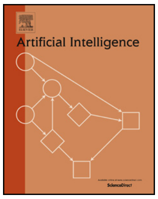
Artificial Intelligence 272 (2019) 101–142

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint



Complexity results for preference aggregation over (*m*)CP-nets: Pareto and majority voting ☆

Thomas Lukasiewicz [a], Enrico Malizia [b,*]

[a] *Department of Computer Science, University of Oxford, UK*
[b] *Department of Computer Science, University of Exeter, UK*

ARTICLE INFO

Article history:
Received 23 June 2018
Received in revised form 9 December 2018
Accepted 21 December 2018
Available online 31 January 2019

Keywords:
Preferences
Combinatorial preferences
Preference aggregation
CP-nets
mCP-nets
Voting
Pareto voting
Majority voting
Computational complexity
Polynomial hierarchy

ABSTRACT

Aggregating preferences over combinatorial domains has many applications in artificial intelligence (AI). Given the inherent exponential nature of preferences over combinatorial domains, compact representation languages are needed to represent them, and (*m*)CP-nets are among the most studied ones. Sequential and global voting are two different ways of aggregating preferences represented via CP-nets. In sequential voting, agents' preferences are aggregated feature-by-feature. For this reason, sequential voting may exhibit voting paradoxes, i.e., the possibility to select sub-optimal outcomes when preferences have specific feature dependencies. To avoid paradoxes in sequential voting, one has often assumed the (quite) restrictive constraint of $\mathcal{O}$-legality, which imposes a shared common topological order among all the agents' CP-nets. On the contrary, in global voting, CP-nets are considered as a whole during the preference aggregation process. For this reason, global voting is immune from the voting paradoxes of sequential voting, and hence there is no need to impose restrictions over the CP-nets' structure when preferences are aggregated via global voting. Sequential voting over $\mathcal{O}$-legal CP-nets received much attention, and $\mathcal{O}$-legality of CP-nets has often been required in other studies. On the other hand, global voting over non-$\mathcal{O}$-legal CP-nets has not carefully been analyzed, despite it was explicitly stated in the literature that a theoretical comparison between global and sequential voting was highly promising and a precise complexity analysis for global voting has been asked for multiple times. In quite a few works, only very partial results on the complexity of global voting over CP-nets have been given. In this paper, we start to fill this gap by carrying out a thorough computational complexity analysis of global voting tasks, for Pareto and majority voting, over not necessarily $\mathcal{O}$-legal acyclic binary polynomially connected (*m*)CP-nets. We show that all these problems belong to various levels of the polynomial hierarchy, and some of them are even in P or LOGSPACE. Our results are a notable achievement, given that the previously known upper bound for most of these problems was the complexity class EXPTIME. We provide various exact complexity results showing tight lower bounds and matching upper bounds for problems that (up to now) did not have any explicit non-obvious lower bound.

☆ Preliminary results of this paper have appeared in the Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16) [61].

* Corresponding author.

E-mail addresses: thomas.lukasiewicz@cs.ox.ac.uk (T. Lukasiewicz), e.malizia@exeter.ac.uk (E. Malizia).

https://doi.org/10.1016/j.artint.2018.12.010

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. [21]), because methods for representing and reasoning about preferences are very important in artificial intelligence (AI) applications, such as recommender systems [70], (group) product configuration [15,30,77], (group) planning [14,72,73,76], (group) preference-based constraint satisfaction [9,12,18], and (group) preference-based query answering/information retrieval [10,28,63,64].

In computer science, the study of preference aggregation has often been based on the solid ground of social choice theory, which is the branch of economics analyzing methods for collective decision making [5,6]. Having a well-founded theory and practice on how to properly and efficiently manage and aggregate preferences of real software agents, and hence support the growth and use of these technologies, has been one of the main drivers for investigating social choice theory from a computational perspective.

In social choice theory, the actual ways of representing agent preferences are rarely taken into consideration, also because the size of the sets of candidates considered is usually, although not always, relatively small. For this reason, most of the insights obtained in the computational social choice literature about the computational properties of preference aggregation functions (or voting procedures) have assumed that agent preferences over the set of candidates are extensively listed (see the survey by Brandt et al. [21] and references therein). Although this is perfectly reasonable when we reason about, e.g., (political) elections among a not too numerous set of human candidates, this is not feasible when the voting domain (i.e., the set of candidates) has a *combinatorial structure* [22,52,55]. By combinatorial structure, we mean that the set of candidates (or *outcomes*) is the Cartesian product of finite value domains for each of a set of *features* (also called *variables*, or *issues*, or *attributes*). The problem of aggregating agents' preferences over combinatorial domains (or *multi-issue* domains) is called a *combinatorial vote* [51,52].

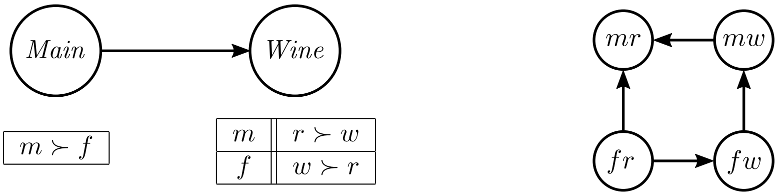
Interestingly, votes over domains exhibiting a combinatorial structure are rather common. For example, in 2012, on the day of the US presidential election, voters in California had also to vote for eleven referenda [55]. As another example, it may be the case that the inhabitants of a town have to make a joint decision about different related issues regarding their community, which could be whether and where to build new public facilities (such as a swimming pool or a library), or whether to levy new taxes. Note that these voting scenarios are often also called *multiple elections* or *multiple referenda* [19,22,54,55,83,84]. Other examples are group product configurations and group planning [55,77]. As for the latter, consider, e.g., a situation in which multiple autonomous agents have to agree upon a shared plan of actions to reach a goal that is preferred by the group as a whole, such as a group of autonomous robots coordinating during the exploration of a remote area/planet. Each robot has a specific task to accomplish, and the group as a whole coordinates to achieve a common goal. That is, the robots have their own specific preferences over a vast amount of variables/features emerging from the contingency of the situation to complete their individual tasks, however, their individual preferences have to be blended together. These examples show the great relevance of dealing with combinatorial votes, and hence the pressing necessity of finding ways to represent agent preferences over multi-issue domains and algorithms for aggregating them.

Combinatorial domains contain an exponential number of outcomes in the number of features, and hence compact representations for combinatorial preferences are needed [52,55] (see also Section 6.1 for more background). The graphical model of *CP-nets* [11] is among the most studied of these representations, as proven by a vast literature on them. In CP-nets, the vertices of a graph represent features, and an edge from vertex A to vertex B models the influence of the value of feature A on the choice of the value of feature B. Intuitively, this model captures preferences like "if the rest of the dinner is the same, with a fish dish (A's value), I prefer a white wine (B's value)", also called *conditional ceteris paribus preferences*; a more detailed example is given below.

Example 1.1. Assume that we want to model one's preferences for a dinner with a main dish and a wine [11]. In the CP-net in Fig. 1a, an edge from vertex *Main* to vertex *Wine* models that the value of feature *Main* influences the choice of the value of feature *Wine*. More precisely, m and f are the possible values of feature *Main*, and they denote "*m*eat" and "*f*ish", respectively, while r and w are the possible values of feature *Wine*, and they denote "*r*ed (wine)" and "*w*hite (wine)", respectively. The table associated with feature *Wine* specifies that when a meat dish is chosen, then a red wine is preferred to a white one, and when a fish main is chosen, then a white wine is preferred to a red wine. The table associated with feature *Main* indicates that a meat dish is preferred to a fish one. These tables are called *CP tables*. A CP-net like this one can represent the above conditional ceteris paribus preference "given that the rest of the dinner does not change, with a meat dish (*Main*'s value), I prefer a red wine (*Wine*'s value)".

Every CP-net has an associated *induced preference graph* [11], whose vertices are all the possible outcomes of the domain, and whose edges connect outcomes differing on only one value. More precisely, there is a directed edge from an outcome to another if the latter is preferred to the former according to the preferences encoded in the tables of the CP-net. Fig. 1b shows the induced preference graph of the CP-net in Fig. 1a, having as vertices all the possible combinations for the dinner, and there is, e.g., an edge from *mw* to *mr*, because the combination meat and red wine is preferred to the combination meat and white wine. The preferences encoded in a CP-net are the transitive closure of its induced preference graph.

Intuitively, an outcome α is preferred to an outcome β according to the preferences of a CP-net, if there is a directed path from β to α in the induced preference graph, in which case α is said to dominate β. ◁



(a) A CP-net modeling dinner preferences.

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

Fig. 1. A CP-net and its preference graph.

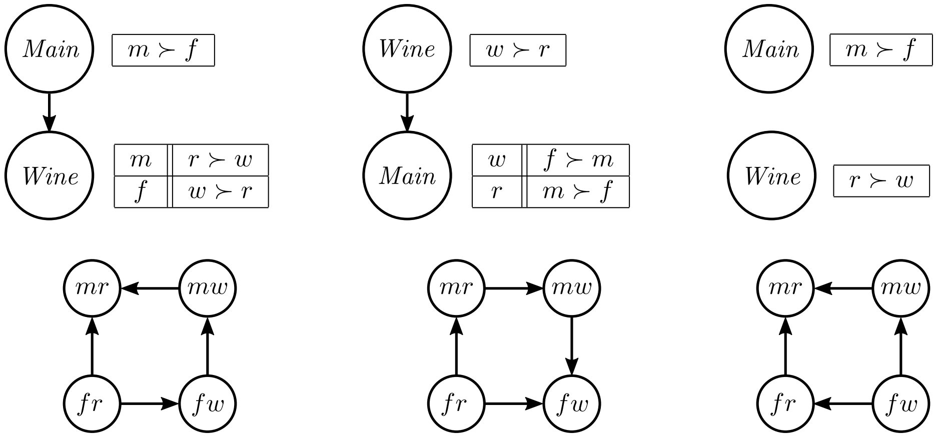


Fig. 2. Dinner preferences of Alice, Bob, and Chuck (in this order) modeled via CP-nets (above) and their induced preference graphs (below).

CP-nets are also used to model preferences of *groups* of individuals, obtaining a multi-agent model, called *m*CP-nets [71], which is a set, or *profile*, of CP-nets, one for each agent. The preference semantics of *m*CP-nets is defined via voting schemes: through its own individual CP-net, every agent votes whether an outcome is preferred to another. Various voting schemes were proposed for *m*CP-nets [59,71], and different voting schemes give rise to different dominance semantics for *m*CP-nets. In this paper, we consider *Pareto* and *majority voting* as they were defined by Rossi et al. [71]. In the voting schemes proposed for *m*CP-nets, the voting protocol adopted, i.e., the actual way in which votes are collected [24], is *global voting* [53,55]. In this protocol, the results of the voting procedure are computed by having as input the CP-nets as a whole (see Section 6.2 for related works on different voting protocols over CP-nets).

