More complexity results about reasoning over (*m*)CP-nets

Enrico Malizia
Department of Computer Science, University of Exeter
Exeter, UK
e._lastname_@exeter.ac.uk

ABSTRACT

Aggregating preferences over combinatorial domains has several applications in artificial intelligence. Due to the exponential nature of combinatorial preferences, compact representations are needed, and (*m*)CP-nets are among the most studied formalisms. Unlike CP-nets, which received an extensive complexity analysis, *m*CP-nets, as mentioned several times in the literature, lacked such a thorough characterization. An initial complexity analysis for *m*CP-nets was carried out only recently. In this paper, we further investigate the complexity of *m*CP-nets. In particular, we prove the Σ_3^P-completeness of checking the existence of max optimal outcomes, which was left as an open problem. We furthermore prove that various tasks known to be feasible in polynomial time are actually P-complete. This shows that these problems are inherently sequential, and hence they cannot benefit from highly parallel computation. The P-completeness results here proven are among the very first of this kind in the computational social choice literature.

KEYWORDS

CP-nets; mCP-nets; combinatorial preferences; preference aggregation; complexity; rank voting; max voting; P-completeness.

1 INTRODUCTION

The problem of managing and aggregating agent preferences has attracted extensive interest in the computer science community [10], because methods for representing and reasoning about preferences are significant in AI applications, such as recommender systems [42], (group) product configuration [7, 16, 48], (group) planning [6, 44, 45, 47], (group) preference-based constraint satisfaction [2, 4, 8], and (group) preference-based query answering/information retrieval [15, 35, 36], just to name a few. In computer science, the study of preference aggregation has often been based on social choice theory [10]. In this theory, it is common to assume that agents' preferences are explicitly represented. Although this is reasonable when small sets of candidates are considered, this is not feasible when the voting domain is combinatorial, i.e., the set of candidates, or outcomes, is the Cartesian product of finite value domains for each of a set of features [25, 28].

Combinatorial domains contain an exponential number of outcomes in the number of features, and hence compact representations for combinatorial preferences are needed [25, 28]. CP-nets [3], which are a graph model, are among the most studied of these representations, as proven by a vast literature on them. In CP-nets, graph vertices represent features, and an edge from vertex *A* to vertex *B* models the influence of the value of feature *A* on the choice of the value of feature *B*. Intuitively, this model captures preferences like "if the rest of the dinner is the same, with a fish dish (*A*'s value), I prefer a white wine (*B*'s value)".

CP-nets were used to model preferences of *groups*, obtaining the *m*CP-nets [43]. This multi-agent model is a set of CP-nets, one for each agent. The preference semantics of *m*CP-nets is defined via voting schemes: Through their own individual CP-nets, each agent votes whether an outcome is preferred to another. Various voting schemes were proposed for *m*CP-nets [32, 43] and different voting schemes give rise to different dominance semantics for *m*CP-nets. In the voting schemes proposed for *m*CP-nets, the voting protocol adopted, i.e., the actual way in which votes are collected [13], is global voting [26] over the CP-nets of the single players. In this protocol, the outcomes of the vote are computed by having in input the whole CP-nets (see Section 7 for related works on different voting protocols over CP-nets). In the literature, a comparison between global voting and other protocols over CP-nets was explicitly asked for and stated to be highly promising [26]. However, global voting over CP-nets has not been thoroughly investigated as other protocols (see Section 7). In fact, unlike CP-nets, which were extensively analyzed, a precise complexity analysis of *m*CP-nets was missing for long time, as explicitly mentioned several times in the literature [26, 29–32]. An initial complexity analysis of voting tasks over *m*CP-nets was carried out only recently [33]. For example, deciding Pareto dominance was shown co-NP-complete, and deciding the existence of weak Condorcet winners was proven Σ_2^P-complete. The aim of this paper is to further explore the complexity of *m*CP-nets (and hence the complexity of global voting over CP-nets).

Contributions. In this paper, we focus on acyclic binary polynomially connected *m*CP-nets with standard CP-nets, i.e., the constituent CP-nets of an *m*CP-net rank all the features, and they are not partial CP-nets. Therefore, in this paper the dominance semantics of *m*CP-nets is precisely global voting over CP-nets. Our contributions are briefly as follows:

- ▷ Via a non-trivial reduction, we show the Σ_3^P-completeness of deciding the existence of max optimal outcomes;
- ▷ We prove that various voting tasks over *m*CP-nets known to be feasible in polynomial time are actually P-complete.

Furthermore, as a side result of our investigation:

- ▷ We define the P-complete Tʜ-CVP problem of deciding, given a Boolean circuit $\mathscr{C}$, a Boolean vector $\mathbf{x}$, and an integer k, whether the number of logical gates of $\mathscr{C}$ evaluating to *true* when $\mathbf{x}$ is given in input to $\mathscr{C}$ is at least k. This problem can be very useful in reductions showing P-hardness of problems involving counting tasks.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. Andre, S. Koenig (eds.), July 2018, Stockholm, Sweden

https://doi.org/doi

Organization of the paper. In Section 2, we provide an overview of our results. Preliminaries on CP-nets and *m*CP-nets are given in Section 3. We show P-completeness results over CP-nets and *m*CP-nets in Sections 4 and 5, respectively. In Section 6, we study max voting. We close our paper with a discussion on related works in Section 7, and we draw our conclusion in Section 8. For space constraints, we only provide proof intuitions for various results. Details will be provided in a forthcoming extended paper.

2 OVERVIEW AND DISCUSSION OF THE RESULTS

In Tables 1 and 2 there is a summary of our results. Definitions of the concepts mentioned in this section are given in the preliminaries.

We prove that deciding the existence of max optimal outcomes is Σ_3^P-complete. This supports that, in *m*CP-nets, max voting is computationally more demanding than majority voting, for which deciding the existence of optimal outcomes is Σ_2^P-complete [33]. The increase in the complexity is due to the need in max dominance of precisely counting the number of agents preferring an outcome to another, whereas this precision is not required in majority voting (majority and max dominance are NP-complete and Θ_2^P-complete, respectively [33, 34]). This complexity dissimilarity is carried over to the complexity of deciding the existence of optimal outcomes.

Besides this, we obtain several P-completeness results, which are quite interesting. Let us consider a group planning scenario [6], in which multiple autonomous agents have to agree upon a shared plan of actions to reach a goal that is preferred by the group as a whole. For example, a group of autonomous robots coordinating during the exploration of a remote area/planet. Each robot has a specific task to accomplish, and the group, as a whole, coordinates to achieve a common goal. Robots have their own specific preferences over a vast amount of variables/features emerging from the contingency of the situation to complete their individual tasks. However, their individual preferences have to be blended in all together, so that the course of action of an agent does not interfere with the tasks of the other agents and the mission is successful.

Managing huge amount of data could be tackled by using parallel algorithms. However, there are problems that, although solvable in polynomial time, are inherently sequential and hence do not benefit from highly parallel processing [21]. By saying that a problem L does not benefit from parallel processing, it is not meant that L does not admit parallel algorithms for its solution, but it means that parallel algorithms for L would not provide a speedup comparable with the increase in the amount of processing hardware available [21]. Decision problems of this type are the P-hard ones, which are often said to be non-parallelizable [21]. For this reason, P-complete problems are quite interesting, because they are in P, and hence they are regarded as "easy", but they are not parallelizable, which could be an issue when the input is of remarkable size.

P-time voting has attracted extensive consideration. However, to the best of our knowledge, P-hardness has not been carefully investigated so far in the computational social choice literature (see Section 7). In fact, it may well be the case that P-time voting schemes are actually P-hard, which would be a sign that these voting procedures would not scale up over huge input instances. Here we show that this is the case for some voting tasks over *m*CP-nets. Hence, the P-completeness results reported here, not only characterize more precisely the complexity of *m*CP-nets, but they also point out a significant issue, which is whether P-time voting schemes can benefit from parallel algorithms or not.

Table 1: Summary of the results for CP-nets.

Problem	Complexity
Feat-Value-Opt	P-complete
Same-Opt	P-complete
Rank-Bound	P-complete
Compare-Rank	P-complete

Table 2: Summary of the results for *m*CP-nets. *Membership shown in [33].

Problem	Complexity
Exists-Pareto-Optimum	P-complete*
Rank-Dominance	P-complete*
Is-Rank-Optimal	P-complete*
Is-Rank-Optimum	P-complete*
Exists-Rank-Optimum	P-complete*
Exists-Max-Optimal	Σ_3^P-complete*
Exists-Max-Optimum	in Σ_3^P

Observe that we show P-completeness already for the evaluation of the optimal outcome and the rank of outcomes on single CP-nets. Therefore, the P-completeness of preference aggregation based on these concepts derives from the P-hardness of the underlying concepts on single CP-nets. This points out that, to have parallelizable preference aggregation semantics, we need simpler semantics that are parallelizable (e.g., in LogSpace) already on single CP-nets.

