Complexity Results for Preference Aggregation over (m)CP-nets: Max and Rank Voting

Published Version:
Complexity Results for Preference Aggregation over (m)CP-nets: Max and Rank Voting / Lukasiewicz, Thomas; Malizia, Enrico. - In: ARTIFICIAL INTELLIGENCE. - ISSN 0004-3702. - ELETTRONICO. - 303:(2022), pp. 103636.1-103636.34. [10.1016/j.artint.2021.103636]

Availability:
This version is available at: https://hdl.handle.net/11585/840223 since: 2021-12-26

Published:
DOI: http://doi.org/10.1016/j.artint.2021.103636

(Article begins on next page)

Complexity Results for Preference Aggregation over (m)CP-nets: Max and Rank Voting*

Thomas Lukasiewicz[a] and Enrico Malizia[b]

[a]Department of Computer Science, University of Oxford, UK
[b]DISI, University of Bologna, Italy

Abstract

Aggregating preferences over combinatorial domains has a plethora of applications in AI. Due to the exponential nature of combinatorial preferences, compact representations are needed, and conditional ceteris paribus preference networks (CP-nets) are among the most studied compact representation formalisms. Unlike the problem of outcome dominance over individual CP-nets, which received an extensive complexity analysis in the literature, mCP-nets (and global voting/preference aggregation over CP-nets) lacked such a thorough complexity characterization, despite this being reported multiple times in the literature as an open problem. An initial complexity analysis for mCP-nets was carried out only recently, where Pareto and majority dominance semantics were studied. In this paper, we further explore the complexity of mCP-nets, giving a precise complexity analysis of the dominance semantics in mCP-nets when the max and rank voting schemes are considered. In particular, we show that deciding dominance under max voting is Θ_2^{P}-complete, while deciding optimal outcomes and their existence under max voting is complete for Π_2^{P} and Σ_3^{P}, respectively. We also show that, under max voting, deciding optimum outcomes is Π_2^{P}-complete, and deciding their existence is Π_2^{P}-hard and in Σ_3^{P}. As for rank voting, apart from deciding whether mCP-nets have rank optimal outcomes, which is a trivial problem, as all mCP-nets have rank optimal outcomes, all the other rank voting tasks considered are tractable and in P. Interestingly, we show here that these problems are not only in P, but also P-hard (and hence P-complete). Furthermore, we show that deciding whether mCP-nets have Pareto optimum outcomes, which was known to be feasible in polynomial time, is actually P-complete, as well as that various tasks for CP-nets are P-complete. These results provide interesting insights, as P-complete problems are (currently believed to be) inherently sequential, and hence they cannot benefit from highly parallel computations.

1 Introduction

The problem of managing and aggregating agent preferences has attracted extensive interest in the computer science community (see, e.g., the comprehensive survey by Brandt et al. [16]), because methods for representing and reasoning about preferences are very important in AI applications, such as recommender systems [52], (group) product configuration [11, 23, 60], (group) planning [10, 55, 57, 59], (group) preference-based constraint satisfaction [6, 9, 12], and (group) preference-based query answering/information retrieval [7, 20, 46, 47].

Social choice theory, which is the branch of economics studying methods for collective decision making [3, 4], has often been employed in the computer science literature to have a solid theoretical ground upon which to properly build the study of agent preference aggregation. For this reason, social choice has received extensive investigation from its computational perspective. However, the sets of candidates considered in social choice theory are usually (although not always) small in size, and hence in this theory, not much attention has been devoted to the actual ways to represent agent preferences, which has an important impact from a computational perspective. The underlying assumption has often been that agent preferences are extensively represented (see the survey by Brandt et al. [16] and the references therein), and hence most of the results achieved in computational social choice hold for this kind of representations. Extensively representing preferences is completely reasonable when we deal with small sets of candidates, like, e.g., in political elections. However, it is not feasible when the voting domain (i.e., the set of candidates) has a *combinatorial structure* [17, 34, 37], which means that the set of candidates (or *outcomes*) is the Cartesian product of finite value domains for each of a set of *features* (also

*Preliminary results of this paper have appeared in Proc. of AAAI [43] and in Proc. of AAMAS [48].

called *variables*, or *issues*, or *attributes*). The problem of aggregating agents' preferences over combinatorial domains (or *multi-issue* domains) is called a *combinatorial vote* [33, 34].

Interestingly, votes over domains exhibiting a combinatorial structure are rather common. For example, on the day of the 2012 US presidential election, voters in California expressed their preferences also for 11 referenda [37]. Making a joint decision regarding different related issues in a community, like whether and where to build a new swimming pool or library, is another example of a combinatorial vote. Other names given to these voting scenarios are *multiple elections* or *multiple referenda* [13, 17, 36, 37, 65, 66]. Another example is the following group planning scenario. During the exploration of a remote area/planet, a set of autonomous robots, each of which has a specific task to accomplish, coordinate to achieve a common goal. To complete their individual tasks, the robots have their own specific preferences over a vast amount of variables/features emerging from the contingency of the situation, and their individual preferences have to be blended in all together. Only in this way, the overall mission can be successful. These examples show that combinatorial votes are rather important, and hence there is the need of finding ways to represent the combinatorial preferences of agents and algorithms to aggregate them.

By definition, the number of outcomes in a combinatorial domain is exponential in the number of features. For this reason, we need compact representations for combinatorial preferences [34, 37]. CP-nets [8] are a graphical model for representing combinatorial preferences, and they are one of the most studied compact representations for combinatorial preferences, as a vast literature on them demonstrates. In CP-nets, features are represented via vertices of a graph, and an edge from vertex A to vertex B indicates that feature A's value influences the choice of feature B's value. Intuitively, the preferences represented via CP-nets are of the kind "keeping everything else equal, if I am having fish for dinner, then I would rather pair it with a white wine than with a red wine". In this scenario, A is the type of dinner, and B (depending on A) is the type of wine. This kind of preferences are called *conditional ceteris paribus preferences*.

Rossi et al. [53] introduced mCP-nets as an extension of CP-nets to groups of agents. An mCP-net is essentially a *profile* of CP-nets, one for each agent. Voting procedures are at the base of the group dominance semantics of mCP-nets: every agent, with its own CP-net, expresses its preferences for an outcome over another. Various voting schemes were proposed for mCP-nets [42, 53], and different voting schemes give rise to different group dominance semantics for mCP-nets. The specific way in which votes are collected from the agents in order to implement a voting rule is called *voting protocol* [18]. The voting protocol implementing mCP-nets' group dominance semantics is *global voting* [35, 37], which assumes that the whole CP-nets are available during the process of preference aggregation—as the dominance semantics for mCP-nets is global voting over CP-nets, in the following, we use "global voting over CP-nets" and "(group) dominance semantics for mCP-nets" interchangeably. *Sequential voting* is a different voting protocol for CP-nets, in which preference aggregation is carried out feature-by-feature. A comparison between sequential voting and global voting over CP-nets was explicitly asked for in the literature and stated to be highly promising [35]. However, global voting over CP-nets has not received as much attention as sequential voting. A precise complexity analysis of global voting was missing for a long time, as explicitly mentioned several times in the literature [35, 38, 39, 40, 42, 58]. Only recently, a thorough complexity analysis of Pareto and majority global voting over CP-nets was carried out in [45].

In this paper, we continue our thorough complexity investigation started in [45] by considering *max* and *rank voting* as defined by Rossi et al. [53]. We expand our previous work and further explore the complexity of mCP-nets (and hence of global voting over CP-nets). As in our previous work, we study acyclic binary mCP-nets, whose constituent CP-nets are the standard ones (and not partial CP-nets, which instead were allowed in the original definition of mCP-nets [53]). Many works in the literature assume the CP-net profiles to be $\mathcal{O}$-legal, which imposes a common topological order to all the CP-nets of the profile. $\mathcal{O}$-legality is required when sequential voting is considered. Otherwise, voting paradoxes (i.e., the selection in the aggregation process of suboptimal outcomes) can happen. $\mathcal{O}$-legality is a quite stringent constraint, which we can avoid to assume in this work, as it was also not assumed in [45], because we consider global voting. Our main goal in this paper is to carry out a precise complexity analysis of the dominance semantics in mCP-nets when the max and rank voting schemes [53] are considered. For each of these voting schemes, we investigate the complexity of deciding the dominance relation, the complexity of deciding whether a given outcome is optimal or optimum, and the complexity of deciding whether the mCP-net admits optimal or optimum outcomes. In the course of this, we also prove that various tasks for CP-nets are P-complete and that deciding whether mCP-nets have Pareto optimum outcomes, which was known to be feasible in polynomial time, is actually P-complete as well.

The rest of this paper is organized as follows. We provide some preliminaries in the next section, and an overview of the complexity results obtained in this paper in Section 3. Studying the rank semantics over mCP-nets requires that various problems regarding the evaluation of optimal outcomes and outcomes' rank in (individual) CP-nets have been addressed first, and this is carried out in Section 4. We are then ready to investigate the complexity of rank voting over mCP-nets in Section 5. In Section 6, we analyze the complexity of max voting over mCP-nets. In Section 7, we discuss related works, and Section 8 summarizes the main results and gives an outlook on future research. The proofs of two results in Section 6 require several pages and are provided in full detail in the appendix.

2 Preliminaries

In this section, we give some preliminaries on conditional preference nets (CP-nets), CP-nets for groups of m agents (mCP-nets), and the complexity classes that we will encounter in our complexity results.

In this paper, a *preference relation* $\mathcal{R}$ over a set of outcomes $\mathcal{O}$ is a strict order over $\mathcal{O}$, i.e., $\mathcal{R}$ is a binary relation over $\mathcal{O}$ that is irreflexive (i.e., $\langle \alpha, \alpha \rangle \notin \mathcal{R}$), asymmetric (i.e., if $\langle \alpha, \beta \rangle \in \mathcal{R}$, then $\langle \beta, \alpha \rangle \notin \mathcal{R}$), and transitive (i.e., if $\langle \alpha, \beta \rangle \in \mathcal{R}$ and $\langle \beta, \gamma \rangle \in \mathcal{R}$, then $\langle \alpha, \gamma \rangle \in \mathcal{R}$). A *preference ranking* $\mathcal{R}$ is a preference relation that is total (i.e., either $\langle \alpha, \beta \rangle \in \mathcal{R}$ or $\langle \beta, \alpha \rangle \in \mathcal{R}$ for any two different outcomes α and β).

2.1 CP-nets

CP-nets are a formalism to encode conditional ceteris paribus preferences over combinatorial domains.

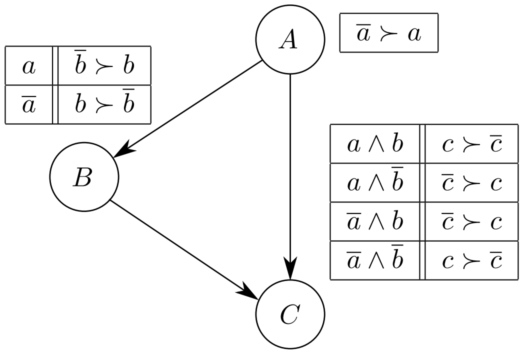
Definition 2.1. A *CP-net* $N = \langle \mathcal{G}_N, \mathit{Dom}_N, (\mathit{CPT}_N^F)_{F \in \mathcal{F}_N} \rangle$ consists of a directed graph $\mathcal{G}_N = \langle \mathcal{F}_N, \mathcal{E}_N \rangle$ whose vertices $\mathcal{F}_N$ represent the *features* of a combinatorial domain, a function Dom_N, and a family of functions $(\mathit{CPT}_N^F)_{F \in \mathcal{F}_N}$, where Dom_N associates a *(value) domain* $\mathit{Dom}_N(F)$ with every feature F, while every function CPT_N^F is the *CP table* for F, defined below.

The value domain of a feature F is the set of all values that F may assume in the possible outcomes. In this paper, features are *binary*, i.e., the domain of each feature F contains exactly two values, usually denoted $\overline{f}$ and f, which we call the *overlined* and the *non-overlined* value (of F), respectively. For a set of features $\mathcal{S}$, $\mathit{Dom}_N(\mathcal{S}) = \times_{F \in \mathcal{S}} \mathit{Dom}_N(F)$ denotes the Cartesian product of the domains of the features in $\mathcal{S}$. Thus, an *outcome* is an element of $\mathcal{O}_N = \mathit{Dom}_N(\mathcal{F}_N)$. Given a feature F and an outcome α, we denote by $\alpha[F]$ the value of F in α, while, given a set of features $\mathcal{F}$, $\alpha[\mathcal{F}]$ is the projection of α to $\mathcal{F}$, i.e., the sub-outcome obtained from α in which only values of features in $\mathcal{F}$ are retained. For two outcomes α and β, and a set of features $\mathcal{F}$, we denote by $\alpha[\mathcal{F}] = \beta[\mathcal{F}]$ that $\alpha[F] = \beta[F]$ for all $F \in \mathcal{F}$; we write $\alpha[\mathcal{F}] \neq \beta[\mathcal{F}]$, otherwise, i.e., when there is at least one feature $F \in \mathcal{F}$ such that $\alpha[F] \neq \beta[F]$.

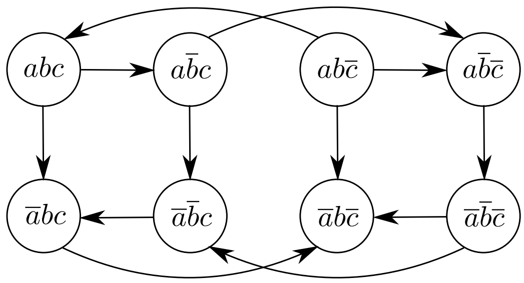
The CP tables encode preferences over feature values. Intuitively, the CP table of a feature F specifies how the values of the parent features of F influence the preferences over the values of F. More formally, for a feature F, we denote by $\mathit{Par}_N(F) = \{G \in \mathcal{F}_N \mid \langle G, F \rangle \in \mathcal{E}_N\}$ the set of all features in $\mathcal{G}_N$ from which there is an edge to F. We call $\mathit{Par}_N(F)$ the set of the *parents* of F (in N). We denote by $\mathit{Ord}_N(F)$ the set of all the preference rankings over the elements of $\mathit{Dom}_N(F)$. Each function $\mathit{CPT}_N^F \colon \mathit{Dom}_N(\mathit{Par}_N(F)) \to \mathit{Ord}_N(F)$ maps every element of $\mathit{Dom}_N(\mathit{Par}_N(F))$ to a preference ranking over the domain of F. If $\mathit{Par}_N(F) = \varnothing$, then CPT_N^F is a single preference ranking over $\mathit{Dom}_N(F)$. Note that indifferences between feature values are not admitted in (classical) CP-nets. Each function CPT_N^F is represented via a two-column table, in which, given a row, the element in the first column is the input value of the function CPT_N^F, and the element in the second column is the associated preference ranking over $\mathit{Dom}_N(F)$. Since CPT_N^F is total, in the table representing its function there is a row for any combination of values of the parent features, i.e., for every feature F, there are $2^{|\mathit{Dom}_N(\mathit{Par}_N(F))|}$ rows in the CP table of F.

In the following, when we define CP tables, we often use a logical notation to identify for which specific values of the parent features a particular row in the CP table has to be considered. Although this is an idea on which generalized propositional CP-nets [24] are based on, here it is used only for notational convenience. In this paper, we always assume that CP tables are explicitly represented in the input instances. In the second column of CP tables, $\overline{f} \succ f$ denotes $\overline{f}$ being preferred to f. In particular, the logical notation $\overline{a} \oplus \overline{b}$ in the first column of CP tables is verified when *exactly* one feature among A and B has an overlined value (recall that our features are always binary). If A, B, and C are three features, with A and B parents of C, when we say that the CP table of C contains "$(a \wedge \overline{b}) \to \overline{c} \succ c$", we mean that in the CP table of C there is a row in which the element in first column is "$a \wedge \overline{b}$" and the element in second column is "$\overline{c} \succ c$" (see the second row of the CP table of C in Figure 1a); and when we add that the CP table of C is "$c \succ \overline{c}$ otherwise", we mean that in all the remaining rows the element in the second column is "$c \succ \overline{c}$". For a feature without parents, like feature A in the CP-net in Figure 1a, we say that its CP table is "$\overline{a} \succ a$", meaning that its CP table is constituted by this single preference ranking (see above).

The preference semantics of CP-nets can be defined in several different but equivalent ways [8]. Here, we adopt the concept of improving (or alternatively worsening) flip [8, Definition 4]: let F be a feature, and let α be an outcome. Intuitively, flipping the value of F in α from $\alpha[F]$ to a different one is an improving flip, if the new value of F is preferred, given the values in α of the parent features of F. More formally, flipping F from $\alpha[F]$ to a different value f' is an *improving flip*, if $f' \succ \alpha[F]$ holds in $\mathit{CPT}_N^F(\alpha[\mathit{Par}_N(F)])$. Given two outcomes α and β



(a) A CP-net N with three features.



(b) The CP-net's induced preference graph G_N.

Figure 1: A CP-net and its induced preference graph.

differing only on the value of a feature F, there is an *improving flip* from α to β, denoted $\alpha \xrightarrow{F}_N \beta$, if flipping the value of F from $\alpha[F]$ to $\beta[F]$ is an improving flip. In the following, we often omit the feature F and simply write $\alpha \rightarrow_N \beta$; and when we say that we flip a feature, then we often mean that the flipping is improving. The *induced preference graph of* N is the graph $G_N = \langle V_N, E_N \rangle$, where the nodes V_N are all the possible outcomes of N, and, given two outcomes $\alpha, \beta \in V_N$, a directed edge from α to β belongs to E_N iff $\alpha \rightarrow_N \beta$.

For an agent whose preferences are encoded through a CP-net N, we say that the agent *prefers* β *to* α, or that β *dominates* α *(in* N*)*, denoted $\beta \succ_N \alpha$, if there is an improving flipping sequence from α to β, or, equivalently, there is a path in the induced preference graph G_N from the vertex associated with α to the vertex associated with β. If for two outcomes α and β, neither $\alpha \succ_N \beta$ nor $\beta \succ_N \alpha$, then α and β are *incomparable* (in N), which is denoted by $\alpha \bowtie_N \beta$. Note here that, as there are no indifferences between feature values in (classical) CP-nets, in the standard dominance semantics, for any two distinct outcomes, either one dominates the other, or they are incomparable.

In a CP-net, an outcome is *optimal* if it is not dominated by any other outcome; while an outcome is *optimum* if it dominates all other outcomes (in which case, it is also optimal).

Example 2.2. Consider the CP-net N and its induced preference graph shown in Figure 1. For the outcomes $\alpha = abc$ and $\beta = \overline{a}bc$, it holds that $\beta \succ_N \alpha$, because $\alpha \xrightarrow{A}_N \beta$. For the outcomes $\alpha = abc$ and $\gamma = ab\overline{c}$, it holds that $\gamma \not\succ_N \alpha$, because there is no path from α to γ in G_N. On the other hand, $\alpha \succ_N \gamma$, because $\gamma \xrightarrow{C}_N \alpha$, and hence it is not the case that $\alpha \bowtie_N \gamma$. Consider now the outcomes $\alpha = abc$ and $\delta = \overline{a}\overline{b}\overline{c}$. Then, $\delta \succ_N \alpha$ by the improving flipping sequence $abc \rightarrow a\overline{b}c \rightarrow a\overline{b}\overline{c} \rightarrow \overline{a}\overline{b}\overline{c}$. By looking at the induced preference graph, we can recognize the outcome $\overline{a}b\overline{c}$ as optimal, because there are no outgoing edges from the associated vertex, and it is also optimum, because there is a path from any vertex to $\overline{a}b\overline{c}$. $\lhd$

A CP-net is *binary*, if all its features are binary. The *in-degree* of a CP-net N is the maximum number of edges entering into a node of the graph $\mathcal{G}_N$. A CP-net N is *singly connected*, if, for any two distinct features G and F, there is at most one path from G to F in $\mathcal{G}_N$. A class $\mathscr{C}$ of CP-nets is *polynomially connected*, if there exists a polynomial p such that, for any CP-net $N \in \mathscr{C}$ and for any two features G and F of N, there are at most $p(\|N\|)$ distinct paths from G to F in $\mathcal{G}_N$, where $\|N\|$ denotes the size of a CP-net N, i.e., the space in terms of bits required to represent the whole net N (which includes features, edges, feature domains, and CP tables). A CP-net N is *acyclic*, if $\mathcal{G}_N$ is acyclic. It is well known that, for acyclic CP-nets N, their induced preference graph G_N is acyclic, the preferences encoded by N are consistent (i.e., there is no outcome α such that $\alpha \succ_N \alpha$), and there is a unique optimal outcome o_N, which is also optimum, that can be computed in polynomial time [8].

Based on the definition of optimal outcome of an acyclic CP-net, the notion of a rank of an outcome can be defined. The term rank is used in the literature to name also different concepts (see Section 7 for more details). However, in this paper, we refer to the definition of Rossi et al. [53]. Intuitively, the rank of an outcome in an acyclic CP-net is a measure of how much worse the outcome is compared to the optimum one. More formally, the *rank* of an outcome α in an acylic CP-net N, denoted $Rank_N(\alpha)$, is the length of the shortest path from α to the optimum outcome o_N in the induced preference graph of N, which is also the least number of improving flips necessary to transform α into o_N. For example, in the CP-net of Example 2.2, the rank of $ab\overline{c}$ is 1, while the rank of abc is 2.

If we compare outcomes according to their rank, we can define a dominance semantics between outcomes in CP-nets that is different from the standard dominance semantics of CP-nets described above. It is easy to exhibit small examples in which the two dominance semantics differ. An interesting difference between the standard dominance semantics and the dominance semantics based on rank comparisons is that, in the latter, outcomes cannot be incomparable, as they are indifferent when they have the same rank [53].

It is known that dominance testing, i.e., deciding, for any two given outcomes α and β, whether $\beta \succ \alpha$, is in NP over polynomially connected classes of acyclic binary CP-nets [8]. However, it is unknown whether dominance testing is in NP over non-polynomially-connected classes of acyclic binary CP-nets. In this respect, Allen's conjecture [1, 2] states that, in general (non-polynomially-connected) acyclic binary CP-nets, if an outcome α dominates an outcome β, then the length of the shortest flipping sequence from β to α is $O(n^2)$, where n is the number of features in the CP-net. This would imply the membership in NP of the problem also for this class of CP-nets. Also, the complexity of dominance testing for non-binary CP-nets is currently unknown. Regarding the tractable cases, it is known that dominance testing can be carried out in polynomial time on acyclic binary CP-nets whose graph is a tree or a polytree (which means that the graph obtained by making undirected the edges of the graph of the CP-net is acyclic) [8]. Regarding the hardness of the dominance test, it is known that the problem is NP-hard already for the quite simple class of acyclic binary singly connected CP-nets whose in-degree is at most three [45]. For an extension of CP-nets, called generalized CP-nets, dominance testing is PSPACE-complete [24].

Given that the dominance semantics based on rank comparisons and the standard dominance semantics are different, one may wonder why use the rank dominance rather than the standard one. As we show later in Section 4, a non-negligible advantage of rank comparison over the standard dominance semantics is that the evaluation of the former can be carried out in polynomial time over any acyclic CP-net, whereas the latter is in general NP-hard over acyclic CP-nets (see above).

In the rest of this paper, we consider only acyclic binary classes of CP-nets. We specify when the class considered is also polynomially connected. When the CP-net N is clear from the context, we often omit the subscript "N" from the notations introduced above.

2.2 mCP-nets

mCP-nets [53] are a formalism to reason about conditional ceteris paribus preferences when a group of multiple agents is considered. Intuitively, an mCP-net is a profile of m (individual) CP-nets, one for each agent of the group. The original definition of mCP-nets also allows for partial CP-nets. Here, we consider only mCP-nets consisting of a collection of standard CP-nets. The difference is that we do not allow for non-ranked features in agents' CP-nets, and hence there is no distinction between private, shared, and visible features (see the work by Rossi et al. [53] for definitions), i.e., all features are ranked in all the individual CP-nets of an mCP-net.

As underlined by Rossi et al. [53], the "m" of an mCP-net stands for multiple agents and also indicates that the preferences of m agents are modeled, so a 3CP-net is an mCP-net with $m = 3$. Formally, in this paper, an mCP-net $\mathcal{M} = \langle N_1, \ldots, N_m \rangle$ consists of m CP-nets $N_1, \ldots, N_m$, all of them defined over the same set of features, which, in turn, have the same domains. If $\mathcal{M}$ is an mCP-net, we denote by $\mathcal{F}_{\mathcal{M}}$ the set of all features of $\mathcal{M}$, and by $Dom_{\mathcal{M}}(F)$ the domain of feature F in $\mathcal{M}$. Given this notation, $\mathcal{F}_{N_i} = \mathcal{F}_{\mathcal{M}}$, for all $1 \leq i \leq m$, and $Dom_{N_i}(F) = Dom_{\mathcal{M}}(F)$, for all features $F \in \mathcal{F}_{\mathcal{M}}$ and all $1 \leq i \leq m$. Although the features of the individual CP-nets are the same, their graphical structures may be different, i.e., the edges between the features in the various individual CP-nets may vary. We underline here that, unlike other papers in the literature, we do *not* impose that the individual CP-nets of the agents share a common topological order (i.e., we do not restrict the profiles of CP-nets to be $\mathcal{O}$-legal).

An *outcome* for an mCP-net is an assignment to all the features of the CP-nets, and given an mCP-net $\mathcal{M}$, we denote by $\mathcal{O}_{\mathcal{M}}$ the set of all the outcomes in $\mathcal{M}$. The preference semantics of mCP-nets is defined through global voting over CP-nets. In particular, via their own individual CP-net, each agent votes whether an outcome dominates another, and hence different ways of considering votes (i.e., different voting schemes) give rise to different group dominance semantics for an mCP-net [42, 53].

Let $\mathcal{M} = \langle N_1, \ldots, N_m \rangle$ be an mCP-net, and let α and β be two outcomes. 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 $S^{\bowtie}_{\mathcal{M}}(\alpha, \beta) = \{i \mid \alpha \bowtie_{N_i} \beta\}$ are the sets of the agents of $\mathcal{M}$ preferring α to β, preferring β to α, and for which α and β are incomparable, respectively. The notion of rank of an outcome can be extended to mCP-nets. In particular, we denote by $Rank_{\mathcal{M}}(\alpha) = \sum_{1 \leq i \leq m} Rank_{N_i}(\alpha)$ the *rank* of α in $\mathcal{M}$ [53].

Consider an mCP-net $\mathcal{M} = \langle N_1, \ldots, N_m \rangle$, and let α and β be two outcomes. The Pareto, max, and rank (and for comparison, also majority) dominance semantics are defined by Rossi et al. [53] as follows:

Pareto: β *Pareto dominates* α, denoted $\beta \succ^{p}_{\mathcal{M}} \alpha$, if *all* the agents of $\mathcal{M}$ prefer β to α, i.e., $|S^{\succ}_{\mathcal{M}}(\beta, \alpha)| = m$.[1]

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

[1] In the literature, this form of Pareto dominance is often also called *strong* Pareto dominance [54].

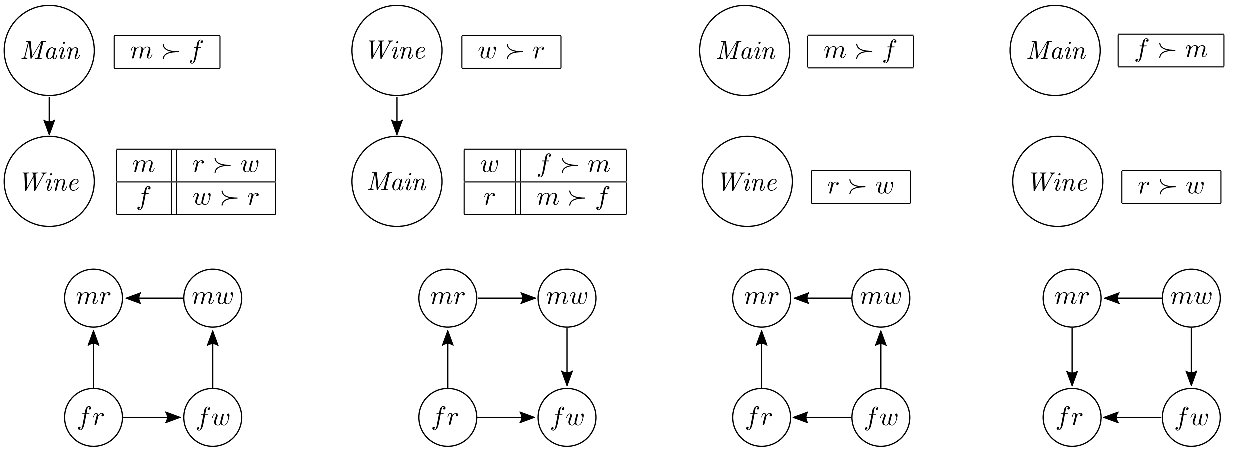


Figure 2: Dinner preferences of Alice, Bob, Chuck, and Deborah (in this order) modeled via CP-nets (above) and their induced preference graphs (below).