Example 1.2. Consider again the dinner scenario, and assume that there are three agents (Alice, Bob, and Chuck), expressing their preferences via CP-nets (see Fig. 2). In Pareto voting, an outcome α dominates an outcome β, if *all* agents prefers α to β. In majority voting, an outcome α dominates an outcome β, if the *majority* of agents prefers α to β.

The outcome *fr* is not *Pareto optimal*, because there is an outcome (namely *mr*), which is preferred to *fr* by *all* the agents. The outcome *mw*, instead, is Pareto optimal, because there is no outcome Pareto dominating *mw*. Hence, from a Pareto perspective, *mw* is better than *fr*. The outcome *mw*, however, is not *majority optimal*, because *mr* majority dominates *mw* (Alice and Chuck prefer *mr* to *mw*). On the other hand, *mr* is majority optimal, because there is no outcome majority dominating *mr*. Hence, from a majority perspective, *mr* is better than *mw*. Moreover, again according to the majority voting scheme, *mr* is a very good outcome, because *mr* is also *majority optimum*, which means that *mr* majority dominates all other outcomes. On the contrary, in this example, there is no Pareto optimum outcome, i.e., there is no outcome Pareto dominating all other outcomes. ◁

In the literature, a comparison between sequential voting (which is another voting protocol; see Section 6.2) and global voting over CP-nets was explicitly asked for and stated to be highly promising [53]. However, global voting over CP-nets has not been as thoroughly investigated as sequential voting. In fact, unlike CP-nets, which were extensively analyzed, a precise complexity analysis of *m*CP-nets has been missing for a long time, as explicitly mentioned several times in the literature [53,56–59,74]—since the dominance semantics for *m*CP-nets is global voting over CP-nets, in the following, we use them interchangeably. Furthermore, it was conjectured that the complexity of computing majority optimal and majority optimum outcomes in *m*CP-nets is harder than NP and co-NP [56,58].

Contributions The aim of this paper is to explore the complexity of *m*CP-nets (and hence of global voting over CP-nets). In particular, we focus on acyclic binary polynomially connected *m*CP-nets (see Section 2 for these notions) built with standard

	Problem	**Complexity**
Pareto	Pareto-Dominance	NP-complete
	Is-Pareto-Optimal	co-NP-complete
	Exists-Pareto-Optimal	$\Theta(1)^*$
	Is-Pareto-Optimum	in LOGSPACE
	Exists-Pareto-Optimum	in P
Majority	Majority-Dominance	NP-complete
	Is-Majority-Optimal	co-NP-complete
	Exists-Majority-Optimal	Σ_2^P-complete
	Is-Majority-Optimum	Π_2^P-complete
	Exists-Majority-Optimum	Π_2^P-hard, in D_2^P

Fig. 3. Summary of the complexity results obtained in this paper for global voting over CP-nets. Membership results above P are valid for any representation scheme whose dominance test is in NP (for example, polynomially connected classes of CP-nets). *A different proof is provided by Rossi et al. [71].

CP-nets, i.e., the constituent CP-nets of an mCP-net rank all the features, and they are not partial CP-nets (which instead were allowed in the original definition of mCP-nets [71]). Unlike what is often assumed in the literature, in this work, we do not restrict the profiles of CP-nets to be $\mathcal{O}$-legal (which means that there is a topological order common to all the CP-nets of the profile; see Section 6.3). We carry out a thorough complexity analysis for the (a) Pareto and (b) majority voting schemes, as defined by Rossi et al. [71], of deciding (1) dominance, (2) optimal and (3) optimum outcomes, and (4) the existence of optimal and (5) optimum outcomes. Deciding the dominance for a voting scheme s means deciding, given two outcomes, whether one dominates the other according to s. Deciding whether an outcome is optimal or optimum for a voting scheme s means deciding whether the outcome is not dominated or dominates all others, respectively, according to s. Deciding the existence of optimal and optimum outcomes is the natural extension of the previous problems.

A summary of the complexity results obtained in this paper is provided in Fig. 3. More precisely, deciding dominance and optimal outcomes is complete for NP and co-NP, respectively, for both Pareto and majority voting, while deciding the existence of optimal outcomes can be done in constant time for Pareto voting and is complete for Σ_2^P for majority voting. Furthermore, deciding optimum outcomes and their existence is in LOGSPACE and P for Pareto voting, and complete for Π_2^P and between Π_2^P and D_2^P for majority voting, respectively.

It thus turns out that Pareto voting is the easiest voting scheme to evaluate among the two analyzed here. More precisely, both Pareto and majority dominance are NP-complete, however, only the complexity of majority dominance carries over to deciding optimal and optimum outcomes and their existence, and causes a substantial increase of their complexity, e.g., deciding the existence of majority optimal and optimum outcomes is hard for Σ_2^P and Π_2^P, respectively. This is due to the fact that majority voting is structurally more complex than Pareto voting. Intuitively, Pareto voting is based on unanimity, hence, to disprove Pareto dominance between two outcomes, it suffices to find *one* agent that does not agree with the dominance relationship. This particular structure of Pareto voting makes the other tasks not more difficult than the dominance test or even tractable. Our results hence prove the conjecture posed by Li et al. [56,58] about majority voting tasks over (m)CP-nets being harder than NP and co-NP.

We show completeness results for most cases, and we provide tight lower bounds for problems that (up to date) did not have any explicit lower bound transcending the obvious hardness due to the dominance test over the underlying CP-nets. Many of our results are intractability results, where the problems are put at various levels of the polynomial hierarchy. However, although intractability is usually "bad" news, these results are quite interesting, as for most of these tasks, only EXPTIME upper bounds were known in the literature to date [71]. Even more interestingly, some of these problems are actually tractable, as they are in P or even LOGSPACE, which is a huge leap from EXPTIME.

Our hardness results are given for binary acyclic polynomially connected (m)CP-nets. This means that our hardness results extend to classes of (m)CP-nets encompassing the CP-nets considered here, and in particular also to general mCP-nets with partial CP-nets or multi-valued features. More generally, the hardness results proven here extend to any representation scheme as "expressive and succinct" as the class of CP-nets used in the proofs (see Section 2.5). Moreover, the membership results above P that we prove here extend to any "NP-representation" scheme (see Section 2.5). Our hardness results on the existence of optimal and optimum outcomes also provide lower bounds for the computational problems. Indeed, actually computing optimal or optimum outcomes cannot be easier than the bounds shown here, because otherwise it would be possible to decide their existence more efficiently.

Characterizing the precise computational complexity of these problems, apart from providing the analysis requested in the literature [53], means understanding their sources of complexity. This information allows us to identify tractable instances by limiting some sources of intractability, and to design and analyze algorithms. For instance, note that any Σ_2^P-complete problem exhibits two orthogonal sources of intractability. To give an idea of a practical consequence of this fact, assume that we would like to solve such a problem on a standard (deterministic) machine. Then, the theory tells us that we cannot design any flat-backtracking algorithm for our problem, unless P = NP. Indeed, any algorithm with a search-space tree having a polynomial number of levels (and such that moving along the tree edges does not take exponential time) should solve a nested co-NP-hard problem to check whether a leaf node is a solution or not. That is, checking leaf-feasibility is an intractable problem as well, and hence it requires a nested call to a further backtracking procedure (or to another kind of procedure with an exponential-time worst case).

Organization of the paper The rest of this paper is organized as follows. Section 2 provides some preliminaries. In Section 3, we prove some basic complexity results for CP-nets. Sections 4 and 5 analyze the complexity of Pareto and majority voting, respectively: first, we analyze the complexity of dominance testing; then, we study the complexity of deciding whether an outcome is optimal and whether there exists an optimal outcome; and we conclude by dealing with the complexity of deciding whether an outcome is optimum and whether there exists an optimum outcome. In Section 6, we discuss related works. Section 7 summarizes the main results and gives an outlook on future research. For several results, we give only proof sketches in the body of the paper, while detailed proofs are provided in Appendix A.

2. Preliminaries

In this section, we give some preliminaries, briefly recalling from the literature preference relations and aggregation, 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. We also define a formal framework for preference representation schemes, because our membership results will be given for generic representations whose dominance test is in NP.

2.1. Preference relations and aggregation

Before dwelling upon the details of CP-nets, which is the specific preference representation analyzed in this paper, we now give an introductory overview of the general concepts of preferences and their aggregation.

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). Usually, given two outcomes α and β, their preference relationship stated in $\mathcal{R}$ is denoted by $\alpha \succ_{\mathcal{R}} \beta$, instead of $\langle\alpha,\beta\rangle \in \mathcal{R}$, which means that, in $\mathcal{R}$, α is strictly preferred to β, or α *dominates* β. On the other hand, $\alpha \not\succ_{\mathcal{R}} \beta$ means that $\langle\alpha,\beta\rangle \notin \mathcal{R}$, and $\alpha \bowtie_{\mathcal{R}} \beta$ means that $\langle\alpha,\beta\rangle \notin \mathcal{R}$ *and* $\langle\beta,\alpha\rangle \notin \mathcal{R}$, i.e., α and β are *incomparable* in $\mathcal{R}$. Observe that in a preference ranking, it cannot be the case that two outcomes are incomparable. Given a preference relation $\mathcal{R}$, an outcome α is *optimal* in $\mathcal{R}$ if there is no outcome β such that $\beta \succ_{\mathcal{R}} \alpha$. We say that α is *optimum* in $\mathcal{R}$, if for all outcomes β such that $\beta \neq \alpha$, it holds that $\alpha \succ_{\mathcal{R}} \beta$. If there is an optimum outcome in $\mathcal{R}$, then it is unique. For notational convenience, if the preference relation $\mathcal{R}$ is clear from the context, we do not explicitly mention $\mathcal{R}$ as a subscript in the notations above. In the following, if not stated otherwise, when we speak of preferences structures, we mean preference relations.

In preference aggregation, we deal with preferences of multiple agents. A preference *profile* $\mathscr{P} = \langle\mathcal{R}_1,\dots,\mathcal{R}_m\rangle$ is a set[1] of m preference relations. We assume that all the preferences $\mathcal{R}_i$ of $\mathscr{P}$ are defined over the same set of outcomes, i.e., the agents express their preferences over the same set of candidates. In this paper, we focus on voting procedures based on comparisons of pairs of outcomes (see, e.g., the work by Baumeister and Rothe [7] for a classification of different kinds of preference aggregation procedures). For this reason, we need to define the following sets of agents. For a profile $\mathscr{P} = \langle\mathcal{R}_1,\dots,\mathcal{R}_m\rangle$, we denote by $S^{\succ}_{\mathscr{P}}(\alpha,\beta) = \{i \mid \alpha \succ_{\mathcal{R}_i} \beta\}$, $S^{\prec}_{\mathscr{P}}(\alpha,\beta) = \{i \mid \alpha \prec_{\mathcal{R}_i} \beta\}$, and $S^{\bowtie}_{\mathscr{P}}(\alpha,\beta) = \{i \mid \alpha \bowtie_{\mathcal{R}_i} \beta\}$, the sets of agents preferring α to β, preferring β to α, and for which α and β are incomparable, respectively.

