More Complexity Results about Reasoning over (m)CP-nets

Enrico Malizia
Department of Computer Science, University of Exeter
Exeter, UK
e.malizia@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, mCP-nets, as mentioned several times in the literature, lacked such a thorough characterization. An initial complexity analysis for mCP-nets was carried out only recently. In this paper, we further investigate the complexity of mCP-nets. In particular, we show the \( \Sigma^p_3 \)-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 cannot benefit from highly parallel computation.

CCS CONCEPTS
• Theory of computation → Problems, reductions and completeness; Algorithmic game theory; • Computing methodologies → Multi-agent systems;

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

 ACM Reference Format:

1 INTRODUCTION
Managing and aggregating agent preferences have attracted extensive interest given their importance in AI applications, such as recommender systems [47], (group) product configuration [6, 16], (group) planning [5, 50, 52], (group) preference-based constraint satisfaction [2, 4, 8], and (group) preference-based query answering/information retrieval [15, 40, 41]. In computer science, preference aggregation has often been based on social choice theory [10]. In this theory, agents’ preferences are usually assumed to be extensively represented. Although this is reasonable with a small set of candidates, this is not feasible with a combinatorial domain, i.e., when the set of candidates, or outcomes, is the Cartesian product of finite value domains for each of a set of features [30, 33].

Combinatorial domains contain an exponential number of outcomes in the number of features. Hence, compact representations for combinatorial preferences are needed [30, 33]. CP-nets [3] are among the most studied of these representations, as proven by a vast literature on them. In CP-nets, vertices of a graph 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 \( B \). 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 mCP-nets [48]. An mCP-net is a set of CP-nets, one for each agent. The preference semantics of mCP-nets is defined via voting: through their own individual CP-nets, each agent votes whether an outcome is preferred to another. Various voting schemes were proposed for mCP-nets [37, 48] and different voting schemes give rise to different dominance semantics for mCP-nets. In the voting schemes proposed for mCP-nets, the voting protocol adopted, i.e., the actual way in which votes are collected [13], is global voting [31] over the mCP-nets of the single players. In this protocol, the results of the vote are computed by having in input the whole CP-nets (see Section 7 for related works on sequential voting, which is a different voting protocol). In the literature, a comparison between global voting and sequential voting over CP-nets was explicitly asked for and stated to be highly promising [31]. However, global voting over CP-nets has not been thoroughly investigated as sequential voting (see Section 7). In fact, unlike CP-nets, which were extensively analyzed, a precise complexity analysis of mCP-nets was missing for long time, as explicitly mentioned several times in the literature [31, 34–37]. An initial complexity analysis of voting tasks over mCP-nets was carried out only recently [38]. For example, deciding Pareto dominance was shown co-NP-complete, and deciding the existence of majority optimal outcomes was proven \( \Sigma^p_3 \)-complete. The aim of this paper is to further explore the complexity of mCP-nets (and hence the complexity of global voting over CP-nets).

Contributions. We focus on acyclic binary polynomially connected mCP-nets whose constituent are standard CP-nets. Hence, in this paper the dominance semantics of mCP-nets is precisely global voting over CP-nets. Our contributions are briefly as follows:

- We show the \( \Sigma^p_3 \)-completeness of deciding the existence of max optimal outcomes in mCP-nets;

- We prove that various voting tasks over mCP-nets known to be feasible in polynomial time are actually P-complete.

Furthermore, as a side result of our investigation:

- We define the problem Th-CVP: given a Boolean circuit \( C \), a Boolean vector \( x \), and an integer \( k \), decide whether the number of gates of \( C \) evaluating to true when \( x \) is given in input to \( C \) is at most (resp., at least) \( k \). Th-CVP is here
shown P-complete, and hence it can be useful in reductions showing P-hardness of problems involving counting tasks.

**Organization of the paper.** In Section 2, we provide an overview of our results. Preliminaries on CP-nets and mCP-nets are given in Section 3. We show P-completeness results over CP-nets and mCP-nets in Sections 4 and 5, respectively. In Section 6, we study max voting. Related works are discussed in Section 7. We draw our conclusions and outline possible future research in Section 8. For space reasons, we provide only proof intuitions for various results.

