to-right direction, assume that IC has the SMP and consider an arbitrary valuation v such that $v \models \neg$ IC. Suppose, for reductio, that the consequent does not hold. Hence, either there is no prime implicant of \neg IC of size 2 or there is one, but the negation of its literals is not a prime implicant of \neg IC. In both cases, this would contradict our assumption that IC has the SMP.

For the right-to-left direction, assume that IC has not the SMP. This means that either it has not the MP, or it has the MP but there is a prime implicant of \neg IC such that its negated literals are not also a prime implicant of \neg IC. In the first case, we would get by Proposition 7 that there is a valuation v such that $v \not\models \mathsf{Pl}(\{p_i, p_k\}, \neg$ IC) thus making the consequent false. In the second case, we would have that $v \not\models [\mathsf{flip}(p_i); \mathsf{flip}(p_k)]\mathsf{Pl}(\{p_i, p_k\}, \neg$ IC), thus falsifying the consequent again. Therefore, $\not\models \bigvee_{p_i, p_k \in \mathbb{P}_{\mathrm{IC}}} (\mathsf{Pl}(\{p_i, p_k\}, \neg$ IC) \land [$\mathsf{flip}(p_i); \mathsf{flip}(p_k)$] $\mathsf{Pl}(\{p_i, p_k\}, \neg$ IC)). \Box

6. CONCLUSIONS AND FUTURE WORK

In this paper we showed how to translate the framework of judgment aggregation, in its model of binary aggregation with integrity constraints, into the propositional dynamic logic DL-PA. The key ideas of our translation consisted in turning profiles of individual ballots into a specific type of valuation, and aggregation rules into DL-PA programs modifying the truth value of a set of variables for the outcome. We then provided compact representations for a number of aggregation rules from the literature. Next, we focused on the axiomatic characterisation of aggregation rules as well as the safety of the agenda problem in DL-PA.

Our work paves the way to further investigations from both a computational and an agent-based perspective. First of all, a significant characteristics of DL-PA is that this modal logic is grounded on propositional logic. In other words, this means that there exists a procedure to translate any DL-PA formula as a formula of propositional logic [10, 3]. Therefore, thanks to the work presented here we now have a chain of translations from aggregation problems to DL-PA, and from DL-PA to propositional logic — which yields us the tool of SAT solvers to enhance research in judgment aggregation. As we anticipated in the introduction, this computer-based approach has already been proven successful in Computational Social Choice [31, 15, 5].

In the second place, our translation allows us to also model the winner determination problem for aggregation rules [13, 23]: i.e., computing the outcome of a rule on a given profile. The formulation of this problem differs between resolute and irresolute aggregators. As an example, for a resolute aggregation rule F the problem is usually formulated as checking for each issue j whether $F(\mathbf{B})_j = 1$ for profile \mathbf{B} . This would hence translate into DL-PA as checking whether $v_{\mathbf{B}} \models [f(\mathbb{B}^{n,m})]p_j$ is the case. Following our previous consideration, it would then be possible to translate instances of such formula into propositional logic.

As far as the questions related to agent-based reasoning are concerned, we propose a possible generalisation and a direction for future research. It is easily seen that our framework could be generalised to deal with a setting where agents are allowed to abstain on the issues. Specifically, it would be sufficient to consider an additional set of propositional variables for the profile, to keep track of the issues on which the agents abstain. This would hence result in two copies of the profile to fully cover the information about abstentions and individual opinions.

Finally, it would be interesting to provide a DL-PA treatment of strategy-proofness for aggregation rules. Given that we have a way to store the values of propositional variables, to compute the Hamming distance and to use counters, incorporating this kind of study in our setting would be fairly straightforward — of course, in case we assume Hamming distance type of preferences over possible outcomes for the agents. We thus leave these questions for future investigation.

REFERENCES

- T. Ågotnes, W. van der Hoek, and M. Wooldridge. On the logic of preference and judgment aggregation. *Autonomous Agents and Multi-Agent Systems*, 22(1):4–30, 2011.
- [2] K. J. Arrow. Social Choice and Individual Values. John Wiley & Sons, 2nd edition, 1963.
- [3] P. Balbiani, A. Herzig, and N. Troquard. Dynamic logic of propositional assignments: a well-behaved variant of PDL. In *Proceedings of the 28th Annual* ACM/IEEE Symposium on Logic in Computer Science, (LICS), 2013.
- [4] P. Blackburn, M. de Rijke, and Y. Venema. Modal Logic. Cambridge University Press, 2001.
- [5] F. Brandt and C. Geist. Finding strategyproof social choice functions via SAT solving. *Journal of Artificial Intelligence Research (JAIR)*, 55:565–602, 2016.
- [6] F. Brandt, C. Geist, and D. Peters. Optimal bounds for the no-show paradox via SAT solving. In Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS), 2016.
- [7] G. Ciná and U. Endriss. A syntactic proof of Arrow's theorem in a modal logic of social choice functions. In Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, (AAMAS), 2015.
- [8] F. Dietrich and C. List. Judgment aggregation by quota rules: majority voting generalized. *Journal of Theoretical Politics*, 19(4):391–424, 2007.
- [9] S. Doutre, A. Herzig, and L. Perrussel. A dynamic logic framework for abstract argumentation. In Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning, (KR), 2014.
- [10] J. van Eijck. Making things happen. Studia Logica, 66(1):41–58, 2000.
- [11] U. Endriss. Logic and social choice theory. In A. Gupta and J. van Benthem, editors, *Logic and Philosophy Today*, volume 2, pages 333–377. College Publications, 2011.
- [12] U. Endriss. Judgment aggregation. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*. Cambridge University Press, 2016.
- [13] U. Endriss, U. Grandi, and D. Porello. Complexity of judgment aggregation. *Journal of Artificial Intelligence Research*, 45:481–514, 2012.
- [14] B. Gaudou, A. Herzig, E. Lorini, and C. Sibertin-Blanc. How to do social simulation in logic: modelling the segregation game in a dynamic logic of assignments. In *Proceedings of the 12th International Workshop on Multi-Agent Systems and Agent-Based Simulation*, 2011.
- [15] C. Geist and U. Endriss. Automated search for impossibility theorems in social choice theory: Ranking sets of objects. *Journal of Artificial Intelligence Research*, 40:143–174, 2011.
- [16] U. Grandi and U. Endriss. Binary aggregation with

integrity constraints. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence, (IJCAI), 2011.

- [17] U. Grandi and U. Endriss. First-order logic formalisation of impossibility theorems in preference aggregation. *Journal of Philosophical Logic*, 42(4):595–618, 2013.
- [18] D. Grossi and G. Pigozzi. Judgment aggregation: a primer. Synthesis Lectures on Artificial Intelligence and Machine Learning, 8(2):1–151, 2014.
- [19] A. Herzig. Belief change operations: a short history of nearly everything, told in dynamic logic of propositional assignments. In Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning, (KR), 2014.
- [20] A. Herzig, E. Lorini, F. Moisan, and N. Troquard. A dynamic logic of normative systems. In Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI), 2011.
- [21] J. G. Kemeny. Mathematics without numbers. Daedalus, 88(4):577–591, 1959.
- [22] L. A. Kornhauser and L. G. Sager. The one and the many: adjudication in collegial courts. *California Law Review*, 81(1):1–59, 1993.
- [23] J. Lang and M. Slavkovik. How hard is it to compute majority-preserving judgment aggregation rules? In Proceedings of the 21st European Conference on Artificial Intelligence, (ECAI), 2014.
- [24] C. List and P. Pettit. Aggregating sets of judgments: an impossibility result. *Economics and Philosophy*, 18(01):89–110, 2002.
- [25] J. Marchi, G. Bittencourt, and L. Perrussel. Prime forms and minimal change in propositional belief bases. Annals of Mathematics and Artificial Intelligence, 59(1):1–45, 2010.
- [26] P. Marquis. Consequence finding algorithms. In Handbook of Defeasible Reasoning and Uncertainty Management Systems, pages 41–145. Springer, 2000.
- [27] T. Nipkow. Social choice theory in HOL: Arrow and Gibbard-Satterthwaite. *Journal of Automated Reasoning*, 43(3):289–304, 2009.
- [28] A. Novaro. Judgment aggregation in dynamic logic of propositional assignments. Master's thesis, University of Amsterdam, 2016.
- [29] T. Perkov. Natural deduction for modal logic of judgment aggregation. Journal of Logic Language and Information, pages 1–20, 2016.
- [30] A. Rubinstein. The single profile analogues to multi profile theorems: mathematical logic's approach. *International Economic Review*, pages 719–730, 1984.
- [31] P. Tang and F. Lin. Computer-aided proofs of Arrow's and other impossibility theorems. *Artificial Intelligence*, 173(11):1041–1053, 2009.
- [32] N. Troquard, W. van der Hoek, and M. Wooldridge. Reasoning about social choice functions. *Journal of Philosophical Logic*, 40(4):473–498, 2011.
- [33] F. Wiedijk. Arrow's impossibility theorem. *Formalized*
- Mathematics, 15(4):171-174, 2007.