The voting schemes considered in this paper are Pareto and majority. The definition of their dominance semantics over preference profiles, reported below, is a generalization of the respective definition over mCP-nets given by Rossi et al. [71].

Pareto: An outcome β *Pareto dominates* an outcome α, denoted $\beta \succ^{p}_{\mathscr{P}} \alpha$, if *all* agents prefer β to α, i.e., $|S^{\succ}_{\mathscr{P}}(\beta,\alpha)| = m$.

Majority An outcome β *majority dominates* an outcome α, denoted $\beta \succ^{maj}_{\mathscr{P}} \alpha$, if the *majority* of the agents prefer β to α, i.e., $|S^{\succ}_{\mathscr{P}}(\beta,\alpha)| > |S^{\prec}_{\mathscr{P}}(\beta,\alpha)| + |S^{\bowtie}_{\mathscr{P}}(\beta,\alpha)|$.

For a preference profile $\mathscr{P}$ and a voting scheme s, if outcome β does not s dominate outcome α, we denote this by $\beta \not\succ^{s}_{\mathscr{P}} \alpha$. An outcome α is *s optimal* in $\mathscr{P}$, if for all $\beta \neq \alpha$, it holds that $\beta \not\succ^{s}_{\mathscr{P}} \alpha$, while α is *s optimum* in $\mathscr{P}$, if for all $\beta \neq \alpha$, it holds that $\alpha \succ^{s}_{\mathscr{P}} \beta$. Note that optimum outcomes, if they exist, are unique.

2.2. CP-nets

We now focus on CP-nets, which is the preference representation that we will more closely investigate in this work. As mentioned in the introduction, the set of outcomes of a preference relation is often defined as the Cartesian product of finite value domains for each of a set of features. Conditional preference nets (CP-nets) [11] are a formalism to encode

[1] More formally, a profile is a multi-set, because different agents may have the very same preference relation.

conditional ceteris paribus preferences over such combinatorial domains. The distinctive element of CP-nets is that a directed graph, whose vertices represent the features of a combinatorial domain, is used to intuitively model the conditional part of conditional ceteris paribus preference statements. Below, we recall the syntax, semantics, and some properties of CP-nets; see Section 6.1 for more on conditional ceteris paribus preferences and preference representations in general.

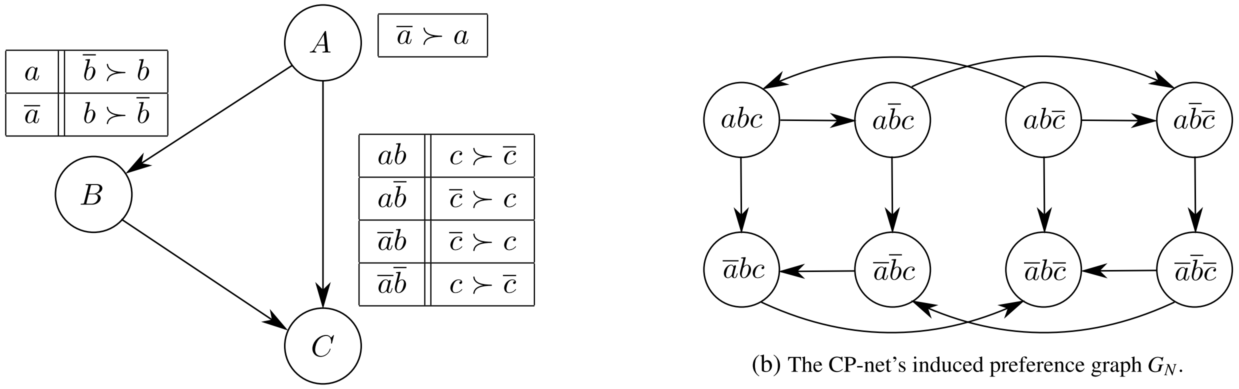
Syntax of CP-nets A *CP-net* N is a triple $\langle \mathcal{G}_N, Dom_N, (CPT^F_N)_{F\in\mathcal{F}_N}\rangle$, where $\mathcal{G}_N = \langle \mathcal{F}_N, \mathcal{E}_N\rangle$ is a directed graph whose vertices $\mathcal{F}_N$ represent the *features* of a combinatorial domain, and Dom_N and $(CPT^F_N)_{F\in\mathcal{F}_N}$ are a function and a family of functions, respectively. The function Dom_N associates a *(value) domain* $Dom_N(F)$ with every feature F, while the functions CPT^F_N are the *CP tables* for every feature F, which are 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, we assume features to be *binary*, i.e., the domain of each feature F contains exactly two values, usually denoted $\overline{f}$ and f, and called the *overlined* and the *non-overlined* value (of F), respectively. For a set of features $\mathcal{S}$, $Dom_N(\mathcal{S}) = \times_{F\in\mathcal{S}} Dom_N(F)$ denotes the Cartesian product of the domains of the features in $\mathcal{S}$. Thus, an *outcome* is an element of $\mathcal{O}_N = Dom_N(\mathcal{F}_N)$. 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 over $\mathcal{F}$. 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 $\alpha[\mathcal{F}] \neq \beta[\mathcal{F}]$, when this is not the case, 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 $Par_N(F) = \{G \in \mathcal{F}_N \mid \langle G, F\rangle \in \mathcal{E}_N\}$ the set of all features in $\mathcal{G}_N$ from which there is an edge to F. We call $Par_N(F)$ the set of the *parents* of F (in N). We denote by $Ord_N(F)$ the set of all the (strict) preference rankings over the elements of $Dom_N(F)$. Each function $CPT^F_N\colon Dom_N(Par_N(F)) \to Ord_N(F)$ maps every element of $Dom_N(Par_N(F))$ to a (strict) preference ranking over the domain of F. If $Par_N(F) = \varnothing$, then CPT^F_N is a single (strict) preference ranking over $Dom_N(F)$. Note that indifferences between feature values are not admitted in (classical) CP-nets. Each function CPT^F_N 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^F_N, and the element in the second column is the associated (strict) preference ranking over $Dom_N(F)$. Since CPT^F_N is total, in the table representing the function, there is a row for any combination of values of the parent features, i.e., for a feature F, there are $2^{|Dom_N(Par_N(F))|}$ rows in the 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 the notation on which generalized propositional CP-nets [33] are based on, it is used here only for notational convenience. In this paper, we always assume that CP tables are explicitly represented in the input instances. In the CP tables, $\overline{f} \succ f$ denotes $\overline{f}$ being preferred to f. We denote by $\|N\|$ the size of CP-net N, i.e., the space in terms of bits required to represent the whole net N (which includes features, edges, feature domains, and CP tables).

Semantics of CP-nets The preference semantics of CP-nets can be defined in several different but equivalent ways [11]. A first definition has a model-theoretic flavor [11, Definitions 2 and 3]. Intuitively, a preference ranking $\mathcal{R}$ violates a CP-net N, if there are two outcomes α and β that according to the CP tables of N should be ranked $\beta \succ \alpha$, but they are not ranked in such a way in $\mathcal{R}$ (i.e., $\alpha \succ_{\mathcal{R}} \beta$, since $\mathcal{R}$ is total). Formally, a preference ranking $\mathcal{R}$ *violates* a CP-net N, if there are two distinct outcomes α, β and a feature F such that $\alpha[\mathcal{F}_N \setminus \{F\}] = \beta[\mathcal{F}_N \setminus \{F\}]$ (i.e., α and β differ only on the value of F), $\beta[F] \succ \alpha[F]$ in the order $CPT^F_N(\alpha[Par_N(F)])$, and $\alpha \succ_{\mathcal{R}} \beta$. A preference ranking $\mathcal{R}$ *satisfies* a CP-net N, if $\mathcal{R}$ does not violate N. Given two outcomes α and β, a CP-net N *entails* the preference $\alpha \succ \beta$, denoted $N \models \alpha \succ \beta$, if $\alpha \succ_{\mathcal{R}} \beta$ for every preference ranking $\mathcal{R}$ over $\mathcal{O}_N$ that satisfies N. The preference semantics of CP-nets can be equivalently defined via the concept of improving (or alternatively worsening) flip [11, 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 $CPT^F_N(\alpha[Par_N(F)])$. Given two outcomes α and β differing only on the value of a feature F, there is an improving flip from α to β, denoted $\alpha \to_{N[F]} \beta$, 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 $\alpha \to_N \beta$; 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 pair $G_N = \langle V_N, E_N\rangle$, where the nodes V_N are all the possible outcomes of N, and, given two outcomes $\alpha, \beta \in V_N$, the directed edge from α to β belongs to E_N if and only if $\alpha \to_N \beta$.

It can be shown that, for a CP-net N and two outcomes α and β, $N \models \beta \succ \alpha$ if and only if there is a sequence of improving flips from α to β [11, Theorems 7 and 8]. Therefore, 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 N entails $\beta \succ \alpha$, or, equivalently, if there is an improving flipping sequence from α to 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), denoted $\alpha \bowtie_N \beta$ (which is equivalent to the existence of preference rankings $\mathcal{R}_1$ and $\mathcal{R}_2$ that both satisfy N such that $\alpha \succ_{\mathcal{R}_1} \beta$ and $\beta \succ_{\mathcal{R}_2} \alpha$).

Note here that, since there are no indifferences between features values in (classical) CP-nets, for any two outcomes α and β, either one dominates the other, or they are incomparable.



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

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

Fig. 4. A CP-net and its induced preference graph.

Example 2.1. Consider the CP-net N shown in Fig. 4. For the outcomes $\alpha = abc$ and $\beta = \bar{a}bc$, it holds that $\beta \succ_N \alpha$, because $\alpha \rightarrow_{N[A]} \beta$. For the outcomes $\alpha = abc$ and $\beta = ab\bar{c}$, it holds that $\beta \not\succ_N \alpha$, because there is no path from α to β in G_N. However, $\alpha \succ_N \beta$, because $\beta \rightarrow_{N[C]} \alpha$, and hence it is not the case that $\alpha \bowtie_N \beta$. Consider now the outcomes $\alpha = abc$ and $\beta = \bar{a}\bar{b}\bar{c}$. Then, $\beta \succ_N \alpha$ by the improving flipping sequence $abc \rightarrow a\bar{b}c \rightarrow a\bar{b}\bar{c} \rightarrow \bar{a}\bar{b}\bar{c}$. $\triangleleft$

Properties of CP-nets A CP-net is *binary*, if all its features are binary. The *indegree* of a CP-net N is the maximum number of edges entering in a node of the graph of 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 $\mathscr{F}$ of CP-nets is *polynomially connected*, if there exists a polynomial p such that, for any CP-net $N \in \mathscr{F}$ and for any two features G and F of N, there are at most $p(\|N\|)$ distinct paths from G to F in $\mathcal{G}_N$. A CP-net N is *acyclic*, if $\mathcal{G}_N$ is acyclic. It is well known that acyclic CP-nets N always have a preference ranking satisfying 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 optimum outcome o_N dominating all other outcomes (and not dominated by any other), which can be computed in polynomial time [11].