## 2 OVERVIEW 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 \( \Sigma^p_3 \)-complete. This supports that, in mCP-nets, max voting is computationally more demanding than majority voting, for which deciding the existence of optimal outcomes is \( \Sigma^p_2 \)-complete [38]. 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 \( \Theta^p \)-complete, respectively [38, 39]). Hence, this dissimilarity in the complexity of dominance checking carries 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 [5], 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, such as 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. That is, the 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, some problems are inherently sequential and do not benefit from highly parallel processing [25]. Saying that a problem \( L \) does not benefit from parallel processing does not mean 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 [25]. Decision problems of this kind are the P-hard ones, which are often said to be non-parallelizable [25]. 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 carefully been 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 mCP-nets. Hence, the P-completeness results reported here, not only characterize more precisely the complexity of mCP-nets, but they also point out a significant issue, which is whether P-time voting schemes can benefit from parallel algorithms or not.

Our results 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 \( \langle G_N, \text{Dom}_N, (\text{CPT}_N^F)_{F \in \mathcal{F}_N} \rangle \), where \( G_N = (\mathcal{F}_N, \mathcal{E}_N) \) is a directed graph whose vertices \( \mathcal{F}_N \) represent the features of the combinatorial domain, \( \text{Dom}_N \) is a function, and \( (\text{CPT}_N^F)_{F \in \mathcal{F}_N} \) is a family of functions. For a feature \( F \), \( \text{Dom}_N \) associates a (value) domain \( \text{Dom}_N(F) \) with \( F \), while \( \text{CPT}_N^F \) 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 \( F \)).

The domain \( \text{Dom}_N(S) \) of an outcome \( S \) is \( \times_{F \in S} \text{Dom}_N(F) \). An outcome is an element of the set \( \Omega_N = \text{Dom}_N(\mathcal{F}_N) \). For a feature \( F \in \mathcal{F}_N \) and an outcome \( \alpha \), \( \alpha[F] \) is \( F \)'s value in \( \alpha \). For a feature set \( S \subseteq \mathcal{F}_N \) and an outcome \( \alpha \), \( \alpha[S] \) is the projection of \( \alpha \) over \( S \).

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 \( G_N \); in each row there is a total order over \( \text{Dom}_N(F) \). This order encodes agent's preferences for \( F \)'s values when specific values of \( F \)'s parents are considered: \( \overline{f} > f \) denotes \( \overline{f} \) being preferred to \( f \). If \( F \) has no parents, its CP table has only one row.

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

<table>
<thead>
<tr>
<th>Problem</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEAT-VALUE-OPT</td>
<td>P-complete</td>
</tr>
<tr>
<td>SAME-OPT</td>
<td>P-complete</td>
</tr>
<tr>
<td>RANK-BOUND</td>
<td>P-complete</td>
</tr>
<tr>
<td>COMPARE-RANK</td>
<td>P-complete</td>
</tr>
</tbody>
</table>

### Table 2: Summary of the results for mCP-nets. Membership shown in [38].

<table>
<thead>
<tr>
<th>Problem</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXISTS-PARETO-OPTIMUM</td>
<td>P-complete</td>
</tr>
<tr>
<td>RANK-DOMINANCE</td>
<td>P-complete</td>
</tr>
<tr>
<td>IS-RANK-OPTIMAL</td>
<td>P-complete</td>
</tr>
<tr>
<td>IS-RANK-OPTIMUM</td>
<td>P-complete</td>
</tr>
<tr>
<td>EXISTS-RANK-OPTIMUM</td>
<td>P-complete</td>
</tr>
<tr>
<td>EXISTS-MAX-OPTIMAL</td>
<td>( \Sigma^p_3 )-complete</td>
</tr>
<tr>
<td>EXISTS-MAX-OPTIMUM</td>
<td>in ( \Sigma^3 )</td>
</tr>
</tbody>
</table>
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 report CP tables in figures, we will use a logical 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 $\alpha, \beta$ 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 $\alpha$ of the parents of $F$, $\beta(F) > \alpha(F)$. Outcome $\beta$ is preferred to $\alpha$, or $\beta$ dominates $\alpha$ (in $N$), denoted $\beta >_N \alpha$, iff there is a sequence of improving flips from $\alpha$ to $\beta$, otherwise $\beta$ does not dominate $\alpha$, denoted $\beta \not>_N \alpha$; $\beta$ and $\alpha$ are incomparable, denoted $\beta \equiv_N \alpha$, iff $\beta \not>_N \alpha$ and $\alpha \not>_N \beta$. Observe that, since there are no indifferences between features values in (classical) CP-nets, for any two outcomes $\alpha$ and $\beta$, 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 $\mathcal{F}$ of CP-nets is polynomially connected iff there exists a polynomial $p$ such that, for any CP-net $N \in \mathcal{F}$ 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 $\alpha_N$, dominating all others, that can be computed in polynomial time [3].

The rank of an outcome $\alpha$ in a CP-net $N$, $\text{Rank}_N(\alpha)$, is the length of the shortest improving flipping sequence from $\alpha$ to $\alpha_N$ [48]. Unless stated otherwise, we consider acyclic binary CP-nets.