Approval in the Echo Chamber

Ben Armstrong and Kate Larson Cheriton School of Computer Science University of Waterloo Waterloo, Ontario, Canada {ben.armstrong, kate.larson}@uwaterloo.ca

ABSTRACT

Recently there has been interest in iterative voting, where voters are able to update their votes based on voting information from previous rounds. In this paper we conduct a series of empirical studies in order to understand the strategic issues which arise when agents, voting to approve a set of k candidates, can base their voting or approval decisions on information from their neighbours in a social network. We illustrate that the k-approval voting rule often results in cyclic voting behaviour, that social network structure matters in terms of strategization, and that homophily in the network decreases strategization for the k-approval voting rule.

Keywords

Social choice, Social simulation, Emergent behaviour, Strategic Voting, Iterative Voting, Approval Voting, Homophily

1. INTRODUCTION

Major elections in recent years have seen uncommonly high levels of divisiveness across the electorate. This tendency to associate with only those considered similar to oneself is called homophily and can cause individuals to be surrounded primarily by others with similar opinions, leading all groups to be convinced that they have the majority of support. A possible cause for the recent levels of homophily exhibited in the world is social networks such as Facebook or Twitter [5].

The recent rapid growth in prevalence of real-world social networks makes them excellent sources of information for understanding effects such as homophily. Social networks as a general tool are used to model human interactions by representing personal information about existence and strength of relationships, and the distance between two people while also allowing insight into societal trends.

An active area of research studies how social networks affect elections by modeling networks of voters with the connections between voters representing the relationships the voters have. The political opinions of a voter's neighbours can have an impact on the voter's decisions throughout the election, also referred to as the "strategy" of the voter. Strategic voting occurs when a voter submits a ballot that is not entirely honest. In plurality voting systems this most often manifests as voting for a candidate that you prefer over the candidate you believe is going to win even though neither of those is your most preferred candidate.

In addition to plurality there are voting systems such as approval voting. In approval voting, the voters may approve of as many candidates as they wish; in some cases strategization might involve approving the 2 or 3 most preferred candidates. Approval voting has many advantages over plurality voting and other voting rules. It allows a voter to safely vote for their favourite candidate while also allowing strategization. Approval voting has been used in a variety of situations including papal elections for over 300 years and by the American Mathematical Society. Many voting rules are quite complex while approval voting is praised for its simplicity [15]. In this paper we focus on k-approval voting, in which voters must approve of exactly k candidates.

This paper studies the intersection of homophily, social networks, strategic voting, and approval voting. Through simulations we explore the effect that homophily has on a variety of differently structured social networks in which the voters employ strategic k-approval voting. Our findings suggest that homophily leads to a much lower social welfare, having a larger number of candidates leads to a better outcome, and that strategization increases with the number of candidates.

2. RELATED WORK

This work builds upon work done by Tsang and Larson [16] which uses a similar model and focuses on the plurality scoring rule, which is equivalent to 1 - approval. It was shown that under the plurality rule strategization leads to an improvement in social welfare over the truthful outcome, and that the presence of homophily decreased the occurrence of strategization. Overall, similar results are shown in this paper.

The model used in this paper has been inspired by previous work, particularly that of Chopra et al.[3] which introduces a *knowledge graph* containing voters and an edge (i,j) when voter *i* is able to observe the current preference of voter *j*. The difference is that Chopra et al. are focused purely on the strategic behaviour of each voter rather than the behaviour of the entire system.

The model of voter decision-making is based on the work done in [10]. The authors study voting equilibria which occur when all voters in a population cause an outcome they have no incentive to switch away from. It is shown that ap-

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

proval voting in a three candidate election leads to a winner located in the median of the voter positions. Certainly, in our model when voters must approve 2 of the 3 candidates one of their honest approvals will be for the median candidate, making that candidate almost certainly the winner.

Clough studies Duverger's Law using a model of iterative voting very similar to ours in [4]. In her model voters are on a grid network, which is neither small-world or scalefree, and use information from their neighbours in order to participate in iterative plurality voting. The focus in her work is on Duverger's Law which is primarily of interest under plurality voting rules and not studied here though her results do suggest there may be less strategization when voters have less information which is not seen in our results. The primary differences between her work and our own are the network structure, use of plurality voting, and a response function that considers only ties rather than ties and near ties.

Several papers show that iterative voting does not necessarily converge under most common voting rules [9, 7, 12, 13]. In particular, approval voting is shown to have no guarantee of convergence. However, the models used in these papers contained voters with complete information that modified their ballots only when they believed the modification would change the outcome of the election.

On the subject of strategic voting, much work has been done. In particular, Smith provided results from a Bayesian regret analysis of approximately 2.2 million simulations showing approval voting to be better than most common voting rules in the presence of strategic voters [15]. Interestingly, his data also showed that in all common voting rules strategic voting leads to more unhappiness than truthful voting which is in contrast to the findings of our model. The difference between these results could be due to the presence of iterated voting in our model and the lack thereof in Smith's. Slinko and White provide a study of an effect observed in our model [14]: Often, in the first round of updating, a majority of voters will approve of a candidate that has very little honest support. This is done in an attempt to remove support for candidates that are not the voter's favourite candidate but nonetheless have a large amount of support. The result is that often after a single round of updates the least popular candidate has the most approvals. Slinko and White refer to this as "strategic overshooting" and provide results indicating that there is often a "safe" ballot that will be better than an unsafe ballot.

Approval voting has been studied in several contexts, Poundstone performed a Bayesian regret analysis on a variety of voting methods and concluded that unrestricted approval voting was much simpler and led to more satisfaction than most other voting methods [11]. Our work restricts the number of approvals voters are permitted.

3. MODEL

In this section we describe the voting problem analysed in this paper. Let a set, V, of n voters be situated in some social network, G = (V, E), where G is a directed graph such that $(i, j) \in E$ means that voter i observes voter j, and thus may be influenced by the voting behavior or opinions of voter j. We define the out-neighbours of i to be the set $\mathcal{N}(i) = \{j | (i, j) \in E\}$, and thus this is the set of voters who may influence voter i.

Let $C = \{1, \ldots, m\}$ be the set of candidates or alter-

natives over which the voters in V may cast votes, and let each candidate $c_j \in C$ be associated with some *position* $p(c_j) \in [0, 100]$. Furthermore, our voters, V, have single-peaked preferences over the candidate set. Each voter, $i \in V$, has a preferred position p_i , and thus its utility if some candidate is selected with position \hat{p} is

$$u_i(p_i, \hat{p}) = -|p_i - \hat{p}|^2.$$

Each voter casts a ballot, b, from a set of admissible ballots \mathcal{B} . A social choice function, $\mathcal{F}: \mathcal{F} \mapsto \mathcal{P}(C)$ is used to aggregate the voters' ballots and select a subset of candidates as winners. We are interested in situations where voting is iterative and progresses in rounds. In round t, each voter i simultaneously casts a ballot $b_i^{(t)} \in \mathcal{B}$ which is chosen in response to the previous ballot $b_j^{(t-1)}$ of each out-neighbour $j \in \mathcal{N}(i)$. If all voters refrain from updating their ballots in a given round, voting stops (and the system is considered stable). Otherwise, voting continues until r rounds have passed. Each round could be considered as a formal poll, or as a more informal update of decisions by voters that happens naturally over time. After voting stops, the winning set of candidates is decided using the voting function, \mathcal{F} .

3.1 Ballot Formation

In this paper we are particularly interested in how voters form their ballots as part of the iterative voting procedure and we make the observation that voter i can base its ballot decisions on the previous ballots of members of $\mathcal{N}(i)$. We argue that each voter believes that their neighbours are representative of the rest of the network, and thus if a fraction f of their neighbours approve of a particular candidate the voter assumes that the same fraction of voters in V approve that candidate and so may strategically cast a vote accordingly.

More formally, in this paper we study the k-approval voting process in order to understand how iterative voting may lead to strategization amongst networked agents. K-approval voting is a member of a larger class of voting rules, scoring rules, in which ballots are vectors representing a score given to each candidate by that voter. A scoring rule uses a vector of the form $(\alpha_1, \alpha_2, ..., \alpha_m)$ where $\alpha_i \ge \alpha_{i+1}$. The candidate ranked first by the voter is given α_1 points, the candidate ranked second given α_2 points, and so on. The candidate with the highest aggregate score wins. In k-approval voting, a ballot has the form $\{1, ..., 1, 0, ..., 0\}$.

If
$$s_j$$
 is the total score for candidate c_j in $\mathcal{N}(i) \cup i$ then i believes that the fraction of support in the entire network for c_j is $\frac{s_j+1}{S}$ where $S = \mathcal{N}(i)+m$. We use Laplace smoothing to ensure that all candidates have a non-zero chance of winning and thus the vector $\mathbf{s} = (\frac{s_1+1}{S}, \frac{s_2+1}{S}, ..., \frac{s_m+1}{S})$ represents the level of support for each candidate.