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 binary acyclic CP-nets [11]. However, it is an open problem whether dominance testing is in NP over non-polynomially-connected classes of binary acyclic CP-nets. Allen's conjecture [2,4] states that in general binary acyclic 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. Also, the complexity of dominance testing for non-binary CP-nets is currently still open. Whereas dominance testing for the class of acyclic binary singly connected CP-nets whose indegree is at most six is NP-hard [11]—we improve this result in Section 3, requiring only indegree three. Dominance testing can be carried out in polynomial time on acyclic binary CP-nets whose graph is a tree or a polytree [11]. For an extension of CP-nets, called generalized CP-nets, dominance testing is PSPACE-complete [33].

In the rest of this paper, we consider only binary acyclic (and often polynomially connected) classes of CP-nets. When the CP-net N is clear from the context, we often omit the subscript "N" from the notations introduced above.

2.3. mCP-nets

In this section, we focus on mCP-nets [71], which 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 of Rossi et al. [71] for definitions), i.e., all features are ranked in all the individual CP-nets of an mCP-net.

As underlined by Rossi et al. [71], 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, an mCP-net $\mathcal{M} = \langle N_1, \dots, N_m \rangle$ consists of m CP-nets $N_1, \dots, N_m$, 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 in 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); see Section 6 for more on $\mathcal{O}$-legality.

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 its own individual CP-net, each agent votes whether an outcome dominates another, and hence different ways of collecting votes (i.e., different voting schemes) give rise to different group dominance semantics for an mCP-net. Let $\mathcal{M} = \langle N_1, \dots, N_m \rangle$ be an mCP-net, and let α and β be two outcomes. With a notation similar to the one defined above, $S^{\succ}_{\mathcal{M}}(\alpha, \beta) = \{i \mid \alpha \succ_{N_i} \beta\}$, $S^{\prec}_{\mathcal{M}}(\alpha, \beta) = \{i \mid \alpha \prec_{N_i} \beta\}$, 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.

Rossi et al. [71] and Li et al. [59] proposed and analyzed various voting schemes to define multi-agent dominance semantics for mCP-nets. In this paper, we focus on two of them, namely, Pareto and majority voting, whose dominance semantics definitions are the natural specializations to mCP-nets of Pareto and majority dominance semantics defined above.

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

2.4. Computational complexity

We now give some notions from computational complexity theory, which will be required for the complexity analysis carried out in this paper. First, we briefly recall the complexity classes that we will encounter in this paper (along with some closely related ones), and then we recall the notion of polynomial-time reductions among decision problems, and some decision problems that are hard for some of these complexity classes. We assume that the reader has some elementary 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 [47] and Papadimitriou [67].

Complexity classes The class P is the set of all decision problems that can be solved by a deterministic Turing machine in polynomial time with respect to the input size, i.e., with respect to the length of the string that encodes the input instance. For a given input string s, its size is usually denoted by $\|s\|$. The class of decision problems that can be solved by nondeterministic Turing machines in polynomial time is denoted by NP. They enjoy a remarkable property: any "yes"-instance s has a *certificate* for being a "yes"-instance, which has polynomial length and can be checked in deterministic polynomial time (in $\|s\|$). For example, deciding whether a Boolean formula $\phi(X)$ over the Boolean variables $X = \{x_1, \dots, x_n\}$ is satisfiable, i.e., whether there exists some truth assignment to these variables making ϕ true, is a well-known problem in NP; in fact, any satisfying truth assignment for ϕ is a certificate that ϕ is a "yes"-instance, i.e., that ϕ is satisfiable.

For a complexity class C, we denote by co-C the complementary class to C, i.e., the class containing the complementary languages of those in C. For example, the problem of deciding whether a Boolean formula ϕ is *not* satisfiable is in co-NP. The class P is contained in both NP and co-NP, i.e., $\mathrm{P} \subseteq \mathrm{NP} \cap \text{co-NP}$.

By LOGSPACE, we denote the set of decision problems that can be solved by deterministic Turing machines in logarithmic space. For such machines, it is assumed that the input tape is read-only, and that these machines have a read/write tape, called work tape, for intermediate computations. The logarithmic space bound is given on the space available on the work tape. The class LOGSPACE is contained in P.

The class $\mathrm{D^P}$, defined originally by Papadimitriou and Yannakakis [68], is the class of problems that are a "conjunction" of two problems, one from NP and one from co-NP, i.e., $\mathrm{D^P} = \{L \mid L = L' \cap L'', L' \in \mathrm{NP}, L'' \in \text{co-NP}\}$. The class co-$\mathrm{D^P}$ is the class of problems whose complements are in $\mathrm{D^P}$, equivalently, it can be defined as the class of problems that are a "disjunction" of two problems, one from NP and one from co-NP, i.e., co-$\mathrm{D^P} = \{L \mid L = L' \cup L'', L' \in \mathrm{NP}, L'' \in \text{co-NP}\}$.

The classes Σ^{P}_k, Π^{P}_k, and Δ^{P}_k, for all $k \geq 0$, constituting the *polynomial hierarchy* (PH) [78], are defined as follows: $\Sigma^{\mathrm{P}}_0 = \Pi^{\mathrm{P}}_0 = \Delta^{\mathrm{P}}_0 = \mathrm{P}$, and, for all $k \geq 1$, $\Sigma^{\mathrm{P}}_k = \mathrm{NP}^{\Sigma^{\mathrm{P}}_{k-1}}$, $\Delta^{\mathrm{P}}_k = \mathrm{P}^{\Sigma^{\mathrm{P}}_{k-1}}$, and $\Pi^{\mathrm{P}}_k = \text{co-}\Sigma^{\mathrm{P}}_k$. Here, Σ^{P}_k (resp., Δ^{P}_k) is the set of decision problems solvable by nondeterministic (resp., deterministic) polynomial-time Turing machines with an oracle to recognize, at unit cost, a language in $\Sigma^{\mathrm{P}}_{k-1}$. Note that $\Sigma^{\mathrm{P}}_1 = \mathrm{NP}^{\Sigma^{\mathrm{P}}_0} = \mathrm{NP}^{\mathrm{P}} = \mathrm{NP}$, $\Pi^{\mathrm{P}}_1 = \text{co-}\Sigma^{\mathrm{P}}_0 = \text{co-NP}$, and $\Delta^{\mathrm{P}}_1 = \mathrm{P}^{\Sigma^{\mathrm{P}}_0} = \mathrm{P}^{\mathrm{P}} = \mathrm{P}$. Sometimes a bound is imposed on the number of calls that are allowed to be issued to the oracle. For example, $\Theta^{\mathrm{P}}_k = \mathrm{P}^{\Sigma^{\mathrm{P}}_{k-1}[O(\log n)]}$ denotes the set of decision problems solvable by a deterministic polynomial-time Turing machine that is allowed to query a $\Sigma^{\mathrm{P}}_{k-1}$ oracle at most logarithmically many times (in the size of the input). By definition, $\Theta^{\mathrm{P}}_k \subseteq \Delta^{\mathrm{P}}_k$.[2]

The classes $\mathrm{D^P}$ and co-$\mathrm{D^P}$ can be generalized to the classes $\mathrm{D}^{\mathrm{P}}_k = \{L \mid L = L' \cap L'', L' \in \Sigma^{\mathrm{P}}_k, L'' \in \Pi^{\mathrm{P}}_k\}$ and co-$\mathrm{D}^{\mathrm{P}}_k = \{L \mid L = L' \cup L'', L' \in \Sigma^{\mathrm{P}}_k, L'' \in \Pi^{\mathrm{P}}_k\}$, respectively, for $k \geq 1$, that are the conjunction and the disjunction, respectively, of Σ^{P}_k and Π^{P}_k; in particular, $\mathrm{D}^{\mathrm{P}}_1 = \mathrm{D^P}$. Note also that $\mathrm{D}^{\mathrm{P}}_k \subseteq \Theta^{\mathrm{P}}_{k+1}$.

[2] For the complexity class Θ^{P}_k, an interesting characterization has recently been provided: Θ^{P}_k is the class of languages involving the counting and comparison of the number of "yes"-instances in two sets containing instances of $\Sigma^{\mathrm{P}}_{k-1}$ or Π^{P}_{k-1} languages [62]. This is quite useful for reductions in voting settings where votes have to be counted and compared [61].

$$\begin{tikzcd}
 & & \text{co-NP} & & \Pi_k^P \arrow[r] \arrow[dr] & \text{co-D}_k^P \arrow[dr] & & & \Pi_{k+1}^P & \cdots \arrow[dr] & & \\
\text{LOGSPACE} \arrow[r] & \text{P} \arrow[ur] \arrow[dr] & & \cdots & & & \Theta_{k+1}^P \arrow[r] & \Delta_{k+1}^P \arrow[ur] \arrow[dr] & & & \text{PSPACE} \arrow[r] & \text{EXPTIME} \\
 & & \text{NP} & & \Sigma_k^P \arrow[r] \arrow[ur] & \text{D}_k^P \arrow[ur] & & & \Sigma_{k+1}^P & \arrow[ur] & &
\end{tikzcd}$$

Fig. 5. Inclusion relationships among complexity classes: an arrow from class $\mathcal{C}'$ to class $\mathcal{C}''$ means that $C' \subseteq C''$.

Given their definitions, for all $k \geq 1$, the relationships among the mentioned classes are as follows (see Fig. 5 for an illustration): $(\Sigma_k^{\mathrm{P}} \cup \Pi_k^{\mathrm{P}}) \subseteq \mathrm{D}_k^{\mathrm{P}}$, $\text{co-D}_k^{\mathrm{P}} \subseteq \Theta_{k+1}^{\mathrm{P}} \subseteq \Delta_{k+1}^{\mathrm{P}} \subseteq (\Sigma_{k+1}^{\mathrm{P}} \cap \Pi_{k+1}^{\mathrm{P}})$ (see, e.g., the works by Wagner [79,80]).

PSPACE is the class of decision problems that can be solved by deterministic Turing machines in polynomial space. Similarly to LOGSPACE machines, PSPACE machines have a read-only input tape and a read/write work tape. The polynomial space bound is issued over the space used on the work tape. The class PSPACE contains PH.

The class EXPTIME includes all the decision problems that can be solved by deterministic Turing machines in exponential time. The class EXPTIME contains PSPACE.

A *transducer* is a (possibly non-deterministic) Turing machine T on the alphabet Σ with a read-only input tape, a read/write work tape, and a write-only output tape. For any string $x \in \Sigma^*$, we say that T accepts x, if T has an accepting computation path on x. For each $x \in \Sigma^*$ accepted by T, we denote by $T(x)$ the set of all strings that are written by T on the output tape in its accepting computation paths on the input string x. Thus, every transducer is associated with some multi-valued function f (we say that T *computes* f) such that, for each $x \in \Sigma^*$, $f(x) = T(x)$, if x is accepted by T; otherwise, f is undefined at x (i.e., $x \notin dom(f)$).