**mCP-nets.** An mCP-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 mCP-net is the agents’ number, so a 3CP-net is an mCP-net with $m = 3$. Originally, partial CP-nets were allowed to be constituent of mCP-nets [48]. Here, we assume only standard CP-nets to be part of mCP-nets, and we do not assume CP-nets to be $O$-legal (i.e., we do not assume that the CP-nets of an mCP-net have a common topological order of the features). mCP-nets’ semantics is based on voting. Let $M = \{N_1, \ldots, N_m\}$ be an mCP-net, and let $\alpha, \beta$ be two outcomes. We define the sets $S_M^\alpha(\beta, \alpha) = \{i | \alpha >_{N_i} \beta\}, S_M^{\alpha\beta}(\alpha, \beta) = \{i | \alpha >_{N_i} \beta\}$, and $S_M^{\alpha\beta\alpha}(\alpha, \beta) = \{i | \alpha >_{N_i} \beta\}$, as the sets of agents preferring $\alpha$ to $\beta$, preferring $\beta$ to $\alpha$, and for which $\alpha$ and $\beta$ are incomparable, respectively. $\text{Rank}_M(\alpha) = \sum_{1 \leq i \leq m} \text{Rank}_{N_i}(\alpha)$ [48]. The dominance semantics considered are as [48]:

- **Pareto:** $\beta$ Pareto dominates $\alpha$, denoted by $\beta >_P^M \alpha$, iff all the agents of $M$ prefer $\beta$ to $\alpha$, i.e., $|S_M^\beta(\alpha, \alpha)| = m$.
- **Majority:** $\beta$ majority dominates $\alpha$, denoted by $\beta >_M^\alpha \alpha$, iff the majority of the agents of $M$ prefers $\beta$ to $\alpha$, i.e., $|S_M^\beta(\alpha, \alpha)| > |S_M^{\alpha\beta\alpha}(\alpha, \beta)| + |S_M^{\alpha\beta}(\alpha, \beta)|$.
- **Max:** $\beta$ max dominates $\alpha$, denoted by $\beta >_M^\alpha \alpha$, iff the group of the agents of $M$ preferring $\beta$ to $\alpha$ is the biggest, i.e., $|S_M^\beta(\alpha, \alpha)| > \max(|S_M^{\alpha\beta}(\alpha, \beta)|, |S_M^{\alpha\beta\alpha}(\alpha, \beta)|)$.

**Rank:** $\beta$ rank dominates $\alpha$, denoted by $\beta >_R^M \alpha$, iff $\text{Rank}_M(\beta) < \text{Rank}_M(\alpha)$.

For a voting scheme $s$, an outcome $\alpha$ is $s$ optimal in $M$ iff $\beta >_s^M \alpha$ for all $\beta \neq \alpha$, whereas $\alpha$ is $s$ optimum in $M$ iff $\alpha >_s^M \beta$ for all $\beta \neq \alpha$. Optimum outcomes, if they exist, are unique.

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

**Complexity Classes.** We assume basic knowledge of computational complexity and of the polynomial hierarchy; see [27, 45]. A language $L$ is P-hard iff, for all languages $L'$ in $P$, there is a log-space reduction from $L'$ to $L$; $L$ 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) [29], for a Boolean circuit $\mathcal{C}$ and a Boolean vector $x$, we have to decide whether $\mathcal{C}$’s output is true when receiving $x$ as input. In the literature, various ways to represent circuits were illustrated. Here, we use a representation that is a mix of those in [29, 44, 49]. A circuit $\mathcal{C} = \{C_1, \ldots, C_m\}$ is a sequence of logic gates, which are represented through formulas:

1. If $C_1 = x_j$, $C_1$ is an input gate fed with the $j$th input bit; (ii) if $C_1 = C_j \land C_k$ (resp., $C_1 = C_j \lor C_k$), $C_1$ is an AND (resp., OR) gate, whose inputs are the outputs of $C_j$ and $C_k$ (with $j, k < i$); (iii) if $C_1 = \neg C_j$, $C_1$ is a NOT gate, whose input is the output of $C_j$ (with $j < i$).

The Boolean values of gates $C_j$ when $x$ is given in input to $\mathcal{C}$, denoted by $\nu_\mathcal{C}(C_j, x)$, are defined in the natural way.

