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 PCP-nets 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 $p \geq 1$ 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 $\mathbf{M}$ and the wine $\mathbf{W}$. 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 CP-net, 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 CP-nets (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 (PCP-nets) 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}	P (trivial)	P (Prop. 1)
L_N		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.	Acyclic	Cyclic
L_{0-1}	P [2]	NP-complete (Prop. 3)
L_N		
L_G		P (Prop. 4)

(a) CP-nets

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.

1. 0-1 loss function (L_{0-1}): the loss is 1 if $\vec{d}$ 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_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 CP-net (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 OPTDECISION (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.

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 (Thm. 6), P for shared tree-structured dependency graph. (Thm. 7)	
L_G	coNP-hard (Thm. 8)	

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 $\Pi_{i \le p} D(X_i)$. 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 $D(S) = \Pi_{X_i \in S} D(X_i)$, and $D(-S) = \Pi_{X_i \in I \setminus S} D(X_i)$. 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 $\succ^i_{C,\vec{u}}$ 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 $\Pi_{i \le p} D(X_i)$ is the transitive closure of $\{(a_i, \vec{u}, \vec{z}) \succ (b_i, \vec{u}, \vec{z})\} : i \le p; a_i, b_i \in D(X_i); \vec{u} \in D(Pa(X_i)); \vec{z} \in D(-(Pa(X_i) \cup \{X_i\}))\}$. 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 $\vec{b} \not\succ_C \vec{a}$.*

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 $\vec{a} \not\succ_C \vec{b}$, 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.

Main dish preference
$M_b \succ M_f$

Main dish **M**	Wine preference
M_b	$W_r \succ W_w$
M_f	$W_w \succ W_r$

$$\mathbf{M} \longrightarrow \mathbf{W}$$

$$\begin{tikzcd} M_bW_r \arrow[r] \arrow[d] & M_bW_w \arrow[d] \\ M_fW_r & M_fW_w \arrow[l] \end{tikzcd}$$

Figure 1: A CP-net representing preferences for dinner consisting of a main dish ($\mathbf{M}$) and wine ($\mathbf{W}$). 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 $f^i_{Q,\vec{u}}$ 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 CP-nets 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,

$$f_Q(C) = \prod_{X_i} \prod_{\vec{u} \in D(Pa_Q(X_i))} f^i_{Q,\vec{u}}(\succ^i_{C,\vec{u}})$$

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.*

M pref.	$Pr.$
$M_b \succ M_f$	0.7
$M_f \succ M_b$	0.3

M	**W** pref.	$Pr.$
M_b	$W_r \succ W_w$	0.6
M_b	$W_w \succ W_r$	0.4
M_f	$W_r \succ W_w$	0.3
M_f	$W_w \succ W_r$	0.7

$$\mathbf{M} \longrightarrow \mathbf{W}$$

(Q)

M pref.
$M_f \succ M_b$

M	**W** pref.
M_b	$W_r \succ W_w$
M_f	$W_r \succ W_w$

$$\mathbf{M} \longrightarrow \mathbf{W}$$

(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 I such 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 $L_N(C, \vec{d}) = |\{\vec{d'} : \exists i : d'_i \succ_C d_i$ 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 $\vec{d} \not\succ_C \vec{d'}|$.*

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_fW_r has a neighborhood loss of 2, and a global loss of 3. M_bW_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 ASSIGNMENTS

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 PCP-nets, 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 PCP-nets.*

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

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

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 PCP-nets.*

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

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 non-dominated 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 $L_G(C, \vec{d}) \geq 1$. This corresponds to a reduction to L_G-LOSS where $k = 0$. $\square$

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 $\overline{L_G\text{-LOSS}}$, 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 $\overline{L_G\text{-LOSS}}$ (see Figure 3), beginning with the construction of a CP-net C as follows:

- $I = \{V_1, \bar{V}_1, ..., V_m, \bar{V}_m\} \cup \{C_1, ..., C_n\} \cup \{D_0, D_1, ..., D_{2m+n}\}$ 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 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 $2 \leq i \leq n$, $Pa(C_i) = Pa(C_i) \cup \{C_{i-1}\}$.
- For all $1 \leq i \leq 2m+n$, we let $Pa(D_0) = C_n$ and $Pa(D_i) = \{D_0\}$.

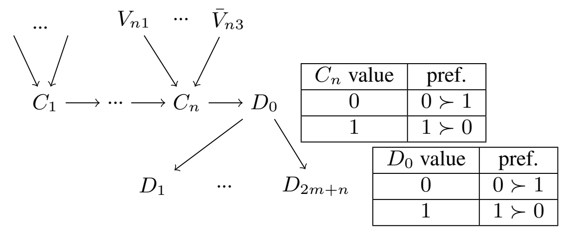

Figure 3: The CP-net used in the proof of Theorem 2.

We populate the associated CP-tables as follows:

- The CPTs for all $V_i, \bar{V}_i$ 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) $V_{i_k} \neq \bar{V}_{i_k}$, (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 $V_i, \bar{V}_i$ 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 $D_{1 \leq i \leq 2m+n}$ may flip to 1 independently. Together, they account for a loss of 2^{2m+n}. The formal proof works as follows.