A function f is computable in (deterministic) polynomial time, if there is a deterministic transducer computing f in polynomial time. We denote by FP the class of all functions computable in deterministic polynomial time. A function f is computable in (deterministic) logarithmic space, if there is a deterministic transducer computing f and using no more than logarithmic space on its work tape.

Reductions and hard problems A decision problem L_1 is *(Karp) reducible*, or *many-one reducible*, to a decision problem L_2, denoted $L_1 \leq L_2$, if there is a computable function h, called *(Karp/many-one) reduction*, such that, for every string s, $h(s)$ is defined, and s is a "yes"-instance of L_1 if and only if $h(s)$ is a "yes"-instance of L_2. A decision problem L_1 is *polynomially (Karp/many-one) reducible* to a decision problem L_2, denoted $L_1 \leq_p L_2$, if there is a polynomial-time (Karp/many-one) reduction from L_1 to L_2. In this paper, we consider only Karp/many-one reductions.

To prove hardness for a problem in a complexity class, we show reductions from various problems known to be complete for the complexity classes that they belong to. We next define such problems, so that we can later refer to them by name.

Deciding the satisfiability of Boolean formulas, denoted Sat, is the prototypical NP-complete problem, which remains NP-hard even if only 3CNF formulas are considered [32,48], i.e., Boolean formulas in conjunctive normal form with three literals per clause. The complementary problem Unsat of deciding whether a given Boolean formula is *not* satisfiable is co-NP-complete. It remains co-NP-hard even if only 3CNF formulas are considered, and it is the equivalent to the problem Taut of deciding whether a 3DNF formula is a tautology. A 3DNF formula is a Boolean formula in disjunctive normal form with three literals per term. CNF and DNF formulas are actually linked, and formulas in one form can be transformed into the other form via De Morgan's laws, and in some cases via a process called dualization [34,35]. These transformations are useful when we prefer formulas in a specific form rather than the other (see the discussion below). Transforming formulas can however yield, in some circumstances, an exponential growth in the size of the transformed formulas.

The prototypical Σ_k^{P}- and Π_k^{P}-complete $\mathrm{QBF}_{Q_1,k}$ problems are defined as follows: given a quantified Boolean formula (QBF) $\Phi = (Q_1 X_1)(Q_2 X_2)\dots(Q_k X_k)\phi(X_1, X_2, \dots, X_k)$, where

- $Q_1, Q_2, \dots, Q_k$ is a sequence of k alternating quantifiers $Q_i \in \{\exists, \forall\}$, and
- $\phi(X_1, X_2, \dots, X_k)$ is a (non-quantified) Boolean formula over k disjoint sets $X_1, X_2, \dots, X_k$ of Boolean variables,

decide whether the formula Φ is valid. The problem $\mathrm{QBF}_{\exists,k}$ is Σ_k^{P}-complete [78,81], while $\mathrm{QBF}_{\forall,k}$ is Π_k^{P}-complete [78,81]. These problems remain hard for their respective classes even if $\phi(X_1, X_2, \dots, X_k)$ is in 3CNF, when $Q_k = \exists$, and if $\phi(X_1, X_2, \dots, X_k)$ is in 3DNF, when $Q_k = \forall$ [78,81]. We denote by $\mathrm{QBF}_{k,\exists}^{CNF}$ (resp., $\mathrm{QBF}_{k,\forall}^{DNF}$)[3] the problem of deciding the validity of formulas $\Phi = (Q_1 X_1)\dots(Q_k X_k)\phi(X_1, \dots, X_k)$, where Q_k is $\exists$ (resp., $\forall$), and $\phi(X_1, \dots, X_k)$ is in 3CNF (resp., 3DNF). For odd k, $\mathrm{QBF}_{k,\exists}^{CNF}$ (resp., $\mathrm{QBF}_{k,\forall}^{DNF}$) is complete for Σ_k^{P} (resp., Π_k^{P}), while, for even k, $\mathrm{QBF}_{k,\exists}^{CNF}$ (resp., $\mathrm{QBF}_{k,\forall}^{DNF}$) is complete for Π_k^{P} (resp., Σ_k^{P}). Observe that $\mathrm{QBF}_{1,\exists}^{CNF}$ (resp., $\mathrm{QBF}_{1,\forall}^{DNF}$) is equivalent to Sat (resp., Taut).

Sometimes, it is preferable that in QBF formulas the non-quantified formula is CNF rather than DNF, or vice-versa. For example, to show Σ_2^{P}-hardness it might be the case that we would prefer to start our reduction from formulas $\Phi = (\exists X)(\forall Y)\phi(X, Y)$ with $\phi(X, Y)$ being in CNF, rather than in DNF, as required by $\mathrm{QBF}_{2,\forall}^{DNF}$. To achieve this, we can exploit De Morgan's laws. Indeed, we have that $\Phi = (Q_1 X_1)\dots(Q_k X_k)\phi(X_1, \dots, X_k)$ is logically equivalent to $\Phi' = (Q_1 X_1)\dots$

[3] Note the difference in the subscripts of the notations $\mathrm{QBF}_{Q_1,k}$ and $\mathrm{QBF}_{k,Q_k}^{CNF}$ (resp., $\mathrm{QBF}_{k,Q_k}^{DNF}$). In the former notation, Q_1 is the *first* quantifier of the sequence, and, for notational convenience, we place "Q_1" before "k" in the subscript. On the other hand, in the latter notation, Q_k is the *last* quantifier of the sequence, and, for notational convenience, we place "Q_k" after "k" in the subscript.

$(Q_k X_k)\neg\phi'(X_1,\ldots,X_k)$, where $\phi'(X_1,\ldots,X_k)=\neg\phi(X_1,\ldots,X_k)$. We thus extend the notation above. We denote by $\mathrm{QBF}^{DNF}_{k,\exists,\neg}$ (resp., $\mathrm{QBF}^{CNF}_{k,\forall,\neg}$) the problem of deciding the validity of formulas $\Phi=(Q_1X_1)\ldots(Q_kX_k)\neg\phi(X_1,\ldots,X_k)$, where Q_k is $\exists$ (resp., $\forall$), and $\phi(X_1,\ldots,X_k)$ is in 3DNF (resp., 3CNF). For odd k, $\mathrm{QBF}^{DNF}_{k,\exists,\neg}$ (resp., $\mathrm{QBF}^{CNF}_{k,\forall,\neg}$) is complete for Σ^{P}_k (resp., Π^{P}_k), while, for even k, $\mathrm{QBF}^{DNF}_{k,\exists,\neg}$ (resp., $\mathrm{QBF}^{CNF}_{k,\forall,\neg}$) is complete for Π^{P}_k (resp., Σ^{P}_k).

2.5. A framework for preference representation schemes

In this paper, most of the membership results that we will show hold for generic preference representation schemes. For this reason, we now introduce the general framework of representation schemes that we will refer to. Inspired by the concept of compact representations introduced by Chevaleyre et al. [22], Greco et al. [40,42], and Lang and Xia [55], we define preference representation schemes $\mathscr{S}$ as suitable encodings for a class of preference relations, denoted $\mathcal{C}(\mathscr{S})$. Formally, a *preference representation scheme* $\mathscr{S}$ defines a computable representation function $\xi^{\mathscr{S}}(\cdot)$ and a computable Boolean function $\mathit{Pref}^{\mathscr{S}}(\cdot,\cdot,\cdot)$ such that, for any relation $\mathcal{R}\in\mathcal{C}(\mathscr{S})$, $\xi^{\mathscr{S}}(\mathcal{R})$ is the encoding of $\mathcal{R}$ according to $\mathscr{S}$, and $\mathit{Pref}^{\mathscr{S}}(\xi^{\mathscr{S}}(\mathcal{R}),\alpha,\beta)$ evaluates to 1, if $\langle\alpha,\beta\rangle\in\mathcal{R}$ (i.e., if $\alpha\succ_{\mathcal{R}}\beta$), and to 0, otherwise. By $\|\xi^{\mathscr{S}}(\mathcal{R})\|$, we denote the size of the representation of $\mathcal{R}$ via $\mathscr{S}$.

Let $\mathscr{S}_1$ and $\mathscr{S}_2$ be two preference representation schemes. We say that $\mathscr{S}_2$ is *at least as expressive* (and *succinct*) *as* $\mathscr{S}_1$, denoted $\mathscr{S}_1\precsim_e\mathscr{S}_2$, if there exists a function f in FP (i.e., computable in deterministic polynomial time) that translates any preference relation $\xi^{\mathscr{S}_1}(\mathcal{R})$ represented in $\mathscr{S}_1$ into an equivalent preference relation $\xi^{\mathscr{S}_2}(\mathcal{R})$ represented in $\mathscr{S}_2$, i.e., into a preference relation over the same outcomes and with the same preference relationships between them. More precisely, we require that $\xi^{\mathscr{S}_2}(\mathcal{R})=f(\xi^{\mathscr{S}_1}(\mathcal{R}))$ and $\mathit{Pref}^{\mathscr{S}_1}(\xi^{\mathscr{S}_1}(\mathcal{R}),\alpha,\beta)=\mathit{Pref}^{\mathscr{S}_2}(\xi^{\mathscr{S}_2}(\mathcal{R}),\alpha,\beta)$, for each pair of outcomes α and β. Observe that f belonging to FP entails that there exists a constant c_f (depending on f) such that $\|\xi^{\mathscr{S}_2}(\mathcal{R})\|\le\|\xi^{\mathscr{S}_1}(\mathcal{R})\|^{c_f}$, i.e., the size of $\xi^{\mathscr{S}_2}(\mathcal{R})$ is polynomially bounded in the size of $\xi^{\mathscr{S}_1}(\mathcal{R})$.[4]

A P- and an NP-*representation* $\mathscr{S}$ is a preference representation scheme whose function $\mathit{Pref}^{\mathscr{S}}$ is in P and NP, respectively. For example, the polynomially connected classes of acyclic binary CP-nets are NP-representation schemes [11].

When a compact representation $\mathscr{S}$ is clear from the context, we often simply write $\mathcal{R}$ instead of $\xi^{\mathscr{S}}(\mathcal{R})$, and $\alpha\succ_{\mathcal{R}}\beta$ and $\alpha\not\succ_{\mathcal{R}}\beta$ instead of $\mathit{Pref}^{\mathscr{S}}(\xi^{\mathscr{S}}(\mathcal{R}),\alpha,\beta)=1$ and $\mathit{Pref}^{\mathscr{S}}(\xi^{\mathscr{S}}(\mathcal{R}),\alpha,\beta)=0$, respectively. While doing so, we are identifying the preference relation with its actual representation.

3. Complexity of basic tasks on CP-nets

To precisely characterize the complexity of Pareto and majority voting tasks in Sections 4 and 5, we need to understand the complexity of deciding, given a CP-net N and two outcomes α and β, whether α dominates β or whether α and β are incomparable. Here, we prove that the former problem is NP-complete, while the latter is co-NP-complete. To achieve this, after giving some preliminary definitions on how to encode Boolean formulas into CP-nets, we show that deciding the satisfiability of Boolean formula can be reduced to the problem of deciding dominance between outcomes in CP-nets. This allows us to prove the NP-hardness of the dominance test in CP-nets and the co-NP-hardness of deciding incomparability is shown as a byproduct of this property. The NP-hardness of dominance in CP-nets was already proven by Boutilier et al. [11]. Here, we show this result, because the construction proposed here, which also allows us to prove a stricter result, is different from the one available in the literature, and it is required in multiple reductions in the rest of the paper.