We assume that the problem CVP is defined as in [25]. A CVP instance $I = (\mathcal{C}, x, C_{out})$, where $\mathcal{C} = \{C_1, \ldots, C_m\}$ is a circuit, $x = (x_1, \ldots, x_n)$ is a vector, and $C_{out} \in \mathcal{C}$ is the output gate, is a ‘yes’-instance iff $\nu_\mathcal{C}(C_{out}, x) = true$. 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 $C_m$ [25, 29, 44].

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

- If $C_i$ is an input gate with $C_i = x_j$, there is no edge entering in $D_i$, $x_j = true$, $\overline{d_i} > d_i$; $x_j = false$, $d_i > \overline{d_i}$.
- If $C_i$ is an AND (resp., OR) gate, with $C_i = C_j \land C_k$ (resp., $C_i = C_j \lor C_k$), then there are two edges entering in $D_i$, one from $D_j$ and one from $D_k$. If $C_i = C_j \land C_k$, for $D_i$, $d_i > d_j$ iff both $D_j$ and $D_k$ have overlined values. If $C_i = C_j \lor C_k$, for $D_i$, $\overline{d_i} > d_j$ if $D_j$ 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 $D_i$, $d_i > \overline{d_i}$ if $D_j$ has value $\overline{d_i}$, $\overline{d_i} > d_i$ otherwise.
Observe that $N(\vec{v}, x)$ is binary, acyclic, and can be computed in logarithmic space from $\vec{v}$ and $x$ (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 $N(\vec{v}, x)$ is bounded by a constant as well). 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 $\vec{v}$, it can be shown that, in $N(\vec{v}, x)$, a feature $D_i$ has value $\overline{d_i}$ in the optimum outcome iff $v_\vec{v}(C_i, x) = true$.

**Lemma 4.1.** Let $\vec{v} = \{C_1, \ldots, C_m\}$ be a circuit, and let $x$ be an input vector. For any gate $C_i$, $v_\vec{v}(C_i, x) = true$ iff $o_N(\vec{v}, x)[D_i] = \overline{d_i}$.

From this key property follows the P-hardness of the problem **Feat-Value-Opt:** for a CP-net $N$, a feature $F \in F_N$, 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 feature corresponding to the output gate and replicates its value, instead, in $N_2$, $O$ has a specific preferred value, say $\overline{d}$. In this case, $o_N_1 = o_N_2$ iff the circuit outputs $true$.

**Theorem 4.3.** **Same-Opt** is P-complete.

Let $TG(\vec{v}, x)$ denote the number of $\vec{v}$’s gates evaluating to true when $x$ is given in input to $\vec{v}$. Consider the problem **Th-CVP** (Threshold CVP): given a Boolean circuit $\vec{v}$, an input vector $x$, and an integer $k$, decide whether $TG(\vec{v}, x) \leq k$. Th-CVP can be seen as the decision version of the computation problem e-CTGP in [49]. We show that Th-CVP is P-hard via a reduction similar to the one used to prove the P-hardness of e-CTGP.

**Theorem 4.4.** **Th-CVP** is P-complete. Hardness holds even if the threshold number $k$ is such that $k < |\vec{v}|/2$.

**Proof.** Th-CVP is in P, because gates’ values can be evaluated in polynomial time [25, 29], 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 $(\vec{v}, x, C_{out})$ of CVP, where $\vec{v} = \{C_1, \ldots, C_m\}$, into an instance $(\vec{v}', x', k)$ of Th-CVP. $\vec{v}'$ consists of $2m$ gates, whose first $m$ gates are identical (for function and wiring) to those of $\vec{v}$. The remaining $m$ gates of $\vec{v}'$ replicate the value of $C_{out} = C_{out}$. More formally, $C_{m+i} = C_{m+i} \land C_{m+i}$ and, for all $2 \leq i \leq m$, $C_{m+i} = C_{m+i} \lor C_{m+i}$ and $C_{m+i} = C_{m+i}$. The input vector $x'$ equals $x$, and $k = m - 1$. Clearly, the reduction can be computed in logarithmic space. Observe that $k = m - 1 < |\vec{v}'|/2$, where $2m = |\vec{v}'|$. 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 $(\vec{v}, x)$ is a ‘yes’-instance of CVP, i.e., $v_\vec{v}(C_{out}, x) = false$, then $v_\vec{v}(C_{out}, x') = v_\vec{v}(C_{m+1}, x') = \cdots = v_\vec{v}(C_{2m}, x') = false$. Hence, $TG(C', x') \leq |\vec{v}'| = (m + 1) = m - 1 = k$, and thus $(\vec{v}', x', k)$ is a ‘yes’-instance of Th-CVP as well.