3 PRELIMINARIES

CP-nets. A CP-net N is formally defined as a triple $\langle \mathcal{G}_N, Dom_N, (CPT_N^F)_{F \in \mathcal{F}_N} \rangle$, where $\mathcal{G}_N = \langle \mathcal{F}_N, \mathcal{E}_N \rangle$ is a directed graph whose vertices $\mathcal{F}_N$ represent the features of the combinatorial domain, Dom_N is a function, and $(CPT_N^F)_{F \in \mathcal{F}_N}$ is a family of functions. For a feature F, Dom_N associates a (value) domain $Dom_N(F)$ with F, while CPT_N^F is the so called "CP table" of F.

The domain of a feature F is the set of values that F may have in the outcomes. Here, we assume features to be *binary*, i.e., each feature's domain contains two values. We denote by $\overline{f}$ and f the two values of F, called the overlined and the non-overlined value (of F), respectively. For a feature set $\mathcal{S} \subseteq \mathcal{F}_N$, $Dom_N(\mathcal{S}) = \times_{F \in \mathcal{S}} Dom_N(F)$. An outcome is an element of the set $O_N = Dom_N(\mathcal{F}_N)$. For a feature $F \in \mathcal{F}_N$ and an outcome α, $\alpha[F]$ is F's value in α. For a feature set $\mathcal{S} \subseteq \mathcal{F}_N$ and an outcome α, $\alpha[\mathcal{S}]$ is the projection of α over $\mathcal{S}$.

CP tables encode preferences over feature values. The CP table of feature F has a row for *any* possible combination of values of *all* the parent features of F in $\mathcal{G}_N$; in each row there is a total order over $Dom_N(F)$. This order encodes agent's preferences for F's values when specific values of F's parents are considered: $\overline{f} \succ f$ denotes $\overline{f}$ being preferred to f. If F has no parents, its CP table has only one row with a total order over $Dom_N(F)$. Note that indifferences between features values are not admitted in (classical) CP-nets. When we will outline CP tables in figures, we will use a "logic notation" to identify, for which values of the parents of the features, a particular CP table row has to be considered. Although this is the notion which generalized propositional CP-nets are based on [17], in this paper it is used only for notational convenience. Here we always assume that CP tables are extensively and explicitly represented in the input instances. We denote by $\|N\|$ the size of CP-net N, i.e., the space in terms of bits required to represent the whole net N (which includes, features, links, feature domains, and CP tables).

CP-nets' preference semantics is based on "improving flips". Let F be a feature, and let α, β be two outcomes differing *only* on F's value. Flipping F from $\alpha[F]$ to $\beta[F]$ is an *improving flip* (of F in N) iff, in the row of F's CP table associated with the values in α of the parents of F, $\beta[F] \succ \alpha[F]$. Outcome β is preferred to α, or β *dominates* α (in N), denoted $\beta \succ_N \alpha$, iff there is a sequence of improving flips from α to β, otherwise β does not dominate α, denoted $\beta \not\succ_N \alpha$; β and α are *incomparable*, denoted $\beta \bowtie_N \alpha$, iff $\beta \not\succ_N \alpha$ and $\alpha \not\succ_N \beta$. Observe that, since there are no indifferences between features values in (classical) CP-nets, for any two outcomes α and β, either one dominates the other, or they are incomparable.

A CP-net N is *binary* iff all its features are binary; N is *singly connected* iff, for any two features G and F of N, there is at most one path from G to F in $\mathcal{G}_N$. A class $\mathscr{F}$ of CP-nets is *polynomially connected*, iff there exists a polynomial p such that, for any CP-net $N \in \mathscr{F}$ and for any two features G and F of N, there are at most $p(\|N\|)$ distinct paths from G to F in $\mathcal{G}_N$. A CP-net N is *acyclic* iff $\mathcal{G}_N$ is acyclic. Acyclic CP-nets N have a unique optimum outcome o_N, dominating all others, that can be computed in polynomial time [3]. The rank of an outcome α in a CP-net N, $Rank_N(\alpha)$, is the length of the shortest improving flipping sequence from α to o_N [43]. Unless stated otherwise, we consider acyclic binary CP-nets.

***m*CP-nets.** An *m*CP-net is a set of m CP-nets defined over the same set of features having, in turn, the same domain. The "m" of an *m*CP-net is the agents' number, so a 3CP-net is an *m*CP-net with $m = 3$. Originally, *partial* CP-nets were allowed to be constituent of *m*CP-nets [43]. We assume only standard CP-nets to be part of *m*CP-nets, and we do not assume CP-nets to be $\mathcal{O}$-legal (i.e., we de not assume that the CP-nets of an *m*CP-net have a common topological order of the features). *m*CP-nets' semantics is based on voting. Let $\mathcal{M} = \langle N_1, \dots, N_m \rangle$ be an *m*CP-net, and let α, β be two outcomes. We define the sets $S^{\succ}_{\mathcal{M}}(\alpha,\beta) = \{i \mid \alpha \succ_{N_i} \beta\}$, $S^{\prec}_{\mathcal{M}}(\alpha,\beta) = \{i \mid \alpha \prec_{N_i} \beta\}$, and $S^{\bowtie}_{\mathcal{M}}(\alpha,\beta) = \{i \mid \alpha \bowtie_{N_i} \beta\}$, as the sets of agents preferring α to β, preferring β to α, and for which α and β are incomparable, respectively. $Rank_{\mathcal{M}}(\alpha) = \sum_{1 \le i \le m} Rank_{N_i}(\alpha)$ [43]. The following are the dominance semantics considered:

Pareto: β *Pareto dominates* α, denoted by $\beta \succ^p_{\mathcal{M}} \alpha$, iff *all* the agents of $\mathcal{M}$ prefer β to α, i.e., $|S^{\succ}_{\mathcal{M}}(\beta,\alpha)| = m$.

Majority: β *majority dominates* α, denoted by $\beta \succ^{maj}_{\mathcal{M}} \alpha$, iff the *majority* of the agents of $\mathcal{M}$ prefers β to α, i.e., $|S^{\succ}_{\mathcal{M}}(\beta,\alpha)| > |S^{\prec}_{\mathcal{M}}(\beta,\alpha)| + |S^{\bowtie}_{\mathcal{M}}(\beta,\alpha)|$.

Max: β *max dominates* α, denoted by $\beta \succ^{max}_{\mathcal{M}} \alpha$, iff the group of the agents of $\mathcal{M}$ preferring β to α is the *biggest*, i.e., $|S^{\succ}_{\mathcal{M}}(\beta,\alpha)| > \max(|S^{\prec}_{\mathcal{M}}(\beta,\alpha)|, |S^{\bowtie}_{\mathcal{M}}(\beta,\alpha)|)$.

Rank: β *rank dominates* α, denoted by $\beta \succ^r_{\mathcal{M}} \alpha$, iff $Rank_{\mathcal{M}}(\beta) < Rank_{\mathcal{M}}(\alpha)$.

For a voting scheme s, α is s optimal in $\mathcal{M}$ iff $\beta \not\succ^s_{\mathcal{M}} \alpha$ for all $\beta \neq \alpha$, whereas α is s optimum in $\mathcal{M}$ iff $\alpha \succ^s_{\mathcal{M}} \beta$ for all $\beta \neq \alpha$. Optimum outcomes, if they exist, are unique.

An *m*CP-net is acyclic, binary, and singly connected, iff all its CP-nets are acyclic, binary, and singly connected, respectively. A class $\mathscr{F}$ of *m*CP-nets is polynomially connected, iff the set of CP-nets of the *m*CP-nets in $\mathscr{F}$ is polynomially connected. Unless stated otherwise, the considered *m*CP-nets are acyclic, binary, and belong to polynomially connected classes of *m*CP-nets.

Complexity Classes. We assume basic knowledge of computational complexity and of the polynomial hierarchy (PH); see [22, 40] for an overview. A language L is P-hard iff, for all languages L' in P, there is a log-space reduction from L' to L. A language L is P-complete iff L is in P and is P-hard.

4 P-COMPLETE PROBLEMS ON CP-NETS

In this section, we show the P-completeness of various tasks over CP-nets. To prove these results, we will exploit the P-completeness of the classical CVP problem defined below.

In the Circuit Value Problem (CVP) [24], for a Boolean circuit $\mathscr{C}$ and a Boolean vector $\mathbf{x}$, we have to decide whether $\mathscr{C}$'s output is *true* when receiving $\mathbf{x}$ as input. In the literature, various ways to represent circuits were illustrated. Here, we use a representation that is a mix of those in [24, 39]. A circuit $\mathscr{C} = \{C_1, \dots, C_m\}$ is a sequence of logic gates, which are represented through formulas: (i) if $C_i = x_j$, C_i is an input gate fed with the j^{th} input bit; (ii) if $C_i = C_j \wedge C_k$ (resp., $C_i = C_j \vee C_k$), C_i is an AND (resp., OR) gate, whose inputs are the outputs of C_j and C_k (with $j, k < i$); (iii) if $C_i = \neg C_j$, C_i is a NOT gate, whose input is the output of C_j (with $j < i$). The Boolean values of gates C_i when $\mathbf{x}$ is given in input to $\mathscr{C}$, denoted by $v_{\mathscr{C}}(C_i, \mathbf{x})$, are defined in the natural way.