3.1. Preliminaries

We first introduce a notation mapping Boolean assignments to outcomes of CP-nets; this notation will frequently be used later in the paper. In particular, to prove the hardness of voting tasks on mCP-nets, we often provide reductions from problems regarding the satisfiability (or validity) of (quantified) Boolean formulas. For this reason, mCP-nets will often have sets of features associated with sets of Boolean variables. For example, for a Boolean formula $\phi(X)$ over the set of Boolean variables $X=\{x_1,\ldots,x_n\}$, we often define an mCP-net $\mathcal{M}(\phi)$ that has $\mathcal{V}=\bigcup\{\{V_i^T,V_i^F\}\mid x_i\in X\}$ as a subset of its set of features. Then, for a (partial or complete) assignment σ_X over X, an outcome α_{σ_X} of $\mathcal{M}(\phi)$ encoding σ_X over the features set $\mathcal{V}$ is such that, for the features in $\mathcal{V}$, if $\sigma_X[x_i]=\mathtt{true}$, then $\alpha_{\sigma_X}[V_i^T V_i^F]=\overline{v_i^T}v_i^F$; if $\sigma_X[x_i]=\mathtt{false}$, then $\alpha_{\sigma_X}[V_i^T V_i^F]=v_i^T\overline{v_i^F}$; and if $\sigma_X[x_i]$ is undefined, then $\alpha_{\sigma_X}[V_i^T V_i^F]=v_i^T v_i^F$. The values of the features $F\notin\mathcal{V}$ in α_{σ_X} will be specified in each particular case.

We next define *formula nets*, which will be used in hardness proofs, and which are intuitively CP-nets aiming at having a particular preference relationship between two outcomes depending on the satisfiability of associated Boolean formulas in CNF. This will allow us to show that deciding dominance in CP-nets is NP-hard.

Formally, let $\phi(X)$ be a Boolean formula in 3CNF defined over the set of Boolean variables $X=\{x_1,\ldots,x_n\}$, and whose set of clauses is $C=\{c_1,\ldots,c_m\}$. We often omit the variable set from the notation of the Boolean formula, i.e., we write ϕ

[4] Note that the above definitions are slightly different from the ones by Lang and Xia [55]: the counterpart of this paper's function *Pref* in [55] is not required to be computable, and the transformation function f in [55] is only required to be polynomially bounded, but not polynomially computable.

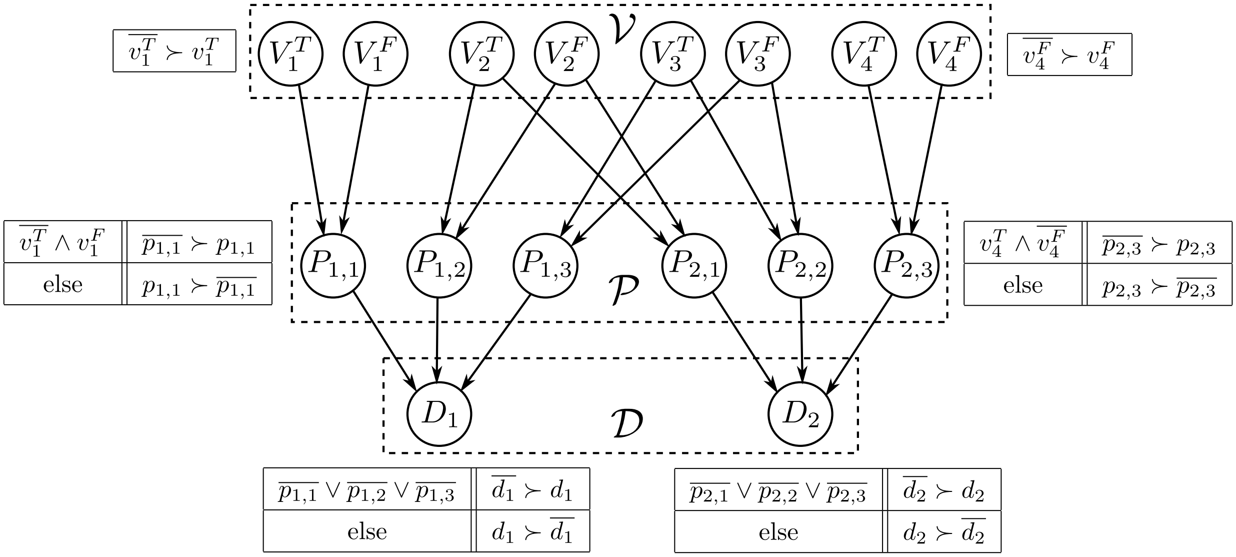

$\overline{v_1^T} \succ v_1^T$
$\mathcal{V}$
V_1^T V_1^F V_2^T V_2^F V_3^T V_3^F V_4^T V_4^F
$\overline{v_4^F} \succ v_4^F$
$\overline{v_1^T} \wedge v_1^F$ | $\overline{p_{1,1}} \succ p_{1,1}$
else | $p_{1,1} \succ \overline{p_{1,1}}$
$P_{1,1}$ $P_{1,2}$ $P_{1,3}$ $\mathcal{P}$ $P_{2,1}$ $P_{2,2}$ $P_{2,3}$
$v_4^T \wedge \overline{v_4^F}$ | $\overline{p_{2,3}} \succ p_{2,3}$
else | $p_{2,3} \succ \overline{p_{2,3}}$
D_1 $\mathcal{D}$ D_2
$\overline{p_{1,1}} \vee \overline{p_{1,2}} \vee \overline{p_{1,3}}$ | $\overline{d_1} \succ d_1$
else | $d_1 \succ \overline{d_1}$
$\overline{p_{2,1}} \vee \overline{p_{2,2}} \vee \overline{p_{2,3}}$ | $\overline{d_2} \succ d_2$
else | $d_2 \succ \overline{d_2}$

Fig. 6. The CP-net $F(\phi)$, where $\phi(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_4)$. Not all the CP tables are reported in the figure.

instead of $\phi(X)$, if this does not cause ambiguity. We denote by $\ell_{j,k}$ the k-th literal of the j-th clause. From $\phi(X)$, we build the CP-net $F(\phi)$ in the following way (see Fig. 6 for an example). The features of $F(\phi)$ are:

- for each variable $x_i \in X$, there are features V_i^T and V_i^F (called *variable features*), and we denote by $\mathcal{V}_\phi$ the set of variable features;
- for each clause $c_j \in C$, there is a feature D_j (called *clause feature*), and we denote by $\mathcal{D}_\phi$ the set of clause features; and
- for each literal $\ell_{j,k}$, there is a feature $P_{j,k}$ (called *literal feature*), and we denote by $\mathcal{P}_\phi$ the set of literal features.

All features are binary, with the usual notation for their values. When the formula ϕ is clear from the context, we often omit the subscript "ϕ" from the notation of the sets of features illustrated above. The edges of $F(\phi)$ are: for each literal $\ell_{j,k} = x_i$ or $\ell_{j,k} = \neg x_i$, there are edges $\langle V_i^T, P_{j,k}\rangle$, $\langle V_i^F, P_{j,k}\rangle$, and $\langle P_{j,k}, D_j\rangle$. The CP tables of $F(\phi)$ are:

- for each variable $x_i \in X$, features V_i^T and V_i^F have the CP tables

  $\boxed{\overline{v_i^T} \succ v_i^T}$ and $\boxed{\overline{v_i^F} \succ v_i^F}$, respectively;
- for each literal $\ell_{j,k}$, if $\ell_{j,k} = x_i$, then feature $P_{j,k}$ has the CP table

$\overline{v_i^T} \wedge v_i^F$	$\overline{p_{j,k}} \succ p_{j,k}$
else	$p_{j,k} \succ \overline{p_{j,k}}$

  ;

  otherwise (i.e., $\ell_{j,k} = \neg x_i$) $P_{j,k}$ has the CP table

$v_i^T \wedge \overline{v_i^F}$	$\overline{p_{j,k}} \succ p_{j,k}$
else	$p_{j,k} \succ \overline{p_{j,k}}$

  ;
- for each clause $c_j \in C$, feature D_j has the CP table

$\overline{p_{j,1}} \vee \overline{p_{j,2}} \vee \overline{p_{j,3}}$	$\overline{d_j} \succ d_j$
else	$d_j \succ \overline{d_j}$

  .

Note that $F(\phi)$ is binary, acyclic, singly connected, its indegree is three, and the CP-net can be built in polynomial time in the size of ϕ.

The following lemma and corollary show an important property of formula nets, which is that ϕ is satisfiable if and only if a particular outcome dominates others in $F(\phi)$.

Lemma 3.1. *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 $F(\phi)$ encoding σ_X on the feature set $\mathcal{V}$, and assigning non-overlined values to all other features, and let $\overline{\beta}$ be the outcome assigning overlined values to all and only variable and clause features. Then:*

(1) *There is an extension of σ_X to X satisfying $\phi(X)$ if and only if $\overline{\beta} \succ_{F(\phi)} \alpha_{\sigma_X}$;*
(2) *There is no extension of σ_X to X satisfying $\phi(X)$ if and only if $\overline{\beta} \bowtie_{F(\phi)} \alpha_{\sigma_X}$.*

Proof (sketch). The idea at the base of this proof is that the CP tables in $F(\phi)$ are designed so that the features enact the role of variables, literals, and clauses of a CNF Boolean formula. Details of the proof are at page 131. □

Corollary 3.2. *Let $\phi(X)$ be a Boolean formula in 3CNF defined over a set X of Boolean variables, and let α and $\overline{\beta}$ be two outcomes of $\mathcal{O}_{F(\phi)}$ assigning non-overlined values to all features and overlined values to all and only variable and clause features, respectively. Then:*

- *ϕ is satisfiable if and only if $\overline{\beta} \succ_{F(\phi)} \alpha$;*
- *ϕ is unsatisfiable if and only if $\overline{\beta} \bowtie_{F(\phi)} \alpha$.*

Proof. Observe that, when the empty assignment σ_X is considered, outcomes α_{σ_X} and $\overline{\beta}_{\sigma_X}$ of the statement of Lemma 3.1 coincide with outcomes α and $\overline{\beta}$ of the statement of this corollary, respectively. Moreover, any assignment over X is an extension of the empty one, and hence Lemma 3.1 applies. □

3.2. Complexity of dominance, incomparability, and optimality on CP-nets

As mentioned above, via formula nets, it is possible to show the NP-hardness and the co-NP-hardness of dominance and incomparability on CP-nets, respectively. We start by showing the NP-hardness of dominance on CP-nets. More formally, consider the following problem on CP-nets.