$(\Leftarrow)$ On the other hand, if $(\vec{v}, x)$ is a ‘no’-instance of CVP, i.e., $v_\vec{v}(C_{out}, x) = true$, then $v_\vec{v}(C_{out}, x') = v_\vec{v}(C_{m+1}, x') = \cdots = v_\vec{v}(C_{2m}, x') = true$. Hence, $TG(C', x') \geq m + 1 > m - 1 = k$, and thus $(\vec{v}', x', k)$ is a ‘no’-instance of Th-CVP as well.

Observe that, since $P$ is closed under complement, also deciding whether $TG(\vec{v}, x) \geq k$ is P-complete.

Consider the problem **Rank-Bound:** for a CP-net $N$, an outcome $\alpha \in O_N$, and an integer $k$, decide whether $Rank_N(\alpha) \leq k$. For an acyclic CP-net $N$, it is known that

$$Rank_N(\alpha) = |\{F \in F_N \land \alpha[F] \neq o_N[F]\}|,$$

i.e., $\alpha$’s rank in $N$ is the number of features whose value in $\alpha$ differs from their respective values in $o_N$ [38]. **Rank-Bound**’s P-hardness follows from Theorem 4.4 and from Lemma 4.1 and Equation (1), by which the number of overlined values in the optimum outcome of $N(\vec{v}, x)$ equals $TG(\vec{v}, x)$.

**Theorem 4.5.** **Rank-Bound** is P-complete.

Consider the problem **Compare-Rank:** for a CP-net $N$ and two outcomes $\alpha, \beta \in O_N$, decide whether $Rank_N(\alpha) < Rank_N(\beta)$.

**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 [38], 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 $(N, F, v)$ of **Feat-Value-Opt** into the instance $(N', a, \beta)$ of **Compare-Rank** as follows (assume w.l.o.g. that $v = f$): $N' = N \cup a \beta$ 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 $\alpha$ and $\beta$ 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 $(N, F, v)$ is a ‘yes’-instance of **Feat-Value-Opt**, $o_N[F] = f \neq \overline{f}$. Hence, $Rank_N(\beta) < Rank_N(\alpha)$.

$(\Leftarrow)$ If $(N, F, v)$ is a ‘no’-instance of **Feat-Value-Opt**, $o_N[F] = \overline{f} \neq f$. Hence, $Rank_N(\alpha) < Rank_N(\beta)$.

**5.** **P-COMPLETE PROBLEMS ON MCP-NETS**

First, we focus on a Pareto voting task. Consider the problem **Exists-Pareto-Optimum:** given an mCP-net $M$, decide whether $M$ 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 [38]. 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 2CP-nets.

**Exists-Pareto-Optimum** is also in P [38], hence it is P-complete.

We now prove the hardness of rank voting tasks over mCP-nets. Consider the problem **Rank-Dominance:** for an mCP-net $M$ and two outcomes $\alpha, \beta \in O_M$, decide whether $\beta >^M \alpha$, i.e., decide whether $Rank_M(\beta) < Rank_M(\alpha)$. Remember that, for an mCP-net $M = \langle N_1, \ldots, N_m \rangle$, $Rank_M(\alpha) = \sum_{1 \leq i \leq m} Rank_{N_i}(\alpha)$.  

1543
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.

Rank-Dominance is also in P [38], hence it is P-complete.

Consider now the problem Is-Rank-Optimal (resp., Is-Rank-Optimum): for an mCP-net $M$ and an outcome $\alpha \in O_M$, decide whether $\alpha$ is rank optimal (resp., optimum) in $M$. We recall some definitions from [38]. A value $v$ of a feature $F$ is average optimal iff $v \in \text{arg min}_{v \in \text{Dom}(F)} \{ i \mid 1 \leq i \leq m \land v \neq \alpha [F] \}$, 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 $\alpha$ is average optimal iff, for each feature $F$, $\alpha[F]$ is average optimal.

An outcome is rank optimal if it is average optimal [38]. 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 $M$ has two average optimal outcomes, then $M$ has two rank optimal outcomes, and hence $M$ 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 to prove the $P$-hardness of Is-Rank-Optimal and Is-Rank-Optimum, we use a CP-net that is designed to have a desired optimum outcome. Let $S$ be a set of binary features, and let $\alpha \in \text{Dom}(S)$ be an outcome. The “direct” net $D(\alpha)$ has as features the set $S$ and has no edge. The CP table of feature $F$ is $f > f$, if $\alpha[F] = f$; the CP table of feature $F$ is $f \neq f$, if $\alpha[F] = \bar{f}$.