Rank: β *rank dominates* α (in $\mathcal{M}$), denoted $\beta \succ^r_{\mathcal{M}} \alpha$, if the rank of β in $\mathcal{M}$ is lower than the rank of α, i.e., $Rank_{\mathcal{M}}(\beta) < Rank_{\mathcal{M}}(\alpha)$.

Observe the difference between the majority and the max dominance semantics. Max dominance is closer to a relative majority dominance semantics. In this semantics, an outcome β dominates an outcome α if the group of agents expressing the preference of β over α is the biggest group, irrespective of the fact that these agents are more than half of the whole voting population. For an analysis of the relationships between these voting schemes, see the work by Rossi et al. [53].

Similarly to what happens for the rank dominance semantics on CP-nets, also for the rank group dominance semantics over mCP-nets outcomes are indifferent when their overall rank is the same. For this reason, any two outcomes are never rank-incomparable in an mCP-net. For the other semantics (i.e., Pareto, majority, and max), outcomes can be incomparable when there is no dominance between them.

For an mCP-net $\mathcal{M}$ and a voting scheme s, an outcome α is s*-optimal (in* $\mathcal{M}$*)*, if there is no outcome β such that $\beta \succ^s_{\mathcal{M}} \alpha$, while α is s*-optimum (in* $\mathcal{M}$*)*, if, for all outcomes $\beta \neq \alpha$, $\alpha \succ^s_{\mathcal{M}} \beta$.

An mCP-net is *acyclic*, *binary*, and *singly connected*, if all its CP-nets are acyclic, binary, and singly connected, respectively. A class $\mathscr{C}$ of mCP-nets is *polynomially connected*, if the set of CP-nets constituting the mCP-nets in $\mathscr{C}$ is a polynomially connected class of CP-nets. The *in-degree* of an mCP-net is the maximum in-degree of its constituent individual CP-nets. Unless stated otherwise, we consider only acyclic binary mCP-nets. We specify when the class considered is also polynomially connected. When the mCP-net $\mathcal{M}$ is clear from the context, we often omit the subscript "$\mathcal{M}$" from the above notations.

Example 2.3. Consider a multi-agent dinner scenario, with agents Alice, Bob, Chuck, and Deborah, expressing their preferences via CP-nets (see Figure 2). The features considered in these CP-nets are the *Main* and the *Wine* for the dinner, where the possible values for *Main* are m and f, denoting "$m$eat" and "$f$ish", respectively, and the possible values for *Wine* are r and w, denoting "$r$ed (wine)" and "$w$hite (wine)", respectively. Observe that the CP-nets in this example are not $\mathcal{O}$-legal, i.e., they do not share a common topological order.

No outcome is Pareto dominated by another, which implies that all outcomes are Pareto optimal. If we consider majority voting, mr majority dominates fr, because Alice, Bob, and Chuck prefer mr to fr. For this reason, fr is not majority optimal. On the other hand, mr does not majority dominates fw, because only Alice and Chuck prefer mr to fw. It is not difficult to verify that no outcome majority dominates fw, and hence fw is majority optimal. If we consider max voting, mr max dominates fw, because Alice and Chuck prefer mr to fw, Bob prefers fw to mr, while mr and fw are incomparable for Deborah. Hence, fw is not max optimal, although it is majority optimal (see above). It is not hard to see that mr is both majority and max optimal. Furthermore, mr is even a max optimum, because it max dominates all other outcomes. Outcomes mw and fr are neither majority optimal nor max optimal. If we consider rank voting, mr rank dominates mw, because mr's total rank is 3, and mw's total rank is 5. For this reason, mw is not rank optimal. Outcomes mr and fr are rank optimal, because there is no other outcome having a lower total rank. $\lhd$

2.3 Complexity classes

We assume that the reader has some background in computational complexity theory, including the notions of Boolean formulas and quantified Boolean formulas, Turing machines, and hardness and completeness of a problem for a complexity class, as can be found, e.g., in the works of Johnson [29] and Papadimitriou [50].

We only briefly recall the complexity classes (and some closely related ones) that we encounter in this paper. P (resp., LOGSPACE, PSPACE, EXPTIME) is the class of all decision problems that can be decided in polynomial time (resp., logarithmic space, polynomial space, exponential time) by a deterministic Turing machine. NP is the class of all decision problems that are decidable in polynomial time by a nondeterministic Turing machine, and co-NP is its complementary class, where 'yes' and 'no' instances are interchanged; NP and co-NP are (currently believed to be) distinct. LOGSPACE, P, PSPACE, and EXPTIME (as they are classes characterized by deterministic machines) are closed under complement, which means that the complement problems reside in the very same classes. The class Θ_2^{P} is the class of all decision problems that can be decided in polynomial time by a deterministic Turing machine using a logarithmic number of calls to an NP oracle; Θ_2^{P} is closed under complement, because the machine calling the NP oracle is deterministic. The class Σ_2^{P} (resp., Σ_3^{P}) is the class of all decision problems that can be decided in polynomial time by a nondeterministic Turing machine using an NP (resp., Σ_2^{P}) oracle, and Π_2^{P} (resp., Π_3^{P}) is the complement of Σ_2^{P} (resp., Σ_3^{P}). Σ_2^{P} and Π_2^{P} (resp., Σ_3^{P} and Π_3^{P}) are (currently believed to be) distinct classes. The class D^{P} (resp., $\mathrm{D}_2^{\mathrm{P}}$) is the class of all problems that are the intersection of a problem in NP (resp., Σ_2^{P}) and a problem in co-NP (resp., Π_2^{P}), more formally, $\mathrm{D}^{\mathrm{P}} = \{L \mid L = L' \cap L'', L' \in \mathrm{NP}, L'' \in \text{co-NP}\}$ (resp., $\mathrm{D}_2^{\mathrm{P}} = \{L \mid L = L' \cap L'', L' \in \Sigma_2^{\mathrm{P}}, L'' \in \Pi_2^{\mathrm{P}}\}$).

The inclusion relationships (which are all currently believed to be strict) for the above-mentioned complexity classes are: $\mathrm{LOGSPACE} \subseteq \mathrm{P} \subseteq \mathrm{NP}, \text{co-NP} \subseteq \mathrm{D}^{\mathrm{P}} \subseteq \Theta_2^{\mathrm{P}} \subseteq \Sigma_2^{\mathrm{P}}, \Pi_2^{\mathrm{P}} \subseteq \mathrm{D}_2^{\mathrm{P}} \subseteq \Sigma_3^{\mathrm{P}}, \Pi_3^{\mathrm{P}} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME}$.[2]

A problem is $\mathcal{C}$-complete for a complexity class $\mathcal{C}$, if the problem belongs to $\mathcal{C}$ and is moreover $\mathcal{C}$-hard. A problem L is $\mathcal{C}$-hard for a complexity class $\mathcal{C} \supseteq \mathrm{NP}$, if all problems in $\mathcal{C}$ can be reduced to L in polynomial time. A problem L is P-hard, if all problems in P can be reduced to L in logarithmic space.

3 Overview of complexity results

We now give an overview of the complexity results obtained in this paper, namely, P-completeness results for CP-nets, and complexity results for the rank and the max dominance semantics in mCP-nets. We recall that, in this paper, we analyze mCP-nets whose constituent CP-nets are standard CP-nets and not partial CP-nets, as in the definition by Rossi et al. [53]; moreover, the (m)CP-nets considered here are always binary and acyclic.

3.1 Rank dominance in CP-nets

To subsequently analyze the complexity of rank voting in mCP-nets, we first explore the complexity of the several problems over CP-nets, namely, deciding whether a feature has a specific value in the optimum outcome of a CP-net (Feat-Value-Opt), deciding whether two CP-nets have the same optimum outcomes (Same-Opt), deciding whether the rank of an outcome does not exceed a given threshold (Rank-Bound), and deciding whether the rank of an outcome is smaller than the rank of another outcome (Compare-Rank), which are here all shown to be P-complete; see Figure 3.

These tractability results (memberships in P) are quite important. Recall from the preliminaries that the concept of outcome rank induces a total non-strict order over the outcomes of a CP-net, as outcomes can be ordered according to their ranks, and that the preference orders induced by the rank semantics and the standard dominance semantics are different. Hence, one may wonder which semantics should be preferred over the other, especially because, up to now, it was believed that the computational effort needed to evaluate the two semantics was similar. Indeed, standard dominance for CP-nets is known to be NP-hard [8], and the algorithm known until now to compute the rank of outcomes over acyclic CP-nets [53], which is necessary to decide rank dominance, requires exponential time. This latter algorithm requires exponential time, as it exhaustively explores the whole space of all outcomes to incrementally compute their ranks [53]. More precisely, the actual algorithm proposed by Rossi et al. [53] to compute the rank of outcomes is thought to work over partial acyclic CP-nets. However, their algorithm deals with the part of the net making the CP-net "partial" in a first phase, while in a second phase the algorithm processes the rest of the partial CP-net as it were a standard CP-net, i.e., the fact of actually being a partial CP-net does not matter here. This means that the algorithm proposed by Rossi et al. [53] can be run on standard CP-nets with only simple adaptations. More specifically, its "second phase" can be run on standard CP-nets without any modification after an initial quick computation that replaces the "first phase". In particular, this initial amendment simply requires to compute (in polynomial time) the optimal outcome of the standard CP-net. The execution time of the "second phase" remains exponential, irrespective of the actual computation carried out before as "first phase".

Interestingly, we show here that, over acyclic binary CP-nets, outcome ranks can be computed in polynomial time, which is a big leap from exponential time. We obtain this by highlighting an interesting property of outcome ranks in acyclic binary CP-nets that has remained unnoticed up to now. In particular, we prove

[2] For these inclusion relationships, the notation "$A \subseteq B, C \subseteq D$" is a shorthand for $A \subseteq B$, $A \subseteq C$, $B \subseteq D$, $C \subseteq D$, $B \nsubseteq C$, and $C \nsubseteq B$.

Problem		Complexity
Feat-Value-Opt:	Given a CP-net N, a feature $F \in \mathcal{F}_N$, and a value $v \in Dom_N(F)$ for F, decide whether $o_N[F] = v$.	P-complete
Same-Opt:	Given two CP-nets N_1 and N_2 (defined over the same set of features, having the same domain in the two nets), decide whether $o_{N_1} = o_{N_2}$.	P-complete
Rank-Bound:	Given a CP-net N, an outcome $\alpha \in \mathcal{O}_N$, and an integer k, decide whether $Rank_N(\alpha) \leq k$.	P-complete
Compare-Rank:	Given a CP-net N and two outcomes $\alpha, \beta \in \mathcal{O}_N$, decide whether $Rank_N(\beta) < Rank_N(\alpha)$.	P-complete

Figure 3: Complexity results obtained in this paper for tasks over CP-nets.

that, in acyclic binary CP-nets, the rank of an outcome α is equal to the number of feature values in α that differ from the respective feature values in the optimum outcome. This property allows to avoid the exhaustive exploration of the space of the outcomes to compute their ranks. Being capable of computing ranks in polynomial time makes it possible to compare outcomes according to their ranks in polynomial time. Therefore, with this result, we now know that an advantage of the rank dominance semantics over the standard dominance semantics in acyclic binary CP-nets is that the former can be evaluated in polynomial time over any class of acyclic binary CP-nets, polynomially connected or not, whereas the latter is NP-hard already for classes of quite simple acyclic binary singly connected CP-nets of in-degree at most three [45].

Furthermore, also the hardness results (the P-hardnesses) are quite interesting. In fact, even if polynomial-time voting schemes are adopted in real systems, autonomous agents often interact with a huge number of peers, and they coordinate and aggregate preferences over even larger domains. This may be tackled by using parallel algorithms on parallel hardware. However, some problems, although solvable in polynomial time, are inherently sequential, and so do not benefit from highly parallel processing [27]. Saying that a problem L does not benefit from highly 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 available [27] (to give a rough example, having two processors, instead of just one, would not halve the algorithm's execution time). Intuitively, this is due to the fact that, in such problems, the intermediate steps needed to compute the final answer essentially have to be performed in sequence, because a step needs the results of the previous ones, before it can be actually executed. P-hard problems are the ones currently believed to be non-parallelizable, as the complexity class P is currently believed to be distinct from NC, which is the class of the highly parallelizable problems [27]. 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 big.

The complexity results of these problems on CP-nets allow us also to show that deciding whether an mCP-net has a Pareto optimum outcome is P-hard (see Figure 4), which is known to be in P, but its hardness was left open [45].

To obtain the P-hardness results, we define and analyze the complexity of the problem Th-CVP: given a Boolean circuit $\mathscr{C}$, a Boolean vector $\mathbf{x}$, and an integer k, decide whether the number of gates of $\mathscr{C}$ evaluating to `true` when $\mathbf{x}$ is given in the input to $\mathscr{C}$ is at most (resp., at least) k. Th-CVP is shown to be P-complete, and so it can be very useful in reductions showing P-hardness of problems involving counting tasks.

3.2 Rank voting in mCP-nets

The rank semantics in CP-nets can be extended to mCP-nets, and in fact the former was introduced for the purpose of defining the rank group dominance semantics in mCP-nets by Rossi et al. [53]. Hence, based on the above P-completeness results for CP-nets, we then explore the complexity of the rank dominance semantics in mCP-nets, namely, the complexity of the problems of deciding rank dominance (Rank-Dominance), of deciding whether an outcome is rank optimal (Is-Rank-Optimal) and whether an mCP-net has a rank optimal outcome (Exists-Rank-Optimal), and of deciding whether an outcome is rank optimum (Is-Rank-Optimum) and whether an mCP-net has a rank optimum outcome (Exists-Rank-Optimum), which are here all shown to be P-complete as well, except for Exists-Rank-Optimal, which is trivial; see Figure 4.

Apart from the problem of deciding whether mCP-nets have rank optimal outcomes, which is shown to be a trivial problem as all mCP-nets have a rank optimal outcome, the tractability of ranks' computation in acyclic CP-nets is a stepping stone for us to show also the tractability of the other tasks for rank voting over mCP-nets, for which only exponential time algorithms were known up to now [53]. Note that also these algorithms by Rossi et al. [53] are tailored for mCP-nets made by partial CP-nets, but these algorithms, again, do not heavily rely on

	Problem		Complexity
	Exists-Pareto-Optimum:	Given an mCP-net $\mathcal{M}$, decide whether $\mathcal{M}$ has a Pareto optimum outcome.	P-complete[+]
Rank	Rank-Dominance:	Given an mCP-net $\mathcal{M}$ and two outcomes $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$, decide whether $\beta \succ^{r}_{\mathcal{M}} \alpha$.	P-complete
	Is-Rank-Optimal:	Given an mCP-net $\mathcal{M}$ and an outcome $\alpha \in \mathcal{O}_{\mathcal{M}}$, decide whether α is a rank optimal outcome in $\mathcal{M}$.	P-complete
	Exists-Rank-Optimal:	Given an mCP-net $\mathcal{M}$, decide whether $\mathcal{M}$ has a rank optimal outcome.	$\Theta(1)$[*]
	Is-Rank-Optimum:	Given an mCP-net $\mathcal{M}$ and an outcome $\alpha \in \mathcal{O}_{\mathcal{M}}$, decide whether α is rank optimum in $\mathcal{M}$.	P-complete
	Exists-Rank-Optimum:	Given an mCP-net $\mathcal{M}$, decide whether $\mathcal{M}$ has a rank optimum outcome.	P-complete
Max	Max-Dominance:	Given an mCP-net $\mathcal{M}$ and two outcomes $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$, decide whether $\beta \succ^{x}_{\mathcal{M}} \alpha$.	Θ_2^{P}-complete
	Is-Max-Optimal:	Given an mCP-net $\mathcal{M}$ and an outcome $\alpha \in \mathcal{O}_{\mathcal{M}}$, decide whether α is max optimal in $\mathcal{M}$.	Π_2^{P}-complete
	Exists-Max-Optimal:	Given an mCP-net $\mathcal{M}$, decide whether $\mathcal{M}$ has a max optimal outcome.	Σ_3^{P}-complete
	Is-Max-Optimum:	Given an mCP-net $\mathcal{M}$ and an outcome $\alpha \in \mathcal{O}_{\mathcal{M}}$, decide whether α is max optimum in $\mathcal{M}$.	Π_2^{P}-complete
	Exists-Max-Optimum:	Given an mCP-net $\mathcal{M}$, decide whether $\mathcal{M}$ has a max optimum outcome.	Π_2^{P}-hard, in Σ_3^{P}

Figure 4: Complexity results obtained in this paper for global voting over CP-nets. The memberships to complexity classes above P are valid for polynomially connected classes of acyclic binary mCP-nets. [+]Membership result in [45]. [*]A different proof is provided in [53].

the fact that the input CP-nets are partial (see above). We prove the tractability of these problems by showing another interesting property, in this case for rank optimal outcomes. In particular, we prove that a rank optimal outcome α for an acyclic binary mCP-net can be computed feature by feature in polynomial time, and the value of each feature F in α is the one appearing most frequently in the optimum outcomes of the individual CP-nets constituting the mCP-net.

By these results, rank voting is easier to compute than majority, max, and Pareto (in most cases). Interestingly, also for these problems, we show here that these problems are not only in P, but they are also P-hard (and hence P-complete).

3.3 Max voting in *m*CP-nets

We finally explore the complexity of the max dominance semantics in mCP-nets, namely, the complexity of deciding max dominance in mCP-nets (Max-Dominance), of deciding whether an outcome is max optimal (Is-Max-Optimal) and whether an mCP-net has a max optimal outcome (Exists-Max-Optimal); and of deciding whether an outcome is max optimum (Is-Max-Optimum) and whether an mCP-net has a max optimum outcome (Exists-Max-Optimum). The precise complexity of these problems is shown to range from Θ_2^{P} (Max-Dominance) over Π_2^{P} (Is-Max-Optimal and Is-Max-Optimum) to Σ_3^{P} (Exists-Max-Optimal), and the problem Exists-Max-Optimal is shown to be Π_2^{P}-hard and to belong to Σ_3^{P}; see Figure 4. The membership parts of these complexity results hold for polynomially connected classes of (acyclic binary) mCP-nets, whereas the above memberships in P of the rank voting tasks over mCP-nets hold also for non-polynomially connected classes of (acyclic binary) mCP-nets.

The Θ_2^{P}-completeness of Max-Dominance is an interesting result, given that majority dominance is NP-complete [45] (this and also the other complexity results of [45] about majority voting mentioned in the following hold for polynomially connected classes of acyclic binary mCP-nets). This increase in complexity is due to the need in max dominance of precisely counting the number of agents preferring an outcome to another, as this is needed to evaluate the size of the biggest group, while this precision is not required in majority voting. Observe that, if on the one hand, majority dominance is co-NP-hard even over classes of mCP-nets having a bounded number of agents [45], on the other hand, max dominance cannot be Θ_2^{P}-hard over classes of mCP-nets having a bounded number of agents. In fact, the most difficult problems of the complexity class Θ_2^{P} need for their solution logarithmically-many calls to the NP oracle, and an essential part of the hardness of max dominance is counting the exact number of agents preferring an outcome to another. If we considered a class of

mCP-nets with a bounded number of agents, then it would be possible to count the agents' preferences through a constant number of calls to a suitable NP oracle, which would imply that, over this specific class of instances, max dominance would not be among the most difficult problems of Θ_2^{P}. This increased complexity of max dominance carries over to the complexity of deciding max optimality and deciding the existence of max optimal outcomes. Indeed, these two problems are complete for Π_2^{P} and Σ_3^{P}, respectively, while the corresponding ones for majority voting are complete for co-NP and Σ_2^{P}, respectively. Deciding whether an outcome is max optimum is Π_2^{P}-complete, which is the same complexity of deciding majority optimum outcomes.

One may wonder why deciding majority optimum outcomes is more complex than deciding majority optimality (complete for Π_2^{P} and co-NP, respectively), whereas the complexity of recognizing max optimum and optimal outcomes is the same (Π_2^{P}-complete). To explain this, observe first that for both majority and max voting, to *disprove* an outcome α to be (1) optimal or (2) optimum, it is sufficient to find a different outcome β that either dominates α (to disprove α being optimal), or that is not dominated by α (to disprove α being optimum). The key point here is the complexity of the dominance test, and more specifically whether checking dominance (needed for task (1)) and checking non-dominance (needed for task (2)) can be carried out in the same complexity class or not. For majority voting, checking dominance is NP-complete. This is a non-deterministic class, and hence there is an asymmetry between the complexity of checking dominance and the complexity of checking non-dominance (complete for NP and co-NP, respectively). When we have to decide whether an outcome α is (not) majority optimal, we can find a disprover outcome β majority dominating α. This can be carried out by an NP machine that guesses β along with a witness for β majority dominating α, and then checks the correctness of the guess. When we have to decide whether an outcome α is (not) majority optimum, we need to find a disprover outcome β that is not majority dominated by α. Also in this case, we can guess β via an NP machine, however, now checking that α does not majority dominate β requires a co-NP test, and since NP and co-NP are (believed to be) distinct classes, this check cannot be carried out by the same NP machine guessing β, but this check has to be delegated to a co-NP oracle. From this, it follows that the problem is in Π_2^{P}. On the other hand, there is no complexity asymmetry between checking max dominance and checking max non-dominance, because Θ_2^{P} is closed under complement. For this reason, to decide both max optimal and max optimum outcomes, after guessing (in NP) the disprover outcome β, there is the need to carry out a check in $\Theta_2^{\mathrm{P}} = \mathrm{P}^{\mathrm{NP}[O(\log n)]}$ in both cases. The P part of $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$ can be performed by the very same machine having guessed β, while the NP part of $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$ can be performed by an NP oracle. Thus, deciding max optimal and max optimum outcomes are both in Π_2^{P}. To conclude the overview of our results for max voting, we show in the paper that deciding whether an mCP-net has a max optimum outcome is between Π_2^{P} and Σ_3^{P}, whereas for majority voting, deciding the existence of optimums is between Π_2^{P} and $\mathrm{D}_2^{\mathrm{P}}$. The above results support that adopting the relative majority flavor of the max semantics requires an increased computational complexity compared to the complexity of the majority semantics. As evidenced in our work, the harder computational complexity of the max semantics is due to the necessity of evaluating the precise size of the biggest group of agents expressing the same preference over a pair of outcomes.

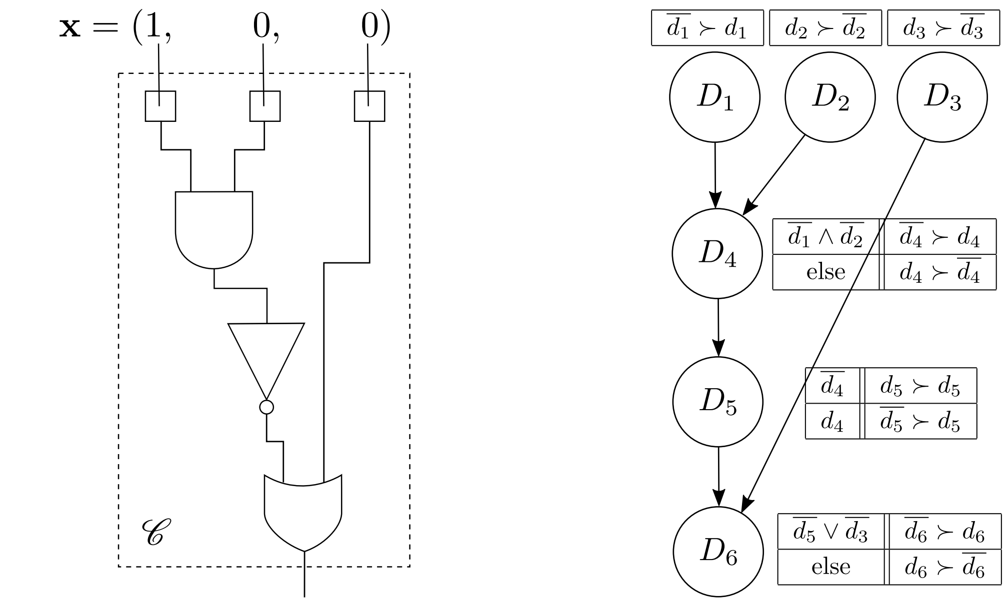
4 Optimal outcomes and rank dominance in CP-nets

This section focuses on optimal outcomes and the rank dominance semantics in CP-nets. As for the former, we are interested in analyzing the complexity of decision problems related to the computation of the optimum outcome in CP-nets, like, for example, deciding whether a feature has a specific value in the optimum outcome (Feat-Value-Opt), or decide whether two CP-nets have the same optimum outcome (Same-Opt). As for the rank dominance semantics, we investigate the complexity of computing the rank of outcomes (Rank-Bound), which sheds light also on the complexity of comparing outcomes according to their rank (Compare-Rank).

4.1 Preliminaries

To prove the P-completeness results of this section, we will exploit the P-completeness of the classical *circuit value problem* (CVP) [30]: given a Boolean circuit $\mathscr{C}$ and a Boolean vector $\mathbf{x}$, decide whether $\mathscr{C}$'s output is `true` when receiving the Boolean vector $\mathbf{x}$ as input. In the literature, different ways to represent circuits were illustrated. Here, we use a representation that is a mix of those adopted by Ladner [30], Serna [56], and Miyano et al. [49]. A circuit $\mathscr{C} = (C_1, \dots, C_m)$ is a sequence of logic gates C_i, which are represented through formulas:

- if the formula is '$C_i = x_j$', then C_i is an input gate fed with the j^{th} input bit;
- if the formula is '$C_i = C_j \wedge C_k$', then C_i is an AND gate, whose inputs are the outputs of the (non-necessarily distinct) gates C_j and C_k (with $j, k < i$);
- if the formula is '$C_i = C_j \vee C_k$', then C_i is an OR gate, whose inputs are the outputs of the (non-necessarily distinct) gates C_j and C_k (with $j, k < i$); and



(a) An instance of the problem CVP, consisting of a circuit $\mathscr{C}$ and an input vector $\mathbf{x}$.

(b) The CP-net $N(\mathscr{C}, \mathbf{x})$ for the circuit $\mathscr{C}$ and the input vector $\mathbf{x}$ of Figure (a).

Figure 5: An instance of the problem CVP (a) and its transformation into a CP-net (b).

- if the formula is '$C_i = \neg C_j$', then 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 $v_{\mathscr{C}}(C_i, \mathbf{x})$, are defined as usual.

In this paper, we assume that the problem CVP is defined as in [27]: 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 of Boolean values, and $C_{out} \in \mathscr{C}$ is the output gate, is a "yes"-instance iff $v_{\mathscr{C}}(C_{out}, \mathbf{x}) = \texttt{true}$, or iff $v_{\mathscr{C}}(C_{out}, \mathbf{x}) = \texttt{false}$, since CVP $\in$ P, and P is closed under complement. We consider the task of checking the circuit's output to be either `true` or `false` depending on which is the most suitable for our aims. CVP is known to be P-complete, and its hardness holds even if various restrictions are issued over the circuit structure, among which the acyclicity of the circuit and that gates have fan-in 2, and even if the output is fixed to be C_m [27, 30, 49].

For the following results, we need CP-nets mimicking the behavior of Boolean circuits when specific vectors are in the input. Let $\mathscr{C} = (C_1, \dots, C_m)$ be a circuit and $\mathbf{x} = (x_1, \dots, x_n)$ 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 transformation's intuition is that values $\overline{d_i}$ and d_i of D_i have the meaning "feature 'active' when the value is overlined", and are hence associated with gate C_i evaluating to `true` and `false`, respectively.

The edges between the features and the CP tables of the features are the following.