Problem: DOMINANCE
Instance: A CP-net N, and two outcomes $\alpha, \beta \in \mathcal{O}_N$.
Question: Is $\beta \succ_N \alpha$?

DOMINANCE is known to be in NP for some classes of instances, and in particular it is in NP for polynomially connected classes of acyclic binary CP-nets [11]. For these classes of CP-nets, it was shown that DOMINANCE is NP-hard, and hardness holds even if the considered CP-nets are singly connected and the indegree of each feature in the net is at most six [11]. Moreover, DOMINANCE 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) [11]. However, the exact complexity of DOMINANCE for general (non polynomially-connected classes of) acyclic binary CP-nets is still an open problem [11], and in particular it is unknown whether it belongs to NP or not. Allen [2] and Allen et al. [4] conjectured that in general binary acyclic 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. If this were the case, then dominance testing would be in NP also for binary acyclic CP-nets.

First, we give an improved result on the hardness of dominance testing in CP-nets. In particular, we show that the NP-hardness holds even if the indegree of the CP-net is three, while the minimum indegree previously required to show the hardness was six [11].

Theorem 3.3 *(improved over [11]). Let N be a CP-net, and let $\alpha, \beta \in \mathcal{O}_N$ be two outcomes. Deciding whether $\beta \succ_N \alpha$ is NP-hard. Hardness holds even if N is acyclic, binary, singly connected, and its indegree is three.*

Proof. To show that DOMINANCE is NP-hard, we prove that SAT $\leq_p$ DOMINANCE. Let ϕ be a Boolean formula in 3CNF. Consider the CP-net $F(\phi)$ (defined in Section 3.1) and outcomes α and β such that in α the values of all features are non-overlined, and in β the values of all and only variable and clause features are overlined. By Corollary 3.2, ϕ is satisfiable if and only if $\beta \succ_{F(\phi)} \alpha$. □

We now focus on the problem of testing incomparability between outcomes in CP-nets and show its co-NP-completeness. We show hardness via the properties of formula nets. More formally, consider the following problem.

Problem: INCOMPARABILITY
Instance: A CP-net N, and two outcomes $\alpha, \beta \in \mathcal{O}_N$.
Question: Is $\alpha \bowtie_N \beta$?

The following theorem shows that deciding whether two outcomes are incomparable in a preference relation represented via an NP-representation scheme is in co-NP.

Theorem 3.4. *Let $\mathcal{R}$ be a preference relation represented via an NP-representation scheme, and let α and β be two outcomes. Deciding whether $\alpha \bowtie_{\mathcal{R}} \beta$ is in co-NP.*

Proof. We show that disproving $\alpha \bowtie \beta$ is in NP. In fact, if $\alpha \not\bowtie \beta$, then either $\alpha \succ \beta$ or $\beta \succ \alpha$. In these cases, since $\mathcal{R}$ is represented via an NP-representation, there is a polynomial certificate either witnessing $\alpha \succ \beta$ or witnessing $\beta \succ \alpha$. To conclude, observe that such a certificate can be checked in polynomial time. □

We now focus on CP-nets and show that deciding incomparability is co-NP-hard.

Theorem 3.5. *Let N be a CP-net, and let α and β be two outcomes. Deciding whether $\alpha \bowtie_N \beta$ is co-NP-hard. Hardness holds even if N is acyclic, binary, singly connected, and its indegree is three.*

Proof. To show that Incomparability is co-NP-hard, we prove that Unsat $\leq_p$ Incomparability. Let ϕ be a Boolean formula in 3CNF. Consider the CP-net $F(\phi)$ (defined in Section 3.1) and outcomes α and β such that in α the values of all features are non-overlined, and in β the values of all and only variable and clause features are overlined. By Corollary 3.2, ϕ is unsatisfiable if and only if $\alpha \bowtie_{F(\phi)} \beta$. □

By combining the previous two results, we obtain that testing incomparability over polynomially connected classes of acyclic CP-nets is co-NP-complete.

Corollary 3.6. *Let $\mathcal{C}$ be a polynomially connected class of acyclic CP-nets. Let $N \in \mathcal{C}$ be a CP-net, and let α and β be two outcomes. Deciding whether $\alpha \bowtie_N \beta$ is co-NP-complete.*

We emphasize here that checking the incomparability between two outcomes is different from deciding the "ordering query" defined by Boutilier et al. [11]. For a CP-net N and two outcomes α and β, an ordering query is deciding whether there is at least a preference raking $\mathcal{R}$ satisfying N such that $\alpha \succ_{\mathcal{R}} \beta$. As noticed by Boutilier et al. [11], this is tantamount to deciding whether $N \not\models \beta \succ \alpha$. Since $N \not\models \beta \succ \alpha$ if and only if $\beta \not\succ_N \alpha$, deciding an ordering query is actually co-NP-hard, because dominance testing is NP-hard. Therefore, it comes as no surprise that the polynomial algorithm proposed by Boutilier et al. [11] to decide ordering queries is actually "partially complete" (as said by [11]). In fact, given the co-NP-hardness of the ordering query problem, there is no sound and complete deterministic polynomial-time algorithm for this problem (unless a major breakthrough in complexity theory occurs, showing that P = NP).

We conclude this section by looking at the complexity of deciding whether an outcome is optimal in a CP-net. This result is needed in Section 4 to characterize the complexity of one of the voting tasks of Pareto voting. Here, we show that this problem can be decided in LOGSPACE. More formally, consider the following problem.

Problem:	Optimality-Testing
Instance:	A CP-net N, and an outcome $\alpha \in \mathcal{O}_N$.
Question:	Is α optimal in N?

Recall that for acyclic CP-nets, there is an outcome that is optimum [11]. This outcome is also the only optimal one in a CP-net. Therefore, checking whether an outcome is optimal is tantamount to checking whether the outcome is optimum. It was shown that, given an acyclic CP-net N, computing the unique optimal outcome o_N of N can be carried out in deterministic polynomial time, more precisely in linear time, through the "forward sweep" procedure [11]. Hence, as pointed out by Rossi et al. [71], a simple procedure to decide whether a given outcome α is optimal in an acyclic CP-net N is to compute o_N (in polynomial time) and then to compare α to o_N. However, this problem actually belongs to a complexity class that is a below P when acyclic CP-nets are considered.

Theorem 3.7. *Let N be an acyclic CP-net, and let $\alpha \in \mathcal{O}_N$ be an outcome. Deciding whether α is optimal in N is in LOGSPACE.*

Proof. As N is acyclic, if an outcome α is not optimal in N, then there is an improving flipping sequence from α to the optimum outcome, and hence there is at least a feature whose value can be flipped in α to obtain a better outcome. Therefore, to decide whether α is optimal, it suffices to consider in turn all features F and check whether it is possible to perform an improving flip according to the CP table of F. If no feature can be flipped to improve the outcome, then α is optimal. This procedure requires only logarithmic space to be carried out. □

Now that we have analyzed the complexity of dominance, incomparability, and optimality in CP-nets, we can devote our focus to the complexity of Pareto and majority voting on CP-nets in the next two sections.

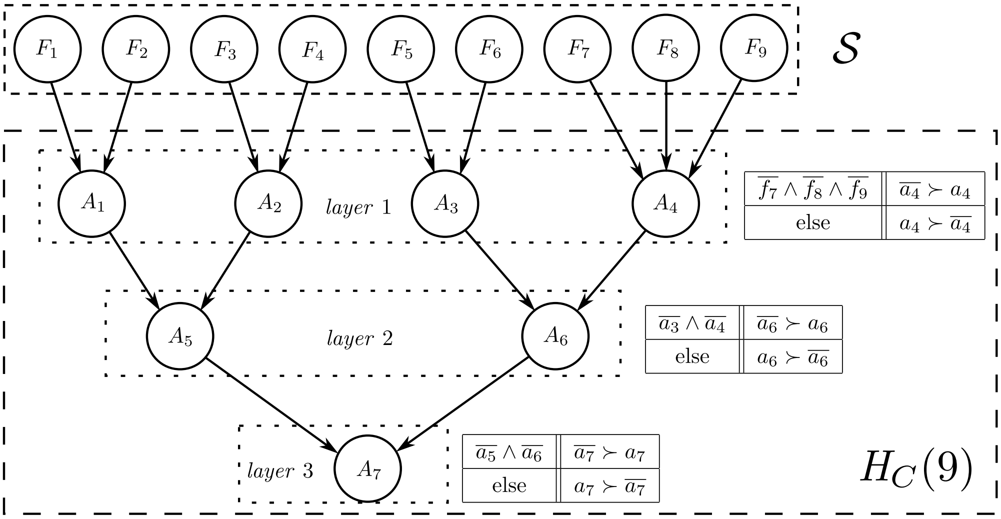

F_1
F_2
F_3
F_4
F_5
F_6
F_7
F_8
F_9
$\mathcal{S}$
A_1
A_2
layer 1
A_3
A_4
$\overline{f_7} \wedge \overline{f_8} \wedge \overline{f_9}$ | $\overline{a_4} \succ a_4$
else | $a_4 \succ \overline{a_4}$
A_5
layer 2
A_6
$\overline{a_3} \wedge \overline{a_4}$ | $\overline{a_6} \succ a_6$
else | $a_6 \succ \overline{a_6}$
layer 3
A_7
$\overline{a_5} \wedge \overline{a_6}$ | $\overline{a_7} \succ a_7$
else | $a_7 \succ \overline{a_7}$
$H_C(9)$

Fig. 7. An interconnecting net $H_C(9)$. Not all the CP tables are reported in the figure.

4. Complexity of Pareto voting on *m*CP-nets

In this section, we characterize the complexity of Pareto voting tasks on *m*CP-nets. In particular, after giving some preliminaries on specific structures of CP-nets that we will use in our reductions, we analyze the complexity of Pareto dominance, which is proven NP-complete. Then, we devote our analysis to the problems related to Pareto optimal outcomes, namely, deciding whether an outcome is Pareto optimal, and deciding whether an *m*CP-net has a Pareto optimal outcome. We prove the former co-NP-complete, while for the latter, we are able to show that every *m*CP-net has a Pareto optimal outcome, which implies that the problem is trivial (i.e., can be answered in constant time). To conclude, we study the complexity of problems on Pareto optimum outcomes, namely, deciding whether an outcome is Pareto optimum, and deciding whether an *m*CP-net has a Pareto optimum outcome. Both problems are proven to be tractable, and in particular we show the former to be in LOGSPACE and the latter to be in P. We recall that the Pareto voting semantics is based on the concept of unanimity, i.e., given an *m*CP-net $\mathcal{M}$ and two outcomes $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$, it holds that $\beta \succ^{p}_{\mathcal{M}} \alpha$, if all agents prefer β to α, i.e., $|S^{\succ}_{\mathcal{M}}(\beta, \alpha)| = m$.