We assume that the problem CVP is defined as in [21]. A CVP instance $\mathcal{I} = \langle \mathscr{C}, \mathbf{x}, C_{out} \rangle$, where $\mathscr{C} = \{C_1, \dots, C_m\}$ is a circuit, $\mathbf{x} = \{x_1, \dots, x_n\}$ is a vector, and $C_{out} \in \mathscr{C}$ is the output gate, is a 'yes'-instance iff $v_{\mathscr{C}}(C_{out}, \mathbf{x}) = true$. CVP is known to be P-complete and its hardness holds even if various restrictions are issued over the circuit structure and even if the output is fixed to be C_m [21, 24, 39].

For the following results, we need CP-nets mimicking the behavior of circuits when specific vectors are given in input. Let $\mathscr{C} = \{C_1, \dots, C_m\}$ be a circuit and let $\mathbf{x} = \{x_1, \dots, x_n\}$ be an input vector. The CP-net $N(\mathscr{C}, \mathbf{x})$, defined from $\mathscr{C}$ and $\mathbf{x}$, is as follows. For each gate $C_i \in \mathscr{C}$, there is a feature $D_i \in \mathcal{F}_{N(\mathscr{C},\mathbf{x})}$, and D_i's domain is $\{d_i, \overline{d_i}\}$. The intuition of the transformation is that values $\overline{d_i}$ and d_i of D_i are associated with gate C_i evaluating to *true* and *false*, respectively, when $\mathbf{x}$ is given in input to $\mathscr{C}$.

- If C_i is an input gate with $C_i = x_j$, there is no edge entering in D_i; if $x_j = true$, $\overline{d_i} \succ d_i$; if $x_j = false$, $d_i \succ \overline{d_i}$.
- If C_i is an AND (resp., OR) gate, with $C_i = C_j \wedge C_k$ (resp., $C_i = C_j \vee C_k$), then there are two edges entering in D_i, one from D_j and one from D_k. If $C_i = C_j \wedge C_k$, for D_i, $\overline{d_i} \succ d_i$ iff both D_j and D_k have overlined values. If $C_i = C_j \vee C_k$, for D_i, $\overline{d_i} \succ d_i$ iff D_j or D_k has an overlined value.

- If C_i is a NOT gate with $C_i = \neg C_j$, there is an edge from D_j to D_i; for D_i, $d_i \succ \overline{d_i}$ if D_j has value $\overline{d_j}$, $\overline{d_i} \succ d_i$ otherwise.

Observe that $N(\mathscr{C}, \mathbf{x})$ is binary, acyclic, and can be computed in logarithmic space from $\mathscr{C}$ and $\mathbf{x}$ (because the indegree of each feature is at most 2, i.e., it is bounded by a constant, and hence the number of rows in the CP tables of $N(\mathscr{C}, \mathbf{x})$ is bounded by a constant). Therefore, all the hardness results shown here hold even on acyclic binary (m)CP-nets with indegree 2. Via induction on the gates' levels in $\mathscr{C}$, it can be shown that, in $N(\mathscr{C}, \mathbf{x})$, a feature D_i has value $\overline{d_i}$ in the optimum outcome iff $v_{\mathscr{C}}(C_i, \mathbf{x}) = true$.

LEMMA 4.1. *Let $\mathscr{C} = \{C_1, \ldots, C_m\}$ be a circuit, and let* $\mathbf{x}$ *be an input vector. For any gate C_i, $v_{\mathscr{C}}(C_i, \mathbf{x}) = true$ iff $o_{N(\mathscr{C},\mathbf{x})}[D_i] = \overline{d_i}$.*

From this key property follows the P-hardness of the problem FEAT-VALUE-OPT: For a CP-net N, a feature $F \in \mathcal{F}_N$, and a value $v \in Dom_N(F)$ for F, decide whether the value of F in the optimum outcome of N is v, i.e., $o_N[F] = v$.

THEOREM 4.2. *FEAT-VALUE-OPT is* P-complete.

Consider now the problem SAME-OPT: Given two (different) CP-nets N_1 and N_2 defined over the same set of features, which, in turn, have the same domain in the two nets, decide whether the optimum outcome of N_1 equals the optimum outcome of N_2, i.e., $o_{N_1} = o_{N_2}$. We can show that SAME-OPT is P-complete. The intuition behind the P-hardness proof (a reduction from CVP) is to encode the same circuit in N_1 and N_2 with an additional feature O. In N_1, O is attached to the output gate and replicates its value, instead, in N_2, O has a specific preferred value, say $\overline{o}$. In this case, $o_{N_1} = o_{N_2}$ iff the circuit outputs *true*.

THEOREM 4.3. *SAME-OPT is* P-complete.

Let us denote by $TG(\mathscr{C}, \mathbf{x})$ the number of $\mathscr{C}$'s gates C_i such that $v_{\mathscr{C}}(C_i, \mathbf{x}) = true$. Consider this new problem TH-CVP (Threshold CVP): Given a Boolean circuit $\mathscr{C}$, an input vector $\mathbf{x}$, and an integer k, decide whether $TG(\mathscr{C}, \mathbf{x}) \leq k$. The P-hardness can be shown via a reduction from CVP. The idea behind the proof is to modify the original circuit by attaching to the original output gate a cascade of (enough) gates replicating the output's value. Then, when the output is *true*, a number of gates greater than k evaluates to *true*.

THEOREM 4.4. *TH-CVP is* P-complete. *Hardness holds even if the threshold number k is such that $k < \lfloor |\mathscr{C}|/2 \rfloor$.*

PROOF. TH-CVP is in P, because gates' values can be evaluated in polynomial time [21, 24], and then we can count those evaluating to *true* and compare the count with k (in polynomial time).

Hardness can be shown via a reduction from CVP. Consider the following reduction transforming an instance $\langle \mathscr{C}, \mathbf{x}, C_{out} \rangle$ of CVP, where $\mathscr{C} = \{C_1, \ldots, C_m\}$, into an instance $\langle \mathscr{C}', \mathbf{x}', k \rangle$ of TH-CVP. $\mathscr{C}'$ consists of $2m$ gates, whose first m gates are identical (for function and wiring) to those of $\mathscr{C}$. The remaining m gates of $\mathscr{C}'$ essentially replicate the value of $C'_{out} = C_{out}$. More formally, $C'_{m+1} = C'_{out} \wedge C'_{out}$, and, for all $2 \leq i \leq m$, $C'_{m+i} = C'_{m+i-1} \wedge C'_{m+i-1}$. The input vector $\mathbf{x}'$ equals $\mathbf{x}$, and $k = m - 1$. Clearly, the reduction can be computed in logarithmic space. Observe that $k = m-1 < \lfloor 2m/2 \rfloor$, where $2m = |\mathscr{C}'|$. Given that P is closed under complement, in this case we assume that 'yes'-instances of CVP are those in which the output of the circuit is *false*.

($\Rightarrow$) If $\langle \mathscr{C}, \mathbf{x} \rangle$ is a 'yes'-instance of CVP, i.e., $v_{\mathscr{C}}(C_{out}, \mathbf{x}) = false$, then $v_{\mathscr{C}'}(C'_{out}, \mathbf{x}') = v_{\mathscr{C}'}(C'_{m+1}, \mathbf{x}') = \cdots = v_{\mathscr{C}'}(C'_{2m}, \mathbf{x}') = false$. Hence, $TG(C', \mathbf{x}') \leq |\mathscr{C}'| - (m+1) = m-1 = k$, and thus $\langle \mathscr{C}', \mathbf{x}', k \rangle$ is a 'yes'-instance of TH-CVP as well.

($\Leftarrow$) On the other hand, if $\langle \mathscr{C}, \mathbf{x} \rangle$ is a 'no'-instance of CVP, i.e., $v_{\mathscr{C}}(C_{out}, \mathbf{x}) = true$, then $v_{\mathscr{C}'}(C'_{out}, \mathbf{x}') = v_{\mathscr{C}'}(C'_{m+1}, \mathbf{x}') = \cdots = v_{\mathscr{C}'}(C'_{2m}, \mathbf{x}') = true$. Hence, $TG(C', \mathbf{x}') \geq m + 1 > m - 1 = k$, and thus $\langle \mathscr{C}', \mathbf{x}', k \rangle$ is a 'no'-instance of TH-CVP as well. □

Problem RANK-BOUND is, for a CP-net N, an outcome $\alpha \in O_N$, and an integer k, decide whether $Rank_N(\alpha) \leq k$. For acyclic CP-nets,

$$Rank_N(\alpha) = |\{F \mid F \in \mathcal{F}_N \wedge \alpha[F] \neq o_N[F]\}|, \quad (1)$$

i.e., α's rank in N is the number of features whose value in α differs from its value in o_N [33]. RANK-BOUND's P-hardness follows from Lemma 4.1 and Equation (1), by which the number of overlined values in the optimum outcome of $N(\mathscr{C}, \mathbf{x})$ equals $TG(\mathscr{C}, \mathbf{x})$.

THEOREM 4.5. *RANK-BOUND is* P-complete.

Problem COMPARE-RANK is, for a CP-net N and two outcomes $\alpha, \beta \in O_N$, decide whether $Rank_N(\beta) < Rank_N(\alpha)$. Its P-hardness can be shown from FEAT-VALUE-OPT. In fact, by Equation (1), for a CP-net N, two outcomes α and β differing only on the value of a feature F are such that $Rank_N(\beta) < Rank_N(\alpha)$ iff $\beta[F]$ is $o_N[F]$.