- If C_i is an input gate with '$C_i = x_j$', then there is no edge entering in D_i. If $x_j = \texttt{true}$, then D_i's CP table is $\overline{d_i} \succ d_i$; and if $x_j = \texttt{false}$, then D_i's CP table is $d_i \succ \overline{d_i}$.
- If C_i is an AND gate, with '$C_i = C_j \wedge C_k$', then, if $C_j \neq C_k$, there are two edges entering in D_i, one from D_j and one from D_k; if $C_j = C_k$, there is one edge from D_j to D_i. If $C_j \neq C_k$, then the CP table of D_i contains $(\overline{d_j} \wedge \overline{d_k}) \to \overline{d_i} \succ d_i$, and is $d_i \succ \overline{d_i}$, otherwise. If $C_j = C_k$, then the CP table of D_i contains $(\overline{d_j}) \to \overline{d_i} \succ d_i$, and $(d_j) \to d_i \succ \overline{d_i}$.
- If C_i is an OR gate, with '$C_i = C_j \vee C_k$', then, if $C_j \neq C_k$, there are two edges entering in D_i, one from D_j and one from D_k; if $C_j = C_k$, there is one edge from D_j to D_i. If $C_j \neq C_k$, the CP table of D_i contains $(\overline{d_j} \vee \overline{d_k}) \to \overline{d_i} \succ d_i$, and is $d_i \succ \overline{d_i}$, otherwise. If $C_j = C_k$, then the CP table of D_i contains $(\overline{d_j}) \to \overline{d_i} \succ d_i$ and $(d_j) \to d_i \succ \overline{d_i}$.
- If C_i is a NOT gate with '$C_i = \neg C_j$', then there is an edge from D_j to D_i. The CP table of D_i contains $(\overline{d_j}) \to d_i \succ \overline{d_i}$ and $(d_j) \to \overline{d_i} \succ d_i$.

Observe that $N(\mathscr{C}, \mathbf{x})$ is binary, acyclic, its in-degree is two, and can be computed in logarithmic space from $\mathscr{C}$ and $\mathbf{x}$ (because the in-degree 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 as well).

Example 4.1. Consider the Boolean function $\neg(x_1 \wedge x_2) \vee x_3$. The circuit $\mathscr{C}$ encoding this function is shown in Figure 5a, where also the input vector $\mathbf{x} = (1, 0, 0)$ is evidenced at the top. For this specific instance of CVP, made by the circuit $\mathscr{C}$ and the input vector $\mathbf{x}$, the corresponding CP-net $N(\mathscr{C}, \mathbf{x})$ is shown in Figure 5b. Recall that the CP-net $N(\mathscr{C}, \mathbf{x})$ is obtained starting from a circuit *and* an input vector for the circuit. The input vector is necessary to define the CP tables of the features without parents in the CP-net. $\triangleleft$

The CP-nets $N(\cdot, \cdot)$ faithfully replicate the behavior of Boolean circuits. In particular, in $N(\mathscr{C}, \mathbf{x})$, every feature D_i has the value $\overline{d_i}$ in the optimum outcome iff $v_{\mathscr{C}}(C_i, \mathbf{x}) = \texttt{true}$, as expressed by the following result.

Lemma 4.2. *Let $\mathscr{C} = (C_1, \dots, C_m)$ be a circuit, and $\mathbf{x}$ be an input vector. For any gate C_i, $v_{\mathscr{C}}(C_i, \mathbf{x}) = \texttt{true}$ iff $o_{N(\mathscr{C},\mathbf{x})}[D_i] = \overline{d_i}$.*

Proof. We perform an induction on the gates' levels in $\mathscr{C}$. We partition $\mathscr{C}$'s gates into levels as follows. The input gates of $\mathscr{C}$ are at level 0. The non-input gates C_i of $\mathscr{C}$ are at level $\ell + 1$ iff the gates from which C_i receives its input are at most at level ℓ, and at least one of them is at level ℓ. Since there is a one-to-one correspondence between gates in $\mathscr{C}$ and features in $N(\mathscr{C}, \mathbf{x})$, we can speak about levels of features in $N(\mathscr{C}, \mathbf{x})$ as well. In particular, a feature D_i in $N(\mathscr{C}, \mathbf{x})$ is at level ℓ iff the gate C_i is at level ℓ in $\mathscr{C}$.

Base of induction: Consider level 0, which is the level of input gates in $\mathscr{C}$, and let C_i be any input gate of $\mathscr{C}$. By definition of $N(\mathscr{C}, \mathbf{x})$, the feature D_i has the value $\overline{d_i}$ in the optimum outcome iff $v_{\mathscr{C}}(C_i, \mathbf{x}) = \texttt{true}$.

Inductive hypothesis: Assume that for any gate C_i at level at most $\ell - 1$, the feature D_i has the value $\overline{d_i}$ in the optimum outcome iff $v_{\mathscr{C}}(C_i, \mathbf{x}) = \texttt{true}$.

Inductive step: Consider any gate C_i at level ℓ. We show that the feature D_i has the value $\overline{d_i}$ in the optimum outcome iff $v_{\mathscr{C}}(C_i, \mathbf{x}) = \texttt{true}$. Assume that C_i is an AND gate, and denote by C_j and C_k the gates whose output are wired to the input of C_i. By the definition of the CP table of D_i, feature D_i has value $\overline{d_i}$ in the optimum outcome iff both D_j and D_k have overlined values in the optimum outcome. Since D_j and D_k are at most at level $\ell - 1$, by the inductive hypothesis, they have an overlined value in the optimum outcome iff the values of the associated gates are $\texttt{true}$. Thus, the feature D_i has the value $\overline{d_i}$ in the optimum outcome iff $v_{\mathscr{C}}(C_i, \mathbf{x}) = \texttt{true}$. Similarly, it can be shown that this property holds also for the OR and NOT gates at level ℓ. $\square$

We now focus on the problem of counting the number of a circuit's gates evaluating to $\texttt{true}$ when a vector is given in the input to the circuit (this is the problem ϵ-CTGP by Serna [56]), and in particular we analyze its decision variant. Let $TG(\mathscr{C}, \mathbf{x})$ denote the number of $\mathscr{C}$'s gates evaluating to $\texttt{true}$ when $\mathbf{x}$ is given in input to $\mathscr{C}$, i.e., $TG(\mathscr{C}, \mathbf{x}) = |\{C_i \in \mathscr{C} \mid v_{\mathscr{C}}(C_i, \mathbf{x}) = \texttt{true}\}|$. Consider the following problem THRESHOLD-CVP (TH-CVP):

Problem:	TH-CVP
Instance:	A Boolean circuit $\mathscr{C}$, an input vector $\mathbf{x}$, and an integer k.
Question:	Does $TG(\mathscr{C}, \mathbf{x}) \leq k$ hold?

The following result shows in particular that the problem TH-CVP is P-complete. Observe that, since P is closed under complement, also deciding whether $TG(\mathscr{C}, \mathbf{x}) > k$ is P-complete.

Theorem 4.3. *Given a Boolean circuit $\mathscr{C}$, an input vector $\mathbf{x}$, and an integer k, deciding whether $TG(\mathscr{C}, \mathbf{x}) \leq k$ is* P*-complete.*

Proof. TH-CVP is in P, because gates' values can be evaluated in polynomial time [27, 30], and then we can count those evaluating to $\texttt{true}$ and compare the count with k (in polynomial time).

P-hardness is shown via a reduction from CVP, similar to the one used to prove the P-hardness of ϵ-CTGP. Consider the following transformation of an instance $\langle \mathscr{C}, \mathbf{x}, C_{out} \rangle$ of CVP into an instance $\langle \mathscr{C}', \mathbf{x}', k \rangle$ of TH-CVP. Assume that $\mathscr{C} = (C_1, \dots, C_m)$. The circuit $\mathscr{C}' = (C'_1, \dots, C'_{2m})$ consists of $2m$ gates, whose first m gates are identical (for function and wiring) to those of $\mathscr{C}$. The remaining m gates of $\mathscr{C}'$ 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 $k = m - 1$. The reduction can be computed in logarithmic space. 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 $\texttt{false}$.

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

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

4.2 Complexity of optimum outcomes in CP-nets

In this section, we analyze the complexity of tasks on CP-nets that are associated with the computation of the optimum outcomes. We start by looking at the problem of computing the optimum outcome in a CP-net, and in particular on its decision variant FEAT-VALUE-OPT. The following result shows that it is P-complete.

Theorem 4.4. *Given an acyclic binary CP-net N, an outcome $\alpha \in \mathcal{O}_N$, a feature $F \in \mathcal{F}_N$, and a value $v \in Dom_N(F)$, deciding whether $o_N[F] = v$ is* P-complete. *Hardness holds even on CP-nets with in-degree two.*

Proof. As for membership in P, observe that o_N can be computed in polynomial time in acyclic binary CP-nets [8], and then we can verify whether $o_N[F] = v$.

Hardness for P holds by a reduction from CVP. Let $\langle \mathscr{C}, \mathbf{x}, C_{out} \rangle$ be an instance of CVP, and consider the instance $\langle N, \widetilde{F}, val \rangle$ of FEAT-VALUE-OPT, where $N = N(\mathscr{C}, \mathbf{x})$, $\widetilde{F} = D_{out}$, and $val = \overline{d_{out}}$. The transformation can be computed in logarithmic space. By Lemma 4.2, $v_{\mathscr{C}}(C_{out}, \mathbf{x}) = \texttt{true}$ iff $o_N[D_{out}] = \overline{d_{out}} = val$. □

We next focus on the problem of deciding whether two CP-nets have the same optimum outcomes, namely, the problem SAME-OPT. We will use the complexity of this problem to show the complexity of deciding whether an mCP-net has a Pareto optimum outcome.

We show that SAME-OPT is P-complete. In particular, the P-hardness holds by a reduction from CVP, encoding 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, whereas 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 4.5. *Given two acyclic binary CP-nets N_1 and N_2 defined over the same set of features, having the same domain in the two nets, deciding whether $o_{N_1} = o_{N_2}$ is* P-complete. *Hardness holds even on CP-nets with in-degree two.*

Proof. Membership in P is again obtained by the fact that computing the optimum outcome of an acyclic binary CP-net is feasible in polynomial time [8], and then we can compare the computed outcomes.

The P-hardness follows from a reduction from CVP. Let $\langle \mathscr{C}, \mathbf{x}, C_{out} \rangle$ be an instance of CVP, and consider the instance $\langle N_1, N_2 \rangle$ of SAME-OPT computed as follows. The nets N_1 and N_2 are very similar: N_1 contains within itself a net $N(\mathscr{C}, \mathbf{x})$, plus a fresh feature O. The feature O has an entering edge from the feature D_{out}, and the CP table of O contains $(\overline{d_{out}}) \to \overline{o} \succ o$ and $(d_{out}) \to o \succ \overline{o}$. Also N_2 contains within itself a net $N(\mathscr{C}, \mathbf{x})$, plus feature O. However, in N_2, the feature O is not linked to any other feature, and its CP table is $\overline{o} \succ o$. Observe that N_1 and N_2 can be computed in logarithmic space, and they are defined over the same set of features, which, in turn, have the same domain in the two nets. By the definition of N_1 and N_2, for any feature $F \neq O$, $o_{N_1}[F] = o_{N_2}[F]$, which implies that $o_{N_1} = o_{N_2}$ iff $o_{N_1}[O] = o_{N_2}[O]$. By the definition of N_2, $o_{N_2}[O] = \overline{o}$. It is not difficult to see that, by construction, $o_{N_1}[O] = \overline{o}$ iff $o_{N_1}[D_{out}] = \overline{d_{out}}$ iff $\langle \mathscr{C}, \mathbf{x}, C_{out} \rangle$ is a "yes"-instance of CVP. □

4.3 Complexity of outcomes' rank in CP-nets

In this section, we study the problems on CP-nets relative to the rank of outcomes. We first focus on the problem of, given a CP-net and an outcome, deciding whether the rank of the outcome does not exceed a given threshold (RANK-BOUND).

To show that this problem is in P, we first prove the following characterization of the rank of an outcome: the rank of an outcome α in an acyclic binary net N is equal to the number of features F having in α a value that is different from the one that F has in the optimum outcome of N.

Lemma 4.6. *Let N be an acyclic binary CP-net, and let $\alpha \in \mathcal{O}_N$ be an outcome. Then,*

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

Proof. First, in any flipping sequence from α to o_N in G_N, the features F for which $\alpha[F] \neq o_N[F]$ have to be flipped at least once. Therefore, a flipping sequence from α to o_N in which, (1) features F such that $\alpha[F] \neq o_N[F]$ are flipped exactly once, and (2) features F such that $\alpha[F] = o_N[F]$ are never flipped, is one of the shortest.

We now show that flipping sequences in G_N from α to o_N satisfying the above two conditions actually exist. Since N is acyclic, there exist topological orders of its features. Let $(F_1, \dots, F_n)$ be any topological order of the features of N. Consider the following sequence of flips, which we show next to be actually improving in N. Each feature F_i is processed in turn according to the topological order given, and if $\alpha[F_i] \neq o_N[F_i]$, then F_i is flipped, otherwise F_i is left as it is. This procedure flips exactly once features F such that $\alpha[F] \neq o_N[F]$, and it never flips features F such that $\alpha[F] = o_N[F]$. Therefore, this sequence of flips, if it is actually improving in N, must be one of the shortest improving flipping sequences from α to o_N in N (see above).

Assume that the proposed flipping sequence is $\pi\colon \delta_0 \to \ldots \to \delta_k$, where $\delta_0 = \alpha$ and $\delta_k = o_N$, and assume by contradiction that π is not improving in N. This implies that there is a feature's flip that is not improving. Let i be the first step in π characterized by a non-improving flip, and let F_j (with j not necessarily equal to i) be the feature flipped in the i-th step. Since features are considered according to a topological order, if a parent of F_j is flipped in π, then it is flipped before F_j, and it is flipped to match its value in o_N. This means that $\delta_i[Par(F_j)] = o_N[Par(F_j)]$. However, if flipping F_j in δ_i from $\delta_i[F_j]$ to $o_N[F_j]$ is not improving, then the value $\delta_i[F_j]$ is better than $o_N[F_j]$. Because $\delta_i[Par(F_j)] = o_N[Par(F_j)]$, the value of F_j can be improved in o_N as well, which means that o_N is not optimal: a contradiction. Thus, π is actually an improving flipping sequence in G_N, and, since it fulfills the mentioned two conditions, it is one of the shortest improving flipping sequences from α to o_N, which proves the statement. □

We next show that Rank-Bound is P-complete. The membership in P follows from the characterization of the rank of an outcome via Equation (1) and the P-hardness from the P-hardness of Th-CVP, from Lemma 4.2, 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 4.7. *Given an acyclic binary CP-net N, an outcome $\alpha \in \mathcal{O}_N$, and an integer k, deciding whether $Rank_N(\alpha) \leq k$ is* P-complete. *Hardness holds even on CP-nets with in-degree two.*

Proof. Membership in P follows from the fact that computing $Rank_N(\alpha)$ in acyclic binary CP-nets is feasible in polynomial time (by Lemma 4.6), and then we can compare it with k.

We show the P-hardness via a reduction from Th-CVP. Consider the reduction transforming an instance $\langle \mathscr{C}, \mathbf{x}, k\rangle$ of Th-CVP into the instance $\langle N, \alpha, k'\rangle$ of Rank-Bound as follows: $N = N(\mathscr{C}, \mathbf{x})$, α is the outcome assigning non-overlined values to all features, and $k' = k$. The reduction is computable in logarithmic space.

By Lemma 4.2, the number of features having overlined values in o_N equals $TG(\mathscr{C}, \mathbf{x})$. By Equation (1), $Rank_N(\alpha)$ is precisely the number of features having overlined values in o_N, and hence $Rank_N(\alpha) = TG(\mathscr{C}, \mathbf{x})$. Thus, since $k' = k$, $TG(\mathscr{C}, \mathbf{x}) \leq k$ iff $Rank_N(\alpha) \leq k'$. □

We finally look at the problem of comparing the rank of two outcomes (Compare-Rank). This analysis will allow us to characterize the complexity of rank dominance over mCP-nets.

We now prove that Compare-Rank is P-complete. In particular, the P-hardness can be shown via a reduction from Feat-Value-Opt. Indeed, 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 4.8. *Given an acyclic binary CP-net N and two outcomes $\alpha, \beta \in \mathcal{O}_N$, deciding whether $Rank_N(\beta) < Rank_N(\alpha)$ is* P-complete. *Hardness holds even on CP-nets with in-degree two.*

Proof. Membership in P follows from the fact that computing outcome ranks in acyclic binary CP-nets is feasible in polynomial time (by Lemma 4.6), and then we can compare them.

We show the P-hardness of Rank-Bound 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., $|Rank_{N'}(\beta) - Rank_{N'}(\alpha)| = 1$. It is not difficult to see that $Rank_{N'}(\beta) < Rank_{N'}(\alpha)$ iff $o_N[F] = f = v$ iff $\langle N, F, v\rangle$ is a "yes"-instance of Feat-Value-Opt. □

4.4 Complexity of the existence of Pareto optimum outcomes in mCP-nets

We now look at the complexity of deciding the existence of Pareto optimum outcomes in mCP-nets (Exists-Pareto-Optimum).

We know that an acyclic binary mCP-net $\mathcal{M}$ has a Pareto optimum outcome iff all the individual CP-nets of $\mathcal{M}$ have the very same individual optimum outcome [45, Lemma 4.9]. By this, the P-hardness of Exists-Pareto-Optimum follows from the P-hardness of Same-Opt.

Theorem 4.9. *Given an acyclic binary mCP-net $\mathcal{M}$, deciding whether there is a Pareto optimum outcome in $\mathcal{M}$ is* P-complete. *Hardness holds even on mCP-nets with in-degree at most two and at most two agents.*

The non-parallelizability of Exists-Pareto-Optimum tightly depends on the non-parallelizability of the task of computing the optimum of a CP-net (see Theorem 4.4). That difficulty is linked to the intricacy of the net and the number of features. In complex environments, where agents have to deal with many features, computing the optimum of a CP-nets could manifest challenges, as parallel algorithms cannot be exploited.

5 Rank voting

In this section, we analyze the complexity of rank voting tasks over mCP-nets. First, we focus on deciding rank dominance (Rank-Dominance). Next, we characterize the complexity of rank optimality, namely, the problems of deciding whether an outcome is rank optimal (Is-Rank-Optimal), and whether an mCP-net has a rank optimal outcome (Exists-Rank-Optimal). Finally, we focus on rank optimums, namely, the problems of deciding whether an outcome is rank optimum (Is-Rank-Optimum) and whether an mCP-net has a rank optimum outcome (Exists-Rank-Optimum).

Recall that, given an mCP-net $\mathcal{M} = \langle N_1, \dots, N_m \rangle$ and two outcomes $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$, $\beta \succ^r_{\mathcal{M}} \alpha$ iff $Rank_{\mathcal{M}}(\beta) < Rank_{\mathcal{M}}(\alpha)$, where, for any outcome γ, $Rank_{\mathcal{M}}(\gamma) = \sum_{1 \leq i \leq m} Rank_{N_i}(\gamma)$.

5.1 Complexity of rank dominance in mCP-nets

From the tractability of computing the rank of outcomes (Lemma 4.6) and the P-hardness of comparing the rank of outcomes on (individual) CP-nets (Theorem 4.8), we obtain that Rank-Dominance is P-complete.

Theorem 5.1. *Given an acyclic binary mCP-net $\mathcal{M}$ and two outcomes $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$, deciding whether $\beta \succ^r_{\mathcal{M}} \alpha$ is* P-complete. *Hardness holds even on acyclic binary mCP-nets with in-degree at most two and at most one agent.*

5.2 Complexity of rank optimality in mCP-nets

To show the complexity of Is-Rank-Optimal, we first consider the task of actually computing a rank optimal outcome in an mCP-net. We show that this task is feasible in polynomial time over acyclic binary mCP-nets. To achieve this, we start by providing a characterization for rank optimal outcomes for acyclic binary mCP-nets.

Given an acyclic binary mCP-net $\mathcal{M} = \langle N_1, \dots, N_m \rangle$ and an outcome $\alpha \in \mathcal{O}_{\mathcal{M}}$, we have that:

$$\begin{aligned} Rank_{\mathcal{M}}(\alpha) &= \sum_{1 \leq i \leq m} Rank_{N_i}(\alpha) \\ &= \sum_{1 \leq i \leq m} |\{F \in \mathcal{F}_{\mathcal{M}} \mid \alpha[F] \neq o_{N_i}[F]\}| \\ &= \sum_{F \in \mathcal{F}_{\mathcal{M}}} |\{i \mid 1 \leq i \leq m \wedge \alpha[F] \neq o_{N_i}[F]\}|. \end{aligned} \tag{2}$$

The last expression in Equation (2) suggests a way to characterize rank optimal outcomes. Indeed, any outcome α minimizing the value of the last expression of Equation (2) is clearly rank optimal. An outcome α is *average optimal*, if, for each feature $F \in \mathcal{F}_{\mathcal{M}}$,

$$\alpha[F] \in \arg\min_{v \in Dom_{\mathcal{M}}(F)} |\{i \mid 1 \leq i \leq m \wedge v \neq o_{N_i}[F]\}| = \arg\max_{v \in Dom_{\mathcal{M}}(F)} |\{i \mid 1 \leq i \leq m \wedge v = o_{N_i}[F]\}|.$$

Intuitively, an outcome α is average optimal, if in α the value of each feature F is the most frequent among the values of F in the optimum outcomes of the individual CP-nets of the mCP-net. For a feature F, the *average optimal values* of F are the values of F maximizing $|\{i \mid 1 \leq i \leq m \wedge v = o_{N_i}[F]\}|$.

We now prove that an outcome is rank optimal iff it is average optimal.

Lemma 5.2. *Let $\mathcal{M}$ be an acyclic binary mCP-net, and let $\alpha \in \mathcal{O}_{\mathcal{M}}$ be an outcome. Then, α is rank optimal in $\mathcal{M}$ iff α satisfies the average optimality condition.*

Proof. ($\Rightarrow$) Assume that α is a rank optimal outcome. We now show that α is average optimal as well. Assume by contradiction that α does not satisfy the average optimality condition, which means that there is a feature F such that $\alpha[F] \notin \arg\min_{v \in Dom_{\mathcal{M}}(F)} |\{i \mid 1 \leq i \leq m \wedge v \neq o_{N_i}[F]\}|$. Consider the outcome α' such that $\alpha'[G] = \alpha[G]$, for all features $G \neq F$, and $\alpha'[F] \in \arg\min_{v \in Dom_{\mathcal{M}}(F)} |\{i \mid 1 \leq i \leq m \wedge v \neq o_{N_i}[F]\}|$. By Equation (2), $Rank_{\mathcal{M}}(\alpha') < Rank_{\mathcal{M}}(\alpha)$, and hence $\alpha' \succ^r_{\mathcal{M}} \alpha$, which implies that α is not rank optimal in $\mathcal{M}$, which is a contradiction. Therefore, α is an average optimal outcome.

($\Leftarrow$) Assume that α is an average optimal outcome. We now show that α is rank optimal. Assume by contradiction that α is not rank optimal. This means that there is an outcome $\beta \neq \alpha$ such that $\beta \succ^r_{\mathcal{M}} \alpha$, and hence that $Rank_{\mathcal{M}}(\beta) < Rank_{\mathcal{M}}(\alpha)$. By Equation (2), we know that $Rank_{\mathcal{M}}(\alpha) = \sum_{F \in \mathcal{F}_{\mathcal{M}}} |\{i \mid 1 \leq i \leq m \wedge \alpha[F] \neq o_{N_i}[F]\}|$ and $Rank_{\mathcal{M}}(\beta) = \sum_{F \in \mathcal{F}_{\mathcal{M}}} |\{i \mid 1 \leq i \leq m \wedge \beta[F] \neq o_{N_i}[F]\}|$. Since $Rank_{\mathcal{M}}(\beta) < Rank_{\mathcal{M}}(\alpha)$, there must exist a feature F such that $|\{i \mid 1 \leq i \leq m \wedge \beta[F] \neq o_{N_i}[F]\}| < |\{i \mid 1 \leq i \leq m \wedge \alpha[F] \neq o_{N_i}[F]\}|$, which contradicts that α is an average optimal outcome. Therefore, α is rank optimal. $\square$

Given the strong characterization of rank optimal outcomes as average optimal, we are now ready to derive the complexity of the problems on rank optimality. In the coming proofs, we need the definition of *direct nets* $D(\alpha)$, which are acyclic binary CP-nets having α as their optimum outcome (see Section 5.1 in [45]).

The following result shows that Is-Rank-Optimal is P-complete. In particular, the P-hardness is shown by reduction 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. *Given an acyclic binary mCP-net $\mathcal{M}$ and an outcome $\alpha \in \mathcal{O}_M$, deciding whether α is rank optimal in $\mathcal{M}$ is* P-complete*. Hardness holds even on mCP-nets with in-degree at most two and at most three agents.*

Proof. As for membership in P, by Lemma 5.2, deciding whether α is rank optimal is tantamount to checking whether α is average optimal. Therefore, for each feature F, we verify whether $\alpha[F] \in \arg\max_{v \in Dom_{\mathcal{M}}(F)} |\{i \mid 1 \leq i \leq m \wedge v = o_{N_i}[F]\}|$. To verify this condition, we simply need to compute all the individual optimum outcomes (feasible in polynomial time [8]) and perform some counting operations. This is feasible in polynomial time.

Hardness for P is 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 as follows (assume w.l.o.g. that $v = f$): $\mathcal{M} = \langle N_1, N_2, N_3 \rangle$ is a 3CP-net, α is the outcome defined over the features in N and assigning non-overlined values to all features, $N_1 = N$, $N_2 = D(\alpha)$, 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 $G \neq F$, because, for all features $G \neq F$, $\alpha[G] = \beta[G]$, and α and β are the optimum outcomes of $D(\alpha)$ and $D(\beta)$, respectively. Since $\alpha[F] = f$ and $\beta[F] = \overline{f}$, α is rank optimal in $\mathcal{M}$ iff $o_{N_1}[F] = o_N[F] = f = v$. ☐

We next focus on the problem Exists-Rank-Optimal, which is trivial, because every acyclic binary mCP-net has an average optimal outcome that is also rank optimal. This is a different proof from the one in [53].

Lemma 5.4. *Let $\mathcal{M}$ be an acyclic binary mCP-net. Then, $\mathcal{M}$ has (always) a rank optimal outcome.*

5.3 Complexity of rank optimums in *m*CP-nets

We now analyze the complexity of rank optimums in mCP-nets. To this end, we first observe the following fact. Since, by Lemma 5.2, all and only the average optimal outcomes are rank optimal, if in an mCP-net there were more than one average optimal outcome, then there would be no rank optimum outcome, because different rank optimal outcomes would not rank dominate each other (which is required to be rank optimum).

The following result states that Is-Rank-Optimum is P-complete. In particular, hardness for P is shown via the same reduction used to prove the P-hardness of Is-Rank-Optimal with the additional observation that in mCP-nets with an odd number of agents, there is always a unique average optimal outcome, and hence there is a unique rank optimal outcome that is also rank optimum.

Theorem 5.5. *Given an acyclic binary mCP-net $\mathcal{M}$ and an outcome $\alpha \in \mathcal{O}_{\mathcal{M}}$, deciding whether α is rank optimum in $\mathcal{M}$ is* P-complete*. Hardness holds even on mCP-nets with in-degree at most two and at most three agents.*

Proof. As for membership in P, the following procedure deciding in polynomial time whether α is rank optimum uses the fact that there is a rank optimum outcome iff there is a unique average optimal outcome. First, we compute all the individual optimum outcomes for all agents of $\mathcal{M}$ (feasible in polynomial time [8]). Next, for every feature F, we check that there is only one value of F in $Dom_{\mathcal{M}}(F)$ such that $|\{i \mid 1 \leq i \leq m \wedge v = o_{N_i}[F]\}|$ is maximized (feasible in polynomial time). Then, we check that α is average optimal (feasible in polynomial time).

Hardness for P is shown via the same reduction from Feat-Value-Opt used in the proof of Theorem 5.3. In particular, since the mCP-net $\mathcal{M}$ in that reduction contains an odd number of CP-nets, $\mathcal{M}$ has only one average optimal outcome, which is also rank optimum. Hence, α is rank optimal in $\mathcal{M}$ iff α is rank optimum in $\mathcal{M}$. ☐

We finally focus on the problem Exists-Rank-Optimum. The following result states that it is P-complete. Its proof is based on the observation that mCP-nets with unique average optimal outcomes have a rank optimum outcome (see above). The P-hardness is again shown via a reduction from Feat-Value-Opt by exploiting direct nets.