**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 $(N, F, v)$ of Feat-Value-Opt into the instance $(M, \alpha)$ of Is-Rank-Optimum as follows (assume w.l.o.g. that $v = f$): $M = \langle N_1, N_2, N_3 \rangle$ is a 4CP-net, where $N_1 = N_2 = N$, $N_3 = D(\alpha)$, with $\alpha$ being an outcome defined over the features in $N$ and assigning non-overlined values to all features, and $N_4 = D(\beta)$, with $\beta$ assigning overlined values to all features but $F$, for which $\beta[F] = f$.

$M$ has a rank optimum outcome iff $M$ has a unique average optimal outcome (see above). For any feature $G \neq F$, since $N_1[G] = N_2[G]$, $N_3[G] = g$, and $N_4[G] = \bar{g}$, the average optimal value is unique and it is $\alpha[G]$. Therefore, $M$ has a unique average optimal outcome iff the average optimal value for feature $F$ is unique in $M$. This implies that $M$ has a unique average optimal outcome which is rank optimal and optimum, and thus $M$ has a rank optimum outcome.

Hence, $\alpha \in \text{Opt}(M)$, and since $M$ has a unique rank optimum outcome, $M$ has a unique optimum. Hence, $\alpha \in \text{Opt}(M)$.

Observe that the value $\alpha[G]$ is the average optimal value for all features $G \neq F$, because, for all features $G \neq F$, $\alpha[G] = \beta[G]$, and $\alpha$ and $\beta$ are the optimum outcomes of $D(\alpha)$ and $D(\beta)$, respectively. Since $\alpha[F] = f$ and $\beta[F] = \bar{f}$, $\alpha$ is rank optimum in $M$ iff $\alpha = \beta[F] = f$. To conclude, since $M$ contains an odd number of CP-nets, $\alpha$ is rank optimum in $M$ iff $\alpha$ is rank optimum in $M$ (see above).

Is-Rank-Optimal and Is-Rank-Optimum are also in P [38], hence they are P-complete.

Consider the problem Exists-Rank-Optimum: for an mCP-net $M$, decide whether $M$ has rank optimum outcomes.

1A different proof of mCP-nets always having rank optimum outcomes is in [48].

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

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

$M$ has a rank optimum outcome iff $M$ has a unique average optimal outcome (see above). For any feature $G \neq F$, since $N_1[G] = N_2[G]$, $N_3[G] = g$, and $N_4[G] = \bar{g}$, the average optimal value is unique and it is $\alpha[G]$. Therefore, $M$ has a unique average optimal outcome iff the average optimal value for feature $F$ is unique in $M$. This implies that $M$ has a unique average optimal outcome which is rank optimal and optimum, and thus $M$ has a rank optimum outcome.

Hence, $\alpha \in \text{Opt}(M)$, and since $M$ has a unique rank optimum outcome, $M$ has a unique optimum. Hence, $\alpha \in \text{Opt}(M)$.

Observe that the value $\alpha[G]$ is the average optimal value for all features $G \neq F$, because, for all features $G \neq F$, $\alpha[G] = \beta[G]$, and $\alpha$ and $\beta$ are the optimum outcomes of $D(\alpha)$ and $D(\beta)$, respectively. Since $\alpha[F] = f$ and $\beta[F] = \bar{f}$, $\alpha$ is rank optimum in $M$ iff $\alpha = \beta[F] = f$. To conclude, since $M$ contains an odd number of CP-nets, $\alpha$ is rank optimum in $M$ iff $\alpha$ is rank optimum in $M$ (see above).

Is-Rank-Optimal and Is-Rank-Optimum are also in P [38], hence they are P-complete.

Consider the problem Exists-Rank-Optimum: for an mCP-net $M$, decide whether $M$ has rank optimum outcomes.
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 $\alpha_{\sigma_X}$ with $\alpha_{\sigma_X} [x_i^T x_i^F] = x_i^T x_i^F$ 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 $F_i(\phi)$ (see [38]).

In particular, the principles of our reduction are:

(a) For any assignment $\sigma_X$ on $X$, the associated outcome is $\overline{\overline{\sigma_X}}$, where $\sigma_X$ is encoded over features $X$, $\overline{\overline{\sigma_X}}[U_1 U_2] = \overline{\overline{\sigma_X}}[U_1 U_2]$, 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 $\sigma_X$ and $\sigma_Y$ on $X$ and $Y$, respectively, the associated outcome is $\overline{\overline{\sigma_X} \sigma_Y}$, where $\sigma_X$ and $\sigma_Y$ are encoded over features $X$ and $\mathcal{Y}$, respectively, $\overline{\overline{\sigma_X} \sigma_Y}[U_1 U_2] = \overline{\overline{\sigma_X}}[U_1 U_2]$, 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 $\overline{\overline{\sigma_X}}$ and $\overline{\overline{\sigma_X} \sigma_Y}$ are two outcomes such that $\sigma_X \neq \sigma_X'$, then $\overline{\overline{\sigma_X} \sigma_Y}$ does not max dominate $\overline{\overline{\sigma_X}}$. This imposes that $\overline{\overline{\sigma_X}}$ might be max dominated only by an outcome encoding the very same assignment for $X$ of $\overline{\overline{\sigma_X}}$.