THEOREM 4.6. *COMPARE-RANK is* P-complete.

PROOF. Membership in P follows from the fact that computing outcome ranks in acyclic CP-nets is feasible in polynomial time [33], and then we can compare them (in polynomial time).

Hardness can be shown via a reduction from FEAT-VALUE-OPT. Consider the reduction transforming an instance $\langle N, F, v \rangle$ of FEAT-VALUE-OPT into the instance $\langle N', \alpha, \beta \rangle$ of COMPARE-RANK as follows (assume w.l.o.g. that $v = f$): $N' = N$, α and β are the outcomes assigning non-overlined values to all features but F, and $\alpha[F] = \overline{f}$, while $\beta[F] = f$. By Equation (1), and since α and β differ only on the value assigned to feature F, there is a difference of exactly 1 between the rank of the two outcomes, i.e., $|Rank_{N'}(\beta) - Rank_{N'}(\alpha)| = 1$.

($\Rightarrow$) If $\langle N, F, v \rangle$ is a 'yes'-instance of FEAT-VALUE-OPT, $o_N[F] = f = v$. Hence, $Rank_{N'}(\beta) < Rank_{N'}(\alpha)$.

($\Leftarrow$) If $\langle N, F, v \rangle$ is a 'no'-instance of FEAT-VALUE-OPT, $o_N[F] = \overline{f} \neq v$. Hence, $Rank_{N'}(\alpha) < Rank_{N'}(\beta)$. □

5 P-COMPLETE PROBLEMS ON *m*CP-NETS

First, we focus on a Pareto voting task. Consider the problem EXISTS-PARETO-OPTIMUM: Given an mCP-net $\mathcal{M}$, decide whether $\mathcal{M}$ has a Pareto optimum outcome. Acyclic mCP-nets have a Pareto optimum outcome iff all their individual CP-nets have the very same individual optimum outcome [33]. By this, the P-hardness of EXISTS-PARETO-OPTIMUM follows from the P-hardness of SAME-OPT.

THEOREM 5.1. *EXISTS-PARETO-OPTIMUM is* P-hard. *Hardness holds even on* 2*CP-nets.*

Since EXISTS-PARETO-OPTIMUM is also in P [33], it is P-complete.

In the rest of the section we prove the hardness results for rank voting over mCP-nets. Consider the problem RANK-DOMINANCE: Given an mCP-net $\mathcal{M}$ and two outcomes $\alpha, \beta \in O_{\mathcal{M}}$, decide whether

$\beta \succ^r_{\mathcal{M}} \alpha$, i.e., decide whether $Rank_{\mathcal{M}}(\beta) < Rank_{\mathcal{M}}(\alpha)$. We remind to the reader that, for an mCP-net $\mathcal{M} = \langle \mathcal{N}_1, \dots, \mathcal{N}_m \rangle$, $Rank_{\mathcal{M}}(\alpha) = \sum_{1 \le i \le m} Rank_{N_i}(\alpha)$. Hence, Rank-Dominance's hardness follows from the P-hardness of Compare-Rank on CP-nets.

Theorem 5.2. *Rank-Dominance is* P-hard. *Hardness holds even on* 1*CP-nets.*

Since Rank-Dominance is also in P [33], it is P-complete.

Consider now problems Is-Rank-Optimal and Is-Rank-Optimum: Given an mCP-net $\mathcal{M}$ and an outcome $\alpha \in O_{\mathcal{M}}$, decide whether α is rank optimal (resp., optimum) in $\mathcal{M}$. We recall some definitions from [33]. A value v of a feature F is average optimal iff v is in $\arg\min_{v \in Dom_{\mathcal{M}}(F)} |\{i \mid 1 \le i \le m \wedge v \ne o_{N_i}[F]\}|$, i.e., iff v minimizes the number of agents i for which v is different from the value of F in the optimum outcome of agent i's CP-net. An outcome α is average optimal iff, for each feature F, $\alpha[F]$ is average optimal.

An outcome is rank optimal iff it is average optimal [33]. Since mCP-nets have always average optimal outcomes, mCP-nets have always rank optimal outcomes.[1] Computing average optimal outcomes of mCP-nets is feasible in polynomial time (we just need to compute the individual optimal outcomes to perform the counting operations). Observe that, if an mCP-net $\mathcal{M}$ has two average optimal outcomes, then $\mathcal{M}$ has two rank optimal outcomes, and hence $\mathcal{M}$ has no rank optimum outcome, because different rank optimal outcomes do not rank dominate each other (which is required to be rank optimum). Thus, binary mCP-nets with an odd number of CP-nets, since they have a unique average optimal outcome, have only one rank optimal outcome which is also rank optimum.

In the reductions for Is-Rank-Optimal and Is-Rank-Optimum, we will use a CP-net that is designed to have a desired optimum outcome. Let $\mathcal{S}$ be a set of binary features, and let $\alpha \in Dom(\mathcal{S})$ be an outcome. The "direct" net $D(\alpha)$ has as features the set $\mathcal{S}$ and has no edge. The CP table of feature F is $f \succ \overline{f}$, if $\alpha[F] = f$; on the other hand the CP table of feature F is $\overline{f} \succ f$, if $\alpha[F] = \overline{f}$.

Thanks to direct nets, P-hardness of Is-Rank-Optimal and Is-Rank-Optimum can be shown from Feat-Value-Opt. In fact, in an mCP-net $\langle N, N', N'' \rangle$, where N' and N'' are designed to have optimum outcomes differing only on the value of a feature F, $o_{N'}$ is average optimal iff $o_N[F]$ is a specific value.

Theorem 5.3. *Is-Rank-Optimal and Is-Rank-Optimum are* P-hard. *Hardness holds even on* 3*CP-nets.*

Proof. Hardness can be shown via a reduction from Feat-Value-Opt. Consider the reduction transforming an instance $\langle N, F, v \rangle$ of Feat-Value-Opt into the instance $\langle \mathcal{M}, \alpha \rangle$ of Is-Rank-Optimal (resp., Is-Rank-Optimum) as follows (assume w.l.o.g. that $v = f$): $\mathcal{M} = \langle N_1, N_2, N_3 \rangle$ is a 3CP-net, where $N_1 = N$, $N_2 = D(\alpha)$, with α being an outcome defined over the features in N and assigning non-overlined values to all features, and $N_3 = D(\beta)$, with β being almost equal to α, except for $\beta[F] = \overline{f}$.

Observe that the value $\alpha[G]$ is the average optimal value for all features $G \ne F$, because, for all features $G \ne F$, $\alpha[G] = \beta[G]$. Since $\alpha[F] = f$ and $\beta[F] = \overline{f}$, α is rank optimal in $\mathcal{M}$ iff $o_N[F] = f = v$. To conclude, since $\mathcal{M}$ contains an odd number of CP-nets, α is rank optimum in $\mathcal{M}$ iff α is rank optimal in $\mathcal{M}$ (see above). ◻

[1] A different proof of mCP-nets always having rank optimal outcomes is in [43].

Is-Rank-Optimal and Is-Rank-Optimum are in P [33], hence they are P-complete.

The P-hardness of Exists-Rank-Optimum can be shown from Feat-Value-Opt via direct nets, as well. In an mCP-net $\langle N, N, N', N'' \rangle$, where N' and N'' have optimum outcomes being equal only on the value of a feature F, o_N is the unique average optimal outcome (and so, rank optimum) iff $o_N[F]$ is a specific value.

Theorem 5.4. *Exists-Rank-Optimum is* P-hard. *Hardness holds even on* 4*CP-nets.*

Proof. Hardness can be shown via a reduction from Feat-Value-Opt. Consider the reduction transforming an instance $\langle N, F, v \rangle$ of Feat-Value-Opt into the instance $\langle \mathcal{M} \rangle$ of Exists-Rank-Optimum as follows (assume w.l.o.g. that $v = f$): $\mathcal{M} = \langle N_1, N_2, N_3, N_4 \rangle$ is a 4CP-net, where $N_1 = N_2 = N$, $N_3 = D(\alpha)$, with α being an outcome defined over the features in N and assigning non-overlined values to all features, and $N_4 = D(\beta)$, with β assigning overlined values to all features but F for which $\beta[F] = f$.

$\mathcal{M}$ has a rank optimum outcome iff $\mathcal{M}$ has a unique average optimal outcome (see above). For any feature $G \ne F$, since $N_1[G] = N_2[G]$, $N_3[G] = g$, and $N_4[G] = \overline{g}$, the average optimal value is unique and it is $o_N[G]$. Therefore, $\mathcal{M}$ has a unique average optimal outcome iff the average optimal value for feature F is unique in $\mathcal{M}$.

($\Rightarrow$) If $\langle N, F, v \rangle$ is a 'yes'-instance of Feat-Value-Opt, $o_N[F] = f = v$. Hence, $o_{N_1}[F] = o_{N_2}[F] = o_{N_3}[F] = o_{N_4}[F] = f$, and f is the unique average optimal value for F in $\mathcal{M}$. This implies that $\mathcal{M}$ has a unique average optimal outcome which is rank optimal and optimum, and thus $\mathcal{M}$ has a rank optimum outcome.