Theorem 5.6. *Given an acyclic binary mCP-net $\mathcal{M}$, deciding whether $\mathcal{M}$ has a rank optimum outcome is* P-complete*. Hardness holds even on mCP-nets with in-degree at most two and at most four agents.*

Proof. As for membership, the following procedure deciding in polynomial time whether $\mathcal{M}$ has rank optimum outcome is based on the observation that there is a rank optimum outcome iff there is a unique average optimal outcome. First, we compute all the individual optimum outcomes for all agents of $\mathcal{M}$ (feasible in polynomial time). Next, for every feature F, we check that there is only one value of F in $Dom_{\mathcal{M}}(F)$ such that $|\{i \mid 1 \le i \le m \wedge v = o_{N_i}[F]\}|$ is maximized (feasible in polynomial time).

Hardness for P is 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 $\beta[F] = f$. We know that $\mathcal{M}$ has a rank optimum outcome iff $\mathcal{M}$ 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 $N_4[G] = \overline{g}$, the average optimal value is unique, and it is $o_N[G] = o_{N_1}[G] = o_{N_2}[G]$. 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, $o_N[F] = \overline{f} \neq v$. 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. ☐

6 Max voting

In this section, we characterize the complexity of max voting tasks on mCP-nets. First, we show that there are mCP-nets without max optimal and optimum outcomes, which implies that deciding the existence of max optimal and optimum outcomes is not a trivial problem. Then, we analyze the complexity of deciding max dominance in mCP-nets (Max-Dominance). Next, we devote our analysis to the problems related to max optimal outcomes, namely, deciding whether an outcome is max optimal (Is-Max-Optimal), and deciding whether an mCP-net has a max optimal outcome (Exists-Max-Optimal). To conclude, we study the complexity of problems on max optimum outcomes, namely, deciding whether an outcome is max optimum (Is-Max-Optimum), and deciding whether an mCP-net has a max optimum outcome (Exists-Max-Optimum).

Recall that, given an mCP-net $\mathcal{M}$ and two outcomes $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$, $\beta \succ^x_{\mathcal{M}} \alpha$ if the set of agents preferring β to α is the biggest, i.e., $|S^{\succ}_{\mathcal{M}}(\beta,\alpha)| > \max(|S^{\prec}_{\mathcal{M}}(\beta,\alpha)|, |S^{\bowtie}_{\mathcal{M}}(\beta,\alpha)|)$.

We start by noticing that there are mCP-nets without max optimal and optimum outcomes. This follows from the fact that there are CP-nets without majority optimal outcomes [45, Theorem 5.1].

Theorem 6.1. *There are acyclic binary singly connected mCP-nets with no max optimal and optimum outcomes.*

Proof. There exists an acyclic binary singly connected 4CP-net $\mathcal{M}_{NoWin}$ that does not have majority optimal outcomes [45, Theorem 5.1]. Consider any outcome $\alpha \in \mathcal{O}_{\mathcal{M}_{NoWin}}$. Since α is not majority optimal in $\mathcal{M}_{NoWin}$, there is an outcome β such that $\beta \succ^m_{\mathcal{M}_{NoWin}} \alpha$. This implies that $\beta \succ^x_{\mathcal{M}_{NoWin}} \alpha$ as well, and hence α is not a max optimal outcome in $\mathcal{M}_{NoWin}$. Because there is no max optimal outcome in $\mathcal{M}_{NoWin}$, there is no max optimum outcome in $\mathcal{M}_{NoWin}$ either. ☐

6.1 Complexity of max dominance in mCP-nets

We first focus on the problem Max-Dominance. The following result shows that it is Θ^{P}_2-complete.

Theorem 6.2. *Given an acyclic binary mCP-net $\mathcal{M}$ belonging to a polynomially connected class of mCP-nets, and two outcomes $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$, deciding whether $\beta \succ^x_{\mathcal{M}} \alpha$ is* Θ^{P}_2-complete*. Hardness holds even if $\mathcal{M}$ is singly connected, and its in-degree is at most three.*

Proof. To show that the problem belongs to Θ^{P}_2, we show that answering this question is feasible in deterministic polynomial time with a logarithmic number of calls to an NP oracle.

Let $\mathcal{M} = \langle N_1, \ldots, N_m\rangle$. Since $|S^{\succ}_{\mathcal{M}}(\beta,\alpha)| + |S^{\prec}_{\mathcal{M}}(\beta,\alpha)| + |S^{\bowtie}_{\mathcal{M}}(\beta,\alpha)| = m$, in order to decide whether $|S^{\succ}_{\mathcal{M}}(\beta,\alpha)| > \max(|S^{\prec}_{\mathcal{M}}(\beta,\alpha)|, |S^{\bowtie}_{\mathcal{M}}(\beta,\alpha)|)$, it suffices to compute $|S^{\succ}_{\mathcal{M}}(\beta,\alpha)|$ and $|S^{\prec}_{\mathcal{M}}(\beta,\alpha)|$, as we can then derive the conclusion by checking that $|S^{\succ}_{\mathcal{M}}(\beta,\alpha)| > |S^{\prec}_{\mathcal{M}}(\beta,\alpha)|$ and that $2|S^{\succ}_{\mathcal{M}}(\beta,\alpha)| > m - |S^{\prec}_{\mathcal{M}}(\beta,\alpha)|$.

We can compute the exact value $|S^{\succ}_{\mathcal{M}}(\beta,\alpha)|$ as follows. First, observe that, for an integer k, deciding whether there are at least k different agents of $\mathcal{M}$ preferring β to α is in NP. Indeed, we can guess k CP-nets in which β is preferred to α, and, since $\mathcal{M}$ is assumed to be binary, acyclic, and belonging to a polynomially connected class of mCP-nets, there are polynomial witnesses for β being preferred to α in each CP-net [8, Theorem 16].

Therefore, the overall guess requires only polynomial space and can be checked in polynomial time. Having this oracle, computing $|S^{\succ}_{\mathcal{M}}(\beta,\alpha)|$ can be done through a binary search in the range $[0,m]$ by calling the above described oracle. Observe that we need only a logarithmic number of calls to the oracle. In a similar way, we can compute the exact value $|S^{\prec}_{\mathcal{M}}(\beta,\alpha)|$. In this case, the query for the oracle is whether there are at least k distinct agents preferring α to β. Once we have computed the required values, we can carry out the final check, which can be done in deterministic polynomial time, and return the answer.

The Θ_2^{P}-hardness of the problem is shown via a reduction from the Θ_2^{P}-complete problem Comp-Sat [44]: given two sets A and B of 3CNF Boolean formulas, decide whether the number of satisfiable formulas in A is greater than the number of satisfiable formulas in B. The Θ_2^{P}-hardness of Comp-Sat holds even if all formulas in A and B are defined over the same set of variables and have the same number of clauses [44].

In the reduction, we use CP-nets that are capable of encoding the problem of satisfiability of Boolean formulas. In particular, for a Boolean formula ϕ, the *formula net* $F(\phi)$ [45, Section 3.1] is a CP-net whose features are associated with variables, literals, and clauses of ϕ. It was shown [45, Corollary 3.2] that, if α and $\overline{\beta}$ are two outcomes of $F(\phi)$ assigning non-overlined values to all features and overlined values to all and only variable and clause features, respectively, then ϕ is satisfiable iff $\overline{\beta} \succ_{F\phi} \alpha$, and ϕ is unsatisfiable iff $\overline{\beta} \bowtie_{F\phi} \alpha$.

Here, we introduce the additional formula net $\overline{F}(\phi)$, which is symmetric to $F(\phi)$ and was not present in [45]. In particular, features and edges in $\overline{F}(\phi)$ are the very same of those in $F(\phi)$, while all CP tables of $\overline{F}(\phi)$ are similar to those of $F(\phi)$ with the only difference that, for all variable and clause features (but not for literal features), non-overlined values are exchanged with overlined values, and vice-versa. Observe that $F(\phi)$ and $\overline{F}(\phi)$ have the same outcomes. By an adaptation of Corollary 3.2 of [45], it is possible to show that, if α and $\overline{\beta}$ are two outcomes of $\overline{F}(\phi)$ assigning non-overlined values to all features, and overlined values to all and only variable and clause features, respectively, then ϕ is satisfiable iff $\alpha \succ_{\overline{F}(\phi)} \overline{\beta}$, and ϕ is unsatisfiable iff $\alpha \bowtie_{\overline{F}(\phi)} \overline{\beta}$.

Let $\langle A,B\rangle$ be a pair of sets of Boolean formulas in 3CNF, with $|A|=a$ and $|B|=b$, where all formulas of A and B are defined over the same set of variables $X=\{x_1,\dots,x_n\}$, and have the same number of clauses $C=\{c_1,\dots,c_m\}$. From $\langle A,B\rangle$, we build the $3(a+b)$CP-net $\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)$ in the following way. Since all the formulas of A and B are 3CNFs having the same variables and the same number of clauses, the set of features of each CP-net of $\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)$ is $\mathcal{V}\cup\mathcal{P}\cup\mathcal{D}$, where $\mathcal{V}=\{V_i^T,V_i^F \mid x_i\in X\}$ (which are the variable features of formula nets), $\mathcal{P}=\{P_{j,1},P_{j,2},P_{j,3}\mid 1\le j\le m\}$ (which are the literal features of formula nets), and $\mathcal{D}=\{D_j\mid c_j\in C\}$ (which are the clause features of formula nets). The agents of $\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)$ are:

- for each formula $\phi_i\in A$, there is an agent whose CP-net is $N_{A,i}=F(\phi_i)$;
- for each formula $\varphi_j\in B$, there is an agent whose CP-net is $N_{B,j}=\overline{F}(\varphi_j)$;
- there are $a+b$ agents whose preferences are encoded by the (same) direct net (mentioned before Theorem 5.3) $D(\alpha)$, with α being the outcome assigning non-overlined values to all features; and
- there are $a+b$ agents whose preferences are encoded by the (same) direct net $D(\beta)$, with β being the outcome assigning overlined values to all and only variable and clause features.

Observe that $\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)$ is binary, acyclic, singly connected, and its in-degree is three. To conclude the construction, consider the outcomes $\alpha,\beta\in\mathcal{O}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}$ such that in α the values of all features are non-overlined, while in β the values of all and only variable and clause features are overlined. The construction is computable in polynomial time. We now show that by the above construction, Comp-Sat reduces to Max-Dominance.

Let $S_A\subseteq A$ be the set of the satisfiable formulas of A, and let $S_B\subseteq B$ be the set of the satisfiable formulas of B. By the discussion above, for each formula $\phi_i\in S_A$, $\beta\succ_{N_{A,i}}\alpha$; for each formula $\phi_i\in(A\setminus S_A)$, $\alpha\bowtie_{N_{A,i}}\beta$; for each formula $\varphi_j\in S_B$, $\alpha\succ_{N_{B,j}}\beta$; and, for each formula $\varphi_j\in(B\setminus S_B)$, $\alpha\bowtie_{N_{B,j}}\beta$. Since for the CP-nets $D(\alpha)$ (resp., $D(\beta)$) the outcome α (resp., β) is preferred to all other outcomes, $|S^{\succ}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|=|S_A|+a+b$, $|S^{\prec}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|=|S_B|+a+b$, and $|S^{\bowtie}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|=|A\setminus S_A|+|B\setminus S_B|\le a+b$. We now show that $\langle A,B\rangle$ is a "yes"-instance of Comp-Sat iff $\beta\succ^{x}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}\alpha$.

($\Rightarrow$) If $\langle A,B\rangle$ is a "yes"-instance of Comp-Sat, then $|S_A|>|S_B|$, and hence $|S^{\succ}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|=|S_A|+a+b>|S_B|+a+b=|S^{\prec}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|$. Moreover, since $|S_A|>|S_B|\ge 0$, it must be the case that $|S_A|\ge 1$. Therefore, $|S^{\succ}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|=|S_A|+a+b>a+b\ge|S^{\bowtie}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|$. Thus, $|S^{\succ}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|>\max(|S^{\prec}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|,|S^{\bowtie}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|)$, and hence $\beta\succ^{x}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}\alpha$.

($\Leftarrow$) If $\langle A,B\rangle$ is a "no"-instance of Comp-Sat, then $|S_A|\le|S_B|$, and hence $|S^{\succ}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|=|S_A|+a+b\le|S_B|+a+b=|S^{\prec}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|$. Therefore, it holds that $|S^{\succ}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|\le\max(|S^{\prec}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|,|S^{\bowtie}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}(\beta,\alpha)|)$, and hence $\beta\not\succ^{x}_{\mathcal{M}_{\mathrm{md}}(\langle A,B\rangle)}\alpha$. ☐

Observe that Max-Dominance, unlike Pareto-Dominance and Majority-Dominance [45], cannot be hard for its complexity class over classes of mCP-nets having a bounded number of agents, because an essential part of its hardness is counting the exact number of agents preferring β to α, or α to β. If we considered a class of Max-Dominance instances with a bounded number of agents, then it would be possible to count the agents' preferences through a constant number of calls to the NP oracle. This would imply that, over that specific class of instances, Max-Dominance would not be among the most difficult problems of Θ_2^{P},[3] and thus none of those instances with a bounded number of agents can be used to show the Θ_2^{P}-hardness of Max-Dominance.

6.2 Complexity of max optimality in mCP-nets

In this section, we analyze the complexity of the problems on max optimal outcomes. We first focus on deciding max optimality of outcomes in mCP-nets (Is-Max-Optimal).

To characterize the precise complexity of the problem, we use the following property, whose detailed proof is provided in the appendix.

Lemma 6.3. *There exists a polynomial-time reduction from the problem of deciding the validity of quantified Boolean formulas $\Phi = (\forall X)(\exists Y)\phi(X,Y)$, where $\phi(X,Y)$ is in 3CNF, to the problem of deciding whether an outcome is max optimal in mCP-nets. The mCP-nets obtained in the reduction are binary, acyclic, formed by four agents, they have in-degree three, and they constitute a polynomially connected class of mCP-nets.*

The following result shows that the problem Is-Max-Optimal is Π_2^{P}-complete.

Theorem 6.4. *Given an acyclic binary mCP-net $\mathcal{M}$ belonging to a polynomially connected class of mCP-nets and an outcome $\alpha \in \mathcal{O}_{\mathcal{M}}$, deciding whether α is max optimal in $\mathcal{M}$ is* Π_2^{P}-complete. *Hardness holds even on mCP-nets with in-degree at most three and at most four agents.*

Proof. As for membership in Π_2^{P}, we show that the complementary problem of deciding whether α is *not* a max optimal outcome in $\mathcal{M}$ is in Σ_2^{P}. If α is not max optimal, then there is an outcome β such that $\beta \succ_{\mathcal{M}}^{x} \alpha$. So, to show that α is not max optimal, it suffices to guess β and then check that $\beta \succ_{\mathcal{M}}^{x} \alpha$. Observe that guessing β requires an NP machine, and then checking $\beta \succ_{\mathcal{M}}^{x} \alpha$ is in Θ_2^{P} (see Theorem 6.2). Hence, the very same NP machine having guessed β can also check the max dominance condition by querying logarithmically-many times an NP oracle. Therefore, the overall procedure is in Σ_2^{P}.

Hardness for Π_2^{P} follows from Lemma 6.3 and the Π_2^{P}-hardness of the problem of deciding the validity of quantified Boolean formulas $\Phi = (\forall X)(\exists Y)\phi(X,Y)$, where $\phi(X,Y)$ is in 3CNF [61, 64]. ☐

Note that, relative to the number of agents, the above result is optimal. Indeed, the max dominance semantics and the majority dominance semantics are equivalent on mCP-nets with $m \leq 3$, and checking majority optimality is co-NP-complete [45]. Thus, it is not possible to show the Π_2^{P}-hardness of Is-Max-Optimal on mCP-nets with $m \leq 3$. Unlike Max-Dominance, the hardness of Is-Max-Optimal holds even on mCP-nets with a bounded number of agents. One may wonder why this is the case, and there is no need to have an "unpredictable" number of agents to count as for Max-Dominance. After all, also in this case, after guessing an outcome β, to disprove the max optimality of α, we have to count the number of agents preferring β to α, or α to β. The reason is subtle. We observed already that, if the number of agents were bounded, then it would be possible to carry out the counting required to decide max dominance through a constant number of calls to an NP oracle. However, this is not relevant in this case, because the class $\mathrm{NP}^{\mathrm{NP}[O(1)]}$, which is the class of the languages recognizable by a nondeterministic polynomial-time Turing machine querying at most a constant number of times an NP oracle, is equal to the class $\mathrm{NP}^{\mathrm{NP}}$, which is the class of the languages recognizable by a nondeterministic polynomial-time Turing machine querying (at most polynomially-many times) an NP oracle [63].

We next analyze the complexity of the problem of deciding the existence of max optimal outcomes in mCP-nets (Exists-Max-Optimal). To characterize the precise complexity of the problem, we use the following property, whose detailed proof is provided in the appendix.

Lemma 6.5. *There exists a polynomial-time reduction from the problem of deciding the validity of quantified Boolean formulas $\Phi = (\exists X)(\forall Y)(\exists Z)\phi(X,Y,Z)$, where $\phi(X,Y,Z)$ is in 3CNF, to the problem of deciding whether mCP-nets admit a max optimal outcome. The mCP-nets obtained in the reduction are binary, acyclic, formed by eight agents, they have in-degree three, and they constitute a polynomially connected class of mCP-nets.*

The following result shows that the problem Exists-Max-Optimal is Σ_3^{P}-complete.

Theorem 6.6. *Given an acyclic binary mCP-net $\mathcal{M}$ belonging to a polynomially connected class of mCP-nets, deciding whether $\mathcal{M}$ has a max optimal outcome is* Σ_3^{P}-complete. *Hardness holds even on mCP-nets with in-degree at most three and at most eight agents.*

[3] In fact, on such restricted instances, Max-Dominance belongs to $\mathrm{P}^{\mathrm{NP}[O(1)]}$, which is the class of languages recognizable by a deterministic Turing machine in polynomial time performing at most a constant number of calls to an NP oracle. It is known that $\mathrm{P}^{\mathrm{NP}[O(1)]} \subseteq \mathrm{P}^{\mathrm{NP}[O(\log n)]} = \Theta_2^{\mathrm{P}}$ [62, 63].

Proof. As for membership in Σ_3^{P}, to prove that $\mathcal{M}$ has a max optimal outcome, it suffices to guess an outcome α and then check that α is actually max optimal. Observe that guessing α requires an NP machine, and the final check can be carried out by an oracle in Π_2^{P} (see Theorem 6.4). Therefore, the overall procedure is in Σ_3^{P}.

Hardness for Σ_3^{P} follows from Lemma 6.5 and the Σ_3^{P}-hardness of the problem of deciding the validity of quantified Boolean formulas $\Phi = (\exists X)(\forall Y)(\exists Z)\phi(X,Y,Z)$, where $\phi(X,Y,Z)$ is in 3CNF [61, 64]. □

6.3 Complexity of max optimums in *m*CP-nets

We now focus on max optimum outcomes. We first consider the problem of deciding whether an outcome is max optimum in an mCP-net (Is-Max-Optimum). The following result shows that it is Π_2^{P}-complete.

Theorem 6.7. *Given an acyclic binary mCP-net $\mathcal{M}$ belonging to a polynomially connected class of mCP-nets and an outcome $\alpha \in \mathcal{O}_{\mathcal{M}}$, deciding whether α is max optimum in $\mathcal{M}$ is* Π_2^{P}-complete. *Hardness holds even on mCP-nets with in-degree at most three and at most three agents.*

Proof. We prove the membership in Π_2^{P} by showing that deciding whether α is *not* max optimum in $\mathcal{M}$ is in Σ_2^{P}. If α is not max optimum, then there is an outcome β such that $\alpha \not\succ^{x}_{\mathcal{M}} \beta$. So, in order to prove that α is not max optimum, it suffices to guess β and then check that $\alpha \not\succ^{x}_{\mathcal{M}} \beta$. Observe that guessing β requires an NP machine, and then checking $\alpha \not\succ^{x}_{\mathcal{M}} \beta$ is in Θ_2^{P} (see Theorem 6.2, and recall that Θ_2^{P} is closed under complement). Hence, the very same NP machine having guessed β can also check the max dominance condition by querying logarithmically-many times an NP oracle. Therefore, the overall procedure is in Σ_2^{P}.

As for hardness, observe that the max dominance semantics and the majority dominance semantics are equivalent on 3CP-nets. Therefore, the Π_2^{P}-hardness of the problem can be stated as a consequence of the Π_2^{P}-hardness of deciding majority optimum outcomes over 3CP-nets [45, Theorem 5.16]. □

Note that, as for the number of agents, the above result is optimal. Indeed, the max and the Pareto dominance semantics are equivalent on mCP-nets with $m \leq 2$, and checking Pareto optimum outcomes is in LOGSPACE [45]. Thus, it is not possible to show the Π_2^{P}-hardness of Is-Max-Optimum on mCP-nets with $m \leq 2$.

To conclude, the following result shows that the problem Exists-Max-Optimum is Π_2^{P}-hard and in Σ_3^{P}.

Theorem 6.8. *Given an acyclic binary mCP-net $\mathcal{M}$ belonging to a polynomially connected class of mCP-nets, deciding whether $\mathcal{M}$ has a max optimum outcome is in Σ_3^{P} and is Π_2^{P}-hard. Hardness holds even on mCP-nets with in-degree at most three and at most three agents.*

Proof. As for membership, to show that $\mathcal{M}$ has a max optimum outcome, it suffices to guess an outcome α and then check that α is actually max optimum. Observe that guessing α requires an NP machine, and the final check is in Π_2^{P} (see Theorem 6.7), which can be carried out by an oracle. Hence, the overall procedure is in Σ_3^{P}.

As for hardness, again, by the fact that the max dominance semantics and the majority dominance semantics are equivalent on 3CP-nets, the Π_2^{P}-hardness of the problem can be stated as a consequence of the Π_2^{P}-hardness of deciding the existence of majority optimum outcomes over 3CP-nets [45, Theorem 5.20]. □

7 Related work

The present work continues our previous one [45] in the complexity analysis of global voting over CP-nets, which has been lacking compared to the abundance of works considering sequential voting over CP-nets. An in-depth analysis of related works considering sequential voting can be found in our preceding paper [45]. Here, we focus on works more closely related to the specificities of the present paper.

Recently, a work by Haret et al. [28] considered global voting over a variant of CP-nets, called *generalized* CP-nets (or gCP-nets) [24]. In gCP-nets, logical expressions, which we used here only for notational convenience, are the distinctive characteristic, and intuitively are used to define the rows of CP tables (which are called CP statements). This representation allows to avoid the complete specification of CP tables, and it can hence be more compact. The gCP-nets considered in the mentioned works can be cyclic, and this adds complexity to the semantics of the model, as the dominance test is PSPACE-complete.

Haret et al. [28] introduced mgCP-nets, which are a generalization of gCP-nets to the multi-agent case. They studied the generalization of the Pareto, majority, max, and rank semantics over the new model. There is a difference for the rank semantics studied by them; indeed, they consider the rank of an outcome as the longest flipping sequence from an outcome to a non-dominated class (they have equivalence classes of outcomes), instead of the shortest flipping sequence. They also consider a richer setting for the dominance relationship; in particular, they study weak and strong dominance, and they carry out a thorough complexity analysis of many tasks for the voting schemes and the dominance variants considered. Interestingly, almost all their results are of PSPACE-completeness/hardness. This is due to the fact that already the dominance test in gCP-nets is PSPACE-complete. Hence, in many cases, the complexity of the dominance test in gCP-nets masks out the complexity of preference aggregation in mgCP-nets. In this respect, our work can be seen as characterizing the complexity of voting tasks when the complexity of the dominance test (which in our case is "only" NP-complete) is not the one dominating the entire preference aggregation tasks. For this reason, the specific complexity of the different voting schemes can stand out, as it is not masked by the intricacy of the dominance test.

The work of Laing et al. [32] focuses on a different concept of rank. In their paper, the rank is defined so to weigh the importance of the features, and features higher up in the topological order of the CP-net are weighed more, and hence a change in their value impacts more on the rank of an outcome. Their definition of rank is such that, for any CP-net N and for any two outcomes $\alpha, \beta \in \mathcal{O}_N$, if $\alpha \succ_N \beta$, then the rank of α is strictly greater than the rank of β. This property can be used to generate consistent orderings of outcomes, i.e., orderings of outcomes in which it never happens that an outcome α preceding another outcome β in the ordering according to the rank is such that β is preferred to α according to the standard dominance semantics of CP-nets. This does not hold for the definition of rank by Rossi et al. [53]. Moreover, in the same work, this new definition of rank is used as a heuristic to speed-up the decision of the dominance relationship. Similar proposals of rank functions to generate consistent orderings can be found in the works of Domshlak et al. [21] and of Li et al. [41] (see also the thorough discussion by Laing [31]). Interestingly, Laing et al. [32] extend their definition of rank also to CP-nets with indifferent values.

Polynomial-time voting has attracted extensive consideration in the literature, precisely for its efficiency. However, to our knowledge, P-hardness has not carefully been investigated so far in the computational social choice literature. In fact, we are aware of only two other P-completeness results in the literature, namely, the complexity of checking the essential set, which is a solution concept, over weak tournaments [14, 15], and the complexity of deciding, for a profile of preference rankings, whether a given outcome/candidate is the winner according to the single transferable vote rule [19].

In fact, it may very well be the case that polynomial-time voting schemes are actually P-hard, which would be a clear sign that these voting procedures would not scale up over huge input instances. In this paper, we show that this is indeed the case for some voting tasks over mCP-nets. Hence, the P-completeness results reported here not only characterize more precisely the complexity of voting over mCP-nets, but they also point out a significant issue, that, in our opinion, has not been investigated enough so far, which is whether polynomial-time voting schemes are highly parallelizable, so that tailored parallel algorithms can scale up over big input instances.

Regarding the first result mentioned, weak tournaments are graphs representing incomplete preference, and they directly encode a dominance relation (after vote aggregation). Intuitively, the data structure in the 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 individual 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 the input. In this respect, our work is different, because we assume that the input contains the preferences of the individual agents.

For the second result mentioned, an interesting property shown is that if the number of agents is fixed, then the problem can be solved in LOGSPACE, and hence it becomes highly parallelizable [19].

Not all the voting schemes known in the literature to be feasible in polynomial time are actually P-complete. Nonetheless, for the vast majority of these polynomial-time voting schemes, it was not investigated either whether they were actually P-complete, or whether they can be decided in subclasses of P. Again, to our knowledge, only Brandt and Fischer [14] and Csar et al. [19] have carried out such a refined analysis.

8 Conclusion

In this work, we have continued the complexity analysis of global voting over acyclic binary CP-nets started in our previous work [45]. In particular, we have investigated the complexity of max and rank semantics in mCP-nets. The problems analyzed for the two mCP-nets semantics are the classic ones in voting scenarios, namely, deciding dominance, deciding optimal and optimum outcomes, and deciding whether optimal and optimum outcomes exist. For almost all of them, we have shown completeness results, and in fact we give tight lower bound for problems that (up to now) did not have any explicit lower bound transcending the obvious hardness due to the dominance test over the underlying CP-nets. The obtained results situate the complexity of the max voting tasks at various levels of the polynomial hierarchy, which is quite interesting, as for most of these tasks, only EXPTIME upper bounds were known to date [53]. For the rank voting scheme, we have provided various P-completeness results. Memberships in P for these problems are quite a big improvement over the previously known algorithms, requiring exponential time. Hardness results for P show that these tasks are inherently sequential and hence not highly parallelizable. This points out a significant issue, which is whether polynomial-time voting schemes are highly parallelizable to tackle big instances.

As our hardness results are obtained over mCP-nets with standard acyclic binary CP-nets, they extend to more general mCP-nets with partial CP-nets and/or multi-valued features. Our hardness results for the max voting semantics can be extended to any representation scheme as succinct and expressive as the class of CP-nets considered here (see [45, Section 2.5] for the definition of the relationship "as succinct and expressive" between representations schemes for preferences). Our membership results for the max voting semantics can be extended to any NP representation scheme (see [45, Section 2.5] for the definition).

Note also that the hardness results shown here for deciding the existence of optimal and optimum outcomes are also lower bounds for the corresponding computational problems. Indeed, computing optimal or optimum outcomes cannot be easier than the lower bounds shown here for deciding whether such outcomes exist, as otherwise there would be a more efficient way to decide their existence. In this paper, we did not assume the quite stringent constraint of $\mathcal{O}$-legality, which was assumed in multiple research papers in the literature; this also makes our results more general.