In order to decide upon which candidates to approve, a voter assigns a prospect rating to each candidate x by enumerating all possible ties between candidates x and y, calculating the likelihood of that outcome and multiplying by the utility gained if x wins. Specifically, the likelihood of a particular outcome from the other n - 1 voters $\mathbf{b} = (b_1, b_2, ..., b_m)$, where b_i is the number of approvals of candidate i, is given by:

$$Pr(\mathbf{b}; n-1; \mathbf{s}) = \frac{(n-1)!}{b_1! b_2! \dots b_m!} \frac{\prod_{i=1}^m (s_i+1)^{b_i}}{S^{n-1}}$$

Let T(y, x) be the chance of a tie between x and y where x and y are both in a winning position, referred to as a winning tie, and $\tilde{T}(x, y)$ be the chance of an outcome where candidate x has one less vote than candidate y and candidate y is winning. Then voters assign each candidate x a prospect rating using lexicographic tie-breaking, given by:

$$C_x = \sum_{y=1}^{m} (\mathbf{1}_{y < x} T(y, x) (u_x - u_y) + \mathbf{1}_{x < y} \widetilde{T}(y, x) (u_x - u_y))$$

 $\mathbf{1}_{x < y}$ is 1 when x lexicographically precedes y and 0 otherwise. $(u_x - u_y)$ gives the marginal utility gain from voting for x over voting for y. In each round voters calculate C_x for each candidate and approve the k candidates with the highest values. An alternative approach for ballot decisionmaking is discussed in Section 7.

3.2 Network Structure and Properties

We will be interested in understanding how network structure and properties influence strategic choices of voters in the context of k-approval iterative voting. We study two different types of random network structure in this paper: Erdös-Renyi (ER) and Barabási-Albert. These graph structures are used as they have one important property in common, both being *small-world* graphs, and differ in another property, being *scale-free*. Further, we study these network both with and without the presence of homophily, the tendency of similar individuals to associate with one another more often than those with differing views.

Graphs are *small-world* if the average distance between any two nodes in the graph grows proportionally to the logarithm of the number of nodes in the graph [6]. This results in nodes typically being connected to other nodes by a very short path. *Scale-free* graphs have a degree distribution which follows a power law, resulting in many nodes with many edges fewer than average and many nodes with a greater than average number of edges [6]. Many real world networks exhibit small-world [17] and scale-free [1] properties.

Erdös-Renyi (ER) graphs are generated by a parameter, pr, representing the probability of attachment. Any two nodes i, j are connected by a directed edge from i to jwith some provided probability pr. ER graphs are smallworld but not scale-free. When studying ER graphs with homophily, we multiply pr by a homophily factor

$$h = 1 - \frac{|p_i - p_j|}{100}$$

in order to increase the probability of a voter being connected to similar voters. Adding homophily has the effect of reducing the edge density by approximately $\frac{2}{3}$ for the same value of pr.

Barabási-Albert (BA) graphs use preferential attachment to generate a larger variance in average degree. An attachment parameter d is decided upon, the graph begins with d vertices, all connected to each other with edges in both directions. The remaining n - d vertices are added one at a time, attaching each one to d existing vertices, selected randomly with probability proportional to the degree of each



Figure 1: A very basic example of a social network. V_1 is influenced by V_2 and V_2 is also influenced by V_1 .

existing vertex. When an edge is added from i to j, the edge from j to i is also added. These graphs are small-world and scale-free. Homophily can be incorporated by multiplying the probability of adding any connection by homophily factor h. This modification has no effect on the edge density of the network.

4. CONVERGENCE OF ITERATIVE *K*-APPROVAL VOTING

The first property we are interested in is whether iterative k-approval voting converges, that is, whether a state is reached where no voter wishes to update their ballot. While it was known that in a number of situations iterative voting was not guaranteed to converge [7], recent experiments using plurality voting showed that non-convergence was rarely an issue in practice [16]. Unfortunately, as we show in this section and later support with experimental findings, iterative k-approval voting is likely to not converge. We illustrate the problem through a simple example.

Under k-approval, each voter must approve of exactly k candidates and does so based upon a prospect rating assigned to each candidate (discussed in Section 3). Consider the simple network shown in Figure 1 under 2-approval. Let V_1 have preferred position 93 and V_2 have preferred position 24. Assume, furthermore, that there are 3 candidates A, B, and C with positions 0, 43, 35 respectively. We can then induce the following preference orderings over candidates for each voter:

$$V_1: B \succ C \succ A V_2: C \succ B \succ A$$

If the voters submit truthful ballots on the first iteration, then candidates B and C are each awarded a score of 2, and lexicographic tie-breaking results in B winning. At first glance, this seems like a reasonable outcome - both voters agreed on their ballots, and so one might expect that no updates would occur in further iterations of the voting process. However, that is not the case.

After observing each others' initial ballots (due to the structure of the network in Figure 1), each voter computes prospect ratings, using the equations in Section 3, for each candidate. These prospect ratings are shown in Table 1. Voters choose the k candidates with the highest prospect ratings so V_1 will not change its ballot, however, V_2 will switch to a ballot approving A and C. Thus, in round 2, candidates A and B have one approval, while C has two, resulting in candidate C (V_2 's preferred candidate) being declared the winner. Prospect ratings are generated for the candidates after this second round of voting (Table 2), and again resulting in a change in ballots. In particular, voter V_2 would prefer to approve candidates B and C, as it did originally, thus beginning a never-ending cycle.

While at first glance, the cyclic voting behaviour of voter V_2 may seem counter-intuitive, it does have a rational un-

	A	В	C
V_1	-257	313	-56
V_2	-15	-43	58

Table 1: Prospect ratings for each candidate after the election at time t = 0.

	A	B	C
V_1	-721	323	398
V_2	-50	-13	63

Table 2: Prospect ratings for each candidate after the election at time t = 1.

derpinning. After the first ballot, V_2 observes equal support for candidates B and C and is aware that the tie-breaking rule favours B. Thus, by reducing support for B, and approving A and C, voter V_2 is able to ensure that C is the winning candidate. However, in the second round of voting, there is now support for all three candidates (2 approvals for C, one approval each for A and B. By shifting its approvals from A and C to B and C, V_2 is able to ensure that its least preferred candidate A will certainly not be a winner since its receives no approvals. Thus, it reverts back to its original ballot.

If the above explanation correctly personifies the "thought process" of V_2 it reveals both an interesting emergent strength and weakness of the model. First, V_2 does correctly identify that its preferred candidate will not win, despite having a large amount of support, and changes an approval from a more preferred candidate to a less preferred candidate, a much more intelligent action than could have been expected. Second, the voter does not seem to realize that while reverting to its original ballot will accomplish the immediate goal of removing A as a contender in the election, it will also cause a return to the original situation in which B, the most preferred candidate of V_2 loses. Thus, a more advanced model might look ahead and see what effect a change in ballot might have or look to history to avoid cyclical situations. It may also be useful to consider a weaker definition of convergence, where the system is considered stable after a candidate wins for a particular number of consecutive rounds.

5. EXPERIMENTAL SETUP

We are particularly interested in deepening our understanding of the relationship between strategization and iterative voting under k-approval on a social network. To this end, we conducted a series of experiments, varying different aspects of the underlying social network of voters and the number of candidates.

For all experiments we set k = 2 and set the number of iterations to be at most 20. Unless otherwise noted, we set the number of voters to be 150, and varied m from 3 to 5. For m = 3, our findings are the average of 200 trials, while for m = 4 and 5 our findings are the average over 100 trials.

In the first set of experiments, we studied what happened as we varied the underlying social-network structure. In particular, for each class of graph (ER, homophily+ER (hER), BA, homophily+BA (hBA)) we set the parameters so that voters had an average out-degree of approximately 12, 20, and 28 for m = 3 and 4. We measured and report several

m	Homophily?	PoH	M:T	PoS	M:O	% Str
Erc	Erdös-Renyi Graph					
9	No	1.234	1.518	1.476	1.809	0.203
5	Yes	1.169	1.493	1.438	1.806	0.177
4	No	0.992	1.083	1.074	1.168	0.385
4	Yes	1.259	1.609	1.370	1.737	0.407
5	No	0.950	0.954	1.053	1.056	0.397
	Yes	1.073	1.225	1.194	1.350	0.384
Ba	Barabási-Albert Graph					
9	No	1.213	1.488	1.460	1.784	0.193
5	Yes	1.144	1.511	1.393	1.811	0.145
4	No	1.008	1.137	1.085	1.222	0.377
	Yes	1.195	1.530	1.298	1.651	0.378
5	No	0.942	0.966	1.041	1.067	0.398
	Yes	1.036	1.190	1.166	1.326	0.379

Table 3: Summary of results for experiments with 3-5 candidates comparing, for graphs with and without homophily, the average Price of Honesty, Mean:Truthful ratio, Price of Stability, Mean:Optimal ratio, and proportion of voters engaging in strategic behaviour.