($\Leftarrow$) If $\langle N, F, v \rangle$ is a 'no'-instance of Feat-Value-Opt, $o_N[F] = \overline{f} \ne v$. Hence, $o_{N_1}[F] = o_{N_2}[F] = \overline{f}$ and $o_{N_3}[F] = o_{N_4}[F] = f$, and both f and $\overline{f}$ are average optimal values for F in $\mathcal{M}$. This implies that $\mathcal{M}$ has two distinct average optimal outcomes, which are rank optimal, and thus $\mathcal{M}$ has no rank optimum outcome. ◻

Exists-Rank-Optimum is in P [33], hence it is P-complete.

6 MAX VOTING

We now show the Σ^P_3-completeness of deciding the existence of max optimal outcomes in mCP-nets. To prove this we need an involved reduction, for which we will give intuitions on the purpose of the key pieces. The starting problem for the reduction is deciding the validity of quantified Boolean formulas $\Phi = (\exists X)(\forall Y)(\exists Z)\phi(X, Y, Z)$, where X, Y, and Z, are three disjoint sets of Boolean variables, and $\phi(X, Y, Z)$ is a non-quantified Boolean formula. This problem is Σ^P_3-complete, and it is Σ^P_3-hard even if ϕ is in 3CNF [49, 50].

In the reduction, we use as building pieces two CP-nets introduced in [33]: An interconnecting net $H_C(m)$ "propagating the information" that all the m features of a set have been flipped to their overlined value; and a net $F_s(\phi)$ encoding the satisfiability of a 3CNF non-quantified Boolean formula ϕ. The other building piece that we need is the direct net introduced in Section 5. For notational convenience, in this section we define direct nets as follows: the direct net $D(\mathcal{A}\overline{\mathcal{B}})$ is defined over feature sets $\mathcal{A}$ and $\mathcal{B}$, for features of $\mathcal{A}$ the non-overlined value is preferred, while for features of $\mathcal{B}$

To transform the validity problem into Exists-Max-Optimal, we need to encode Boolean assignments into outcomes of a suitable

mCP-net. We use three sets of "variable features" $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, associated with the sets of Boolean variables X, Y, and Z, respectively. In particular, $\mathcal{X} = \{X_i^T, X_i^F \mid x_i \in X\}$, $\mathcal{Y} = \{Y_i^T, Y_i^F \mid y_i \in Y\}$, and $\mathcal{Z} = \{Z_i^T, Z_i^F \mid z_i \in Z\}$. We use the following association for the assignments. If we focus on the variables in X, for a (partial) Boolean assignment σ_X over X, an outcome α_{σ_X} encoding σ_X over the features set $\mathcal{X}$ is such that if $\sigma_X[x_i] = \mathit{true}$ then $\alpha_{\sigma_X}[X_i^T X_i^F] = \overline{x_i^T} x_i^F$, if $\sigma_X[x_i] = \mathit{false}$ then $\alpha_{\sigma_X}[X_i^T X_i^F] = x_i^T \overline{x_i^F}$, and if $\sigma_X[x_i]$ is undefined then $\alpha_{\sigma_X}[X_i^T X_i^F] = x_i^T x_i^F$. (An outcome α_{σ_X} with $\alpha_{\sigma_X}[X_i^T X_i^F] = \overline{x_i^T x_i^F}$ will be dealt with so that it will not give issues in the reduction.) We use a similar encoding for the variable sets Y and Z over feature sets $\mathcal{Y}$ and $\mathcal{Z}$, respectively.

The idea of the reduction is to design an mCP-net such that specific outcomes encoding assignments for variables X are max optimal iff the encoded assignments are witnesses of the validity of the quantified formula. All other outcomes that are not in the specific form encoding assignments for X have to be max dominated (and hence not max optimal). Besides the features associated with the Boolean variables, there are various other features supporting the correctness of the reduction. Two of these additional features are U_1 and U_2, which are features belonging to the net $F_s(\phi)$ (see, [33]).

In particular, the principles of our reduction are:

(a) For an assignment σ_X on X, the associated outcome is $\overline{\beta}_{\sigma_X}$, where σ_X is encoded over the features $\mathcal{X}$, $\overline{\beta}_{\sigma_X}[U_1 U_2] = \overline{u_1 u_2}$, and all other features have non-overlined values.

(b) Any outcome in a form different from the one described in Principle (a) is max dominated.

(c) For a pair of assignments σ_X and σ_Y on X and Y, respectively, the associated outcome is $\beta_{\sigma_X,\sigma_Y}$, where σ_X and σ_Y are encoded over the features $\mathcal{X}$ and $\mathcal{Y}$, respectively, $\beta_{\sigma_X,\sigma_Y}[U_1 U_2] = u_1 u_2$, and all other features have non-overlined values.

(d) Any outcome in a form different from the one of Principle (c) does not max dominate an outcome of Principle (a).

(e) If $\overline{\beta}_{\sigma_X}$ and $\beta_{\sigma'_X,\sigma_Y}$ are two outcomes such that $\sigma_X \neq \sigma'_X$, then $\beta_{\sigma'_X,\sigma_Y}$ does not max dominate $\overline{\beta}_{\sigma_X}$. This imposes that $\overline{\beta}_{\sigma_X}$ might be max dominated only by an outcome encoding the very same assignment for X of $\overline{\beta}_{\sigma_X}$.

(f) If $\overline{\beta}_{\sigma_X}$ and $\beta_{\sigma_X,\sigma_Y}$ are two outcomes, then $\beta_{\sigma_X,\sigma_Y}$ max dominates $\overline{\beta}_{\sigma_X}$ iff $\phi(X/\sigma_X, Y/\sigma_Y, Z)$ is not satisfiable.

A reduction following the principles above has the property that only an outcome in the form $\overline{\beta}_{\sigma_X}$ can be max optimal, and $\overline{\beta}_{\sigma_X}$ is max optimal iff σ_X is an assignment such that $(\forall Y)(\exists Z)\phi(X/\sigma_X, Y, Z)$ is valid, i.e., iff σ_X is a witness of the validity of the quantified formula Φ. Therefore, an mCP-net obtained via this reduction has a max optimal outcome iff the quantified formula is valid.

Let us now see the reduction. Let $\Phi = (\exists X)(\forall Y)(\exists Z)\phi(X, Y, Z)$ be a quantified formula. From $\phi(X, Y, Z)$ we define the 8CP-net $\mathcal{M}(\phi) = \langle N_1, \dots, N_8 \rangle$ as follows.

The features of $\mathcal{M}(\phi)$ are:

- The features of a net $F_s(\phi)$ (see, [33]) in which we distinguish three variable feature sets $\mathcal{X} = \{X_i^T, X_i^F \mid x_i \in X\}$, $\mathcal{Y} = \{Y_i^T, Y_i^F \mid y_i \in Y\}$, and $\mathcal{Z} = \{Z_i^T, Z_i^F \mid z_i \in Z\}$ ($\mathcal{P}$ and $\mathcal{D}$ are the literal and clause feature sets, respectively, $\mathcal{A}$ is the set of features of the interconnecting net embedded in $F_s(\phi)$ and A is the apex of the interconnecting net);
- Features $\mathcal{Y}' = \{Y'_i \mid y_i \in Y\}$, $\mathcal{Y}'' = \{Y''_i \mid y_i \in Y\}$;
- Features in set $\mathcal{B}$, which are the features B_i of an interconnecting net $H_C(|\mathcal{Y}'|)$ and its apex is feature B (features B_i and features A_i of the interconnecting net $H_C(m)$ embedded in $F_s(\phi)$ are distinct).

To sum up, $\mathcal{M}(\phi)$'s features are $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Y}' \cup \mathcal{Y}'' \cup \mathcal{Z} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B} \cup \{U_1, U_2\}$ (U_1 and U_2 are features of $F_s(\phi)$).

The CP-nets of $\mathcal{M}(\phi)$ are (we do not report the direct nets in the figures with the schematic representations of these CP-nets):

- $\boldsymbol{N_1}$ is composed by a net $F_s(\phi)$ (for a schematic representation of this net see [33]), in which we distinguish three variable feature sets $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$, and a direct net $D(\mathcal{Y}'\mathcal{Y}''\mathcal{B})$.

This net supports Principle (f). Indeed, we need a CP-net mimicking a Boolean formula to encode the satisfiability of ϕ.

- $\boldsymbol{N_2}$ has, for each $x_i \in X$, the link (X_i^T, X_i^F), and a net $D(\mathcal{Y}\mathcal{Y}'\mathcal{Y}''\mathcal{Z}\mathcal{P}\mathcal{D}\mathcal{A}\mathcal{B}\overline{\{U_1, U_2\}})$. The other CP tables are: for X_i^T, $x_i^T \succ \overline{x_i^T}$; for X_i^F, $x_i^F \succ \overline{x_i^F}$ iff X_i^T has value $\overline{x_i^T}$.
- $\boldsymbol{N_3}$ is similar to N_2 with roles of X_i^T and X_i^F exchanged.