Individuating the precise computational complexity of the above-mentioned problems not only provides the analysis that was requested in the literature [35], but also highlights what are the sources of complexity in the problems. With this information, it is hence possible to design algorithms to solve the problems and to characterize subclasses of instances over which the problems are tractable, because the sources of intractability are individuated. For example, problems/languages L that are Σ_2^{P}-complete are characterized by two independent sources of complexity. As a practical consequence of this, with our current state of knowledge, any correct algorithm for L running on a standard deterministic machine requires two nested backtracking procedures to explore the space of the possible solutions. Intuitively, an outer backtracking procedure is used to generate the candidate solutions, and an inner backtracking procedure must be used (unless P = NP) to verify that the candidate solution is actually a correct one. That is, checking the correctness of the candidate solution is an intractable problem on its own. As an example, it has been shown in this paper that deciding the existence of max optimal outcomes in an mCP-net is Σ_3^{P}-complete. Having this precise information, and not, for example, only a more loose NP-hardness result, tells us important insights of practical relevance. First, unless P = NP (or unless there are collapses in the polynomial hierarchy impacting on the first three layers), to solve this problem we need an algorithm that, if implemented on standard deterministic machines, requires three nested exponential backtracking procedures. No approach using less than three nested exponential searches will ever be able to correctly solve the problem on general instances. Second, it is not possible to encode this problem into a Boolean formula to be solved by a SAT solver (as it was done in the literature for other dominance semantics [38, 39, 40, 42]) without having a double exponential growth in the size of the resulting formula. Or, similarly, if we want to avoid the exponential explosion in the translation, the problem cannot be encoded into a plain Boolean formula, but the problem can be encoded into a quantified Boolean formula with three alternating quantifiers, and the use of heavier QBF solvers to obtain the solution is required.

There are various possibilities for future research. First, the exact complexity of deciding the existence of max optimum outcomes is still not known. The complexity lies between Π_2^{P} and Σ_3^{P}, and it would be interesting to find a lower bound and a matching upper bound. In fact, the complexity of deciding the existence of majority optimum outcomes lies between Π_2^{P} and $\mathrm{D}_2^{\mathrm{P}}$, and therefore finding the exact complexities of these problems will allow to understand whether one task is actually more intricate than the other.

The various tasks analyzed have in general a high computational complexity. Hence, investigating structural restrictions on the structure of CP-nets to identify broader classes of CP-nets where the dominance test is tractable can manifest itself as a quite fruitful direction of research. Another possibility is studying other voting schemes based over majority voting, like Dodgson, Young, and Kemeny (see, e.g., the survey by Brandt et al. [16]) on profiles of CP-nets. Note that the latter three can be seen as variations/generalizations of the majority rule; in some way, they use a measure of how distant an outcome is from the majority optimal outcome, and hence these rules can be used to compute an optimal outcome with a majority flavor even in situations where the standard majority optimal outcome is not available [5]. Kemeny voting has also been used to aggregate website rankings of various search engines [22].

The formalism of mCP-nets in their original definition allows the presence of "non-ranked" features in CP-nets [53]. The authors called this kind of CP-nets partial CP-nets, and they can model a form of indifference between preferences. Completing the analysis of voting complexity over profiles of partial CP-nets would give a clearer picture of the complexity of voting over (m)CP-nets.

Another aspect to investigate more deeply is related to the modeling capabilities of CP-nets, which assume that all outcomes in a domain are attainable. However, this is not always the case, and hence we should take into account what outcomes are feasible in the definition of preference aggregation. To give an example, when majority voting is considered, to decide whether an outcome is dominated by another, we should check whether the latter is feasible. A non-feasible outcome should not be allowed to dominate another outcome. Constraints issued over the outcome domain can be considered prior to the aggregation process or after it, similarly to what was done for NTU cooperative games defined via constraints [25, 26]. NTU games (i.e., non-transferable utility games) are cooperative games in which players do not have complete freedom in sharing the worth that they get by forming a coalition. Constraints were used to compactly define the allowed worth distributions available to the players of coalitions. This approach could be merged with the definition of constrained CP-nets [9, 51].

Acknowledgments

This work was supported by the UK EPSRC grants EP/J008346/1, EP/L012138/1, and EP/M025268/1, by the Alan Turing Institute under the EPSRC grant EP/N510129/1, and by the AXA Research Fund.

We thank Jörg Vogel for pointing out the reduction that we used in the preliminary version [43] of this work to prove that COMP-SAT is Θ_2^{P}-hard. In this paper, we did not rely on a tailored reduction to prove the Θ_2^{P}-hardness of COMP-SAT, but its hardness is a corollary of the $\Theta_{k+1}^{\mathrm{P}}$-hardness of the problem COMP-VALID$_k$ [44].

A Proofs for Section 6

To prove Lemma 6.3, we need to show a transformation for the mentioned problems. This transformation uses the *summarized formula nets* $F_s(\phi)$ (see Section 5.1 in [45] for the details of the definition and see Section 3.1 in [45] for the encoding of the Boolean assignments in these nets), which are CP-nets linking the satisfiability of 3CNF Boolean formulas ϕ with the dominance relationship of two outcomes differing on the value of only two features. This advantage comes at the cost of losing the single connectedness property of the (non-summarized) formula nets (used to show the hardness of the max dominance problem in Section 6.1). We also use *direct nets*, which are CP-nets designed to have specific optimum outcomes (see Section 5.1 in [45]); and also *conjunctive/disjunctive interconnecting nets*, which are nets designed to "check" whether all features or at least one feature of a set, respectively, have overlined values (see Section 4.1 in [45]).

Construction A.1. Let $\Phi = (\forall X)(\exists Y)\phi(X, Y)$ be a quantified formula where $\phi(X, Y)$ is a 3CNF Boolean formula defined over two disjoint sets $X = \{x_1, \dots, x_{n_X}\}$ and $Y = \{y_1, \dots, y_{n_Y}\}$ of Boolean variables, and whose set of clauses is $C = \{c_1, \dots, c_m\}$. From Φ, we define the 4CP-net $\mathcal{M}_{\mathrm{ixl}}(\phi) = \langle N_1^{\mathrm{ixl}}, \dots, N_4^{\mathrm{ixl}}\rangle$ in the following way.

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

- all the features of a net $F_s(\phi)$ in which, in this case, we distinguish two variable feature sets $\mathcal{V} = \{V_i^T, V_i^F \mid x_i \in X\}$ and $\mathcal{W} = \{W_i^T, W_i^F \mid y_i \in Y\}$ (recall that $\mathcal{P}$ and $\mathcal{D}$ are the sets of literal and clause features, respectively, and $\mathcal{A}$ is the set of features of the conjunctive interconnecting net embedded in $F_s(\phi)$);
- all the features of sets $\mathcal{V}' = \{V_i' \mid x_i \in X\}$, $\mathcal{V}'' = \{V_i'' \mid x_i \in X\}$, and $\mathcal{V}''' = \{V_i''' \mid x_i \in X\}$;
- all the features of the set $\mathcal{B}$, which are the features B_i of a conjunctive/disjunctive interconnecting net $H_{\mathrm{C}}(|\mathcal{V}'|)$ (once these features will be used in a conjunctive interconnecting net, and once in a disjunctive interconnecting net), and its apex is feature B (features B_i are distinct from features A_i of the conjunctive interconnecting net $H_{\mathrm{C}}(m)$ embedded in $F_s(\phi)$).

To summarize, all the features of $\mathcal{M}_{\mathrm{ixl}}(\phi)$ are $\mathcal{V} \cup \mathcal{V}' \cup \mathcal{V}'' \cup \mathcal{V}''' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B} \cup \{U_1, U_2\}$.

The CP-nets of $\mathcal{M}_{\mathrm{ixl}}(\phi)$ are:

- $N_1^{\mathrm{ixl}} = \langle \mathcal{F}_{N_1^{\mathrm{exl}}}, \mathcal{E}_{N_1^{\mathrm{exl}}}\rangle$ embeds a net $F_s(\phi)$ with its features, links, and CP tables (but the CP table of U_2, which is defined below).

  The other links of N_1^{ixl} are:
  - for each $x_i \in X$, $\{(V_i', V_i'''), (V_i'', V_i''')\} \subseteq \mathcal{E}_{N_1^{\mathrm{ixl}}}$;
  - a disjunctive interconnecting net $H_{\mathrm{D}}(|\mathcal{V}'''|)$ over the set of features $\mathcal{B}$, which is connected to the set of features $\mathcal{V}'''$;
  - $(B, U_2) \in \mathcal{E}_{N_1^{\mathrm{ixl}}}$.

  Besides the usual CP tables for features of $F_s(\phi)$, the other CP tables of N_1^{ixl} are:
  - the CP table of features $F \in (\mathcal{V}' \cup \mathcal{V}'')$ is $f \succ \overline{f}$;
  - the CP table of features $V_i''' \in \mathcal{V}'''$ contains $(\overline{v_i'} \vee \overline{v_i''}) \to \overline{v_i'''} \succ v_i'''$, and is $v_i''' \succ \overline{v_i'''}$, otherwise;
  - the CP tables for features in $\mathcal{B}$ of the interconnecting net are the usual ones;
  - the CP table of feature U_2 contains $(\overline{a} \vee \overline{b}) \to \overline{u_2} \succ u_2$, and is $u_2 \succ \overline{u_2}$, otherwise.

- N_2^{ixl} is a direct net, in particular $N_2^{\mathrm{ixl}} = D(\overline{\alpha})$, with $\overline{\alpha}$ defined over all features of $\mathcal{M}_{\mathrm{ixl}}(\phi)$, and having overlined values only for features U_1 and U_2.

- $N_3^{\mathrm{ixl}} = \langle \mathcal{F}_{N_3^{\mathrm{ixl}}}, \mathcal{E}_{N_3^{\mathrm{ixl}}}\rangle$ is as follows (see Figure 6 for a schematic illustration).

  Links of N_3^{ixl} are the following:

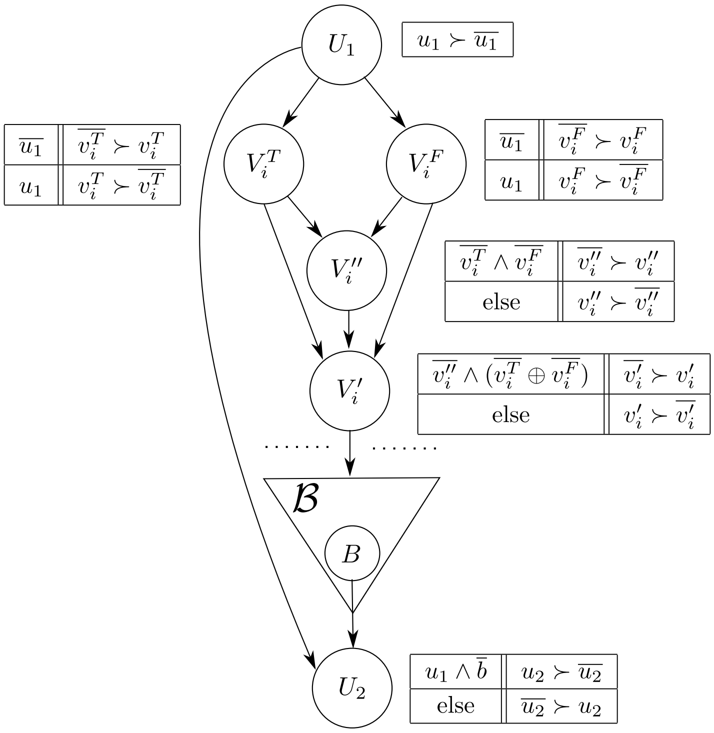


Figure 6: A schematic illustration of part of net N_3^{ixl} of $\mathcal{M}_{\text{ixl}}(\phi)$. Not all features are represented.

- for each $x_i \in X$, $\{(U_1, V_i^T), (U_1, V_i^F), (V_i^T, V_i'), (V_i^F, V_i'), (V_i^T, V_i''), (V_i^F, V_i''), (V_i'', V_i')\} \subseteq \mathcal{E}_{N_3^{\text{ixl}}}$;
- a conjunctive interconnecting net $H_C(|\mathcal{V}'|)$ over the feature set $\mathcal{B}$, which is connected to the features in $\mathcal{V}'$;
- $\{(B, U_2), (U_1, U_2)\} \subseteq \mathcal{E}_{N_3^{\text{ixl}}}$.

The CP tables of N_3^{ixl} are the following:

- the CP table of feature U_1 is $u_1 \succ \overline{u_1}$;
- the CP table of features $F \in \mathcal{V}$ contains $(\overline{u_1}) \to \overline{f} \succ f$ and $(u_1) \to f \succ \overline{f}$;
- the CP table of features $V_i'' \in \mathcal{V}''$ contains $(\overline{v_i^T} \wedge \overline{v_i^F}) \to \overline{v_i''} \succ v_i''$, and is $v_i'' \succ \overline{v_i''}$, otherwise;
- the CP table of features $V_i' \in \mathcal{V}'$ contains $(\overline{v_i''} \wedge (\overline{v_i^T} \oplus \overline{v_i^F})) \to \overline{v_i'} \succ v_i'$, and is $v_i' \succ \overline{v_i'}$, otherwise;
- the CP tables of features in $\mathcal{B}$ of the interconnecting net are the usual ones;
- the CP table of U_2 contains $(u_1 \wedge \overline{b}) \to u_2 \succ \overline{u_2}$, and is $\overline{u_2} \succ u_2$, otherwise;
- there is a direct net $D(\gamma)$, with γ defined over the features $\mathcal{V}''' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A}$ and assigning non-overlined values to all the features.

- N_4^{ixl} is similar to the net N_3^{ixl}, with the only differences that the features U_1 and U_2 are exchanged, and the CP tables of N_4^{ixl} are adjusted to reflect this change.

Observe that $\mathcal{M}_{\text{ixl}}(\Phi)$ is acyclic, binary, its in-degree is three, and can be computed in polynomial time from Φ. Moreover, the class of mCP-nets $\{\mathcal{M}_{\text{ixl}}(\Phi)\}_\Phi$ derived from formulas Φ of the specified kind and according to the reduction shown above is polynomially connected. $\lhd$

We report here an important property of summarized formula nets.

Lemma A.2 ([45, Lemma 5.2])**.** *Let $\phi(X)$ be a Boolean formula in 3CNF defined over a set X of Boolean variables, and let σ_X be an assignment on X. Let α_{σ_X} be the outcome of $F_s(\phi)$ encoding σ_X on the feature set $\mathcal{V}$, and assigning non-overlined values to all other features. Let $\overline{\beta}$ be an outcome of $F_s(\phi)$ such that $\overline{\beta}[U_1U_2] = \overline{u_1u_2}$, assigning any value to the features of $\mathcal{V}$, and assigning non-overlined values to all other features. Then:*
(1) There is an extension of σ_X to X satisfying $\phi(X)$ iff $\overline{\beta} \succ_{F_s(\phi)} \alpha_{\sigma_X}$;
(2) There is no extension of σ_X to X satisfying $\phi(X)$ iff $\overline{\beta} \bowtie_{F_s(\phi)} \alpha_{\sigma_X}$.

We can now prove Lemma 6.3.

Lemma 6.3. *There exists a polynomial-time reduction from the problem of deciding the validity of quantified Boolean formulas $\Phi = (\forall X)(\exists Y)\phi(X, Y)$, where $\phi(X, Y)$ is in 3CNF, to the problem of deciding whether an* *outcome is max optimal in mCP-nets. The mCP-nets obtained in the reduction are binary, acyclic, formed by four agents, they have in-degree three, and they constitute a polynomially connected class of mCP-nets.*

Proof of Lemma 6.3. Let $\Phi = (\forall X)(\exists Y)\phi(X,Y)$ be a quantified Boolean formula, where $\phi(X,Y)$ is in 3CNF. Consider the reduction defined in Construction A.1, which builds the mCP-net $\mathcal{M}_{\text{ixl}}(\Phi)$, and the outcome $\overline{\alpha}$ assigning overlined values only to U_1 and U_2. We show that Φ is valid iff $\overline{\alpha}$ is a max optimal outcome in $\mathcal{M}_{\text{ixl}}(\Phi)$.

To prove this, we have to analyze the max dominance relationship between $\overline{\alpha}$ and all the other outcomes. Remember that in $\overline{\alpha}$ features U_1 and U_2 are overlined. To analyze the max dominance relationships, we partition the set of all possible outcomes into two sets O_d and O_c. In O_c, there are outcomes encoding (partial or complete) truth assignments for the variables X. In particular, any outcome β in O_c assigns non-overlined values to features $\mathcal{V}' \cup \mathcal{V}'' \cup \mathcal{V}''' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B}$, and β encodes a (partial or complete) Boolean assignment σ_X for variables X over features $\mathcal{V}$. Since there is a one-to-one relationship between outcomes in O_c and Boolean assignments σ_X for variables X, we can also denote the outcomes belonging to O_c as β_{σ_X}, where σ_X is the assignment encoded in the outcome.

Intuitively, O_c is the set of outcomes "*c*andidate" to max dominate $\overline{\alpha}$, and for this reason, we use the subscript "c" in O_c, whereas O_d contains outcomes that cannot max dominate $\overline{\alpha}$. The aims of this proof are: showing that all outcomes in O_d cannot max dominate $\overline{\alpha}$; and showing that only outcomes of a subset S (whose precise characterization is given towards the end of the proof) of O_c may max dominate $\overline{\alpha}$. In particular, we show that all outcomes in $(O_d \cup O_c) \setminus S$ do not max dominate $\overline{\alpha}$. Then, we prove that outcomes in S, which may be empty, max dominate $\overline{\alpha}$. Hence, $\overline{\alpha}$ is max optimal iff S is empty. We now formally define the two sets O_d and O_c:

- $O_d = {O_d}' \cup {O_d}'' \cup {O_d}''' \cup {O_d}''''$, where
  - ${O_d}' = \{\beta \in \mathcal{O}_{\mathcal{M}_{\text{ixl}}(\Phi)} \mid (\exists F)(F \in (\mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A}) \wedge \beta[F] = \overline{f})\}$;
  - ${O_d}'' = \{\beta \in \mathcal{O}_{\mathcal{M}_{\text{ixl}}(\Phi)} \mid \beta[U_1U_2] \neq u_1u_2\}$;
  - ${O_d}''' = \{\beta \in \mathcal{O}_{\mathcal{M}_{\text{ixl}}(\Phi)} \mid (\exists F)(F \in (\mathcal{V}' \cup \mathcal{V}'' \cup \mathcal{V}''' \cup \mathcal{B}) \wedge \beta[F] = \overline{f})\}$;
  - ${O_d}'''' = \{\beta \in \mathcal{O}_{\mathcal{M}_{\text{ixl}}(\Phi)} \mid (\exists i)(\beta[V_i^T V_i^F] = \overline{v_i^T v_i^F})\}$.
- $O_c = \{\beta \in \mathcal{O}_{\mathcal{M}_{\text{ixl}}(\Phi)} \mid \beta \notin O_d\}$.

Observe that the sets ${O_d}'$, ${O_d}''$, ${O_d}'''$, and ${O_d}''''$, do not constitute a partition of O_d, because they are not disjoint.

Note that, since $\mathcal{M}_{\text{ixl}}(\Phi)$ is a 4CP-net, $|S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta, \overline{\alpha})| \leq 1$ implies that $\beta \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$. The next four properties show that no outcome $\beta \in O_d$ max dominates $\overline{\alpha}$.

Property 6.3.(1). *Let $\beta' \in {O_d}'$ be an outcome. Then, $\beta' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$.*

Proof. Since $\beta' \in {O_d}'$, there is a feature $F \in (\mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A})$ such that $\beta[F] = \overline{f}$. By the definition of the CP-nets of $\mathcal{M}_{\text{ixl}}(\Phi)$, $\beta' \not\succ_{N_2^{\text{ixl}}} \overline{\alpha}$, $\beta' \not\succ_{N_3^{\text{ixl}}} \overline{\alpha}$, and $\beta' \not\succ_{N_4^{\text{ixl}}} \overline{\alpha}$. Therefore, $|S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta', \overline{\alpha})| \leq 1$, and hence $\beta' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$. □

Property 6.3.(2). *Let $\beta'' \in {O_d}''$ be an outcome. Then, $\beta'' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$.*

Proof. Observe that $\overline{\alpha} \in {O_d}''$, and if $\beta'' = \overline{\alpha}$, then $\beta'' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$. So, let us assume that $\beta'' \neq \overline{\alpha}$ for the rest of the proof of this property. There are three cases: (1) $\beta''[U_1U_2] = \overline{u_1}u_2$, or (2) $\beta''[U_1U_2] = u_1\overline{u_2}$, or (3) $\beta''[U_1U_2] = \overline{u_1u_2}$.

Let us consider Case (1). Let $\overline{\alpha}'$ be the outcome assigning an overlined value only to feature U_1. There are two cases: either (a) $\beta'' = \overline{\alpha}'$, or (b) $\beta'' \neq \overline{\alpha}'$. Consider Case (a). By the definition of the CP-nets of $\mathcal{M}_{\text{ixl}}(\Phi)$, $\beta'' = \overline{\alpha}' \succ_{N_1^{\text{ixl}}} \overline{\alpha}$ (because $\beta'' = \overline{\alpha}'$, and hence $\beta''[AB] = ab$), $\overline{\alpha} \succ_{N_2^{\text{ixl}}} \overline{\alpha}' = \beta''$, $\overline{\alpha} \succ_{N_3^{\text{ixl}}} \overline{\alpha}' = \beta''$ (because $\beta'' = \overline{\alpha}'$, and hence $\beta''[U_1] = \overline{u_1}$), and $\beta'' = \overline{\alpha}' \succ_{N_4^{\text{ixl}}} \overline{\alpha}$. Therefore $|S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta'', \overline{\alpha})| = 2$ and $|S^{\prec}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta'', \overline{\alpha})| = 2$, which implies that $\beta'' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$.

Consider now Case (b). By definition of N_2^{ixl}, $\overline{\alpha} \succ_{N_2^{\text{ixl}}} \beta''$. Let us now focus on net N_3^{ixl}. Since U_1 has no parents in N_3^{ixl}, in any improving flipping sequence in N_3^{ixl}, once U_1 is flipped from $\overline{u_1}$ to u_1, U_1 cannot be flipped back. Hence, in any improving flipping sequence of N_3^{ixl} from $\overline{\alpha}$ to β'' (if it exists), since $\overline{\alpha}[U_1] = \overline{u_1} = \beta''[U_1]$ (we are in Case (1)), feature U_1 cannot be flipped at all. However this implies that, by the definition of the CP table of U_2 in N_3^{ixl}, U_2 cannot be flipped from $\overline{u_2}$ to u_2, which is required to reach β'' (we are in Case (1)). Therefore, in N_3^{ixl}, there is no improving flipping sequence from $\overline{\alpha}$ to β'', which implies that $\beta'' \not\succ_{N_3^{\text{ixl}}} \overline{\alpha}$.

Consider now net N_1^{ixl}. Recall that $\overline{\alpha}'$ is the outcome assigning an overlined value only to feature U_1. By the definition of this net, the only outcome dominating $\overline{\alpha}$ in N_1^{ixl} is $\overline{\alpha}'$. Because we are assuming $\beta'' \neq \overline{\alpha}'$, $\beta'' \not\succ_{N_1^{\text{ixl}}} \overline{\alpha}$.

Therefore, $|S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta'',\overline{\alpha})| \leq 1$, and hence $\beta'' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$.

Let us now consider Case (2). Again, let $\overline{\alpha}'$ be the outcome assigning an overlined value only to feature U_1. By the definition of N_1^{ixl}, the only outcome dominating $\overline{\alpha}$ in N_1^{ixl} is $\overline{\alpha}'$. Also in this case $\beta'' \neq \overline{\alpha}'$, because $\beta''[U_1U_2] = u_1\overline{u_2} \neq \overline{u_1}u_2 = \overline{\alpha}'[U_1U_2]$, and hence $\beta'' \not\succ_{N_1^{\text{ixl}}} \overline{\alpha}$. With a similar argument to the one used in Case (1)(b) for N_3^{ixl}, we can show that in N_4^{ixl} there is no improving flipping sequence from $\overline{\alpha}$ to β'' (simply focus on U_2 instead of U_1), which implies that $\beta'' \not\succ_{N_4^{\text{ixl}}} \overline{\alpha}$. Moreover, we know that $\beta'' \not\succ_{N_2^{\text{ixl}}} \overline{\alpha}$. Therefore, $|S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta'',\overline{\alpha})| \leq 1$, and hence $\beta'' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$.

Let us consider Case (3). First we remind the reader that we are assuming that $\beta'' \neq \overline{\alpha}$. The following is an improving flipping sequence from β'' to $\overline{\alpha}$ in N_1^{ixl}. We can flip, in the proper order, to their non-overlined values all the features of the following sets in the specified sequence: $\mathcal{V}'$, $\mathcal{V}''$, $\mathcal{V}'''$, $\mathcal{B}$, $\mathcal{V}$, $\mathcal{W}$, $\mathcal{P}$, $\mathcal{D}$, and $\mathcal{A}$. Observe that the obtained outcome is precisely $\overline{\alpha}$, and hence $\overline{\alpha} \succ_{N_1^{\text{ixl}}} \beta''$. Remember that by definition of N_2^{ixl}, $\overline{\alpha} \succ_{N_2^{\text{ixl}}} \beta''$. Therefore, $|S^{\prec}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta'',\overline{\alpha})| \geq 2$, and hence $|S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta'',\overline{\alpha})| \leq 2$, which implies that $\beta'' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$. ☐

Property 6.3.(3). *Let $\beta''' \in O_d{}'''$ be an outcome. Then, $\beta''' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$.*

Proof. By Property 6.3.(2), we can assume that $\beta'''[U_1U_2] = u_1u_2$. We will show that in N_1^{ixl} there is an improving flipping sequence from β''' to $\overline{\alpha}$. Since there is a feature $F \in \mathcal{V}' \cup \mathcal{V}'' \cup \mathcal{V}''' \cup \mathcal{B}$ such that $\beta'''[F] = \overline{f}$, the disjunctive interconnecting net in N_1^{ixl} allows us to flip the proper features of the interconnecting net until we can flip also U_2 from u_2 to $\overline{u_2}$. Then, in the proper order, we can flip to their non-overlined values all features in $\mathcal{V}' \cup \mathcal{V}'' \cup \mathcal{V}''' \cup \mathcal{B}$. Next, we flip U_1 from u_1 to $\overline{u_1}$, and after this we can flip to their non-overlined values all the remaining features having overlined values but U_2. Observe that the outcome reached is exactly $\overline{\alpha}$, and hence $\overline{\alpha} \succ_{N_1^{\text{ixl}}} \beta'''$. By definition of N_2^{ixl}, $\overline{\alpha} \succ_{N_2^{\text{ixl}}} \beta'''$ as well.

Therefore, $|S^{\prec}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta''',\overline{\alpha})| \geq 2$, and hence $|S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta''',\overline{\alpha})| \leq 2$, implying that $\beta''' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$. ☐

Property 6.3.(4). *Let $\beta'''' \in O_d{}''''$ be an outcome. Then, $\beta'''' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$.*

Proof. By Property 6.3.(2), we can assume that $\beta''''[U_1U_2] = u_1u_2$. Let us focus on N_3^{ixl}. We will show that $\beta'''' \not\succ_{N_3^{\text{ixl}}} \overline{\alpha}$. Let us assume by contradiction that $\beta'''' \succ_{N_3^{\text{ixl}}} \overline{\alpha}$, hence there must be an improving flipping sequence $\rho\colon \delta_0 \to \ldots \to \delta_z$ from $\overline{\alpha} = \delta_0$ to $\beta'''' = \delta_z$.