(f) For any assignment $\sigma_Y$ on $Y$, if $\overline{\overline{\sigma_X}}$ and $\overline{\overline{\sigma_X} \sigma_Y}$ are two outcomes (encoding the same assignment over $X$), then $\overline{\overline{\sigma_X} \sigma_Y}$ max dominates $\overline{\overline{\sigma_X}}$ 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 $\overline{\overline{\sigma_X}}$ can be max optimal, and $\overline{\overline{\sigma_X}}$ is max optimal iff $\sigma_X$ is an assignment such that $(Y \sigma_Y)(Z \sigma_Y)\phi(X \sigma_X, Y, Z)$ is valid, i.e., iff $\sigma_X$ is a witness of the validity of the quantified formula $\phi$. 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 $\Phi = (\exists X)(\exists Y)(\exists Z)\phi(X, Y, Z)$ be a quantified formula. From $\phi(X, Y, Z)$ we define the $8CP$-net $M(\phi)$ as $\langle N_1, \ldots, N_6 \rangle$ as follows. The features of $M(\phi)$ are:

- The features of a net $F_i(\phi)$ (see [38]) in which we distinguish three variable feature sets $X = \{x_i^T, x_i^F \mid x_i \in X\}$, $Y = \{Y_i^T, Y_i^F \mid y_i \in Y\}$, and $Z = \{Z_i^T, Z_i^F \mid z_i \in Z\}$ ($\mathcal{P}$ and $\mathcal{D}$ are the literal and clause feature sets, respectively, $\mathcal{A}$ is the set of features of the interconnecting net embedded in $F_i(\phi)$ and $A$ is the apex of the interconnecting net);
- Features $Y'$ = $\{Y_i^T \mid y_i \in Y\}$, $Y'' = \{Y_i^F \mid y_i \in Y\}$;
- Features in set $B$, which are the features $B_1$ of an interconnecting net $HC_i(\langle Y' \rangle)$ (see [38]) and its apex is feature $B$.

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

**Figure 1**: A schematic representation for CP-net $N_4$. The expression "($Y_i^T \oplus Y_i^F$)" in the CP table of $Y_i^T$ is satisfied when exactly one feature among $Y_i^T$ and $Y_i^F$ has an overlined value.

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

- $N_1$ is composed by a net $F_i(\phi)$ (for a schematic representation of this net see [38]), in which we distinguish three variable feature sets $X$, $\mathcal{Y}$, 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 $\phi$.

- $N_2$ has, for each $x_i \in X$, the link $(X_i^T, X_i^F)$, and a net $D(\mathcal{Y}', \mathcal{Y}''; \mathcal{P}, \mathcal{A}, \mathcal{B}[U_1, U_2])$. The other CP tables are: for $X_i^T$, $x_i^T > x_i^F$; for $X_i^F$, $x_i^F > x_i^T$ iff $X_i^F$ has value $x_i^F$.

- $N_3$ is similar to $N_2$, but with roles of $X_i^T$ and $X_i^F$ exchanged.

The purpose of these two nets is achieved in conjunction with nets $N_4$ and $N_5$ below. Their aim is supporting Principle (e). For nets $N_2$ and $N_6$, their preferences restricted over $(X_i^T, X_i^F)$ are $x_i^T > x_i^F$, $x_i^F > x_i^T$; while, for nets $N_4$ and $N_5$, their preferences restricted over $(X_i^T, X_i^F)$ are $x_i^T > x_i^T$, $x_i^F > x_i^F$. Therefore, for an outcome $\overline{\overline{\overline{\sigma_X} \sigma_Y}}$, 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 $\overline{\overline{\overline{\sigma_X}}}$ can be max dominated by an outcome $\overline{\overline{\overline{\overline{\sigma_X} \sigma_Y}}}$. Only outcomes $\overline{\overline{\overline{\overline{\sigma_X} \sigma_Y}}}$ with $\sigma_X = \sigma_X'$ may max dominate $\overline{\overline{\overline{\overline{\sigma_X}}}}$, because there will not be contrasting preferences among the agents.