The purpose of these two nets is achieved in conjunction with nets N_6 and N_7 below. Their purpose is supporting Principle (e). For nets N_2 and N_6, their preferences restricted over $\{X_i^T, X_i^F\}$ are $\overline{x_i^T x_i^F} \prec \overline{x_i^T} x_i^F \prec x_i^T x_i^F \prec x_i^T \overline{x_i^F}$; while, for nets N_3 and N_7, their preferences restricted over $\{X_i^T, X_i^F\}$ are $\overline{x_i^T x_i^F} \prec x_i^T \overline{x_i^F} \prec x_i^T x_i^F \prec \overline{x_i^T} x_i^F$. Therefore, for an outcome $\overline{\beta}_{\sigma_X}$, if we focus on a pair of features $\{X_i^T, X_i^F\}$, some of the nets prefer to change the values of $\{X_i^T, X_i^F\}$ in a specific way, and the other nets prefer something different. Hence, intuitively, there will never be a group of agents big enough such that $\overline{\beta}_{\sigma_X}$ can be max dominated by an outcome $\beta_{\sigma'_X,\sigma_Y}$ with $\sigma_X \neq \sigma'_X$. Only outcomes $\beta_{\sigma'_X,\sigma_Y}$ with $\sigma_X = \sigma'_X$ may max dominate $\overline{\beta}_{\sigma_X}$, because there will not be contrasting preferences among the agents (this is Principle (e)).

- $\boldsymbol{N_4}$ (see Figure 1) has, for each $y_i \in Y$, the links (U_1, Y_i^T), (U_1, Y_i^F), (Y_i^T, Y'_i), (Y_i^F, Y'_i), (Y_i^T, Y''_i), (Y_i^F, Y''_i), (Y''_i, Y'_i); the links of a net $H_C(|\mathcal{Y}'|)$ over features $\mathcal{B}$ and connected to features $\mathcal{Y}'$, with apex B linked to U_2; and the link (U_1, U_2). There is the direct net $D(\overline{\mathcal{X}}\mathcal{Z}\mathcal{P}\mathcal{D}\mathcal{A})$. The other CP tables are: for U_1, $u_1 \succ \overline{u_1}$; for $F \in \mathcal{Y}$, $\overline{f} \succ f$ iff U_1 has value $\overline{u_1}$; for $Y''_i \in \mathcal{Y}''$, $\overline{y''_i} \succ y''_i$ iff X_i^T and X_i^F have values $\overline{y_i^T}$ and $\overline{y_i^F}$; for $Y'_i \in \mathcal{Y}'$, $\overline{y'_i} \succ y'_i$ iff Y''_i has value $\overline{y''_i}$ and either Y_i^T or Y_i^F has an overlined value; features $\mathcal{B}$ of the interconnecting net have the usual CP tables; for U_2, $u_2 \succ \overline{u_2}$ iff U_1 and B have values u_1 and $\overline{b}$.
- $\boldsymbol{N_5}$ is similar to N_4 with roles of U_1 and U_2 exchanged.

These two nets are devised to achieve two "contrasting" goals. They are designed so that for an outcome $\overline{\beta}_{\sigma_X}$, if we focus on a pair of features $\{Y_i^T, Y_i^F\}$, it is not possible to have improving flips toward an outcome having overlined value for both Y_i^T and Y_i^F (this is required by Principle (d), because such an outcome would not properly encode an assignment for variables Y). On the other hand,

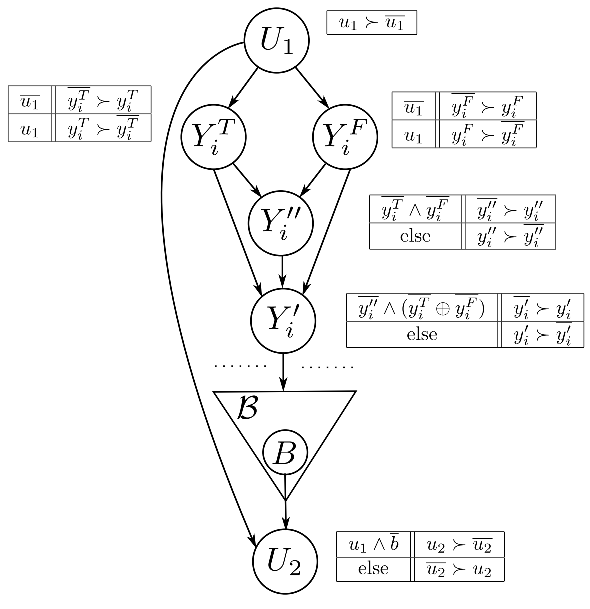

Figure 1: A schematic representation for CP-net N_4. The expression "$(\overline{y_i^T} \oplus \overline{y_i^F})$" in the CP table of Y_i' is satisfied when *exactly* one feature among Y_i^T and Y_i^F has an overlined value.

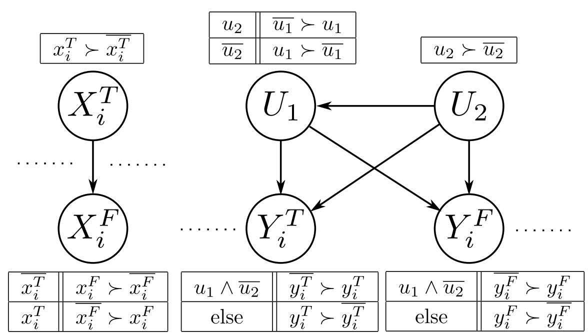

Figure 2: A schematic representation for CP-net N_6.

these two nets allow improving flips toward outcomes properly encoding an assignment σ_Y for Y, and hence toward outcomes for which either Y_i^T or Y_i^F have an overlined value (this is required by Principle (f)). Since these two nets have to exhibit this mixed behavior, N_4 and N_5 have this slightly intricate structure.

- $\boldsymbol{N_6}$ (see Figure 2) has, for each $x_i \in X$, the link (X_i^T, X_i^F); for each $y_i \in Y$, the links $(U_1, Y_i^T), (U_1, Y_i^F), (U_2, Y_i^T), (U_2, Y_i^F)$; and the link (U_2, U_1). There is the direct net $D(\mathcal{Y}'\mathcal{Y}''\mathcal{Z}\mathcal{P}\mathcal{D}\mathcal{A}\mathcal{B})$. The other CP tables are: X_i^T and X_i^F have CP tables as in N_2; for U_2, $u_2 \succ \overline{u_2}$; for U_1, $\overline{u_1} \succ u_1$ iff U_2 has value u_2; for $F \in \mathcal{Y}$, $\overline{f} \succ f$ iff U_1 and U_2 have values u_1 and $\overline{u_2}$.
- $\boldsymbol{N_7}$ has, for each $x_i \in X$, the link (X_i^F, X_i^T); and the link (U_2, U_1). There is the direct net $D(\mathcal{Y}\mathcal{Y}'\mathcal{Y}''\mathcal{Z}\mathcal{P}\mathcal{D}\mathcal{A}\mathcal{B})$. The other CP tables are: X_i^T and X_i^F have CP tables as in N_3; for U_2, $\overline{u_2} \succ u_2$; for U_1, $\overline{u_1} \succ u_1$ iff U_2 has value $\overline{u_2}$.

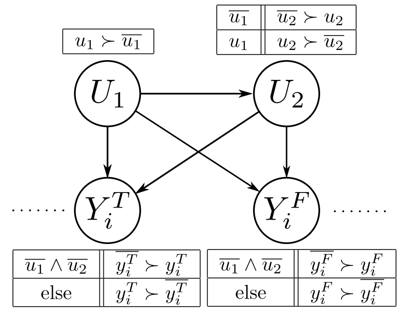

Figure 3: A schematic representation for CP-net N_8.

- $\boldsymbol{N_8}$ (see Figure 3) has, for each $y_i \in Y$, the links (U_1, Y_i^T), $(U_1, Y_i^F), (U_2, Y_i^T), (U_2, Y_i^F)$; and the link (U_1, U_2). There is the direct net $D(\overline{\mathcal{X}}\mathcal{Y}'\mathcal{Y}''\mathcal{Z}\mathcal{P}\mathcal{D}\mathcal{A}\mathcal{B})$. The other CP tables are: for U_1, $u_1 \succ \overline{u_1}$; for U_2, $\overline{u_2} \succ u_2$ iff U_1 has value $\overline{u_1}$; for $F \in \mathcal{Y}$, $\overline{f} \succ f$ iff U_1 and U_2 have values $\overline{u_1}$ and $\overline{u_2}$.

The aim of these nets is supporting the correctness of the reduction and realizing all the principles listed above. This is achieved together with various parts of the other nets.

$\mathcal{M}(\phi)$ is acyclic, binary, its indegree is three, and can be computed in polynomial time from Φ. Moreover, the class of *m*CP-nets derived from quantified formulas according to the reduction shown above is polynomially connected. It is possible to prove the following crucial property of $\mathcal{M}(\phi)$.

Lemma 6.1. *Let $\Phi = (\exists X)(\forall Y)(\exists Z)\phi(X, Y, Z)$ be a quantified formula, where $\phi(X, Y, Z)$ is a 3CNF formula defined over three disjoint sets, X, Y, and Z, of variables. Then, Φ is valid iff $\mathcal{M}(\phi)$ has a max optimal outcome.*

Lemma 6.1 implies the following theorem.