Since $\delta_0[U_2] = \overline{u_2}$ and $\delta_z[U_2] = u_2$ (because we are assuming $\beta''''[U_2] = u_2$), there must be an index s such that $\delta_s[U_2] = \overline{u_2}$, $\delta_{s+1}[U_2] = u_2$, and $\delta_s \xrightarrow{U_2} \delta_{s+1}$. By the CP table of U_2 in N_3^{ixl}, this requires that $\delta_s[U_1B] = u_1\overline{b}$. Observe that all features in $\mathcal{B}$ have non-overlined values in δ_0, therefore, in order for $\delta_s[B] = \overline{b}$ to be true, by the definition of the conjunctive interconnecting net in N_3^{ixl}, there is an index $r < s$ such that in δ_r all features in $\mathcal{V}'$ have overlined values. Consider a feature $V_i' \in \mathcal{V}'$ for which the pair of features $\{V_i^T, V_i^F\}$ is such that $\beta''''[V_i^TV_i^F] = \delta_z[V_i^TV_i^F] = \overline{v_i^Tv_i^F}$. Since $\delta_0[V_i'] = v_i'$ and $\delta_r[V_i'] = \overline{v_i'}$, there must be an index $q < r$ such that $\delta_q[V_i'] = v_i'$, $\delta_{q+1}[V_i'] = \overline{v_i'}$, and $\delta_q \xrightarrow{V_i'} \delta_{q+1}$. By the CP table of V_i' in N_3^{ixl}, this requires that $\delta_q[V_i''] = \overline{v_i''}$ and that either $\delta_q[V_i^TV_i^F] = \overline{v_i^T}v_i^F$ or $\delta_q[V_i^TV_i^F] = v_i^T\overline{v_i^F}$. Since $\delta_0[V_i''] = v_i''$ and $\delta_q[V_i''] = \overline{v_i''}$, there must be and index $p < q$ such that $\delta_p[V_i''] = v_i''$, $\delta_{p+1}[V_i''] = \overline{v_i''}$, and $\delta_p \xrightarrow{V_i''} \delta_{p+1}$. By the CP table of V_i'' in N_3^{ixl}, this requires that $\delta_p[V_i^TV_i^F] = \overline{v_i^Tv_i^F}$. Since $\delta_0[V_i^TV_i^F] = v_i^Tv_i^F$ and $\delta_p[V_i^TV_i^F] = \overline{v_i^Tv_i^F}$, it must be the case that V_i^T and V_i^F are flipped to their overlined values before the p-th step of the sequence.

Observe that U_1 is without parents in N_3^{ixl}, and hence once it is flipped from $\overline{u_1}$ to u_1, it cannot be flipped back. Moreover, V_i^T and V_i^F can be flipped from v_i^T to $\overline{v_i^T}$, and from v_i^F to $\overline{v_i^F}$, respectively, iff U_1 has value $\overline{u_1}$, instead they can be flipped from $\overline{v_i^T}$ to v_i^T, and from $\overline{v_i^F}$ to v_i^F, respectively, iff U_1 has value u_1. Since $\delta_p[V_i^TV_i^F] = \overline{v_i^Tv_i^F}$ and in the q-th step either $\delta_q[V_i^TV_i^F] = \overline{v_i^T}v_i^F$ or $\delta_q[V_i^TV_i^F] = v_i^T\overline{v_i^F}$, it must be the case that U_1 is flipped from $\overline{u_1}$ to u_1 at some p'-th step with $p < p' < q$. We know that U_1 cannot be flipped back to $\overline{u_1}$ after the p'-th step, hence it is not possible to flip the pair of features $\{V_i^T, V_i^F\}$ from either $\overline{v_i^T}v_i^F$ or $v_i^T\overline{v_i^F}$ to $\overline{v_i^Tv_i^F}$ after the p'-th step, which contradicts that $\delta_z[V_i^TV_i^F] = \overline{v_i^Tv_i^F}$.

Therefore, it must be the case that $\beta'''' \not\succ_{N_3^{\text{ixl}}} \overline{\alpha}$.

Similarly, it can be shown that $\beta'''' \not\succ_{N_4^{\text{ixl}}} \overline{\alpha}$. Thus, N_3^{ixl} and N_4^{ixl} do not belong to $S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta'''',\overline{\alpha})$. Moreover, by definition of N_2^{ixl}, $\overline{\alpha} \succ_{N_2^{\text{ixl}}} \beta'''$, and hence $N_2^{\text{ixl}} \in S^{\prec}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta'''',\overline{\alpha})$. Therefore, $|S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta'''',\overline{\alpha})| \leq 1$, and thus $\beta'''' \not\succ^{x}_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$. ☐

Let us now consider the outcomes in O_c. Recall that, by the definition of O_c, each outcome $\beta \in O_c$ is such that there is a (partial or complete) Boolean assignment for the variables X such that $\beta = \beta_{\sigma_X}$, and there is a one-to-one relationship between outcomes in O_c and partial/complete Boolean assignments for variables X. Observe that the variables X are universally quantified in Φ, hence no single assignment σ_X can be a witness of the validity of Φ. Single assignments over X can only be witnesses of the *non*-validity of Φ. Therefore, in this context, when we say that an assignment σ_X is a witness, then it is meant that σ_X is a witness of the *non*-validity of Φ, i.e., σ_X is such that $(\exists Y)\phi(X/\sigma_X, Y)$ is *not* valid, which means that $\phi(X/\sigma_X, Y)$ is *not* satisfiable. Let us denote by $\overline{Witn}_c$ the set of all *complete* assignments σ_X over X such that $(\exists Y)\phi(X/\sigma_X, Y)$ is valid (i.e., such that $\phi(X/\sigma_X, Y)$ is satisfiable). Let $\overline{Witn}$ be the set of all (partial or complete) assignments σ_X over X such that there is an extension of σ_X to X belonging to $\overline{Witn}_c$, and let $Witn$ be the set of all (partial or complete) assignments over X not belonging to $\overline{Witn}$. Remember that if σ_X is a complete assignment over X, then σ_X itself is the unique extension of σ_X to X. Given the above definitions, $O_c^{\overline{Witn}} = \{\beta_{\sigma_X} \in O_c \mid \sigma_X \in \overline{Witn}\}$, and $O_c^{Witn} = \{\beta_{\sigma_X} \in O_c \mid \sigma_X \in Witn\}$ constitute a partition of O_c.

We show in the next three properties that only outcomes belonging to O_c^{Witn} can max dominate $\overline{\alpha}$ in $\mathcal{M}_{\text{ixl}}(\Phi)$. In this respect, O_c^{Witn} is the set S mentioned earlier. We start by proving a basic property.

Property 6.3.(5). *Let $\beta_c \in O_c$ be an outcome. Then, $\beta_c \succ_{N_3^{\text{ixl}}} \overline{\alpha}$ and $\beta_c \succ_{N_4^{\text{ixl}}} \overline{\alpha}$.*

Proof. Let σ_X be the assignment over X such that $\beta_c = \beta_{\sigma_X}$, and let σ'_X be any extension of σ_X to X. First consider net N_3^{ixl}. The following is an improving flipping sequence from $\overline{\alpha}$ to β_c in N_3^{ixl}, which proves that $\beta_c \succ_{N_3^{\text{ixl}}} \overline{\alpha}$.

We first flip all features in $\mathcal{V}$ to their overlined values (remember that $\overline{\alpha}[U_1] = \overline{u_1}$). Then, we flip all features in $\mathcal{V}''$ to their overlined values. After this, we flip U_1 from $\overline{u_1}$ to u_1. Then, we flip the proper features in $\mathcal{V}$ to their non-overlined values in order to obtain an assignment of values for features in $\mathcal{V}$ encoding σ'_X. Observe that we can now flip to their overlined values all features in $\mathcal{V}'$ because σ'_X is a complete assignment (and hence there is no pair of features $\{V_i^T, V_i^F\}$ for which $v_i^T v_i^F$, or $\overline{v_i^T v_i^F}$). Next, we can flip to their overlined values, in the proper order, all features in $\mathcal{B}$ of the interconnecting net (and hence also the apex B). We can now flip U_2 from $\overline{u_2}$ to u_2. After this, we can flip features in $\mathcal{V}$ to values matching those in β_c. Observe that we can do this because the values for the features in $\mathcal{V}$, just before this point of the flipping sequence, reflect the assignment σ'_X that is an extension of σ_X, which, on the other hand, is the assignment encoded in β_c. Therefore, the flips required in this step are from overlined values to non-overlined ones, and this can be done since the value of feature U_1 is u_1. To conclude, we flip, in the proper order, to their non-overlined values all features in $\mathcal{V}''$ (observe that none of the pairs $\{V_i^T, V_i^F\}$ has overlined values for both V_i^T and V_i^F), $\mathcal{V}'$, and $\mathcal{B}$. The obtained outcome is precisely β_c. Similarly, it can be proven that $\beta_c \succ_{N_4^{\text{ixl}}} \overline{\alpha}$. □

For the following two properties it is useful to note that, for any outcome $\beta_c \in O_c$, β_c assigns non-overlined values to all features in $\mathcal{V}' \cup \mathcal{V}'' \cup \mathcal{V}''' \cup \mathcal{B}$. Moreover, also $\overline{\alpha}$ assigns non-overlined values to all features in $\mathcal{V}' \cup \mathcal{V}'' \cup \mathcal{V}''' \cup \mathcal{B}$. Therefore, the part of net N_1^{ixl} over feature sets $\mathcal{V}'$, $\mathcal{V}''$, $\mathcal{V}'''$ and $\mathcal{B}$, does not play an active role in any improving flipping sequence (if exists) either from $\overline{\alpha}$ to β_c, or from β_c to $\overline{\alpha}$, because, in N_1^{ixl}, features in $\mathcal{V}' \cup \mathcal{V}''$ have no parents, and they have already their most preferred values in $\overline{\alpha}$ and β_c. The following property shows that outcomes $\beta_c \in O_c^{\overline{Witn}}$ cannot max dominate $\overline{\alpha}$.

Property 6.3.(6). *Let $\beta_c \in O_c^{\overline{Witn}}$ be an outcome. Then, $\beta_c \not\succ^x_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$.*

Proof. Let $\sigma_X \in \overline{Witn}$ be the (partial or complete) assignment over X such that $\beta_c = \beta_{\sigma_X}$.

Let us focus on net N_1^{ixl}. Consider now the non-quantified formula $\phi(X, Y)$. If we consider the set $X \cup Y$ of all the Boolean variables in ϕ, the assignment σ_X is a partial assignment over $X \cup Y$. Since $\sigma_X \in \overline{Witn}$, $(\exists Y)\phi(X/\sigma_X, Y)$ is valid (i.e., $\phi(X/\sigma_X, Y)$ is satisfiable), and hence there is an extension of σ_X to $X \cup Y$ satisfying ϕ. Therefore, by Lemma A.2, $\overline{\alpha} \succ_{N_1^{\text{ixl}}} \beta_c$.

By the definition of N_2^{ixl}, $\overline{\alpha} \succ_{N_2^{\text{ixl}}} \beta_c$, because $\beta_c \neq \overline{\alpha}$. By Property 6.3.(5), $N_3^{\text{ixl}} \in S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta_c, \overline{\alpha})$ and $N_4^{\text{ixl}} \in S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta_c, \overline{\alpha})$. Therefore, $|S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta_c, \overline{\alpha})| = 2$ and $|S^{\prec}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta_c, \overline{\alpha})| = 2$, and hence $\beta_c \not\succ^x_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$. □

Next we prove that outcomes $\beta_c \in O_c^{Witn}$ max dominate $\overline{\alpha}$.

Property 6.3.(7). *Let $\beta_c \in O_c^{Witn}$ be an outcome. Then, $\beta_c \succ^x_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$.*

Proof. Let $\sigma_X \in Witn$ be the (partial or complete) assignment over X such that $\beta_c = \beta_{\sigma_X}$. Since $\sigma_X \in Witn$, there is no extension σ'_X of σ_X to X such that $(\exists Y)\phi(X/\sigma'_X, Y)$ is valid, (i.e., such that $\phi(X/\sigma'_X, Y)$ is satisfiable).

Let us focus on net N_1^{ixl}. We claim that $\beta_c \bowtie_{N_1^{\text{ixl}}} \overline{\alpha}$. Consider the non-quantified formula $\phi(X, Y)$. If we consider the set $X \cup Y$ of all the Boolean variables in ϕ, the assignment σ_X is a partial assignment over $X \cup Y$. Since $\phi(X/\sigma_X, Y)$ is not satisfiable, there is no extension of σ_X to $X \cup Y$ satisfying ϕ. Therefore, by Lemma A.2, $\beta_c \bowtie_{N_1^{\text{ixl}}} \overline{\alpha}$.

To conclude, observe that, by the definition of N_2^{ixl}, $\overline{\alpha} \succ_{N_2^{\text{ixl}}} \beta_c$, because $\beta_c \neq \overline{\alpha}$. By Property 6.3.(5), $N_3^{\text{ixl}} \in S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta_c, \overline{\alpha})$ and $N_4^{\text{ixl}} \in S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta_c, \overline{\alpha})$. Therefore, $|S^{\succ}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta_c, \overline{\alpha})| = 2$, $|S^{\prec}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta_c, \overline{\alpha})| = 1$, and $|S^{\bowtie}_{\mathcal{M}_{\text{ixl}}(\Phi)}(\beta_c, \overline{\alpha})| = 1$, and hence $\beta_c \succ^x_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$. □

We are now ready to prove that $\Phi = (\forall X)(\exists Y)\phi(X,Y)$ is valid iff $\overline{\alpha}$ is max optimal in $\mathcal{M}_{\text{ixl}}(\Phi)$.

($\Rightarrow$) Assume that Φ is valid, hence there is no assignment in *Witn*, and hence O_c^{Witn} is empty. Therefore, since by all the properties above only outcomes in O_c^{Witn} max dominate $\overline{\alpha}$, $\overline{\alpha}$ is max optimal in $\mathcal{M}_{\text{ixl}}(\Phi)$.

($\Leftarrow$) Assume that Φ is not valid, hence there is an assignment σ_X for the variables in X such that $\sigma_X \in$ *Witn*. By Property 6.3.(7), $\beta_{\sigma_X} \succ^x_{\mathcal{M}_{\text{ixl}}(\Phi)} \overline{\alpha}$, and hence $\overline{\alpha}$ is not max optimal in $\mathcal{M}_{\text{ixl}}(\Phi)$. □

We now focus on the proof of Lemma 6.5. Also the transformation for this proof uses the *summarized formula nets* $F_s(\phi)$ (see Section 5.1 in [45]), the concept of encoding of Boolean assignments in these nets (see Section 3.1 in [45]), direct nets (see Section 5.1 in [45]), and conjunctive/disjunctive interconnecting nets (see Section 4.1 in [45]).

Construction A.3. Let $\Phi = (\exists X)(\forall Y)(\exists Z)\phi(X,Y,Z)$ be a quantified formula, where $\phi(X,Y,Z)$ is a 3CNF Boolean formula defined over three disjoint sets $X = \{x_1,\dots,x_{n_X}\}$, $Y = \{y_1,\dots,y_{n_Y}\}$, and $Z = \{z_1,\dots,z_{n_Z}\}$, of Boolean variables, and whose set of clauses is $C = \{c_1,\dots,c_m\}$. From Φ, we define the 8CP-net $\mathcal{M}_{\text{exl}}(\Phi) = \langle N_1^{\text{exl}},\dots,N_8^{\text{exl}}\rangle$ as follows (intuitions on the aims of the components of this construction can be found in [48]).

The features of $\mathcal{M}_{\text{exl}}(\Phi)$ are:

- all the features of a net $F_s(\phi)$ in which, in this case, we distinguish three variable feature sets $\mathcal{V} = \{V_i^T, V_i^F \mid x_i \in X\}$, $\mathcal{W} = \{W_i^T, W_i^F \mid y_i \in Y\}$, and $\mathcal{T} = \{T_i^T, T_i^F \mid z_i \in Z\}$ (recall that $\mathcal{P}$ and $\mathcal{D}$ are the sets of literal and clause features, respectively, and $\mathcal{A}$ is the set of features of the conjunctive interconnecting net embedded in $F_s(\phi)$); for further reference, we call A the apex of the interconnecting net;
- all the features of the sets $\mathcal{W}' = \{W_i' \mid y_i \in Y\}$ and $\mathcal{W}'' = \{W_i'' \mid y_i \in Y\}$;
- all the features of the set $\mathcal{B}$ which are the features B_i of a conjunctive interconnecting net $H_{\text{C}}(|\mathcal{W}'|)$ and its apex is feature B (the features B_i are distinct from the features A_i of the conjunctive interconnecting net $H_{\text{C}}(m)$ embedded in $F_s(\phi)$).

To summarize, all the features of $\mathcal{M}_{\text{exl}}(\Phi)$ are $\mathcal{V}\cup\mathcal{W}\cup\mathcal{W}'\cup\mathcal{W}''\cup\mathcal{T}\cup\mathcal{P}\cup\mathcal{D}\cup\mathcal{A}\cup\mathcal{B}\cup\{U_1,U_2\}$.

The CP-nets of $\mathcal{M}_{\text{exl}}(\Phi)$ are:

- N_1^{exl} is composed by a net $F_s(\phi)$ with its features, links, and CP tables. There is also a direct net $D(\gamma)$, where γ is defined over the set of features $\mathcal{W}'\cup\mathcal{W}''\cup\mathcal{B}$ and assigns non-overlined values to all of them.

- N_2^{exl}, for every $x_i \in X$, has a link from the feature V_i^T to the feature V_i^F. The CP table for feature V_i^T is $v_i^T \succ \overline{v_i^T}$, while the CP table for feature V_i^F contains $(\overline{v_i^T}) \to v_i^F \succ \overline{v_i^F}$ and $(v_i^T) \to \overline{v_i^F} \succ v_i^F$.

  Then, there is a direct net $D(\gamma)$, with γ defined over all the features of $\mathcal{M}_{\text{ixl}}(\phi)$ but those in $\mathcal{V}$. The outcome γ has non-overlined values for all its features but U_1 and U_2, which have overlined values.

- N_3^{exl} is similar to N_2^{exl} with the difference that the features V_i^T and V_i^F are exchanged, and the CP tables of N_3^{exl} are adjusted accordingly to reflect this change.

- $N_4^{\text{exl}} = \langle \mathcal{F}_{N_4^{\text{exl}}}, \mathcal{E}_{N_4^{\text{exl}}}\rangle$ is as follows (see Figure 6 for a schematic illustration, where features V_i^T, V_i^F, V_i', and V_i'' have to be substituted by W_i^T, W_i^F, W_i', and W_i'', respectively).

  Links of N_4^{exl} are the following:

  - for each $y_i \in Y$, $\{(U_1,W_i^T),(U_1,W_i^F),(W_i^T,W_i'),(W_i^F,W_i'),(W_i^T,W_i''),(W_i^F,W_i''),(W_i'',W_i')\} \subseteq \mathcal{E}_{N_4^{\text{exl}}}$;
  - a conjunctive interconnecting net $H_{\text{C}}(|\mathcal{W}'|)$ over the feature set $\mathcal{B}$ is connected to the features in $\mathcal{W}'$, and the apex B is linked to U_2;
  - $(U_1,U_2) \in \mathcal{E}_{N_4^{\text{exl}}}$.

  The CP tables of N_4^{exl} are the following:

  - the CP table of feature U_1 is $u_1 \succ \overline{u_1}$;
  - the CP table of features $F \in \mathcal{W}$ contains $(\overline{u_1}) \to \overline{f} \succ f$ and $(u_1) \to f \succ \overline{f}$;
  - the CP table of features $W_i'' \in \mathcal{W}''$ contains $(\overline{w_i^T} \wedge \overline{w_i^F}) \to \overline{w_i''} \succ w_i''$, and is $w_i'' \succ \overline{w_i''}$, otherwise;
  - the CP table of features $W_i' \in \mathcal{W}'$ contains $(\overline{w_i''} \wedge (\overline{w_i^T} \oplus \overline{w_i^F})) \to \overline{w_i'} \succ w_i'$, and is $w_i' \succ \overline{w_i'}$, otherwise;
  - the CP tables of features in $\mathcal{B}$ of the interconnecting net are the usual ones;
  - the CP table of U_2 contains $(u_1 \wedge \overline{b}) \to u_2 \succ \overline{u_2}$, and is $\overline{u_2} \succ u_2$, otherwise;
  - there is a direct net $D(\gamma)$, with γ defined over the features $\mathcal{P}\cup\mathcal{D}\cup\mathcal{A}\cup\mathcal{T}\cup\mathcal{V}$ and assigning overlined values to the features in $\mathcal{V}$ and non-overlined values to all the others.

- N_5^{exl} is similar to N_4^{exl} with the difference that the features U_1 and U_2 are exchanged, and the CP tables of N_5^{exl} are adjusted accordingly to reflect this change.

- $N_6^{\text{exl}} = \langle \mathcal{F}_{N_6^{\text{exl}}}, \mathcal{E}_{N_6^{\text{exl}}} \rangle$ is as follows. The links of N_6^{exl} are:
  - for each $x_i \in X$, $(V_i^T, V_i^F) \in \mathcal{E}_{N_6^{\text{exl}}}$;
  - for each $y_i \in Y$, $\{(U_1, W_i^T), (U_1, W_i^F), (U_2, W_i^T), (U_2, W_i^F)\} \subseteq \mathcal{E}_{N_6^{\text{exl}}}$;
  - $(U_2, U_1) \in \mathcal{E}_{N_6^{\text{exl}}}$.

  The CP tables of N_6^{exl} are:
  - the CP tables of features V_i^T and V_i^F are as those in N_2^{exl};
  - the CP table of feature U_2 is $u_2 \succ \overline{u_2}$;
  - the CP table of feature U_1 contains $(\overline{u_2}) \to u_1 \succ \overline{u_1}$ and $(u_2) \to \overline{u_1} \succ u_1$;
  - the CP table of features $F \in \mathcal{W}$ contains $(u_1 \wedge \overline{u_2}) \to \overline{f} \succ f$, and is $f \succ \overline{f}$, otherwise;
  - there is a direct net $D(\gamma)$, with γ defined over all features but $\mathcal{V} \cup \mathcal{W} \cup \{U_1, U_2\}$, and assigning non-overlined values to all of them.

- $N_7^{\text{exl}} = \langle \mathcal{F}_{N_7^{\text{exl}}}, \mathcal{E}_{N_7^{\text{exl}}} \rangle$ is as follows. The links of N_7^{exl} are:
  - for each $x_i \in X$, $(V_i^F, V_i^T) \in \mathcal{E}_{N_7^{\text{exl}}}$;
  - $(U_2, U_1) \in \mathcal{E}_{N_7^{\text{exl}}}$.

  The CP tables of N_7^{exl} are:
  - the CP tables of features V_i^T and V_i^F are as those in N_3^{exl};
  - the CP table of feature U_2 is $\overline{u_2} \succ u_2$;
  - the CP table of feature U_1 contains $(\overline{u_2}) \to \overline{u_1} \succ u_1$ and $(u_2) \to u_1 \succ \overline{u_1}$;
  - there is a direct net $D(\gamma)$, with γ defined over all features but $\mathcal{V} \cup \{U_1, U_2\}$, and assigning non-overlined values to all of them.

- $N_8^{\text{exl}} = \langle \mathcal{F}_{N_8^{\text{exl}}}, \mathcal{E}_{N_8^{\text{exl}}} \rangle$ is as follows. The links of N_8^{exl} are:
  - for each $y_i \in Y$, $\{(U_1, W_i^T), (U_1, W_i^F), (U_2, W_i^T), (U_2, W_i^F)\} \subseteq \mathcal{E}_{N_8^{\text{exl}}}$;
  - $(U_1, U_2) \in \mathcal{E}_{N_8^{\text{exl}}}$.

  The CP tables of N_8^{exl} are:
  - the CP table of feature U_1 is $u_1 \succ \overline{u_1}$;
  - the CP table of feature U_2 contains $(\overline{u_1}) \to \overline{u_2} \succ u_2$ and $(u_1) \to u_2 \succ \overline{u_2}$;
  - the CP tables of features $F \in \mathcal{W}$ contains $(\overline{u_1} \wedge \overline{u_2}) \to \overline{f} \succ f$, and is $f \succ \overline{f}$, otherwise;
  - there is a direct net $D(\gamma)$, with γ defined over all features but $\mathcal{W} \cup \{U_1, U_2\}$ and assigning overlined values to all the features in $\mathcal{V}$ and non-overlined values to all the remaining features.

Observe that $\mathcal{M}_{\text{exl}}(\Phi)$ is acyclic, binary, its in-degree is three, and can be computed in polynomial time from Φ. Moreover, the class of mCP-nets $\{\mathcal{M}_{\text{exl}}(\Phi)\}_{\Phi}$ derived from formulas Φ of the specified kind and according to the reduction shown above is polynomially connected. $\lhd$

We can now prove Lemma 6.5.

Lemma 6.5. *There exists a polynomial-time reduction from the problem of deciding the validity of quantified Boolean formulas $\Phi = (\exists X)(\forall Y)(\exists Z)\phi(X, Y, Z)$, where $\phi(X, Y, Z)$ is in 3CNF, to the problem of deciding whether mCP-nets admit a max optimal outcome. The mCP-nets obtained in the reduction are binary, acyclic, formed by eight agents, they have in-degree three, and they constitute a polynomially connected class of mCP-nets.*

Proof of Lemma 6.5. Let $\Phi = (\exists X)(\forall Y)(\exists Z)\phi(X, Y, Z)$ be a quantified Boolean formula, where $\phi(X, Y, Z)$ is in 3CNF. Consider the reduction defined in Construction A.3, which builds the mCP-net $\mathcal{M}_{\text{exl}}(\Phi)$. We show that Φ is valid iff $\mathcal{M}_{\text{exl}}(\Phi)$ admits a max optimal outcome.

To prove this, we have to analyze the max dominance relationship between all the outcome pairs. Again, to this aim, we partition the set of all possible outcomes into two sets O_d and O_c. In O_c, there are outcomes encoding (in this case) *complete* truth assignments for the variables X. In particular, any outcome β in O_c assigns non-overlined values to features $\mathcal{W} \cup \mathcal{W}' \cup \mathcal{W}'' \cup \mathcal{T} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B}$, β assigns overlined values to U_1 and U_2, and β encodes a complete Boolean assignment σ_X for variables X over features $\mathcal{V}$. Also in this case, as there is a one-to-one relationship between outcomes in O_c and complete Boolean assignments σ_X for variables X, we can also denote the outcomes belonging to O_c as $\overline{\beta}_{\sigma_X}$, where σ_X is the assignment for variables X encoded in the outcome. In this context, as the outcomes in O_c, unlike those in the proof of Lemma 6.3, assigns overlined values to the features U_1 and U_2, we overline the symbol "$\overline{\beta}_{\sigma_X}$" to remind us this fact.

Intuitively, O_c is the set of outcomes "*c*andidate" to be max optimal, and for this reason, we use the subscript "c" in O_c, while O_d contains outcomes that surely are not max optimal. The aims of this proof are: showing that all outcomes in O_d are not max optimal; and showing that only outcomes of a subset S (whose precise characterization will be given toward the end of the proof) of O_c may be max optimal. In particular, we obtain this by proving that all outcomes in $(O_d \cup O_c) \setminus S$ are max dominated by some other outcome, and hence they are not max optimal; and then by proving that all outcomes in S, which might be empty, are not max dominated, and hence they are max optimal. Therefore, $\mathcal{M}_{\text{exl}}(\Phi)$ has a max optimal outcome iff S is not empty. Let us now formally define the two sets O_d and O_c:

- $O_d = O_d' \cup O_d'' \cup O_d''' \cup O_d''''$, where
  - $O_d' = \{\beta \in \mathcal{O}_{\mathcal{M}_{\text{exl}}(\Phi)} \mid (\exists F)(F \in (\mathcal{W}' \cup \mathcal{W}'' \cup \mathcal{T} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B}) \wedge \beta[F] = \overline{f})\}$;
  - $O_d'' = \{\beta \in \mathcal{O}_{\mathcal{M}_{\text{exl}}(\Phi)} \mid \beta[U_1U_2] \neq \overline{u_1u_2}\}$;
  - $O_d''' = \{\beta \in \mathcal{O}_{\mathcal{M}_{\text{exl}}(\Phi)} \mid (\exists F)(F \in \mathcal{W} \wedge \beta[F] = \overline{f})\}$;
  - $O_d'''' = \{\beta \in \mathcal{O}_{\mathcal{M}_{\text{exl}}(\Phi)} \mid (\exists i)(\beta[V_i^T V_i^F] = v_i^T v_i^F \vee \beta[V_i^T V_i^F] = \overline{v_i^T v_i^F})\}$.
- $O_c = \{\beta \in \mathcal{O}_{\mathcal{M}_{\text{exl}}(\Phi)} \mid \beta \notin O_d\}$.

Observe that O_d', O_d'', O_d''', and O_d'''' do not constitute a partition of O_d because they are not disjoint.

The next four properties show that no outcome $\beta \in O_d$ can be max optimal in $\mathcal{M}_{\text{exl}}(\Phi)$, because there is an outcome max dominating β. Note that, since $\mathcal{M}_{\text{exl}}(\Phi)$ is an 8CP-net, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\beta,\alpha)| \geq 5$ implies that $\beta \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \alpha$.