	ER	hER	BA	hBA
m = 3	14.66	16.64	14.23	16.38
m = 4	11.86	15.27	12.12	15.09

Table 4: Average number of updates per agent, as a function of social-network structure.

metrics including the prevalence of strategization and the effect of connectivity on social welfare.

A second set of experiments was run on a smaller population of 60 voters with m = 5 candidates and k = 2. Average voter out-degrees were varied over 4, 12, and 20. These experiments begin to provide hints as to how the number of candidates affects the social welfare of the system. However, due to the limited population size and wide variance in average degree relative to population size these simulations are intended as only a starting point for a study on the effects of the number of available candidates.

6. **RESULTS**

In this section we report our findings. We are interested in understanding the frequency with which voters update their ballots, the amount of strategization that occurs as a function of the underlying social network, and the degree to which strategization is either beneficial or harmful to the system in terms of social welfare. We initially report our findings from experiments with m = 3 and 4 candidates, and then provide a short discussion of our preliminary findings with 5 candidates.

6.1 Updating of Ballots

One measure of interest is the frequency in which voters change their votes over a certain period of time. This provides us with insight into both the level of strategization


Proportion of Strategizers, m=3

Figure 2: Proportion of voters strategizing for each network type.

occurring in the system, as well as the cognitive overhead required by voters as they decide which ballots to submit in each round. Table 4 reports the average number of ballot updates for each voter over a period of 20 iterations. We make several observations. First, the voting rarely converged, as was discussed earlier in the paper, and so voters were still best off updating their ballots after 20 iterations. Second, the number of candidates seemed to influence the number of updates slightly. With 4 candidates, there were consistently fewer updates across the system. Finally, while graph structure (i.e. ER vs BA) did not seem to be a significant influence, the presence of homophily in the network was important. This was somewhat unexpected, as we had thought that being surrounded by voters with similar views should make a voter more confident in their ballot.

6.2 Degree of Strategization

A voter is considered to be voting strategically if its ballot is anything but entirely honest. Figure 2 shows the effect of homophily on strategization. In each graph with homophily, the fraction of voters strategizing is consistently (albeit, very slightly in the case of m = 4) lower than the non-homophily version of that graph.

The fraction of strategizing voters tends to decrease with increasing degree for Erdös-Renyi graphs. In Barabási-Albert graphs, when m = 4, that trend continues however it seems as though when m = 3 the fraction of strategizers increases with the degree. The reason for this is unclear but it does represent a difference from plurality voting in which strategization always increased with edge density (to a plateau) [16]

Also interesting to note is that of every single strategizing

voter, exactly one of their approvals was strategic and the other honest. This is unavoidable when m is 3 but at m = 4 agents are capable of approving their two least favourite candidates but seemingly never consider it useful to do so. This is consistent with the idea that approval voting should always allows you to vote for your favourite candidate while also voting strategically for a "lesser of two evils" of candidates more likely to win than your favourite. It has been shown that when voters are allowed to decide the number of candidates they approve it is always useful to approve one's favourite candidate [15].

6.3 Benefits of Strategization

In this section we report on our findings as to how beneficial strategization is for the entire system. We define the *social welfare* of some candidate \hat{c} with position $p(\hat{c})$ being chosen as

$$SW(V) = \sum_{i \in V} u_i(p_i, p(\hat{c}))$$

where p_i is the preferred position of voter i.

We use several other metrics measured across our experiments. The Price of Honesty (PoH) is defined as the ratio of social welfare of the truthful outcome to that of the strategic outcome [16, 2, 9]. Since both utility values are negative, the larger the PoH, the more costly the truthful outcome is, relative to the strategic outcome. We also define the Price of Stability (PoS) to be the ratio of social welfare of the strategic outcome to that of the optimal outcome [16]. A smaller PoS shows that strategization is more beneficial than honesty; the lowest possible value occurs at 1 when the strategic outcome is the optimal outcome. A PoH larger than 1 indicates that strategic behaviour is more beneficial to the population while a PoH less than one indicates truthtelling is more beneficial. These, or similar, metrics have been used in many settings for evaluating the performance of a system [2, 8]. Both were used in the original analysis of this model for plurality voting and seem quite appropriate when the system has converged to a stable state.

In our experiments, however, voting rarely converged, limiting the usefulness of PoS and PoH. Thus, we propose two additional variants of these measures better suited for nonconverged systems. In place of the Price of Honesty we study the mean social welfare (the mean strategic SW of the winner from each round of an election) divided by the truthful social welfare (Mean:Truthful). Price of Strategy is replaced by the mean SW divided by the optimal SW (Mean:Optimal), pictured in Figure 3 and Figure 4 respectively. These measures provide a more accurate representation of the system, though average values of the PoS and PoH are included in Table 3 to illustrate that they follow the same qualitative trends as our new metrics.

Similarities can be seen between PoH and the Mean:Truthful ratio, and the PoS and Mean:Optimal ratio suggesting that the comparisons are valid. The generally lower values of PoH compared to Mean:Truthful suggest that the final strategic result is not as good as the mean strategic result, or that over time strategies tend to become less beneficial. In general, the opposite trend seems indicated by the comparison of PoS and Mean:Optimal SW which suggests that the final strategic result is closer to optimal than the mean strategic result. This seems somewhat contradictory and warrants closer inspection.

While we see little difference when it comes to whether



Figure 3: Mean Social Welfare over Truthful Social Welfare for each network type.

20 Average Out-Degree 28

1

12







Figure 4: Mean Social Welfare over Optimal Social Welfare for each network type.

the underlying social network was generated using ER or BA, we do note that homophily is important. The difference between graphs with and without homophily can be observed most readily when m = 4. We can see from the Mean:Optimal ratio that without homophily voters benefit much more from strategization than those with homophily. We observe a similar effect from the Mean:Truthful ratio: homophily leads to a lower social welfare.

6.4 Simulations with 5 candidates

While we conducted smaller experiments for the case where m = 5 we still report our preliminary findings as they raise some interesting questions. Our results can be seem in Figure 5.

First, we note that the Mean:Truthful ratio is now consistently below 1, indicating that the actual (strategic) outcome is always better than the honest outcome and suggesting that as the space for strategization grows it becomes more beneficial. Evidence for this is also found by observing that the Mean:Optimal ratio is closer to 1 than in previous experiments.

We also noticed a considerable difference in the proportion of strategic voters. With 4 candidates, a lower degree led to more strategization while with 3 and 5 candidates degree seemed to have little effect on strategization levels. However, with 5 candidates there is significantly more strategization occuring compared to the 3-candidate case; consistently 35-40% of candidates strategize. Interestingly, with 4 candidates, there is a case where strategization is at approximately 50%, much higher than seen here. This difference may be related to the differing population sizes but is mildly surprising as the opportunities for strategization are much larger with a larger ballot. We also noted that there were instances when m = 5 where a voter would not vote for any of their top k-candidates, including one instance where as many as 13 voters in a single round did not approve any of their top k candidates. This observation needs to be investigated further, as it opens up a number of questions with respect to the strategy space of voters.

6.5 Comparison with Plurality

The results found in this paper have both similarities and differences to those found under plurality voting [16]. Average PoS and PoH¹ seem to be quite similar when m = 4 (plurality data is not available for m = 3) for ER and BA graphs with a slightly higher PoS for hER and hBA graphs. In general, homophily tends to reduce the benefit of strategization, however k-approval seems to be affected more strongly than plurality.

Curiously, the fraction of agents voting strategically is quite different in k-approval. In both m = 3 and m = 4, the fraction strategizing was higher than in plurality, however (excluding BA and hBA for m = 3) the graphs follow a different curve. In plurality, strategization goes up with degree and here the trend is the opposite.

As the plurality simulations consistently converged within several rounds the number of updates is quite a bit lower, averaging 40 to 80 ballot updates per election. By contrast, k-approval averaged over 2000 updates per election. This massive increase is explained by the fact that in plural-

¹To simplify comparison between the results, we write PoS and PoH rather than Mean:Optimal and Mean:Truthful for this subsection only.



Mean SW / Truthful SW

-

BA ••• 🛥 •• hBA

•• 🖶 •• hFR

FR

Figure 5: Several metrics showing data for 5 candidates over average voter out-degree of 4, 12, and 20.

12

Average Out-Degree

20

0.1

0

4

ity elections converged within 6 rounds while with approval voting the elections did not converge and were terminated after 20 rounds. In plurality the number of agents strategizing quickly drops to zero while under approval the number of strategizers typically remains constant or cycles between large and small numbers in alternating rounds.

CONCLUSIONS 7.

In this paper we have studied the effects of homophily and a variety of social network structures on strategic voting behaviour in a k-approval iterated voting system. Our model uses a population of voters connected in a social network, each with a preference on some issue and several candidates, also with preferences. In each round, the voters observe the support for each candidate amongst their neighbours and determine the expected utility from a vote for each candidate, selecting the k candidates with the top utilities.