Theorem 6.2. *Let $\mathcal{M}$ be an mCP-net. Deciding whether there is a max optimal outcome in $\mathcal{M}$ is Σ_3^{P}-hard. Hardness holds even on polynomially connected classes of acyclic and binary mCP-nets whose indegree is three, and the number of agents is bounded to 8.*

Exists-Max-Optimal is in Σ_3^{P} [33], hence it is Σ_3^{P}-complete.

Regarding the complexity of deciding the existence of max optimum outcomes, we narrow down the upper-bound shown in the literature. In fact, in order to decide whether an *m*CP-net has a max optimum outcome, it is sufficient to guess an outcome α, and then check, via an oracle call, that α is actually max optimum. The oracle answering the latter question is in Π_2^{P} [33].

Theorem 6.3. *Let $\mathcal{M}$ be an mCP-net. Deciding whether there is a max optimum outcome in $\mathcal{M}$ is in Σ_3^{P}.*

7 RELATED WORKS

The graphical structure of CP-nets evidences that, in general, preferences may exhibit dependencies between features. Dependencies certainly are a critical characteristic to model, however they can become troublesome during preference aggregation. Whether dependencies are actually problematic or not depends on the specific ways in which agents' votes are collected. Two ways of collecting votes over combinatorial domains are the global voting and the sequential voting [28]. In global voting, agents submit the entire representation of their preferences, while, in sequential voting, agents' preferences are collected feature-by-feature. Global voting is the semantics of *m*CP-nets. Feature dependencies are not an issue in global voting, because in this case all the information needed for the aggregation is available. However, global voting can be expensive to evaluate (especially if preferences are extensively unfolded before any further processing). This computational burden can be limited by adopting sequential voting, for which, on the other hand, dependencies can be quite detrimental, to the point that sub-optimal outcomes can be selected [28]. Lacy and Niou [23] showed that these issues in sequential voting can be (partly) avoided if the considered preferences are separable, i.e., they do not have dependencies among features. Clearly, this is a very strong assumption, and it is unlikely to be met in practice [26, 27, 52, 53].

To overcome this limitation, $\mathcal{O}$-legality was proposed by Lang [26] as a weaker restriction. Essentially, if $\mathcal{O} = (F_1, \dots, F_m)$ is a sequence of features, a set $\mathcal{P}$ of agent preferences is $\mathcal{O}$-legal if, for any agent $A \in \mathcal{P}$, and any two features F_i and F_j, $i < j$ implies that A's preferences for F_i do not depends on F_j's value. When preferences are represented via CP-nets, a set of CP-nets is $\mathcal{O}$-legal if $\mathcal{O}$ is a topological order shared among all the CP-nets' graphs. Sequential voting over $\mathcal{O}$-legal CP-nets has extensively been investigated [26, 27, 52, 53], and $\mathcal{O}$-legality of CP-nets has been required in various other studies, e.g., [14, 18, 37, 38]. Of these, an interesting approach to preference aggregation over $\mathcal{O}$-legal CP-nets was proposed in [14], where "probabilistic" CP-nets were used to represent the result of the aggregation. However, also $\mathcal{O}$-legality is somewhat demanding [31, 46, 51], because it imposes that there are no inversions in the preference dependencies.

For example, if in a set of CP-nets encoding preferences for a dinner there were an agent whose choice of the starter influences the choice of the main dish and another agent whose choice of the main dish influences the choice of the starter, then those CP-nets would not be $\mathcal{O}$-legal. To overcome this limitation, the hypercubewise preference aggregation was introduced, however the semantics of hypercubewise aggregation is different from global voting (see, e.g., [12, 31, 32, 51]). Another approach is computing tailored voting agendas to circumvent preference dependencies [1].

Although it was explicitly stated in the literature that a theoretical comparison between global and sequential voting was highly promising [26], global voting over (non-$\mathcal{O}$-legal) CP-nets has not been thoroughly investigated as sequential voting.

The first work studying global voting over (not necessarily $\mathcal{O}$-legal) CP-nets was the one of Rossi et al. [43] in which *m*CP-nets are defined (remember that *m*CP-nets' semantics is global voting over CP-nets). Most of the algorithms considered in [43] were brute-force, hence, those algorithms gave only Exp upper bounds for most of the global voting tasks over CP-nets, and no hardness result was provided. Algorithms exploiting SAT solvers to compute global Pareto optimal outcomes and weak Condorcet winners over CP-nets were proposed in [29, 30]. Li et al. [31] extended those results to computing weak Condorcet winners via SAT solvers even on cyclic CP-nets, while Li et al. [32] introduced also the possibility of multivalued and incomplete CP-nets. Although the mentioned works advanced the research on global voting over CP-nets, still they did not provide precise complexity results. As mentioned above, the complexity of these problems was reported as open several times in the literature [26, 29–32], and only recently a work characterized the exact complexity of some voting tasks over *m*CP-nets [33].

Regarding the P-completeness results, to the best of our knowledge there is only another P-completeness result in the computational social choice literature [9, 11], and it is the complexity of checking the essential set, which is a specific solution concept, over weak tournaments. Weak tournaments are graphs representing incomplete preference relations, and they directly encode a dominance relation (after vote aggregation, we could say). Intuitively, the data structure in input, i.e., the weak tournament, reports whether an alternative is preferred to another via some voting procedure (e.g., majority), but the preferences of the single agents are not explicitly represented in the input. This means that the aggregation of the preferences is assumed to be pre-computed and provided in input. In this respect, our work is different because we assume that the input contains the preferences of the single agents. Moreover, the papers cited above do not mention the consequences of P-completeness in terms of non-parallelizability. In this respect, to the best of our knowledge, our work is the first pointing to the important issue of whether polynomial-time voting procedures can scale up over big instances.

8 SUMMARY AND OUTLOOK

In this paper, we have further analyzed the complexity of *m*CP-nets, whose dominance semantics is global voting over not necessarily $\mathcal{O}$-legal CP-nets. We have proven that deciding the existence of max optimal and max optimum outcomes is Σ_3^P-complete and in Σ_3^P, respectively. We have also shown that various polynomial-time voting tasks over (*m*)CP-nets are actually P-complete, and hence non-parallelizable. This points out a significant issue, which is whether polynomial-time voting schemes are highly parallelizable, so that parallel algorithms can scale up un big instances.

Possible directions for further research are showing the exact complexity of deciding the existence of max optimum and Pareto optimum outcomes in *m*CP-nets. Analyzing *m*CP-nets when partial CP-nets are allowed to be constituent of them will also be important, given that the original definition of *m*CP-nets used the idea of partial CP-nets to model influences between preferences of different agents. Having constraints on outcomes's feasibility is another interesting direction of investigation. Without any constraint, CP-nets model agents' preferences when it is assumed that all outcomes are attainable. However, this is not always the case. During the aggregation precess, we have to take into account what outcomes are feasible and what are not. For example, to decide whether an outcome is majority dominated by another, we have to check that the latter is actually feasible. A similar idea characterized the solution concepts in NTU cooperative games defined via constraints [19]. This approach could be integrated with the definition of constrained CP-nets [5, 41], and a concept of compact representation of constraints (see [20]) could also be introduced.

ACKNOWLEDGMENTS

This work was partly supported by the EPSRC grant EP/M025268/ "VADA: Value Added Data Systems – Principles and Architecture". We thank the anonymous reviewers for their helpful comments.

REFERENCES

[1] Stéphane Airiau, Ulle Endriss, Umberto Grandi, Daniele Porello, and Joel Uckelman. 2011. Aggregating Dependency Graphs into Voting Agendas in Multi-Issue Elections. In *Proc. of IJCAI*. 18–23.

[2] James C. Boerkoel Jr., Edmund H. Durfee, and Keith Purrington. 2010. Generalized solution techniques for preference-based constrained optimization with CP-nets. In *Proc. of AAMAS*. 291–298.

[3] Craig Boutilier, Ronen I Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. 2004. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.* 21 (2004), 135–191. https://doi.org/10.1613/jair.1234

[4] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. 2004. Preference-Based Constrained Optimization with CP-nets. *Comput. Intell.* 20, 2 (2004), 137–157. https://doi.org/10.1111/j.0824-7935.2004.00234.x

[5] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. 2004. Preference-Based Constrained Optimization with CP-Nets. *Comput. Intell.* 20, 2 (2004), 137–157. https://doi.org/10.1111/j.0824-7935.2004.00234.x

[6] Ronen I. Brafman and Yuri Chernyavsky. 2005. Planning with Goal Preferences and Constraints. In *Proc. of ICAPS*. 182–191.

[7] Ronen I. Brafman and Carmel Domshlak. 2002. TCP-nets for Preference-based Product Configuration. In *Proc. of the ECAI 2002 Workshop on Configuration*. 101–106.

[8] Ronen I. Brafman, Francesca Rossi, Domenico Salvagnin, Kristen Brent Venable, and Toby Walsh. 2010. Finding the Next Solution in Constraint- and Preference-Based Knowledge Representation Formalisms. In *Proc. of KR*. 425–433.