Property 6.5.(1). *Let $\beta' \in O_d'$ be an outcome. Then, there is an outcome $\gamma \in \mathcal{O}_{\mathcal{M}_{\text{exl}}(\Phi)}$ such that $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta'$.*

Proof. Let $F \in (\mathcal{W}' \cup \mathcal{W}'' \cup \mathcal{T} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B})$ be a feature such that $\beta'[F] = \overline{f}$, and let γ be the outcome for which, for all features $G \neq F$, $\gamma[G] = \beta'[G]$, and $\gamma[F] = f \neq \overline{f} = \beta'[F]$. We show that $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta'$.

By the definition of the CP-nets of $\mathcal{M}_{\text{exl}}(\Phi)$, $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta'$ because $\gamma \succ_{N_2^{\text{exl}}} \beta'$, $\gamma \succ_{N_3^{\text{exl}}} \beta'$, $\gamma \succ_{N_6^{\text{exl}}} \beta'$, $\gamma \succ_{N_7^{\text{exl}}} \beta'$, ad $\gamma \succ_{N_8^{\text{exl}}} \beta'$. Therefore, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma,\beta')| \geq 5$, and hence $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta'$. ☐

Property 6.5.(2). *Let $\beta'' \in O_d''$ be an outcome. Then, there is an outcome $\gamma \in \mathcal{O}_{\mathcal{M}_{\text{exl}}(\Phi)}$ such that $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta''$.*

Proof. There are three cases: (1) $\beta''[U_1U_2] = u_1u_2$; or (2) $\beta''[U_1U_2] = \overline{u_1}u_2$; or (3) $\beta''[U_1U_2] = u_1\overline{u_2}$.

Consider Case (1). Let γ be the outcome such that, for all features $F \neq U_1$, $\gamma[F] = \beta''[F]$, and $\gamma[U_1] = \overline{u_1} \neq u_1 = \beta''[U_1]$. By the definition of the CP-nets in $\mathcal{M}_{\text{exl}}(\Phi)$, $\gamma \succ_{N_1^{\text{exl}}} \beta''$, $\gamma \succ_{N_2^{\text{exl}}} \beta''$, $\gamma \succ_{N_3^{\text{exl}}} \beta''$, and $\gamma \succ_{N_6^{\text{exl}}} \beta''$ (because $\beta''[U_2] = u_2$). Consider now net N_5^{exl}. By Property 6.5.(1), we can limit our attention to those outcomes β'' such that, for all features $F \in \mathcal{B}$, $\beta''[F] = f$. Therefore, we can assume that feature B, i.e., the apex of the interconnecting net in N_5^{exl}, has value b in β''. Hence, by the definition of the CP table of U_1 in N_5^{exl}, $\gamma \succ_{N_5^{\text{exl}}} \beta''$. Therefore, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma,\beta'')| \geq 5$, and hence $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta''$.

Consider Case (2). Let γ be the outcome such that, for all features $F \neq U_2$, $\gamma[F] = \beta''[F]$, and $\gamma[U_2] = \overline{u_2} \neq u_2 = \beta''[U_2]$. By the definition of the CP-nets of $\mathcal{M}_{\text{exl}}(\Phi)$, $\gamma \succ_{N_2^{\text{exl}}} \beta''$, $\gamma \succ_{N_3^{\text{exl}}} \beta''$, $\gamma \succ_{N_4^{\text{exl}}} \beta''$ (because $\beta''[U_1] = \overline{u_1}$), $\gamma \succ_{N_7^{\text{exl}}} \beta''$, and $\gamma \succ_{N_8^{\text{exl}}} \beta''$ (again, because $\beta''[U_1] = \overline{u_1}$). Therefore, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma,\beta'')| \geq 5$, and hence $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta''$.

Consider Case (3). Let γ be the outcome such that, for all features $F \neq U_1$, $\gamma[F] = \beta''[F]$, and $\gamma[U_1] = \overline{u_1} \neq u_1 = \beta''[U_1]$. By the definition of the CP-nets of $\mathcal{M}_{\text{exl}}(\Phi)$, $\gamma \succ_{N_1^{\text{exl}}} \beta''$, $\gamma \succ_{N_2^{\text{exl}}} \beta''$, $\gamma \succ_{N_3^{\text{exl}}} \beta''$, $\gamma \succ_{N_5^{\text{exl}}} \beta''$ (because $\beta''[U_2] = \overline{u_2}$), and $\gamma \succ_{N_7^{\text{exl}}} \beta''$ (again, because $\beta''[U_2] = \overline{u_2}$). Therefore, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma,\beta'')| \geq 5$, and hence $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta''$. ☐

Property 6.5.(3). *Let $\beta''' \in O_d'''$ be an outcome. Then, there is an outcome $\gamma \in \mathcal{O}_{\mathcal{M}_{\text{exl}}(\Phi)}$ such that $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta'''$.*

Proof. Let $F \in \mathcal{W}$ be a feature such that $\beta'''[F] = \overline{f}$. Let γ be the outcome such that, for all features $G \neq F$, $\gamma[G] = \beta'''[G]$, and $\gamma[F] = f \neq \overline{f} = \beta'''[F]$. By Property 6.5.(2), we can limit our attention to outcomes β''' such that $\beta'''[U_1U_2] = \overline{u_1u_2}$. By the definition of the CP-nets of $\mathcal{M}_{\text{exl}}(\Phi)$, $\gamma \succ_{N_1^{\text{exl}}} \beta'''$ (because $\beta'''[U_1] = \overline{u_1}$), $\gamma \succ_{N_2^{\text{exl}}} \beta'''$, $\gamma \succ_{N_3^{\text{exl}}} \beta'''$, $\gamma \succ_{N_6^{\text{exl}}} \beta'''$ (see the CP tables in N_6^{exl} of features in $\mathcal{W}$, and remember that only feature $F \in \mathcal{W}$ changes its value from β''' to γ), and $\gamma \succ_{N_7^{\text{exl}}} \beta'''$. Therefore, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma,\beta''')| \geq 5$, and hence $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta'''$. ☐

Property 6.5.(4). *Let $\beta'''' \in O_d''''$ be an outcome. Then, there is an outcome $\gamma \in \mathcal{O}_{\mathcal{M}_{\text{exl}}(\Phi)}$ such that $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta''''$.*

Proof. First, let us consider the case in which $\{V_i^T, V_i^F\}$ is a pair of variable features such that $\beta''''[V_i^T V_i^F] = v_i^T \overline{v_i^F}$. Consider the outcome γ such that, for all features $F \notin \{V_i^T, V_i^F\}$, $\gamma[F] = \beta''''[F]$, and $\gamma[V_i^T V_i^F] = v_i^T \overline{v_i^F} \neq v_i^T v_i^F = \beta''''[V_i^T V_i^F]$. By the definition of the CP-nets of $\mathcal{M}_{\mathrm{exl}}(\Phi)$, $\gamma \succ_{N_2^{\mathrm{exl}}} \beta''''$ (because $\beta''''[V_i^T] = v_i^T$), $\gamma \succ_{N_4^{\mathrm{exl}}} \beta''''$, $\gamma \succ_{N_5^{\mathrm{exl}}} \beta''''$, $\gamma \succ_{N_6^{\mathrm{exl}}} \beta''''$ (again, because $\beta''''[V_i^T] = v_i^T$), and $\gamma \succ_{N_8^{\mathrm{exl}}} \beta''''$. Therefore, $|S^{\succ}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\gamma, \beta'''')| \geq 5$, and hence $\gamma \succ^{x}_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \beta''''$.

Let us consider now the case in which there is a pair of features $\{V_i^T, V_i^F\}$ such that $\beta''''[V_i^T V_i^F] = \overline{v_i^T v_i^F}$. Let γ be the outcome such that, for all features $F \notin \{V_i^T, V_i^F\}$, $\gamma[F] = \beta''''[F]$, and $\gamma[V_i^T V_i^F] = v_i^T v_i^F \neq \overline{v_i^T v_i^F} = \beta''''[V_i^T V_i^F]$. By Property 6.5.(2), we can limit our attention to outcomes β'''' such that $\beta''''[U_1 U_2] = \overline{u_1 u_2}$. Since $\beta''''[U_1] = \overline{u_1}$, $\gamma \succ_{N_1^{\mathrm{exl}}} \beta''''$. Let us now consider N_2^{exl}. If in β'''' we flip first V_i^F from $\overline{v_i^F}$ to v_i^F, and then V_i^T from $\overline{v_i^T}$ to v_i^T, we arrive to γ. Hence, $\gamma \succ_{N_2^{\mathrm{exl}}} \beta''''$. The same flipping sequence proves that $\gamma \succ_{N_6^{\mathrm{exl}}} \beta''''$. Similarly, it can be shown that $\gamma \succ_{N_3^{\mathrm{exl}}} \beta''''$ and $\gamma \succ_{N_7^{\mathrm{exl}}} \beta''''$ (first flip V_i^T and then V_i^F). Therefore, $|S^{\succ}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\gamma, \beta'''')| \geq 5$, and hence $\gamma \succ^{x}_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \beta''''$. □

Recall that we aim at showing that a specific subset S of O_c contains outcomes that are max optimal. Therefore, we have to prove that the outcomes in $O_c \setminus S$ are max dominated. In order to achieve this, we show an interesting property of outcomes in O_c: they can be max dominated only by very specific outcomes.

We recall that outcomes in O_c are in one-to-one relationship with complete truth assignments for variables X, and we can hence denote them by $\overline{\beta}_{\sigma_X}$. For a pair of assignments $\langle \sigma_X, \sigma_Y \rangle$ for variables X and Y, respectively, let us denote by $\beta_{\sigma_X,\sigma_Y}$ the outcome assigning non-overlined values to all features but those in $\mathcal{V} \cup \mathcal{W}$, and encoding σ_X over features $\mathcal{V}$ and encoding σ_Y over feature $\mathcal{W}$. In this case, we do not overline the symbol "$\beta_{\sigma_X,\sigma_Y}$" to remind us that in $\beta_{\sigma_X,\sigma_Y}$ the features U_1 and U_2 have non-overlined values. We prove below that a necessary (but not sufficient) condition for an outcome to max dominate an outcome $\overline{\beta}_{\sigma_X}$ of O_c is to be an outcome of the kind $\beta_{\sigma_X,\sigma_Y}$ described above for which, moreover, the encoding of σ_X in $\beta_{\sigma_X,\sigma_Y}$ coincides with the one in $\overline{\beta}_{\sigma_X}$. Essentially, an outcome γ cannot max dominate $\overline{\beta}_{\sigma_X}$ if γ is not in a form of an outcome $\beta_{\sigma'_X,\sigma_Y}$ and moreover $\sigma'_X = \sigma_X$.

The next four properties show the just aforementioned property of outcomes in O_c. Note that, since $\mathcal{M}_{\mathrm{exl}}(\Phi)$ is an 8CP-net, $|S^{\succ}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\beta, \alpha)| \leq 3$ implies that $\beta \not\succ^{x}_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \alpha$.

Property 6.5.(5). *Let $\beta_c \in O_c$ and $\gamma \in \mathcal{O}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}$ be two outcomes such that there is a feature $F \in (\mathcal{W}' \cup \mathcal{W}'' \cup \mathcal{T} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B})$ for which $\beta_c[F] \neq \gamma[F]$. Then, $\gamma \not\succ^{x}_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \beta_c$.*

Proof. Since $\beta_c \in O_c$, $\beta_c[F] = f$ and $\gamma[F] = \overline{f}$. By the definition of the CP-nets in $\mathcal{M}_{\mathrm{exl}}(\Phi)$, $\gamma \not\succ_{N_2^{\mathrm{exl}}} \beta_c$, $\gamma \not\succ_{N_3^{\mathrm{exl}}} \beta_c$, $\gamma \not\succ_{N_6^{\mathrm{exl}}} \beta_c$, $\gamma \not\succ_{N_7^{\mathrm{exl}}} \beta_c$, $\gamma \not\succ_{N_8^{\mathrm{exl}}} \beta_c$. Hence, $|S^{\succ}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\gamma, \beta_c)| \leq 3$, and thus $\gamma \not\succ^{x}_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \beta_c$. □

Before proving the next property, let us carry out the following considerations about the preferences over the values of variable features in $\mathcal{V}$ in nets N_2^{exl}, N_3^{exl}, N_6^{exl}, and N_7^{exl}. In these nets, variable features are connected in pairs $\{V_i^T, V_i^F\}$, in particular either from V_i^T to V_i^F or from V_i^F to V_i^T, and each pair is completely disconnected from the rest of the net. Therefore, in these nets, whether a flip of a feature of the pair $\{V_i^T, V_i^F\}$ is improving or not depends only on the value of the specific features in the pair. This means that, in any improving flipping sequence for N_2^{exl}, N_3^{exl}, N_6^{exl}, or N_7^{exl}, flips cannot violate the preferences' order restricted over each pair of features $\{V_i^T, V_i^F\}$. To give an example, consider net N_2^{exl}. If we restrict our focus over the pair of features $\{V_i^T, V_i^F\}$, preferences of N_2^{exl} projected over these features are $\overline{v_i^T v_i^F} \prec \overline{v_i^T} v_i^F \prec v_i^T v_i^F \prec v_i^T \overline{v_i^F}$. If $\rho\colon \delta_0 \to_{N_2^{\mathrm{exl}}} \ldots \to_{N_2^{\mathrm{exl}}} \delta_z$ is any improving flipping sequence for N_2^{exl}, it cannot be the case that there are two distinct indices i and j such that $i < j$, for which, for example, $\delta_i[V_i^T V_i^F] = v_i^T v_i^F$ and $\delta_j[V_i^T V_i^F] = \overline{v_i^T} v_i^F$, because, in order for this to be true, there would be in ρ flips of features $\{V_i^T, V_i^F\}$ which would not be improving according to their CP tables in N_2^{exl}. For the following discussion, note that for nets N_2^{exl} and N_6^{exl}, their preferences restricted over (V_i^T, V_i^F) are $\overline{v_i^T v_i^F} \prec \overline{v_i^T} v_i^F \prec v_i^T v_i^F \prec v_i^T \overline{v_i^F}$; while, for nets N_3^{exl} and N_7^{exl}, their preferences restricted over (V_i^T, V_i^F) are $\overline{v_i^T v_i^F} \prec v_i^T \overline{v_i^F} \prec v_i^T v_i^F \prec \overline{v_i^T} v_i^F$.

Property 6.5.(6). *Let $\beta_c \in O_c$ and $\gamma \in \mathcal{O}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}$ be two outcomes such that there is a feature $F \in \mathcal{V}$ for which $\beta_c[F] \neq \gamma[F]$. Then, $\gamma \not\succ^{x}_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \beta_c$.*

Proof. Let *Change* $= \{F \in \mathcal{V} \mid \beta_c[F] \neq \gamma[F]\}$ be the set of all variable features in $\mathcal{V}$ changing value from β_c to γ. Let *Up* $= \{F \in \mathcal{V} \mid \beta_c[F] = f \wedge \gamma[F] = \overline{f}\}$ be the subset of *Change* containing variable features in $\mathcal{V}$ changing their value from non-overlined in β_c to overlined in γ. Let *Down* $= \{F \in \mathcal{V} \mid \beta_c[F] = \overline{f} \wedge \gamma[F] = f\}$ be the subset of *Change* containing variable features in $\mathcal{V}$ changing their value from overlined in β_c to non-overlined in γ. The sets *Up* and *Down* constitute a partition of *Change*, and, since from the statement of this property we are assuming that *Change* $\neq \varnothing$, it must be the case that (*Up* $\cup$ *Down*) $\neq \varnothing$. Therefore, there are three cases: (1) *Up* $\neq \varnothing \wedge$ *Down* $\neq \varnothing$, or (2) *Up* $\neq \varnothing \wedge$ *Down* $= \varnothing$, or (3) *Up* $= \varnothing \wedge$ *Down* $\neq \varnothing$. In the following, remember that, since $\beta_c \in O_c$, for each pair of features $\{V_i^T, V_i^F\}$, either $\beta_c[V_i^T V_i^F] = \overline{v_i^T} v_i^F$ or $\beta_c[V_i^T V_i^F] = v_i^T \overline{v_i^F}$.

Consider Case (1). Let us consider N_1^{exl}. Observe that, since $\beta_c \in O_c$, $\beta_c[U_1] = \overline{u_1}$, and, in net N_1^{exl}, $\overline{u_1}$ is the most preferred value of feature U_1 which, moreover, does not have parents in N_1^{exl}. Therefore, in any improving flipping sequence of N_1^{exl} starting from β_c, the value of feature U_1 cannot be flipped at all. This implies that, in any improving flipping sequence of N_1^{exl} starting from β_c, the value of features in $\mathcal{V}$ can be flipped only from overlined to non-overlined. Thus, $\gamma \not\succ_{N_1^{\text{exl}}} \beta_c$, because $Up \neq \varnothing$. Moreover, by the definitions of the CP-nets in $\mathcal{M}_{\text{exl}}(\Phi)$, because $Down \neq \varnothing$, $\gamma \not\succ_{N_4^{\text{exl}}} \beta_c$, $\gamma \not\succ_{N_5^{\text{exl}}} \beta_c$, and $\gamma \not\succ_{N_8^{\text{exl}}} \beta_c$.

Let $F \in Up$ be a feature. First, let us assume that $F = V_i^T$, which implies that $\beta_c[V_i^T V_i^F] = v_i^T \overline{v_i^F}$. Outcome γ is such that either $\gamma[V_i^T V_i^F] = \overline{v_i^T} v_i^F$ or $\gamma[V_i^T V_i^F] = \overline{v_i^T}\,\overline{v_i^F}$, depending on the fact whether V_i^F belongs to *Down* or not, respectively. In both cases, if there were an improving flipping sequence from β_c to γ, it would be against the preferences of N_2^{exl} restricted over (V_i^T, V_i^F) (see above). Therefore, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma, \beta_c)| \leq 3$, and hence $\gamma \not\succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta_c$. On the other hand, if $F = V_i^F$, it follows that $\beta_c[V_i^T V_i^F] = \overline{v_i^T} v_i^F$. Outcome γ is such that either $\gamma[V_i^T V_i^F] = v_i^T \overline{v_i^F}$ or $\gamma[V_i^T V_i^F] = \overline{v_i^T}\,\overline{v_i^F}$, depending on the fact whether V_i^T belongs to *Down* or not, respectively. In both cases, if there were an improving flipping sequence from β_c to γ, it would be against the preferences of N_3^{exl} restricted over (V_i^T, V_i^F) (see above). Therefore, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma, \beta_c)| \leq 3$, and hence $\gamma \not\succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta_c$.

Consider Case (2). We already know that from $Up \neq \varnothing$ follows $\gamma \not\succ_{N_1^{\text{exl}}} \beta_c$ (see Case (1)). Let $F \in Up$ be a feature. Irrespective of the fact whether $F = V_i^T$ or $F = V_i^F$, from $Down = \varnothing$ follows that $\gamma[V_i^T V_i^F] = \overline{v_i^T}\,\overline{v_i^F}$. If there were an improving flipping sequence from β_c to γ, it would be against the preferences of N_2^{exl}, N_3^{exl}, N_6^{exl}, and N_7^{exl} restricted over (V_i^T, V_i^F) (see above). Therefore, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma, \beta_c)| \leq 3$, and hence $\gamma \not\succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta_c$.

Consider Case (3). We already know that from $Down \neq \varnothing$ follows $\gamma \not\succ_{N_4^{\text{exl}}} \beta_c$, $\gamma \not\succ_{N_5^{\text{exl}}} \beta_c$, and $\gamma \not\succ_{N_8^{\text{exl}}} \beta_c$ (see Case (1)). Let $F \in Down$ be a feature. First, let us assume that $F = V_i^T$, which implies $\beta_c[V_i^T V_i^F] = \overline{v_i^T} v_i^F$, and from $Up = \varnothing$ it follows that $\gamma[V_i^T V_i^F] = v_i^T v_i^F$. If there were an improving flipping sequence from β_c to γ, it would be against the preferences of N_3^{exl}, and N_7^{exl}, restricted over (V_i^T, V_i^F) (see above). Therefore, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma, \beta_c)| \leq 3$, and hence $\gamma \not\succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta_c$. On the other hand, if $F = V_i^F$, it follows that $\beta[V_i^T V_i^F] = v_i^T \overline{v_i^F}$, and from $Up = \varnothing$ it follows that $\gamma[V_i^T V_i^F] = v_i^T v_i^F$. If there were an improving flipping sequence from β_c to γ, it would be against the preferences of N_2^{exl}, and N_6^{exl}, restricted over (V_i^T, V_i^F) (see above). Therefore, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma, \beta_c)| \leq 3$, and hence $\gamma \not\succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta_c$. ☐

Property 6.5.(7). *Let $\beta_c \in O_c$ and $\gamma \in \mathcal{O}_{\mathcal{M}_{\text{exl}}(\Phi)}$ be two outcomes such that $\gamma[U_1 U_2] \neq u_1 u_2$. Then, $\gamma \not\succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta_c$.*

Proof. There are three cases: (1) $\gamma[U_1 U_2] = \overline{u_1 u_2}$, or (2) $\gamma[U_1 U_2] = \overline{u_1} u_2$, or (3) $\gamma[U_1 U_2] = u_1 \overline{u_2}$.

Let us consider Case (1). Let us assume by contradiction that $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta_c$, which implies that $\gamma \neq \beta_c$. Since $\gamma[U_1 U_2] = \overline{u_1 u_2} = \beta_c[U_1 U_2]$, by Properties 6.5.(5) and 6.5.(6), it must be the case that the only features changing value from β_c to γ are those in $\mathcal{W}$. Remember that, since $\beta_c \in O_c$, for all features $F \in \mathcal{W}$, $\beta_c[F] = f$. Therefore, for all features $F \in \mathcal{W}$ such that $\beta_c[F] \neq \gamma[F]$, $\gamma[F] = \overline{f} \neq f = \beta_c[F]$. By this, $\beta_c \succ_{N_1^{\text{exl}}} \gamma$ (because $\gamma[U_1] = \beta_c[U_1] = \overline{u_1}$), $\beta_c \succ_{N_2^{\text{exl}}} \gamma$, $\beta_c \succ_{N_3^{\text{exl}}} \gamma$, and $\beta_c \succ_{N_7^{\text{exl}}} \gamma$. Therefore, $|S^{\prec}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma, \beta_c)| \geq 4$, which means that $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma, \beta_c)| \leq 4$, and thus it cannot be the case that $\gamma \succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta_c$: a contradiction.

Consider Case (2). Since $\beta_c[U_1 U_2] = \overline{u_1 u_2}$ and $\gamma[U_1 U_2] = \overline{u_1} u_2$, by the definition of the CP-nets of $\mathcal{M}_{\text{exl}}(\Phi)$, $\gamma \not\succ_{N_2^{\text{exl}}} \beta_c$, $\gamma \not\succ_{N_3^{\text{exl}}} \beta_c$, and $\gamma \not\succ_{N_7^{\text{exl}}} \beta_c$ (because, in N_7^{exl}, $\overline{u_2}$ is the most preferred value of U_2, which is moreover without parents). Consider now net N_4^{exl}. In this net, feature U_1 is without parents. Hence, given the CP table of U_1 in N_4^{exl}, in any improving flipping sequence for N_4^{exl}, once U_1 is flipped from $\overline{u_1}$ to u_1, it cannot be flipped back. Therefore, since $\gamma[U_1] = \beta_c[U_1] = \overline{u_1}$, if there were an improving flipping sequence from β_c to γ in N_4^{exl}, then feature U_1 could not be flipped at all. Given the CP table of feature U_2 in N_4^{exl}, this would imply that also feature U_2 could not be flipped from $\overline{u_2}$ to u_2, which would contradict the existence of an improving flipping sequence from β_c to γ in N_4^{exl}. Therefore, $\gamma \not\succ_{N_4^{\text{exl}}} \beta_c$. Similar considerations allow us to show that $\gamma \not\succ_{N_8^{\text{exl}}} \beta_c$. Thus, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma, \beta_c)| \leq 3$, and hence $\gamma \not\succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta_c$.

Consider now Case (3). Since $\beta_c[U_1 U_2] = \overline{u_1 u_2}$ and $\gamma[U_1 U_2] = u_1 \overline{u_2}$, by the definition of the CP-nets of $\mathcal{M}_{\text{exl}}(\Phi)$, $\gamma \not\succ_{N_1^{\text{exl}}} \beta_c$ (because, in N_1^{exl}, $\overline{u_1}$ is the most preferred value of U_1, which is moreover without parents), $\gamma \not\succ_{N_2^{\text{exl}}} \beta_c$, and $\gamma \not\succ_{N_3^{\text{exl}}} \beta_c$. Consider now net N_5^{exl}. A similar argument to the one for N_4^{exl} in Case (2), but with the roles of features U_1 and U_2 exchanged, shows that $\gamma \not\succ_{N_5^{\text{exl}}} \beta_c$. Similarly, it can be shown that $\gamma \not\succ_{N_7^{\text{exl}}} \beta_c$. Thus, $|S^{\succ}_{\mathcal{M}_{\text{exl}}(\Phi)}(\gamma, \beta_c)| \leq 3$, and hence $\gamma \not\succ^x_{\mathcal{M}_{\text{exl}}(\Phi)} \beta_c$. ☐

Property 6.5.(8). *Let $\beta_c \in O_c$ and $\gamma \in \mathcal{O}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}$ be two outcomes such that there is a pair of variable features $\{W_i^T, W_i^F\}$ for which $\gamma[W_i^T W_i^F] = \overline{w_i^T w_i^F}$. Then, $\gamma \not\succ^x_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \beta_c$.*

Proof. Remember that, since $\beta_c \in O_c$, for all features $F \in \mathcal{W}$, $\beta_c[F] = f$, and hence, any feature of $\mathcal{W}$ changing its value from β_c to γ has an non-overlined value in β_c and an overlined value in γ. This implies that, $\gamma \not\succ_{N_2^{\mathrm{exl}}} \beta_c$, $\gamma \not\succ_{N_3^{\mathrm{exl}}} \beta_c$, and $\gamma \not\succ_{N_7^{\mathrm{exl}}} \beta_c$.

Let us now focus on N_4^{exl}. We will show that $\gamma \not\succ_{N_4^{\mathrm{exl}}} \beta_c$. Let us assume by contradiction that $\gamma \succ_{N_4^{\mathrm{exl}}} \beta_c$, hence there must be an improving flipping sequence ρ: $\delta_0 \to \ldots \to \delta_z$ from $\beta_c = \delta_0$ to $\gamma = \delta_z$.

By Property 6.5.(7), we can limit our attention to those outcomes γ such that $\gamma[U_1U_2] = u_1u_2$. Since $\delta_0[U_2] = \overline{u_2}$ and $\delta_z[U_2] = u_2$, there must be an index s such that $\delta_s[U_2] = \overline{u_2}$, $\delta_{s+1}[U_2] = u_2$, and $\delta_s \xrightarrow{U_2} \delta_{s+1}$. By the CP table of U_2 in N_4^{exl}, this requires that $\delta_s[U_1B] = u_1\overline{b}$. Observe that all features in $\mathcal{B}$ have non-overlined values in δ_0 (because $\beta_c \in O_c$), therefore, in order for $\delta_s[B] = \overline{b}$ to be true, by the definition of the conjunctive interconnecting net in N_4^{exl}, there is an index $r < s$ such that in δ_r all features in $\mathcal{W}'$ have overlined values. Consider precisely the feature $W_i' \in \mathcal{W}'$ for which the pair of features $\{W_i^T, W_i^F\}$ is such that $\gamma[W_i^T W_i^F] = \delta_z[W_i^T W_i^F] = \overline{w_i^T w_i^F}$. Since $\delta_0[W_i'] = w_i'$ and $\delta_r[W_i'] = \overline{w_i'}$, there must be an index $q < r$ such that $\delta_q[W_i'] = w_i'$, $\delta_{q+1}[W_i'] = \overline{w_i'}$, and $\delta_q \xrightarrow{W_i'} \delta_{q+1}$. By the CP table of W_i' in N_4^{exl}, this requires that $\delta_q[W_i''] = \overline{w_i''}$ and that either $\delta_q[W_i^T W_i^F] = \overline{w_i^T} w_i^F$ or $\delta_q[W_i^T W_i^F] = w_i^T \overline{w_i^F}$. Since $\delta_0[W_i''] = w_i''$ and $\delta_q[W_i''] = \overline{w_i''}$, there must be and index $p < q$ such that $\delta_p[W_i''] = w_i''$, $\delta_{p+1}[W_i''] = \overline{w_i''}$, and $\delta_p \xrightarrow{W_i''} \delta_{p+1}$. By the CP table of W_i'' in N_4^{exl}, this requires that $\delta_p[W_i^T W_i^F] = \overline{w_i^T w_i^F}$. Since $\delta_0[W_i^T W_i^F] = w_i^T w_i^F$ and $\delta_p[W_i^T W_i^F] = \overline{w_i^T w_i^F}$, it must be the case that W_i^T and W_i^F are flipped to their overlined values before the p-th step of the sequence.