We notice several interesting patterns emerge from our data. As with previous work under plurality, strategization is slightly more prevalent and much more useful for networks without homophily, indicating that social welfare is increased when a voter's associates hold a wide variety of opinions. We also noted that as the space for strategization increased (when the number of candidates increased) strategization led to a larger utility gain, as might be expected. Curiously, the fraction of voters strategizing tends to either go down or remain approximately constant as the average degree of the network changes. With plurality voting it was observed that there seems to be a ceiling on the amount of strategization occurring in the network, it is possible that our experiments were simply at this ceiling much of the time.

The results of this study indicate that there is significant room for future work in this area. A more in-depth study of approval voting is warranted, as well many additional voting systems are available for study. In future work, it may be desirable to modify the model in such a way that would make simulations more likely to converge and avoid repetitive behaviour.

The model used in our experiments allows for two different ways of constructing a ballot. Currently, a voter views the ballots of all its neighbours, sums the approvals and calculates the prospects of each individual candidate and selects the k candidates with the highest prospects. This method was chosen due to the intuitive aspect of simply approving of the candidates that give the highest expected utility. The disadvantage in this method is that in considering only specific candidates information about which sets of k candidates are most approved by the voter's neighbours is lost. The alternative method involves calculating the prospects for each individual ballot, rather than each candidate. This would retain some useful information however it would vastly increase the computational complexity of the simulations as, in effect, there would be one candidate for each possible ballot.

A metric more relevant to approval voting might be the notion that a voter can always safely approve of their favourite candidate and also approve of other candidates. This property is always satisfied for m = 3 and 4, and only very occasionally not met when m = 5. We hypothesize that if voters were allowed a variable number of approvals, this property would always be met.

Presently, the preference structure of the voters in our model is limiting in several ways. The requirement that

voters have single-peaked preferences means there will be a tendency to elect the candidate with the median opinion, and in fact when $k > \frac{m}{2}$ that candidate will always have the most honest approvals. Removing single-peaked preferences could be difficult while keeping intact the preference structures we have given voters. A simple modification to the model could give voters and candidates multi-dimensional preferences to reflect the fact that each agent may have a distinct opinion on several issues. This allows for slightly more variance in preferences over candidates while leaving the possibility for a simple utility function. Unfortunately, this would likely not remove all bias towards electing the median candidate but it may reduce the likelihood of such an event.

Finally, extending this work to yet more election methods could yield interesting comparisons between the methods. Different voting methods could serve two purposes: First, running experiments with alternative methods would, of course, teach about the behaviour of voters under those methods and may yield surprises as with the lack of convergence with approval voters. Second, different voting methods might serve to highlight aspects of this model that could be further refined or may not generalize well, and may give clues as to how to construct a more accurate model.

8. ACKNOWLEDGEMENTS

We would like to thank Alan Tsang for allowing us to build upon code written for his paper studying this model with the plurality rule and for his answers to numerous questions during the writing this paper.

REFERENCES

- A.-L. Barabási. Scale-free networks: a decade and beyond. science, 325(5939):412–413, 2009.
- [2] S. Brânzei, I. Caragiannis, J. Morgenstern, and A. D. Procaccia. How bad is selfish voting? In AAAI, volume 13, pages 138–144, 2013.
- [3] S. Chopra, E. Pacuit, and R. Parikh. Knowledge-theoretic properties of strategic voting. In European Workshop on Logics in Artificial Intelligence, pages 18–30. Springer, 2004.
- [4] E. Clough. Strategic voting under conditions of uncertainty: A re-evaluation of duverger's law. British Journal of Political Science, 37(02):313–332, 2007.
- [5] Y. Halberstam and B. Knight. Homophily, group size, and the diffusion of political information in social networks: Evidence from Twitter. Technical report, National Bureau of Economic Research, 2014.
- [6] M. O. Jackson et al. Social and economic networks, volume 3. Princeton University Press, Princeton, 2008.
- [7] O. Lev and J. S. Rosenschein. Convergence of iterative voting. In Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 2, pages 611–618. International Foundation for Autonomous Agents and Multiagent Systems, 2012.
- [8] R. Meir, O. Lev, and J. S. Rosenschein. A local-dominance theory of voting equilibria. In Proceedings of the Fifteenth ACM Conference on Economics and Computation, pages 313–330. ACM, 2014.

- [9] R. Meir, M. Polukarov, J. S. Rosenschein, and N. R. Jennings. Convergence to equilibria in plurality voting. In Proc. of 24th Conference on Artificial Intelligence (AAAI-10), pages 823–828, 2010.
- [10] R. B. Myerson and R. J. Weber. A theory of voting equilibria. American Political Science Review, 87(01):102–114, 1993.
- [11] W. Poundstone. Gaming the vote: Why elections aren't fair (and what we can do about it). Macmillan, 2008.
- [12] R. Reyhani and M. C. Wilson. Best-reply dynamics for scoring rules. In 20th European Conference on Artificial Intelligence. IOS Press, 2012.
- [13] S. Sina, N. Hazon, A. Hassidim, and S. Kraus. Adapting the social network to affect elections. In Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), pages 705–713, 2015.
- [14] A. Slinko and S. White. Is it ever safe to vote strategically? Social Choice and Welfare, 43(2):403–427, 2014.
- [15] W. D. Smith. Range voting. The paper can be downloaded from the author's homepage at http://www. math. temple. edu/~ wds/homepage/works. html, 2000.
- [16] A. Tsang and K. Larson. The echo chamber: Strategic voting and homophily in social networks. In Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, pages 368–375. International Foundation for Autonomous Agents and Multiagent Systems, 2016.
- [17] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *nature*, 393(6684):440–442, 1998.

Practical Algorithms for Computing STV and Other Multi-Round Voting Rules

Chunheng Jiang Rensselaer Polytechnic Inst. Dept. of Computer Science jiangc4@rpi.edu Sujoy Sikdar Rensselaer Polytechnic Inst. Dept. of Computer Science sikdas@rpi.edu Hejun Wang Rensselaer Polytechnic Inst. Dept. of Computer Science wangj38@rpi.edu

Lirong Xia Rensselaer Polytechnic Inst. Dept. of Computer Science xial@cs.rpi.edu Zhibing Zhao Rensselaer Polytechnic Inst. Dept. of Computer Science zhaoz6@rpi.edu

ABSTRACT

STV is one of the most commonly-used voting rules for group decision-making, especially for political elections. However, the literature is vague about which tie-breaking mechanism should be used to eliminate alternatives. We propose the first algorithms for computing co-winners under STV, each of which corresponds to the winner under some tie-breaking mechanism. This problem is known as *parallel-universestiebreaking (PUT)-STV*, which is known to be NP-complete to compute [9]. We conduct experiments on synthetic data and Preflib data, and show that standard search algorithms work much better than ILP. We also explore improvements to the search algorithm with various features including pruning, reduction, caching and sampling.

1. INTRODUCTION

Voting is one of the most practical and popular ways for group decision-making, and is one of the major topics under *social choice theory*. In the past decades there has been a growing literature of *computational social choice*, which studies computational aspects of social choice problems and voting rules [4]. More recently, computational social choice, in conjunction with algorithmic game theory, has been recognized as one of the eleven "fundamental methods and application areas" of AI, according to The One Hundred Year Study on Artificial Intelligence [14].

One of the earliest and the most fundamental problems in computational social choice is the computation of winners of well-studied voting rules. In fact, the widely-regarded first paper in computational social choice, published by Bartholdi et al. in 1989 [3], proved that *Dodgson*'s rule and the *Kemeny* rule are NP-hard to compute. In addition, the *Slater* rule is also NP-hard to compute [7].

For political elections, the plurality rule seems to be the most common choice. Perhaps the second one is *Single Transferable Vote (STV)*, also known as *instant runoff voting, alternative vote*, or *ranked choice voting*. According to wikipedia, STV is being used to elect senators in Australia, city councils in San Francisco (CA, USA) and Cambridge (MA, USA), and has recently been approved to be used for state and federal elections in Maine State in the USA.

A typical description of STV is the following. Suppose there are *m* alternatives. In each round, we calculate the *plurality score* for each remaining alternative, which is the number of times it is ranked in the first place. The alternative with the smallest plurality score is eliminated. This has the effect of transferring the ballots in support of the eliminated candidate to their corresponding favorite remaining candidate. The last-standing alternative is the winner.

However, it was not clear from the literature which alternative should be eliminated when two or more alternatives are tied for the last place in a round. For example, in the San Francisco version, "a tie between two or more candidates shall be resolved in accordance with State law" [1]. See [2] for a list of commonly used variants of STV.

Random elimination and fixed-order tie-breaking are two popular tie-breaking mechanisms for STV. Random elimination, as the name suggests, means that whenever multiple alternatives are tied for the last place, the one to be eliminated is chosen uniformly at random. Fixed-order tie-breaking is characterized by a linear order \mathcal{O} , called the *priority order*, over the alternatives. Among all alternatives that are tied for the last place in a certain round, the one that is ranked lowest in \mathcal{O} is eliminated. However, random elimination may result in poor ex-post satisfaction due to randomness. For fixed-order tie-breaking, it is unclear how the priority order should be determined, and the existence of such an order itself is unfair to the alternatives who are ranked low in the priority order. Formally, STV with fixed-order tie-breaking violates *neutrality*.