[9] Felix Brandt, Markus Brill, and Paul Harrenstein. 2016. Tournament Solutions. In *Handbook of Computational Social Choice*. Cambridge University Press, New York, NY, USA, 57–84.

[10] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). 2016. *Handbook of Computational Social Choice*. Cambridge University Press, New York, NY, USA.

[11] Felix Brandt and Felix Fischer. 2008. Computing the minimal covering set. *Math. Soc. Sci.* 56, 2 (2008), 254–268.

[12] Vincent Conitzer, Jérôme Lang, and Lirong Xia. 2011. Hypercubewise Preference Aggregation in Multi-Issue Domains. In *Proc. of IJCAI*. 158–163.

[13] Vincent Conitzer and Tuomas Sandholm. 2005. Communication complexity of common voting rules. In *Proc. of EC*. 78–87.

[14] Cristina Cornelio, Umberto Grandi, Judy Goldsmith, Nicholas Mattei, Francesca Rossi, and Kristen Brent Venable. 2015. Reasoning with PCP-nets in a Multi-Agent Context. In *Proc. of AAMAS*. 969–977.

[15] Tommaso Di Noia, Thomas Lukasiewicz, Maria Vanina Martinez, Gerardo Ignacio Simari, and Oana Tifrea-Marciuska. 2015. Combining Existential Rules with the Power of CP-theories. In *Proc. of IJCAI*. 2918–2925.

[16] Alexander Felfernig, Lothar Hotz, Claire Bagley, and Juha Tiihonen (Eds.). 2014. *Knowledge-Based Configuration: From Research to Business Cases*. Morgan Kaufmann/Elsevier, Waltham, MA, USA.

[17] Judy Goldsmith, Jérôme Lang, Miroslaw Truszczynski, and Nic Wilson. 2008. The computational complexity of dominance and consistency in CP-nets. *J. Artif. Intell. Res.* 33 (2008), 403–432. https://doi.org/10.1613/jair.2627

[18] Umberto Grandi, Hang Luo, Nicolas Maudet, and Francesca Rossi. 2014. Aggregating CP-nets with Unfeasible Outcomes. In *Proc. of CP*. LNCS, Vol. 8656. Springer-Verlag, Berlin Heidelberg, Germany, 366–381.

[19] Gianluigi Greco, Enrico Malizia, Luigi Palopoli, and Francesco Scarcello. 2010. Non-Transferable Utility Coalitional Games via Mixed-Integer Linear Constraints. *J. Artif. Intell. Res.* 38 (2010), 633–685. https://doi.org/10.1613/jair.3060

[20] Gianluigi Greco, Enrico Malizia, Luigi Palopoli, and Francesco Scarcello. 2014. The Complexity of the Nucleolus in Compact Games. *ACM Trans. Comput. Theory* 7, 1 (2014), 3:1–3:52. https://doi.org/10.1145/2692372.2692374

[21] Raymond Greenlaw, H. James Hoover, and Walter L. Ruzzo. 1995. *Limits to Parallel Computation: P-Completeness Theory*. Oxford University Press, New York, NY, USA.

[22] David S. Johnson. 1990. A catalog of complexity classes. In *Handbook of Theoretical Computer Science (Vol. A)*, Jan van Leeuwen (Ed.). Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 67–161.

[23] Dean Lacy and Emerson M. S. Niou. 2000. A Problem with Referendums. *J. Theor. Polit.* 12, 1 (2000), 5–31. https://doi.org/10.1177/0951692800012001001

[24] Richard E. Ladner. 1975. The Circuit Value Problem is Log Space Complete for P. *ACM SIGACT News* 7, 1 (1975), 18–20.

[25] Jérôme Lang. 2004. Logical preference representation and combinatorial vote. *Ann. Math Artif. Intel.* 42, 1 (2004), 37–71. https://doi.org/10.1023/B:AMAI.0000034522.25580.09

[26] Jérôme Lang. 2007. Vote and Aggregation in Combinatorial Domains with Structured Preferences. In *Proc. of IJCAI*. 1366–1371.

[27] Jérôme Lang and Lirong Xia. 2009. Sequential composition of voting rules in multi-issue domains. *Math. Soc. Sci.* 57, 3 (2009), 304–324. https://doi.org/10.1016/j.mathsocsci.2008.12.010

[28] Jérôme Lang and Lirong Xia. 2016. Voting in Combinatorial Domains. In *Handbook of Computational Social Choice*. Cambridge University Press, New York, NY, USA, 197–222.

[29] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. 2010. An effcient Majority-Rule-Based Approach for Collective Decision Making with CP-nets. In *Proc. of KR*. 578–580.

[30] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. 2010. An efficient Procedure for Collective Decision-making with CP-nets. In *Proc. of ECAI*. 375–380.

[31] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. 2011. Majority-rule-based preference aggregation on multi-attribute domains with CP-nets. In *Proc. of AAMAS*. 659–666.

[32] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. 2015. Aggregating multi-valued CP-nets: a CSP-based approach. *J. Heuristics* 21, 1 (2015), 107–140. https://doi.org/10.1007/s10732-014-9276-8

[33] Thomas Lukasiewicz and Enrico Malizia. 2016. On the Complexity of mCP-nets. In *Proc. of AAAI*. 558–564.

[34] Thomas Lukasiewicz and Enrico Malizia. 2017. A novel characterization of the complexity class Θ_k^P based on counting and comparison. *Theor. Comput. Sci.* 694 (2017), 21–33. https://doi.org/10.1016/j.tcs.2017.06.023

[35] Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo Ignacio Simari. 2013. Preference-Based Query Answering in Datalog+/- Ontologies. In *Proc. of IJCAI*. 1017–1023.

[36] Thomas Lukasiewicz, Maria Vanina Martinez, Gerardo Igancio Simari, and Oana Tifrea-Marciuska. 2014. Ontology-Based Query Answering with Group Preferences. *ACM Trans. Internet Technol.* 14, 4 (2014), 25:1–25:24. https://doi.org/10.1145/2677207

[37] Alberto Maran, Maudet. Nicolas, Maria Silvia Pini, Francesca Rossi, and Kristen Brent Venable. 2013. A Framework for Aggregating Influenced CP-Nets and Its Resistance to Bribery. In *Proc. of AAAI*. 668–678.

[38] Nicholas Mattei, Maria Silvia Pini, Francesca Rossi, and Kristen Brent Venable. 2013. Bribery in voting with CP-nets. *Ann. Math Artif. Intel.* 68, 1–3 (2013), 135–160. https://doi.org/10.1007/s10472-013-9330-5

[39] Satoru Miyano, Shuji Shiraishi, and Takayoshi Shoudai. 1990. *A List of P-complete Problems*. TR-CS 17. RIFIS, Kyushu University, Japan.

[40] Christos H. Papadimitriou. 1994. *Computational Complexity*. Addison Wesley, Reading, MA, USA.

[41] Steven David Prestwich, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. 2005. Constraint-Based Preferential Optimization. In *Proc. of AAAI*. 461–466.

[42] Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). 2015. *Recommender Systems Handbook* (2nd ed.). Springer-Verlag, New York, NY, USA.

[43] Francesca Rossi, Kristen Brent Venable, and Toby Walsh. 2004. mCP Nets: representing and reasoning with preferences of multiple agents. In *Proc. of AAAI*. 729–734.

[44] Richard Anthony Russell. 2012. *Planning with preferences using maximum satisfiability*. UCAM-CL-TR 822. Computing Laboratory, University of Cambridge, UK.

[45] Dmitry Shaparau, Marco Pistore, and Paolo Traverso. 2006. Contingent Planning with Goal Preferences. In *Proc. of AAAI*. 927–935.

[46] Sujoy Sikdar, Sibel Adali, and Lirong Xia. 2017. Optimal Decision Making with CP-nets and PCP-nets (Extended Abstract). In *Proc. of AAMAS*. 1736–1738.

[47] Tran Cao Son and Enrico Pontelli. 2006. Planning with Preferences using Logic Programming. *Theor. Pract. Log. Prog.* 6, 5 (2006), 559–607. https://doi.org/10.1017/S1471068406002717

[48] Jacob Stein, Ingrid Nunes, and Elder Cirilo. 2014. Preference-based feature model configuration with multiple stakeholders. In *Proc. of SPLC*. 132–141.

[49] Larry J. Stockmeyer. 1976. The polynomial-time hierarchy. *Theor. Comput. Sci.* 3, 1 (1976), 1–22. https://doi.org/10.1016/0304-3975(76)90061-X

[50] Celia Wrathall. 1976. Complete sets and the polynomial-time hierarchy. *Theor. Comput. Sci.* 3, 1 (1976), 23–33. https://doi.org/10.1016/0304-3975(76)90062-1

[51] Lirong Xia, Vincent Conitzer, and Jérôme Lang. 2008. Voting on Multiattribute Domains with Cyclic Preferential Dependencies. In *Proc. of AAAI*. 202–207.

[52] Lirong Xia, Jérôme Lang, and Mingsheng Ying. 2007. Sequential voting rules and multiple elections paradoxes. In *Proc. of TARK*. 279–288.

[53] Lirong Xia, Jérôme Lang, and Mingsheng Ying. 2007. Strongly Decomposable Voting Rules on Multiattribute Domains. In *Proc. of AAAI*. 776–781.