$\Rightarrow$ 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 $\vec{0}$.

$\Leftarrow$ 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 $\{V_i, \bar{V}_i\}_{i \leq m} \cup \{C_i\}_{i \leq n}$. 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 $\vec{d'}$ 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 $j : V_j \neq \bar{V}_j, V_j, \bar{V}_j \in Pa(C_i)$. If $V_j = 1, \bar{V}_j = 0, V_j, \bar{V}_j \in Pa(C_i)$, set $\phi_j = 1$. Otherwise, if $V_j = 0, \bar{V}_j = 1$, set $\phi_j = 0$. Simultaneously, clause C_i must be satisfied. Once any of the variables $V_i, \bar{V}_i$ is set to 1 in the sequence, 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 $V_i = 1, \bar{V}_i = 0$ in e but $V_i = 0, \bar{V}_i = 1$ in e'. Therefore, when each C_i is flipped to 1 in S, the values of the variables $V_j, \bar{V}_j \in Pa(C_i)$ 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. $\square$

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 non-dominated 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, $L_N(C, \vec{d}) = 0$). $\square$

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 $\vec{d}$ always exists such that $L_G(C, \vec{d}) = 0$. $\square$

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. $\arg\min_{\vec{d}} L_{0-1}(Q, \vec{d}) = \arg\min_{\vec{d}}(1 - \prod_{i=1}^{n} f^{i}_{Q, d_{Pa(X_i)}}(\bar{d}_i \succ d_i)) = 1 - \arg\max_{\vec{d}}(\prod_{i=1}^{n} f^{i}_{Q, d_{Pa(X_i)}}(\bar{d}_i \succ d_i))$. 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]. $\square$

THEOREM 3. *L_N-OPTDECISION is NP-hard for acyclic PCP-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-OPTDECISION:

- $I = \{V_i, \bar{V}_i\}_{1 \leq i \leq m} \cup \{C_i\}_{1 \leq i \leq n} \cup \{D\}$ 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 $2 \leq i \leq n$, we let $Pa(C_i) = Pa(C_i) \cup \{C_{i-1}\}$.
- $Pa(D) = C_n$.

We now define the PCP-tables.

- For all $V_i, \bar{V}_i$, $1 \succ 0$ (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) $V_{i_k} \neq \bar{V}_{i_k}$, (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).

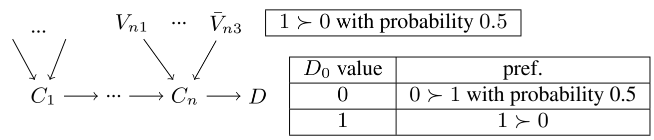

Figure 4: Construction of PCP-net from 3-SAT instance for Theorem 3.

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

$\Rightarrow$ 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 $0.5 \cdot 2n$.

$\Leftarrow$ 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 $V_i, \bar{V}_i$. 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. $\square$

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^w_x 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 $val^w_X = \arg\min_x l^w_x$ by minimizing over $x \in D(X)$. Intuitively, val^w_X 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^w_x. Let Y be the descendants of X. l^w_x is computed as $l^w_x = l^x_{val^x_Y} + f^X_{Q,w}(\bar{x} \succ x)$.

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^x_W.

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 $\mathbf{W}$ *and compute the distribution* $l^{\mathbf{M}}_{\mathbf{W}} = (l^{M_b}_{W_r} = 0.4, l^{M_b}_{W_w} = 0.6, l^{M_f}_{W_r} = 0.7, l^{M_f}_{W_w} = 0.3)$*. We can now compute* $val^{M_b}_{\mathbf{W}} = W_r, val^{M_f}_{\mathbf{W}} = W_w$*. Then we move up one level.*

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

The choice of M_b *guarantees the lowest possible neighborhood loss from* $\mathbf{M}$ *and its descendants. We have that* $val^{M_b}_{\mathbf{W}} = W_r$*. 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, $\overline{L_G\text{-OPTDECISION}}$ defined as: given a PCP-net 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 $V_i, \bar{V}_i$, we now have $1 \succ 0$ with probability 0.5.

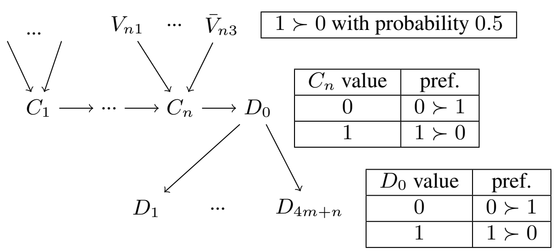

Figure 5: Construction of PCP-net from 3-SAT instance for Theorem 5.

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 $k \geq 1$ 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 $D_i, 1 \leq i \leq 4m + n$ 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}. $\square$

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. $\square$

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 $V_i, \bar{V}_i$, $1 \succ 0$.

• For all C_i, we add the entry $1 \succ 0$ for every assignment to $Pa(C_i)$ where there exists a $k \le 3$ such that all the following conditions are satisfied: (1) $V_{i_k} \neq \bar{V}_{i_k}$, (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 $V_i, \bar{V}_i$, $0 \succ 1$.