A natural solution is to output all alternatives who can be made to win under *some* tie-breaking mechanism. This multi-winner version of STV is called *parallel-universestiebreaking (PUT)-STV* [9], and the same paper proved that computing the winners under PUT-STV is NP-complete. To the best of our knowledge, no practical algorithm exists for computing PUT-STV.

NP-hardness of PUT-STV may not be a critical real issue in political elections, as the frequency of holding such elections is low, the number of alternatives is often large, and the chance of ties may not be high. The NP-hardness becomes more critical in low-stakes and more frequent group decision-making scenarios, such as a group of friends us-

Appears at: 4th Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE 2017). Held as part of the Workshops at the 16th International Conference on Autonomous Agents and Multiagent Systems. May 8th-9th, 2017. Sao Paulo, Brazil.

ing voting to decide the restaurant for dinner using an online voting website, for example Pnyx [5], robovote.org, or opra.tech. In such cases, in addition to computing all winners as soon as possible, a more practical objective is to design anytime algorithms for PUT-STV to encourage early discovery of winners for better user experience and timely decision-making.

To address this problem, we model the problem of determining the set of all co-winners under different run-off voting rules as a search problem in AI. We compare standard AI search algorithms together with various ways of improving the performance w.r.t. the following measures of performance.

- Time taken to discover all winners.
- Early discovery of a large portion of winners.

The first measure is important for high-stakes applications such as political elections, because we want to make sure that all winners are found. The second measure is important for low-stakes applications where we are given limited resources and must output as many winners as possible.

1.1 Our Contributions

We model the PUT-STV problem as a search problem and propose various algorithms with different combinations of features, including, pruning, reduction, cache, and sampling. We employ the following techniques to improve the running time of our search algorithms and to reduce the search space explored:

- **Pruning** cuts all branches that do not lead to new winners.
- **Reduction** tries to remove multiple alternatives in each round.
- **Caching** stores visited states and prevents the same states from being explored again.
- Sampling can be seen as a preprocessing step: we first randomly sample multiple priority orders O and run STV with fixed-order tie-breaking O to compute multiple winners to start with, before running the search algorithm.

All algorithms are tested on three types of datasets: synthetic datasets with i.i.d. rankings chosen uniformly at random, i.i.d. single-peaked rankings, and Preflib data. Our main discoveries are the following.

- 1. Standard search techniques from AI perform better than ILP formulations (Section 5).
- 2. Caching helps increase performance. Unfortunately, reductions and sampling are expensive to compute and do not provide any benefit (Section 4).
- 3. For single-peaked preferences, ties are rare, and the running time grows linearly with the size of the profile (Section 4.3).

We also extend our algorithms to other multi-round voting rules, including Baldwin and Coombs, which use Borda score and veto score in each round, respectively. Computing all winners under PUT-Baldwin or PUT-Coombs is NPhard [13].

1.2 Related Work and Discussions

There is a large literature on computational complexity of winner determination under commonly-studied voting rules. In particular, computing winners of the Kemeny rule has attracted much attention from researchers in AI and theory, see for example [8, 12] and references therein. However, STV has been overlooked in the literature, despite its popularity. We are not aware of previous work on practical algorithms for PUT-STV.

In this paper we do not discuss how to choose a single winner from the output of PUT-STV, such as the president, when multiple alternatives are PUT-STV winners. This is mostly up to the decision-maker's choice. For high-stakes applications, we believe that being able to identify potential co-winners under STV w.r.t. different tie-breaking mechanisms is important in itself, because it can detect and resolve post-election dispute on tie-breaking mechanisms.

As discussed in the Introduction, we believe that the computation of PUT-STV is important not only for political elections, but also, perhaps more importantly, for everyday group decision-making scenarios. In such cases anytime algorithms are necessary, and our search algorithms naturally have anytime guarantee—they can be terminated at any time and output the winners that have been explored so far. This is another advantage of our search algorithms over ILP.

Our work is related to a recent work on computing winners of commonly-studied voting rules by MapReduce [10], where the authors proved that computing STV is P-complete. We note that STV in [10] is with a fixed-order tie-breaking mechanism, while our paper focuses on PUT-STV. Our technique can also be used to compute PUT-Ranked-Pairs, which is NP-compete to compute [6]. See [11] for more discussions on tie-breaking mechanisms in social choice.

2. PRELIMINARIES

An election is given by a pair $E = (\mathcal{A}, \mathcal{N})$ where $\mathcal{A} = \{a_1, \ldots, a_m\}$ is a set of alternatives, and $\mathcal{N} = \{1, \ldots, n\}$ is a set of voters. Let $\mathcal{L}(\mathcal{A})$ denote the set of all possible linear orders on \mathcal{A} . A profile of n voters is a collection $P = (V_1, \ldots, V_n)$ of votes where for each $i \leq n, V_i \in \mathcal{L}(\mathcal{A})$. The set of all profiles on \mathcal{A} is denoted by \mathcal{P} . A voting rule r is a function $r : \mathcal{P} \to \mathcal{A}$ that maps a profile to a unique winning alternative.

A scoring function is identified by a collection of scoring vectors $M = (\vec{s}_1, \ldots, \vec{s}_m)$, where for each $\hat{m}, \vec{s}_{\hat{m}}$ is a vector of non-negative numbers so that for every pair $k, k' \leq \hat{m}$, if k < k', then $\vec{s}_{\hat{m}}(k) \geq \vec{s}_{\hat{m}}(k')$ holds. The scoring function given by M is denoted by $score_M : \mathcal{L}(\mathcal{A}) \times \mathcal{A} \to \mathbb{Z}_{\geq 0}$. For any set of alternatives \mathcal{A} , a linear order $V \in \mathcal{L}(\mathcal{A})$, and any alternative c ranked at position k by V, $score_M(V, c) = \vec{s}_{|\mathcal{A}|}(k)$.

We can view the scoring functions as being defined by an $m \times m$ matrix, where the rows $\hat{m} \leq m$ correspond to the scoring vector $\vec{s}_{\hat{m}}$. As an example, the *Borda* scoring function is defined by a left triangular matrix where each \hat{m} -th row is the vector $(\hat{m} - 1, \hat{m} - 2..., 0, ..., 0)$ as shown in Figure 1.

EXAMPLE 1. Given the linear order $V = 3 \succ 1 \succ 2 \succ 4$ over the set of alternatives $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$, the Borda scoring function assigns a score of 2 to alternative a_1 , denoted by $score_{M_{Borda}}(V, 1) = 2$ which corresponds to the

\vec{s}_1	0				0	0				0
\vec{s}_2	1	0				1	0			
\vec{s}_3	2	1	0			1	1	0		
\vec{s}_4	3	2	1	0		1	1	1	0	
\vec{s}_5	4	3	2	1	0	1	1	1	1	0
M_{Borda}					l	M _{Veto}				

Figure 1: Matrix view of Borda and Veto scoring functions.

(4,2)-th element of the matrix M_{Borda} in Figure 1.

In this paper we will consider the following voting rules.

2.1 Scoring run-off voting rules

A scoring run-off voting rule is defined by a scoring function f, and a priority function $g: 2^{\mathcal{N}} \to \mathcal{N}$ and proceeds in m-1 rounds, where at each round an alternative with the lowest score by f, ties being broken by g, is eliminated and the agents' votes are determined on the remaining alternatives. The remaining alternative is declared as the winner. In this paper, we are interested in the following well-studied voting rules: 1. Single Transferable Vote (STV) where the scoring function is the plurality function, 2. Coomb's rule defined by the veto function, and 3. Baldwin's rule defined by the Borda function.

Notice that the choice of priority function affects the outcome of the voting rule. An alternative is a *co-winner* w.r.t. a scoring run-off rule if there exists a priority function under which the alternative is declared the winner. This leads to the question: Can we determine the set of all possible winning alternatives under a given scoring run-off voting rule?

DEFINITION 1. (PUT-WINNERS) Given a profile P, and a voting rule r, we are asked to compute the set of all cowinners.

An alternate view of run-off voting rules is that given a profile, each voting rule corresponds to an order of eliminating the alternatives. Indeed, a brute force way to determine the set of all co-winners is to explore every possible order in which to eliminate alternatives one after the other.

This suggests two important avenues to pursue:

- Model the PUT-WINNERS problem as a search problem, where starting at the state with all alternatives, we expand the frontier by eliminating one alternative at a time and explore the state space until we reach a state where all but one alternative remains and the corresponding alternative is a co-winner. Tracing the paths to each reachable state gives us the corresponding voting rules.
- Formulate the problem as an ILP where feasible solutions correspond to the elimination of exactly one alternative in each of m-1 rounds. We can test whether an alternative is a co-winner by testing whether a solution where the alternative is not eliminated in any round is feasible.