- $N_4$ (see Figure 1) has, for each $y_i \in Y$, the links $(U_1, Y_i^T)$, $(U_1, Y_i^F)$, $(Y_i^T, Y_i')$, $(Y_i^F, Y_i')$, $(Y_i^T, Y_i'')$, $(Y_i^F, Y_i'')$, $(Y_i', Y_i'')$, $(Y_i'', Y_i')$;
The aim of these last three 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 CP-nets.

$M(\phi)$ is acyclic, binary, its indegree is three, and can be computed in polynomial time from $\Phi$. Moreover, the class of mCP-nets derived from quantified formulas of the mentioned kind and according to the reduction shown above is polynomially connected. It is possible to prove the following crucial property of $M(\phi)$:

**Lemma 6.1.** Let $\Phi = (\exists X)(\exists Y)(\exists Z)\phi(X, Y, Z)$ 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, $\Phi$ is valid iff $M(\phi)$ has a max optimal outcome.

**Theorem 6.2.** Let $M$ be an mCP-net. Deciding whether there is a max optimal outcome in $M$ is $\Sigma^P_3$-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.

**Theorem 6.3.** Let $M$ be an mCP-net. Deciding whether there is a max optimal outcome in $M$ is in $\Sigma^P_3$.

### 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 [33]. 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 mCP-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 [33]. Lacy and Niou [28] 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 [31, 32, 56, 57].

To overcome this limitation, $O$-legality was proposed by Lang [31] as a weaker restriction. If $O = (F_1, \ldots, F_n)$ is a sequence of features, a profile of agent preferences is $O$-legal if, for any agent $A$, and any two features $F_i$ and $F_j$, $i < j$ implies that $A$’s preferences for $F_i$ do not depend on $F_j$’s value. When preferences are represented via CP-nets, a profile of CP-nets is $O$-legal if $O$ is a topological order shared among all the CP-nets’ graphs. When $O$ is a common topological order of the CP-nets in a profile of CP-nets, asking the agents to sequentially vote for the single features following the order in $O$ is very natural. In fact, sequential voting over $O$-legal CP-nets has extensively been investigated [31, 32, 56, 57], and $O$-legality of CP-nets was required in various other works, e.g., [14, 18, 42, 43]. An interesting approach to preference aggregation over $O$-legal CP-nets was proposed in [14], where “probabilistic” CP-nets were used to represent the result of the aggregation. However, also $O$-legality is somewhat demanding [36, 51, 55], because it imposes that there are no inversions in the preference dependencies. For example, if in a profile 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 $O$-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, 36, 37, 55]). 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 [31], global voting over (non-$O$-legal) CP-nets has not thoroughly been investigated as sequential voting.

The first work studying global voting over (not necessarily $O$-legal) CP-nets was the one of Rossi et al. [48], 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 [48] were brute-force, hence, those algorithms gave only exponential time upper bounds for most of the global voting tasks over CP-nets, and no hardness result was provided. Algorithms exploiting SAT solvers to compute Pareto and majority optimal outcomes according to global voting over profiles of CP-nets were proposed in [34, 35]. Li et al. [36] extended those results to computing majority optimal outcomes via SAT solvers on cyclic CP-nets, while Li et al. [37] introduced the possibility of multivalued and incomplete CP-nets. Although these works advanced the research on global voting over CP-nets, they did not provide precise complexity results. As already mentioned, the complexity of these problems was reported as open several times [31, 34–37], and only recently a work characterized the exact complexity of some voting tasks over $m$CP-nets [38].

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.

### 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 $O$-legal CP-nets. We have proven that deciding the existence of max optimal and max optimum outcomes is $\Sigma^P_3$-complete and in $\Sigma^P_3$, 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 on big instances.

A possible direction for further research is 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 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’ 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 process, we should take into account what outcomes are feasible. For example, to decide whether an outcome is majority dominated by another, we should check that the latter is actually feasible. A similar idea characterized the solution concepts in NTU cooperative games defined via constraints [19, 21]. This approach could be merged with the definition of constrained CP-nets [4, 46], and a concept of compact representation of constraints (see [24]) could also be introduced. It will also be interesting investigating structural restrictions on the structure of CP-nets, in the spirit of what was done in [7, 20, 22, 23, 26], to identify broader classes of CP-nets where the dominance test is tractable, whereas, in general, over acyclic CP-nets the dominance test is NP-hard.

### ACKNOWLEDGMENTS

This work was partly supported by the EPSRC grant EP/J008346/1 “ProQAW: Probabilistic Ontological Query Answering on the Web” and the EPSRC grant EP/M025268/1 “VADA: Value Added Data Systems – Principles and Architecture”. We thank the anonymous reviewers for their helpful comments.
REFERENCES