Observe that U_1 is without parents in N_4^{exl}, and hence once it is flipped from $\overline{u_1}$ to u_1, it cannot be flipped back. Moreover, W_i^T and W_i^F can be flipped from w_i^T to $\overline{w_i^T}$ and from w_i^F to $\overline{w_i^F}$, respectively, iff U_1 has value $\overline{u_1}$, instead they can be flipped from $\overline{w_i^T}$ to w_i^T and from $\overline{w_i^F}$ to w_i^F, respectively, iff U_1 has value u_1. Since $\delta_p[W_i^T W_i^F] = \overline{w_i^T w_i^F}$ and in the q-th step either $\delta_q[W_i^T W_i^F] = \overline{w_i^T} w_i^F$ or $\delta_q[W_i^T W_i^F] = w_i^T \overline{w_i^F}$, it must be the case that U_1 is flipped from $\overline{u_1}$ to u_1 at some p'-th step with $p < p' < q$. We know that U_1 cannot be flipped back to $\overline{u_1}$ after the p'-th step, hence it is not possible to flip the pair of features $\{W_i^T, W_i^F\}$ from either $\overline{w_i^T} w_i^F$ or $w_i^T \overline{w_i^F}$ to $\overline{w_i^T w_i^F}$ after the p'-th step, which contradicts that $\delta_z[W_i^T W_i^F] = \overline{w_i^T w_i^F}$.

Therefore, it must be the case that $\gamma \not\succ_{N_4^{\mathrm{exl}}} \beta_c$. Similarly, it can be proven that $\gamma \not\succ_{N_5^{\mathrm{exl}}} \beta_c$. This implies that $|S^{\succ}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\gamma, \beta_c)| \leq 3$, and hence that $\gamma \not\succ^x_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \beta_c$. □

Now that we have identified the outcomes that have the ability to max dominate the outcomes of O_c, we can characterize precisely the subset S of O_c containing the outcomes that are not max dominated and hence are max optimal. We will see that the outcomes of S are those encoding specific truth assignments for X. In particular, we can distinguish Boolean assignments for X into two categories. Since the variables X are existentially quantified in Φ, a single assignment σ_X for X can be a witness of the validity of Φ. More precisely, we say that an assignment σ_X for variables X is a witness of the validity of Φ if $(\forall Y)(\exists Z)\phi(X/\sigma_X, Y, Z)$ is valid. Let *Witn* be the set of all *complete* assignments σ_X for the variables in X for which $(\forall Y)(\exists Z)\phi(X/\sigma_X, Y, Z)$ is valid, and let $\overline{\mathit{Witn}}$ be the set of all *complete* assignments σ_X over X not belonging to *Witn*. Given the above definitions, $O_c^{\mathit{Witn}} = \{\overline{\beta}_{\sigma_X} \in O_c \mid \sigma_X \in \mathit{Witn}\}$, and $O_c^{\overline{\mathit{Witn}}} = \{\overline{\beta}_{\sigma_X} \in O_c \mid \sigma_X \in \overline{\mathit{Witn}}\}$ constitute a partition of O_c. We now show that only outcomes belonging to O_c^{Witn} are not max dominated, and hence they are candidate to be max optimal in $\mathcal{M}_{\mathrm{exl}}(\Phi)$. In this respect, O_c^{Witn} is the set S mentioned earlier. The following two properties show that outcomes in $O_c^{\overline{\mathit{Witn}}}$ are max dominated. The intuition here is that an outcome $\overline{\beta}_{\tilde{\sigma}_X} \in O_c^{\overline{\mathit{Witn}}}$ is max dominated by an outcome $\beta_{\tilde{\sigma}_X, \sigma_Y}$, where σ_Y is the assignment disproving the validity of $(\forall Y)(\exists Z)\phi(X/\tilde{\sigma}_X, Y, Z)$ (recall that the variables Y are universally quantified, and hence a single assignment for Y can be a witness of non-validity for the formula $(\forall Y)(\exists Z)\phi(X/\tilde{\sigma}_X, Y, Z)$).

Property 6.5.(9). *Let $\beta_c \in O_c^{\overline{\mathit{Witn}}}$ be an outcome, let σ_X be the complete assignment over X such that $\beta_c = \overline{\beta}_{\sigma_X}$, and let σ_Y be any complete assignment over Y. Then, $\beta_{\sigma_X,\sigma_Y} \succ_{N_4^{\mathrm{exl}}} \overline{\beta}_{\sigma_X}$, and $\beta_{\sigma_X,\sigma_Y} \succ_{N_5^{\mathrm{exl}}} \overline{\beta}_{\sigma_X}$.*

Proof. First, consider net N_4^{exl}. The following is an improving flipping sequence from $\overline{\beta}_{\sigma_X}$ to $\beta_{\sigma_X,\sigma_Y}$. We first flip all features in $\mathcal{W}$ to their overlined values (we can do this because $\overline{\beta}_{\sigma_X}[U_1] = \overline{u_1}$). Then, we flip all features in $\mathcal{W}''$ to their overlined values. After this, we flip U_1 from $\overline{u_1}$ to u_1. Then, we flip the proper features in $\mathcal{W}$ to their non-overlined values in order to obtain an assignment of values for features in $\mathcal{W}$ identical to that in $\beta_{\sigma_X,\sigma_Y}$ (i.e., in order to encode σ_Y over $\mathcal{W}$). Observe that we can now flip to their overlined values all features in $\mathcal{W}'$ because σ_Y is a complete assignment (and hence there is no pair of features $\{W_i^T, W_i^F\}$ for which $\beta_{\sigma_X,\sigma_Y}[W_i^T W_i^F] = w_i^T w_i^F$, or $\beta_{\sigma_X,\sigma_Y}[W_i^T W_i^F] = \overline{w_i^T w_i^F}$). Next, we can flip to their overlined values, in the proper order, all features in $\mathcal{B}$ of the interconnecting net (and hence also the apex B).

We can now flip U_2 from $\overline{u_2}$ to u_2. To conclude, we flip, in the proper order, to their non-overlined values all features in $\mathcal{W}''$ (observe that none of the pairs $\{W_i^T, W_i^F\}$ has overlined values for both W_i^T and W_i^F), $\mathcal{W}'$, and $\mathcal{B}$. The obtained outcome is precisely $\beta_{\sigma_X,\sigma_Y}$. Similarly, it can be shown that $\beta_{\sigma_X,\sigma_Y} \succ_{N_5^{\mathrm{exl}}} \overline{\beta}_{\sigma_X}$. ☐

For the following two properties, observe that, since $\overline{\beta}_{\sigma_X} \in O_c$, $\overline{\beta}_{\sigma_X}$ assigns non-overlined values to all features in $\mathcal{W}' \cup \mathcal{W}'' \cup \mathcal{B}$. Moreover, also $\beta_{\sigma_X,\sigma_Y}$ assigns non-overlined values to all features in $\mathcal{W}' \cup \mathcal{W}'' \cup \mathcal{B}$. Therefore, the part of net N_1^{exl} over feature sets $\mathcal{W}'$, $\mathcal{W}''$, and $\mathcal{B}$, does not play an active role in any improving flipping sequence (if it exists) either from $\overline{\beta}_{\sigma_X}$ to $\beta_{\sigma_X,\sigma_Y}$, or from $\beta_{\sigma_X,\sigma_Y}$ to $\overline{\beta}_{\sigma_X}$, because, in N_1^{exl}, features in $\mathcal{W}' \cup \mathcal{W}'' \cup \mathcal{B}$ have no parents, and they have already their most preferred values in $\overline{\beta}_{\sigma_X}$ and $\beta_{\sigma_X,\sigma_Y}$.

We now show that outcomes in $O_c^{\overline{Witn}}$ are not max optimal in $\mathcal{M}_{\mathrm{exl}}(\Phi)$.

Property 6.5.(10). *Let $\beta_c \in O_c^{\overline{Witn}}$ be an outcome. Then, there is an outcome $\gamma \in \mathcal{O}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}$ such that $\gamma \succ^x_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \beta_c$.*

Proof. Let $\sigma_X \in \overline{Witn}$ be the complete assignment over X such that $\beta_c = \overline{\beta}_{\sigma_X}$. Since $\sigma_X \in \overline{Witn}$, there is a complete assignment σ_Y over the variables Y such that $(\exists Z)\phi(X/\sigma_X, Y/\sigma_Y, Z)$ is not valid (i.e., $\phi(X/\sigma_X, Y/\sigma_Y, Z)$ is *not* satisfiable). We will show that $\beta_{\sigma_X,\sigma_Y} \succ^x_{\mathcal{M}_{\mathrm{exl}}(\oplus)} \overline{\beta}_{\sigma_X} = \beta_c$.

By the definition of the CP-nets of $\mathcal{M}_{\mathrm{exl}}(\Phi)$, $\beta_{\sigma_X,\sigma_Y} \succ_{N_6^{\mathrm{exl}}} \overline{\beta}_{\sigma_X}$ (we first flip U_1 from $\overline{u_1}$ to u_1, next we flip the proper features in $\mathcal{W}$ to encode σ_Y, and then we flip U_2 from $\overline{u_2}$ to u_2), and $\beta_{\sigma_X,\sigma_Y} \succ_{N_8^{\mathrm{exl}}} \overline{\beta}_{\sigma_X}$ (we first flip the proper features in $\mathcal{W}$ to encode σ_Y, next we flip U_1 from $\overline{u_1}$ to u_1, and then we flip U_2 from $\overline{u_2}$ to u_2). By Property 6.5.(9), $\beta_{\sigma_X,\sigma_Y} \succ_{N_4^{\mathrm{exl}}} \overline{\beta}_{\sigma_X}$, and $\beta_{\sigma_X,\sigma_Y} \succ_{N_5^{\mathrm{exl}}} \overline{\beta}_{\sigma_X}$. Therefore $|S^{\succ}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\beta_{\sigma_X,\sigma_Y}, \overline{\beta}_{\sigma_X})| \geq 4$.

Now consider net N_1^{exl}. We claim that $\beta_{\sigma_X,\sigma_Y} \bowtie_{N_1^{\mathrm{exl}}} \overline{\beta}_{\sigma_X}$. Consider the non-quantified formula $\phi(X, Y, Z)$. If we consider the set $X \cup Y \cup Z$ of all the Boolean variables in ϕ, the joint assignment $\sigma_X \cup \sigma_Y$ is a partial assignment over $X \cup Y \cup Z$. Since $\phi(X/\sigma_X, Y/\sigma_Y, Z)$ is not satisfiable, there is no extension of $\sigma_X \cup \sigma_Y$ to $X \cup Y \cup Z$ satisfying ϕ. Therefore, by Lemma A.2, $\beta_{\sigma_X,\sigma_Y} \bowtie_{N_1^{\mathrm{exl}}} \overline{\beta}_{\sigma_X}$.

By the definition of the CP-nets of $\mathcal{M}_{\mathrm{exl}}(\Phi)$, $\overline{\beta}_{\sigma_X} \succ_{N_2^{\mathrm{exl}}} \beta_{\sigma_X,\sigma_Y}$, $\overline{\beta}_{\sigma_X} \succ_{N_3^{\mathrm{exl}}} \beta_{\sigma_X,\sigma_Y}$, and $\overline{\beta}_{\sigma_X} \succ_{N_7^{\mathrm{exl}}} \beta_{\sigma_X,\sigma_Y}$.

To summarize, we have shown that $|S^{\succ}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\beta_{\sigma_X,\sigma_Y}, \overline{\beta}_{\sigma_X})| = 4$, $|S^{\bowtie}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\beta_{\sigma_X,\sigma_Y}, \overline{\beta}_{\sigma_X})| = 1$, and $|S^{\prec}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\beta_{\sigma_X,\sigma_Y}, \overline{\beta}_{\sigma_X})| = 3$. Therefore $\beta_{\sigma_X,\sigma_Y} \succ^x_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \overline{\beta}_{\sigma_X} = \beta_c$. ☐

We now show that outcomes in O_c^{Witn} are not max dominated by any other outcome in $\mathcal{M}_{\mathrm{exl}}(\Phi)$. The intuition here is that outcomes $\overline{\beta}_{\tilde{\sigma}_X}$ in O_c^{Witn} are associated with assignments $\tilde{\sigma}_X$ that are witnesses for $(\exists X)(\forall Y)(\exists Z)\phi(X, Y, Z)$, and hence there are no disprovers of the validity of $(\forall Y)(\exists Z)\phi(X/\tilde{\sigma}_X, Y, Z)$. For this reason, no outcome $\beta_{\tilde{\sigma}_X,\sigma_X}$ exists that can max dominate $\overline{\beta}_{\tilde{\sigma}_X}$.

Property 6.5.(11). *Let $\beta_c \in O_c^{Witn}$ be an outcome. Then, there is no outcome $\gamma \in \mathcal{O}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}$ such that $\gamma \succ^x_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \overline{\beta}_{\sigma_X}$.*

Proof. Let $\sigma_X \in Witn$ be the complete assignment over X such that $\beta_c = \overline{\beta}_{\sigma_X}$. We know already that, if γ is not in the form of an outcome $\beta_{\sigma_X,\sigma_Y}$, then γ does not max dominate β_c. Let $\gamma \in \mathcal{O}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}$ be any outcome candidate to max dominate $\overline{\beta}_{\sigma_X}$, and let σ_Y be the (partial or complete) assignment over Y such that $\gamma = \beta_{\sigma_X,\sigma_Y}$. We will show that $\gamma = \beta_{\sigma_X,\sigma_Y} \not\succ^x_{\mathcal{M}_{\mathrm{exl}}(\oplus)} \overline{\beta}_{\sigma_X} = \beta_c$.

By Property 6.5.(9), $\{N_4^{\mathrm{exl}}, N_5^{\mathrm{exl}},\} \subseteq S^{\succ}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\beta_{\sigma_X,\sigma_Y}, \overline{\beta}_{\sigma_X})$. In the proof of Property 6.5.(10), we showed that $\{N_6^{\mathrm{exl}}, N_8^{\mathrm{exl}}\} \subseteq S^{\succ}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\beta_{\sigma_X,\sigma_Y}, \overline{\beta}_{\sigma_X})$ and that $\{N_2^{\mathrm{exl}}, N_3^{\mathrm{exl}}, N_7^{\mathrm{exl}}\} \subseteq S^{\prec}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\beta_{\sigma_X,\sigma_Y}, \overline{\beta}_{\sigma_X})$.

Let us now focus on net N_1^{exl}. Consider the non-quantified formula $\phi(X, Y, Z)$. If we consider the set $X \cup Y \cup Z$ of all the Boolean variables in ϕ, the joint assignment $\sigma_X \cup \sigma_Y$ is a partial assignment over $X \cup Y \cup Z$. Since $\sigma_X \in Witn$, $(\forall Y)(\exists Z)\phi(X/\sigma_X, Y/\sigma_Y, Z)$ is valid, and hence, irrespective of σ_Y being actually a partial or a complete assignment over Y, there is an extension of $\sigma_X \cup \sigma_Y$ to $X \cup Y \cup Z$ satisfying ϕ. Therefore, by Lemma A.2, $\overline{\beta}_{\sigma_X} \succ_{N_1^{\mathrm{exl}}} \beta_{\sigma_X,\sigma_Y}$.

Therefore, $|S^{\succ}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\beta_{\sigma_X,\sigma_Y}, \overline{\beta}_{\sigma_X})| = 4$, and $|S^{\prec}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}(\beta_{\sigma_X,\sigma_Y}, \overline{\beta}_{\sigma_X})| = 4$, implying that $\beta_{\sigma_X,\sigma_Y} \not\succ^x_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \overline{\beta}_{\sigma_X}$. Since, for all outcomes γ candidate to max dominate $\overline{\beta}_{\sigma_X}$, there is a (partial or complete) assignment σ_Y over variables in Y such that $\gamma = \beta_{\sigma_X,\sigma_Y}$, and we showed that $\beta_{\sigma_X,\sigma_Y} \not\succ^x_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \overline{\beta}_{\sigma_X}$, there is no outcome $\gamma \in \mathcal{O}_{\mathcal{M}_{\mathrm{exl}}(\Phi)}$ such that $\gamma \succ^x_{\mathcal{M}_{\mathrm{exl}}(\Phi)} \overline{\beta}_{\sigma_X}$. ☐

We are now ready to prove that $\Phi = (\exists X)(\forall Y)(\exists Z)\phi(X, Y, Z)$ is valid iff $\mathcal{M}_{\mathrm{exl}}(\Phi)$ has a max optimal outcome.

($\Rightarrow$) If Φ is valid, then there is a complete assignment σ_X over the variables in X such that $\sigma_X \in Witn$. Thus, by Property 6.5.(11), $\overline{\beta}_{\sigma_X}$ is max optimal in $\mathcal{M}_{\mathrm{exl}}(\Phi)$.

($\Leftarrow$) If Φ is not valid, then $Witn$ is empty, and hence $O_c = O_c^{\overline{Witn}}$. Therefore, by Properties 6.5.(1), 6.5.(2), 6.5.(3), 6.5.(4), and 6.5.(10), $\mathcal{M}_{\mathrm{exl}}(\Phi)$ has no max optimal outcome. ☐

References

[1] T. E. Allen. *CP-nets: From Theory to Practice.* Phd thesis, Department of Computer Science, University of Kentucky, 2016.

[2] T. E. Allen, J. Goldsmith, H. E. Justice, N. Mattei, and K. Raines. Uniform random generation and dominance testing for CP-nets. *J. Artif. Intell. Res.*, 59:771–813, 2017.

[3] K. J. Arrow, A. K. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare, Volume 1*, volume 19 of *Handbooks in Economics.* North Holland/Elsevier, Amsterdam, The Netherlands, 2002.

[4] K. J. Arrow, A. K. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare, Volume 2*, volume 19 of *Handbooks in Economics.* North Holland/Elsevier, Amsterdam, The Netherlands, 2011.

[5] D. Baumeister and J. Rothe. Preference aggregation by voting. In J. Rothe, editor, *Economics and Computation*, chapter 4, pages 197–326. Springer-Verlag, Berlin/Heidelberg, Germany, 2016.

[6] J. C. Boerkoel Jr., E. H. Durfee, and K. Purrington. Generalized solution techniques for preference-based constrained optimization with CP-nets. In *Proc. of AAMAS*, pages 291–298, 2010.

[7] S. Borgwardt, B. Fazzinga, T. Lukasiewicz, A. Shrivastava, and O. Tifrea-Marciuska. Preferential query answering over the Semantic Web with possibilistic networks. In *Proc. of IJCAI*, pages 994–1000, 2016.

[8] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.*, 21:135–191, 2004.

[9] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. Preference-based constrained optimization with CP-nets. *Comput. Intell.*, 20(2):137–157, 2004.

[10] R. I. Brafman and Y. Chernyavsky. Planning with goal preferences and constraints. In *Proc. of ICAPS*, pages 182–191, 2005.

[11] R. I. Brafman and C. Domshlak. TCP-nets for preference-based product configuration. In *Proc. of the ECAI 2002 Workshop on Configuration*, pages 101–106, 2002.

[12] R. I. Brafman, F. Rossi, D. Salvagnin, K. B. Venable, and T. Walsh. Finding the next solution in constraint- and preference-based knowledge representation formalisms. In *Proc. of KR*, pages 425–433, 2010.

[13] S. J. Brams, D. M. Kilgour, and W. S. Zwicker. The paradox of multiple elections. *Soc. Choice Welfare*, 15 (2):211–236, 1998.

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

[15] F. Brandt, M. Brill, and P. Harrenstein. Tournament solutions. In *Handbook of Computational Social Choice*, pages 57–84. Cambridge University Press, New York, NY, USA, 2016.

[16] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors. *Handbook of Computational Social Choice.* Cambridge University Press, New York, NY, USA, 2016.

[17] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. Preference handling in combinatorial domains: From AI to social choice. *AI Mag.*, 29(4):37–46, 2008.

[18] V. Conitzer and T. Sandholm. Communication complexity of common voting rules. In *Proc. of EC*, pages 78–87, 2005.

[19] T. Csar, M. Lackner, R. Pichler, and E. Sallinger. Winner determination in huge elections with mapreduce. In *Proc. of AAAI*, pages 451–458, 2017.

[20] T. Di Noia, T. Lukasiewicz, M. V. Martinez, G. I. Simari, and O. Tifrea-Marciuska. Combining existential rules with the power of CP-theories. In *Proc. of IJCAI*, pages 2918–2925, 2015.

[21] C. Domshlak, F. Rossi, K. B. Venable, and T. Walsh. Reasoning about soft constraints and conditional preferences: Complexity results and approximation techniques. In *Proc. of IJCAI*, pages 215–220, 2003.

[22] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proc. of WWW*, pages 613–622, 2001.

[23] A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, editors. *Knowledge-Based Configuration: From Research to Business Cases.* Morgan Kaufmann/Elsevier, Waltham, MA, USA, 2014.

[24] J. Goldsmith, J. Lang, M. Truszczynski, and N. Wilson. The computational complexity of dominance and consistency in CP-nets. *J. Artif. Intell. Res.*, 33:403–432, 2008.

[25] G. Greco, E. Malizia, L. Palopoli, and F. Scarcello. Constrained coalitional games: formal framework, properties, and complexity results. In *Proc. of AAMAS*, pages 1295–1296, 2009.

[26] G. Greco, E. Malizia, L. Palopoli, and F. Scarcello. Non-transferable utility coalitional games via mixed-integer linear constraints. *J. Artif. Intell. Res.*, 38:633–685, 2010.

[27] R. Greenlaw, H. J. Hoover, and W. L. Ruzzo. *Limits to Parallel Computation: P-Completeness Theory.* Oxford University Press, New York, NY, USA, 1995.

[28] A. Haret, A. Novaro, and U. Grandi. Preference aggregation with incomplete CP-nets. In *Proc. of KR*, pages 308–317, 2018.

[29] D. S. Johnson. A catalog of complexity classes. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science (Vol. A)*, pages 67–161. Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1990.

[30] R. E. Ladner. The circuit value problem is log space complete for P. *ACM SIGACT News*, 7(1):18–20, 1975.

[31] K. Laing. *Conditional Preference Networks: Efficient Dominance Testing and Learning.* PhD Thesis, School of Mathematics, University of Leeds, 2020.

[32] K. Laing, P. A. Thwaites, and J. P. Gosling. Rank pruning for dominance queries in CP-nets. *J. Artif. Intell. Res.*, 64:55–107, 2019.

[33] J. Lang. From preference representation to combinatorial vote. In *Proc. of KR*, pages 277–290, 2002.

[34] J. Lang. Logical preference representation and combinatorial vote. *Ann. Math Artif. Intel.*, 42(1):37–71, 2004.

[35] J. Lang. Vote and aggregation in combinatorial domains with structured preferences. In *Proc. of IJCAI*, pages 1366–1371, 2007.

[36] J. Lang and L. Xia. Sequential composition of voting rules in multi-issue domains. *Math. Soc. Sci.*, 57(3): 304–324, 2009.

[37] J. Lang and L. Xia. Voting in combinatorial domains. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, pages 197–222. Cambridge University Press, New York, NY, USA, 2016.

[38] M. Li, Q. B. Vo, and R. Kowalczyk. An efficient majority-rule-based approach for collective decision making with CP-nets. In *Proc. of KR*, pages 578–580, 2010.

[39] M. Li, Q. B. Vo, and R. Kowalczyk. An efficient procedure for collective decision-making with CP-nets. In *Proc. of ECAI*, pages 375–380, 2010.

[40] M. Li, Q. B. Vo, and R. Kowalczyk. Majority-rule-based preference aggregation on multi-attribute domains with CP-nets. In *Proc. of AAMAS*, pages 659–666, 2011.

[41] M. Li, Q. B. Vo, and R. Kowalczyk. Efficient heuristic approach to dominance testing in CP-nets. In *Proc. of AAMAS*, pages 353–360, 2011.

[42] M. Li, Q. B. Vo, and R. Kowalczyk. Aggregating multi-valued CP-nets: A CSP-based approach. *J. Heuristics*, 21(1):107–140, 2015.

[43] T. Lukasiewicz and E. Malizia. On the complexity of mCP-nets. In *Proc. of AAAI*, pages 558–564, 2016.

[44] T. Lukasiewicz and E. Malizia. A novel characterization of the complexity class Θ_k^P based on counting and comparison. *Theor. Comput. Sci.*, 694:21–33, 2017.

[45] T. Lukasiewicz and E. Malizia. Complexity results for preference aggregation over (m)CP-nets: Pareto and majority voting. *Artif. Intell.*, 272:101–142, 2019.

[46] T. Lukasiewicz, M. V. Martinez, and G. I. Simari. Preference-based query answering in Datalog+/– ontologies. In *Proc. of IJCAI*, pages 1017–1023, 2013.

[47] T. Lukasiewicz, M. V. Martinez, G. I. Simari, and O. Tifrea-Marciuska. Ontology-based query answering with group preferences. *ACM Trans. Internet Technol.*, 14(4):25:1–25:24, 2014.

[48] E. Malizia. More complexity results about reasoning over (m)CP-nets. In *Proc. of AAMAS*, pages 1540–1548, 2018.

[49] S. Miyano, S. Shiraishi, and T. Shoudai. A list of P-complete problems. TR-CS 17, RIFIS, Kyushu University, Japan, 1990.

[50] C. H. Papadimitriou. *Computational Complexity*. Addison Wesley, Reading, MA, USA, 1994.

[51] S. D. Prestwich, F. Rossi, K. B. Venable, and T. Walsh. Constraint-based preferential optimization. In *Proc. of AAAI*, pages 461–466, 2005.

[52] F. Ricci, L. Rokach, and B. Shapira, editors. *Recommender Systems Handbook*. Springer, New York, NY, USA, 2nd edition, 2015.

[53] F. Rossi, K. B. Venable, and T. Walsh. mCP Nets: Representing and reasoning with preferences of multiple agents. In *Proc. of AAAI*, pages 729–734, 2004.

[54] J. Rothe, editor. *Economics and Computation*. Springer, Berlin, Germany, 2016.

[55] R. A. Russell. Planning with preferences using maximum satisfiability. UCAM-CL-TR 822, Computing Laboratory, University of Cambridge, UK, 2012.

[56] M. Serna. Approximating linear programming is log-space complete for P. *Inform. Process. Lett.*, 37(4): 233–236, 1991.

[57] D. Shaparau, M. Pistore, and P. Traverso. Contingent planning with goal preferences. In *Proc. of AAAI*, pages 927–935, 2006.

[58] S. Sikdar, S. Adali, and L. Xia. Optimal decision making with CP-nets and PCP-nets. In *Proc. of EXPLORE*, pages 32–40, 2017.

[59] T. C. Son and E. Pontelli. Planning with preferences using logic programming. *Theor. Pract. Log. Prog.*, 6 (5):559–607, 2006.

[60] J. Stein, I. Nunes, and E. Cirilo. Preference-based feature model configuration with multiple stakeholders. In *Proc. of SPLC*, pages 132–141, 2014.

[61] L. J. Stockmeyer. The polynomial-time hierarchy. *Theor. Comput. Sci.*, 3(1):1–22, 1976.

[62] K. W. Wagner. Bounded query computations. In *Proc. of CoCo*, pages 260–277, 1988.

[63] K. W. Wagner. Bounded query classes. *SIAM J. Comput.*, 19(5):833–846, 1990.

[64] C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theor. Comput. Sci.*, 3(1):23–33, 1976.

[65] L. Xia, J. Lang, and M. Ying. Sequential voting rules and multiple elections paradoxes. In *Proc. of TARK*, pages 279–288, 2007.

[66] L. Xia, V. Conitzer, and J. Lang. Strategic sequential voting in multi-issue domains and multiple-election paradoxes. In *Proc. of EC*, pages 179–188, 2011.