3. MODELING VOTING RULES AS A SEARCH PROBLEM

We can model the class of scoring run-off voting rules as a search problem where:

• States: there are $|2^{\mathcal{A}}| - m$ states, one for each possible elimination of 0 to m - 1 alternatives.

- Start state: no alternatives have been eliminated.

- Successor function: maps the current state to the set of states where an alternative with the lowest score is eliminated.
- Output: a set of winning alternatives.

Beginning from the start state, we add states to the *frontier* using the successor function. At each iteration, we choose a state from the frontier to *explore*, and remove it from the frontier. If all but one alternatives have been eliminated, add the remaining alternative to the set of winners. Otherwise, use the successor function to add new states to the frontier, one for each elimination of an alternative with the lowest score.

We use depth first search and employ the following techniques to improve the performance, and expand on them later.

- (i) pruning involves removing a state from the frontier if all the remaining alternatives are known winners,
- (ii) reduction, involves eliminating more than one alternative,
- (iii) *caching*, involves maintaining a set of states that have been explored, and
- (iv) sampling, where we pre-compute a subset of the possible winners by running the run-off rule using a random priority function.

Reduction Techniques A key idea in run-off voting rules is to eliminate the alternative that has the least support and run the election on the reduced problem with one less alternative. However, there are conditions under which we can remove more than one of the remaining alternatives.

For example, San Francisco STV uses the following condition [1].

If the total number of votes of the two or more candidates credited with the lowest number of votes is less than the number of votes credited to the candidate with the next highest number of votes, those candidates with the lowest number of votes shall be eliminated simultaneously and their votes transferred to the next-ranked continuing candidate on each ballot in a single counting operation.

For STV, we introduce the following generalization of the above reduction technique as follows.

Reduction for STV. In any round, suppose there is an alternative *a* whose plurality score is strictly larger than the total plurality score of all other alternatives with strictly less plurality scores, then those alternatives can be eliminated.

This condition guarantees that no matter what the elimination order is for the alternatives whose plurality score is strictly less than that of a, denoted by A, before alternatives in A are eliminated, none of a or $\mathcal{A} - A \cup \{a\}$ can be eliminated.

Reduction for general multi-round rules. For general multi-round rules we have a weaker reduction condition.

Given a collection of scoring vectors $M = (\vec{s}_1, \ldots, \vec{s}_m)$, and $m^* \leq m$ and any $k \leq m^* - 2$, let $\text{Diff}_M(P, m^*, k)$ denote the maximum reduction in the score difference between a pair of alternatives (a, b), before and after k alternatives have been eliminated in a ranking over m^* alternatives. $\text{Diff}_M(P, m^*, k)$ can be computed in polynomial time by enumerating all positions of a and b and all ways to eliminate k alternatives (there are no more than k^* ways, each of which corresponds to the number of eliminated alternatives that are ranked higher than a and b, between a and b, and after a and b, respectively).

The condition for general multi-round rule with scoring vectors M is: in any round, suppose there exists an alternative a with score s, let s' denote the next highest score and let A denote the alternatives whose scores are strictly less than s. If $s-s' > n \times \text{Diff}_M(P, m^*, |A|)$, then all alternatives in A can be eliminated.

It is not hard to verify the correctness of the two conditions. The condition for STV is stronger than the generic condition for computing PUT-STV.



Figure 2: Comparison of runtime with and without caching on synthetic data for STV.

4. EXPERIMENTAL RESULTS: SEARCH PROBLEM

Each configuration of our experimental setup involves creating datasets of elections with m alternatives and n voters. For each dataset, we conducted experiments to evaluate the performance of depth first search while varying four parameters corresponding to whether the following techniques were used to speedup the algorithm: (i) pruning (P) (ii) reduction (R) (iii) caching (C) (iv) sampling (S), each of which is set to 1 when the technique is used and set to 0 otherwise. Several factors affect the runtime of the search algorithm. At each iteration, we must add a branch for every alternative that is tied with the lowest score. It is easy to see how the number of ties encountered during the running of the algorithm leads to an increase in the size of the search space. In order to mitigate the effect this may have on our results, we decided to focus on the harder cases where there are ties. The profiles in each dataset are marked as (i) easy if there is a unique winner and every round has a unique alternative with the lowest score, and (ii) hard if at some round of the voting rule we encounter more than one alternative tied with the lowest score. We will focus on results for the STV rule

Table 1: Summary of Preflib datasets.

	All profiles	Hard profiles
# profiles	315	49
Avg. $\#$ alternatives	25.23	77.39
Max. $#$ alternatives	242	242
Avg. $\#$ unique orders	28.1	6.37
Max. $\#$ unique orders	4926	30
Avg. $\#$ co-winners	1.1	1.67
Max. $\#$ co-winners	4	4

 Table 2: Average number of co-winners for synthetic datasets

	n									
m	10	20	30	40	50	60	70	80	90	100
10	2.89	2.04	1.88	1.78	1.72	1.64	1.57	1.58	1.53	1.53
20	4.65	4.64	4.2	3.8	3.27	3.07	2.95	2.8	2.81	2.63
30	5.58	7.24	7.4	6.95	6.51	5.85	5.84	5.33	5.05	5.06

on these hard cases.

Prefib Data In order to test the algorithms on real world preference data, we identified profiles from Prefib that have complete preferences. We found 349 profiles with complete preferences, of which 49 or about 15% correspond to hard cases (see Table 1). We find that in the real world data, profiles with ties and multiple co-winners are rare.

Synthetic Data The synthetic datasets were generated as follows: For each value of m and n, we generated profiles with n i.i.d. rankings uniformly at random over m alternatives. We then identified 1000 hard profiles to evaluate the running time and number of nodes explored to discover a given percentage of the co-winners (see Table 2).

4.1 Effect of Caching, Pruning and Reduction

Caching has the most noticeable impact on the running time (see Figure 2). Since it is a natural improvement to apply to any search problem, we leave caching on in all future experiments. The effect of pruning and applying the reduction on synthetic data is summarized in Figure 3 for the STV rule. For every configuration of $m \in \{10, 20, 30\}$ alternatives and $n \in \{10, 20, \ldots, 100\}$ voters, we generate 1000 hard profiles and report the average running time. When we apply pruning, we see a small improvement in the running time (see Figure 3(a)). However, using reductions (see Figure 3(b)) increases the average runtime and a closer inspection reveals that this was due to time spent in evaluating whether the reduction can be applied.

Our experimental results for Preflib data are summarized in Table 3. We found that for STV on real world datasets, the maximum observed running time was only 0.06 seconds.

4.2 Early Discovery and the Effect of Sampling

Our main results are focused on the more practical problem of early discovery where under a given constraint on time or computational resources, we would like to be able to discover as many of the co-winners as possible. We find that the AI search algorithms do have an early discovery property. A large percentage of the co-winners are found early in the exploration. For m = 20, we find that close to 80% of the co-winners are discovered after exploring just 200 states and for m = 30, close to half of all the co-winners are dis-



Figure 3: The effect of pruning (a) and applying the reduction (b) on the running time of the search algorithm on synthetic data for STV.

Table 3: Running time or	n Preflib data for STV.
--------------------------	-------------------------

Running time $(10^{-4}s)$	All profiles	Hard profiles
average	3.35	1.42
minimum	0.39	0.52
maximum	648.93	4.65

Table 4: Average number of ties when running the search algorithm to compute all STV co-winners.

Preferences	m = 10	m = 20	m = 30
Random preferences	4.7255	76.574	1324.301
Single-peaked preferences	1.9354	3.7466	6.4802

covered after exploring only 100 states. Somewhat unsurprisingly even a relatively unsophisticated search strategy without any improvements to reduce the search space other than caching performs significantly better than attempting to discover co-winners by breaking ties at random.

For comparison, we include the running time of the search algorithms for Coomb's rule and Baldwin's rule in Figure 5. Intuitively, we expect to see a lot more ties when computing co-winners for Coomb's rule and fewer ties for Baldwin's rule and we can observe its effect on the running times which are larger than STV in general for Coomb's rule and significantly lower for Baldwin's rule.

4.3 AI Search for Single Peaked Profiles

We generated profiles with i.i.d. single-peaked preferences by following the algorithm developed in [15]. For each configuration of m alternatives and n candidates, we identified 500 hard profiles from a set of randomly generated profiles. Most of the profiles have either a unique winner or only few co-winners (Figure 6(a)) and the median candidate (who is the Condorcet winner) is most frequently the co-winner of the election under the STV rule (Figure 6(b)).

We find that for a given number of candidates, the average running time of the search algorithm to compute all co-winners for single peaked profiles is significantly faster than the average time for profiles with random preferences (Figure 7) and only grows linearly with the size of the profile. For example, with m = 30 alternatives and profiles with 100 voters, the running time of the algorithm to compute all co-winners is under 0.003 seconds on average when preferences are single peaked which is 2 orders of magnitude lower than the average runtime for random preferences which is close to 0.2 seconds. Indeed, while the running time only increased almost linearly with the number of voters for single-peaked preferences, we observe a near exponential increase in running time with the number of candidates in the election when profiles have random preferences. This is not surprising when we consider the number of ties encountered



Figure 4: Early discovery of co-winners on synthetic data for STV.