• For all C_i, we add the entry $1 \succ 0$ for every assignment to $Pa(C_i)$ where there exists a $k \le 3$ such that all the following conditions are satisfied: (1) $V_{i_k} \neq \bar{V}_{i_k}$, (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 $L_N(P, \vec{d}) \le 2n$.

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 $2n$.

$\Rightarrow$ 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 $2n$.

$\Leftarrow$ Let $\mathcal{F}$ be unsatisfiable, and for the sake of contradiction, let $\vec{d}$ be an assignment with loss $L_N(P, \vec{d}) \le 2n$. 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. $\square$

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 $l^w_{i,x}$ 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 $l^w_x = \sum_{1 \le i \le n} l^w_{i,x}$ which tracks the contribution for the entire profile. Note that for a given value w of the variable W, and quantities l^w_x for every $x \in D(X)$, $val^w_X = \arg\min_x l^w_x$ 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^w_x. Let Y be the descendants of X. The quantity l^w_x is computed as: $l^w_x = \sum_{1 \le i \le n} l^x_{i, val^x_Y} + \{1$, if $\bar{x} \succ^X_{P_i, w} x$; 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^x_W. $\square$

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 $\mathcal{F} = C_1 \wedge ... \wedge C_n$, we give a polynomial time reduction to the complement of L_G-OPTJOINTDECISION, $\overline{L_G\text{-OPTJOINTDECISION}}$ 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 $\overline{L_G\text{-OPTJOINTDECISION}}$ 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:

• $I = \{V_i, \bar{V}_i\}_{1 \le i \le m} \cup \{C_i\}_{1 \le i \le n} \cup \{D_i\}_{0 \le i \le 2m+n}$ 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 $\mathcal{F}$.

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

► 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 $V_i, \bar{V}_i \in I$, 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 $Pa(C_i) = Pa(C_i) \cup \{C_{i-1}\}$.

• $Pa(D_0) = \{C_n\}$.

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

▷ We populate the CP-tables of P_0 as follows:

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

• For all C_i, we add the entry $1 \succ 0$ for every assignment to $Pa(C_i)$ where there exists a $k \le 3$ such that all the following conditions are satisfied: (1) $V_{i_k} \neq \bar{V}_{i_k}$, (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 $\bar{P}_j$. We describe the construction of P_j below. The CP-net P_j has the fol-

lowing dependency graph (See Figure 6):
• For all $1 \leq i \leq m$, $Pa(V_i) = Pa(\bar{V}_i) = \emptyset$. For all $1 \leq i \leq n$, $Pa(C_i) = \emptyset$.
• $Pa(D_0) = \{V_j\}$.
• For all $i = 1, ..., n+m$, $Pa(D_i) = \{D_0\}$
▷ We populate the CP-tables of $P_j, 1 \leq j \leq m$ as follows:
• For all $V_i, \bar{V}_i, 0 \succ 1$.
• For all $C_i, 0 \succ 1$.
• For D_0: if $V_j = 1, 1 \succ 0$. 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.

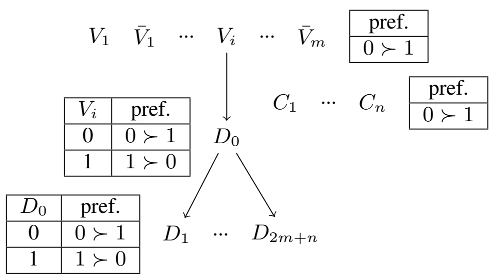

Figure 6: Construction of CP-nets $P_j, 1 \leq j \leq m$ in the proof of Theorem 8. CP-nets $\bar{P}_j$ are constructed in a similar manner.

⇒ 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 $\vec{d}$, $L_G(P, \vec{d}) > 2^{2m+n} - 1$. 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 $d_{D_{1 \leq i \leq 2m+n}} = 0$. Therefore, by construction of P_0, any such $\vec{d}$ has loss $L_G(P_i, \vec{d}) \geq 2^{2m+n}$. Now, consider any decision $\vec{d}$ where for some $1 \leq i \leq m, d_{V_i} = 1$. 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 $\phi_i = 1, d'_{V_i} = 1, d'_{\bar{V}_i} = 0$, and if $\phi_i = 0, d'_{V_i} = 0, d'_{\bar{V}_i} = 1$. 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 $d''_{D_{0 \leq i \leq 2m+n}} = 1$. Therefore, $L_G(P_0, \vec{0}) \geq 2^{2m+n}$. This completes the proof that if $\mathcal{F}$ is satisfiable, then for every decision $\vec{d}$, $L_G(P, \vec{d}) > 2^{2m+n} - 1$.

⇐ Suppose for the sake of contradiction that $\mathcal{F}$ is unsatisfiable and $L_G(P, \vec{0}) > 2^{2m+n} - 1$. 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. □

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 $d'_i = \bar{d}_i$, $L(C', \vec{d'}) > L(C, \vec{d'})$. 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

[1] 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.