Figure 5: Running time of the search algorithm on synthetic data with and without pruning for the Coomb's and Baldwin's rules.

on average by the search algorithm as shown in Table 4.

5. ILP FORMULATION

We model the PUT-WINNERS problem as an ILP where the solutions correspond to the elimination of a single alternative in each of m-1 rounds and we test whether a given alternative is the co-winner by checking if there is a feasible solution when we enforce the constraint that the given alternative is not eliminated in any of the rounds. We present ILP formulations of the STV and Baldwin's voting rules below. The ILP for Coomb's rule is similar to the ILP for STV where the scoring rule is changed from plurality to veto. For each alternative $a_i \in \mathcal{A}$, and for each round $t \leq m-1$, we define the variable $x_i^t \in \{0, 1\}$ to model the elimination of a_i at round t.

m	n	Tab Profiles	le 5: ILP for ILP alternatives	STV rule. Uncertain alternatives	Runtime(s)
10	10	392	6.51	3.52	13.026
10	20	104	8.71	6.91	961.64
10	30	88	9.10	7.49	1799.82
20	10	224	7.93	3.34	177.31
20	20	4	12.25	12	4438.66



(b) Frequency of alternative being the winner.

Figure 6: Single-peaked profiles generated i.i.d.

Table 6: ILP for Coomb's Rule.						
n	# profiles	% Uncertain	Runtime (s)			
		candidates				
10	10	83%	2289.39			
20	5	92%	2503.07			
	10 20	Table 6: ILP forn $\#$ profiles1010205	Table 6: ILP for Coomb's Run# profiles% Uncertain candidates101083%20592%			

5.1 STV and Coomb's rule

- For alternative a_i and rounds $t \leq m$, there is a binary variable x_i^t that represents the elimination order. $x_i^t = 1$ if and only if a_i is eliminated in t-th round.
- For each $i \leq m, 1 \leq t \leq m-1$ and $j \leq n$, there is a binary variable $p_{i,j}^t$ that represents alternative a_i 's plurality score in vote V_j in round t.

The constraints are

- The usual constraints for x_i^t to be a full ranking.
- The constraint for $p_{i,j}^t$ of alternative a_i at the top position: $p_{ij}^1 = 1$ and $\sum_{t'=1}^{t-1} x_i^{t'} + p_{ij}^t = 1$
- The constraint for $p_{i,j}^t$ for alternatives from the second position: Let $K_i^t = \frac{\sum_{i' \succ_j i} \sum_{i'=1}^t x_{i'}^{t'}}{|\{i' \succeq_j i\}|}$, then the con-

straint is,

$$K_i^t - \frac{m-1}{m} \le p_{i,j}^t \le K_i^t$$

• For each t, let $Plu_i^t = \sum_j p_{i,j}^t$. Then for all different i, i', we have

$$(1 + \sum_{t' < t} x_{i'}^{t'} - x_i^t) \times M + Plu_{i'}^t \ge Plu_i^t$$

We determine the set of all co-winners as follows: Pick an alternative a_i . Add constraints $\forall t \leq m - 1, x_i^t = 0$. If ILP is feasible, a_i is a co-winner.

5.2 Baldwin's rule

The variables are: for all $i, t \leq m$, there is a binary variable x_i^t that represents the elimination order. $x_i^t = 1$ if and only if a_i is eliminated in t-th round.

The constraints are

- The usual constraints for x_i^t to be a full ranking.
- For each t, let $Plu_i^t = \sum_j (1 + |\{i' \prec i\}| \sum_{i' \prec i} \sum_{t'=1}^{t-1} x_{i'}^{t'})$. Then for all different i, i', we have

$$(1 + \sum_{t' \le t} x_{i'}^{t'} - x_i^t) \times M + Plu_{i'}^t \ge Plu_i^t$$

The set of co-winners is computed as follows: Pick an alternative a_i . Add constraints $\forall t \leq m - 1, x_i^t = 0$. If ILP is feasible, a_i is a co-winner.

5.3 Comparison of ILP to AI Search

Tables 5, 6 and Figure 8 summarize the experimental results from running the ILPs to solve PUT-WINNERS w.r.t. STV, Coomb's rule and Baldwin's rule respectively. The results were obtained using Matlab's ILP solver. It is clear that the ILP solver takes far longer to solve PUT-WINNERS than even the default formulation of AI search without applying any of our speed up techniques. Another major problem we encountered was that the Matlab's ILP solver frequently terminates without being able to determine if the problem is feasible. A simple comparison with the results in Figure 3 reveals that the running times of standard search algorithms from AI are orders of magnitude lower than the running times of the ILP solvers.

6. SUMMARY AND FUTURE WORK

We have made the first steps of designing practical algorithms for computing all winners under STV and other multi-stage rules. We have shown that standard search algorithms are much faster and more reliable than ILP. By running experiments on synthetic dataset, we observe that cache is the most effective feature for improving running time. The algorithms run much faster on i.i.d. generated single-peaked preferences and the winners are around the median. For Preflib data, about 15% profiles we tested need tie-breaking under STV.

There are many more strategies we plan to explore. Suppose we use priority queue to store and sort the nodes to be explored, what is a good priority function to encourage early discovery of new winners? We tried multiple priority functions, such as various weighted combinations of the



Figure 7: Running time of AI search algorithms on single-peaked preferences.



Figure 8: Average runtime of the ILP for Baldwin's rule on elections with 10 candidates as we vary the number of voters.

depth of the node, the new winners in the set, and other features. Unfortunately none of them is significantly better than the standard search algorithm. The next step is to use machine learning to learn a good priority function. A related approach is to use heuristics along with a search algorithm such as A^* search. We are interested in designing good heuristics to inform the search algorithms.

A major challenge in practice, is the limited cache size. When m = 60 our machine sometimes run out of memory. Even if the size of memory is not an issue, the time for checking whether a node has been visited will become significant when m becomes large. How to design a good search algorithm with limited memory is an interesting and important open question. In addition to multi-round rules, we also plan to extend our techniques to compute PUT-ranked-pairs. We will integrate our algorithms for STV and other multi-stage rules to OPRA for everyday group decision-making tasks.

Acknowledgments

We thank Vincent Conitzer, Dominik Peters, Toby Walsh, for helpful discussions and suggestions.

REFERENCES

- [1] https://www.opavote.com/methods/ instant-runoff-voting#san-francisco-rcv.
- [2] https://www.opavote.com/methods/ single-transferable-vote.
- [3] J. Bartholdi, III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.
- [4] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.
- [5] F. Brandt and G. C. C. Geist. Pnyx:: A Powerful and User-friendly Tool for Preference Aggregation. In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, pages 1915–1916, 2015.
- [6] M. Brill and F. Fischer. The Price of Neutrality for the Ranked Pairs Method. In Proceedings of the National Conference on Artificial Intelligence (AAAI), pages 1299–1305, Toronto, Canada, 2012.
- [7] V. Conitzer. Computing Slater rankings using similarities among candidates. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 613–619, Boston, MA, USA, 2006. Early version appeared as IBM RC 23748, 2005.
- [8] V. Conitzer, A. Davenport, and J. Kalagnanam. Improved bounds for computing Kemeny rankings. In Proceedings of the National Conference on Artificial Intelligence (AAAI), pages 620–626, Boston, MA, USA, 2006.
- [9] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 109–115, Pasadena, CA, USA, 2009.
- [10] T. Csar, M. Lackner, R. Pichler, and E. Sallinger. Winner Determination in Huge Elections with MapReduce. In *Proceedings of the AAAI Conference* on Artificial Intelligence, 2017.
- [11] R. Freeman, M. Brill, and V. Conitzer. General Tiebreaking Schemes for Computational Social Choice. In *Proceedings of the 2015 International*

Conference on Autonomous Agents and Multiagent Systems, pages 1401–1409, 2015.

- [12] C. Kenyon-Mathieu and W. Schudy. How to Rank with Few Errors: A PTAS for Weighted Feedback Arc Set on Tournaments. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, pages 95–103, San Diego, California, USA, 2007.
- [13] N. Mattei, N. Narodytska, and T. Walsh. How hard is it to control an election by breaking ties? In Proceedings of the Twenty-first European Conference on Artificial Intelligence, pages 1067–1068, 2014.
- [14] P. Stone, R. Brooks, E. Brynjolfsson, R. Calo, O. Etzioni, G. Hager, J. Hirschberg,
 S. Kalyanakrishnan, E. Kamar, S. Kraus,
 K. Leyton-Brown, D. Parkes, W. Press, A. Saxenian,
 J. Shah, M. Tambe, and A. Teller. Artificial intelligence and life in 2030. One Hundred Year Study on Artificial Intelligence: Report of the 2015-2016 Study Panel, Stanford University, Stanford, CA, September 2016.
- [15] T. Walsh. Generating single peaked votes. arXiv preprint arXiv:1503.02766, 2015.