4.1. Preliminaries

We now introduce a specific structure of CP-nets that will be used in the forthcoming reductions. In particular, the *conjunctive* $H_C(\cdot)$ and *disjunctive* $H_D(\cdot)$ *interconnecting nets* are CP-nets whose role is intuitively to link different parts of bigger CP-nets: given a set $\mathcal{S}$ of m features, the aim of the CP-nets $H_C(m)$ and $H_D(m)$ is to propagate the information that all features of $\mathcal{S}$ and at least one of the features of $\mathcal{S}$, respectively, have been flipped to their overlined value.

We first introduce the conjunctive interconnecting CP-net $H_C(m)$. See Fig. 7 for an example of a $H_C(9)$ interconnecting net. Such a CP-net is partitioned into layers, and it is a kind of "inverted pyramid". In particular, $H_C(m)$ is an acyclic DAG in which each feature of a layer has two or three distinct parents in the previous layer, and at most one child in the next layer. Features belonging to the same layer have no parents in common, and in every layer at most one feature has three parents. The first layer of $H_C(m)$ is attached to a set $\mathcal{S}$ of m different features of the CP-net that we want to connect. In the first layer of $H_C(m)$, the above described connection properties hold relative to the features of $\mathcal{S}$. The layer with a unique feature, which we call *apex*, is the last layer of the CP-net. All the features of $H_C(m)$ are named A_i with a proper increasing index i (and they are distinct from the features in $\mathcal{S}$), and their values are, as usual, $\{a_i, \overline{a_i}\}$. The CP table for a feature A_i states that value $\overline{a_i}$ is preferred to a_i whenever the value of every parent of A_i is overlined. Otherwise, the value a_i is preferred to $\overline{a_i}$. Let α be an outcome in which the values of all the features A_i are non-overlined. It is not difficult to see that, whenever all the features in $\mathcal{S}$ have overlined values in α, there is an improving flipping sequence starting from α changing the values of all A_i (and hence also the value of the apex) to their overlined values.

The disjunctive interconnecting net $H_D(m)$ is similar to $H_C(m)$. The features and the structure of $H_D(m)$ are the same as those in $H_C(m)$. The only variations are on the CP tables. Since $H_D(m)$ is a disjunctive net, given a feature A_i of $H_D(m)$, value $\overline{a_i}$ is preferred to value a_i, whenever at least one of the parents of A_i has been flipped to its overlined value. Let α be an outcome in which the values of all the features A_i are non-overlined. It is easy to see that, whenever at least a feature in $\mathcal{S}$ has an overlined value in α, there is an improving flipping sequence starting from α and reaching an outcome in which the apex has an overlined value.

Note that $H_{\mathrm{C}}(m)$ and $H_{\mathrm{D}}(m)$ are binary, acyclic, singly connected, their indegree is at most three, and the CP-nets can be built in polynomial time in $|\mathcal{S}| = m$, as the number of their features is polynomial in m (in particular, strictly less than m), and each feature has a bounded number of parents which translates into CP tables of bounded sizes.

4.2. Complexity of Pareto dominance on mCP-nets

First, we analyze the problem of deciding Pareto dominance on CP-nets, which is shown NP-complete. More formally, consider the following problem.

Problem: Pareto-Dominance
Instance: An mCP-net $\mathcal{M}$, and two outcomes $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$.
Question: Is $\beta \succ^{p}_{\mathcal{M}} \alpha$?

The following theorem shows that deciding Pareto dominance over preference profiles represented via an NP-representation scheme is in NP.

Theorem 4.1. *Let $\mathscr{P}$ be a preference profile defined over the same combinatorial domain and represented via an NP-representation scheme, and let α and β be two outcomes. Then, deciding whether $\beta \succ^{p}_{\mathscr{P}} \alpha$ is in NP.*

Proof. To show that this problem resides in NP, we exhibit a concise certificate for it. Let $\mathscr{P} = \langle \mathcal{R}_1, \dots, \mathcal{R}_m \rangle$. If $\beta \succ^{p}_{\mathscr{P}} \alpha$, then, for all $1 \le i \le m$, it holds that $\beta \succ_{\mathcal{R}_i} \alpha$. Since, for every agent i, $\mathcal{R}_i$ is represented via an NP-representation scheme, there is a concise certificate witnessing that $\beta \succ_{\mathcal{R}_i} \alpha$. Therefore, in order to decide whether $\beta \succ^{p}_{\mathscr{P}} \alpha$, it suffices to guess and subsequently check the polynomial witnesses that, for each agent i, it holds that $\beta \succ_{\mathcal{R}_i} \alpha$. The overall guess requires only polynomial space, and it can be checked in polynomial time. □

Observe that on 1CP-nets, Pareto dominance is equivalent to dominance (on simple CP-nets). Therefore, the following result, which follows directly from Theorem 3.3, shows that deciding Pareto dominance in mCP-nets is NP-hard.

Theorem 4.2. *Let $\mathcal{M}$ be an mCP-net, and let α and β be two outcomes. Then, deciding whether $\beta \succ^{p}_{\mathcal{M}} \alpha$ is NP-hard. Hardness holds even on classes of singly connected acyclic binary mCP-nets with indegree at most three and at most one agent.*

By combining the two previous results, we immediately obtain that Pareto dominance over polynomially connected classes of acyclic mCP-nets is NP-complete.

Corollary 4.3. *Let $\mathcal{C}$ be a polynomially connected class of acyclic mCP-nets. Let $\mathcal{M} \in \mathcal{C}$ be an mCP-net, and let α and β be two outcomes. Then, deciding whether $\beta \succ^{p}_{\mathcal{M}} \alpha$ is NP-complete.*

4.3. Complexity of Pareto optimality on mCP-nets

Here, we devote our analysis to problems on Pareto optimal outcomes. In particular, the problems analyzed are deciding whether an outcome is Pareto optimal, and deciding whether an mCP-net has a Pareto optimal outcome. We first focus on deciding Pareto optimality of outcomes in mCP-nets. We show that this problem is co-NP-complete. More formally, consider the following problem.

Problem: Is-Pareto-Optimal
Instance: An mCP-net $\mathcal{M}$, and an outcome $\alpha \in \mathcal{O}_{\mathcal{M}}$.
Question: Is α Pareto optimal in $\mathcal{M}$?

The following result shows that, for a preference profile represented via an NP-representation scheme, deciding the Pareto optimality of an outcome is in co-NP.

Theorem 4.4. *Let $\mathscr{P}$ be a preference profile defined over the same combinatorial domain and represented via an NP-representation scheme, and let α be an outcome. Then, deciding whether α is Pareto optimal in $\mathscr{P}$ is in co-NP.*

Proof. We show that disproving α being Pareto optimal is in NP. If α is not Pareto optimal in $\mathscr{P}$, then there is an outcome β such that $\beta \succ^{p}_{\mathscr{P}} \alpha$. Therefore, we can guess such an outcome β along with the witness that $\beta \succ^{p}_{\mathscr{P}} \alpha$. This guess requires only polynomial space and can be checked in polynomial time (see the proof of Theorem 4.1). □

To prove the co-NP-hardness of Is-Pareto-Optimal, we use a reduction from Unsat. Consider the following construction. Let $\phi(X)$ be a Boolean formula in 3CNF defined over the set of Boolean variables $X = \{x_1, \dots, x_n\}$, and whose set of clauses

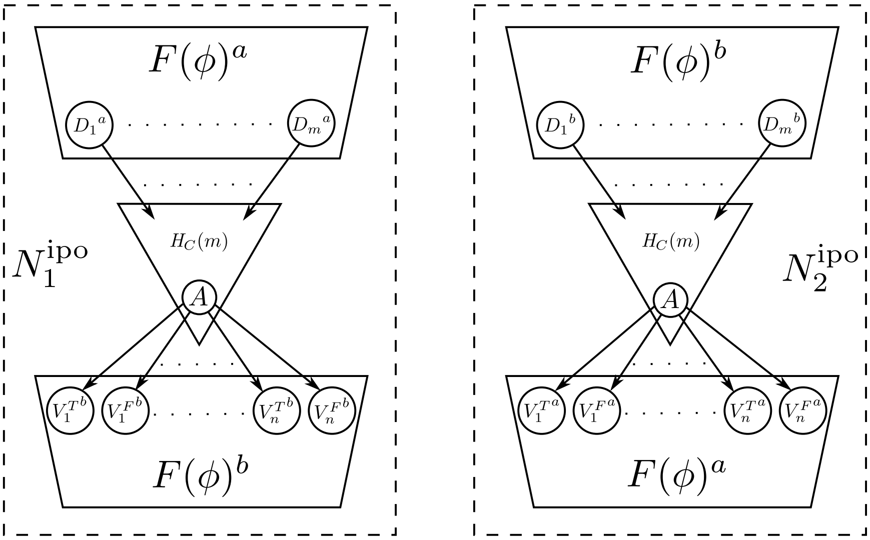


Fig. 8. A schematic representation of the building blocks of the 2CP-net $\mathcal{M}_{\mathrm{ipo}}(\phi) = \langle N_1^{\mathrm{ipo}}, N_2^{\mathrm{ipo}} \rangle$.

is $C = \{c_1, \ldots, c_m\}$. From ϕ, we build the 2CP-net $\mathcal{M}_{\mathrm{ipo}}(\phi) = \langle N_1^{\mathrm{ipo}}, N_2^{\mathrm{ipo}} \rangle$ in the following way. The CP-nets N_1^{ipo} and N_2^{ipo} are built similarly, and we discuss first N_1^{ipo}. See Fig. 8 for a schematic representation of the interconnections between the building blocks of N_1^{ipo} and N_2^{ipo}.

In N_1^{ipo}, there are two complete copies of the net $F(\phi)$ (defined in Section 3.1), with its features, edges, and CP tables. To distinguish these two copies, we append two different superscript to them and obtain $F(\phi)^a$ and $F(\phi)^b$. All features of these two nets have the corresponding superscript a or b to make them different and distinguish them from the others. The clause features of $F(\phi)^a$, i.e., $D_1{}^a, \ldots, D_m{}^a$, are attached to a conjunctive interconnecting net $H_C(m)$ (defined in Section 4.1). The apex A of $H_C(m)$ is attached to *all* the variable features of $F(\phi)^b$. Since the variable features of $F(\phi)^b$ now have one parent (more precisely the same parent, i.e., the apex of $H_C(m)$), the CP tables of these features are a bit different from those in $F(\phi)$. Variable features F of $F(\phi)^b$ have the CP tables

$\overline{a}$	$\overline{f} \succ f$
a	$f \succ \overline{f}$

.

The CP-net N_2^{ipo} is similar to N_1^{ipo}, with the only difference that the conjunctive interconnecting CP-net $H_C(m)$ attaches the clause features of $F(\phi)^b$ to the variable features of $F(\phi)^a$. The CP tables in N_2^{ipo} of the variable features of $F(\phi)^a$ are therefore modified accordingly.

Observe that $\mathcal{M}_{\mathrm{ipo}}(\phi)$ is acyclic, binary, its indegree is three, and can be computed in polynomial time from ϕ. Moreover, the class of mCP-nets $\{\mathcal{M}_{\mathrm{ipo}}(\phi)\}_{\phi}$ derived from formulas ϕ of the specified kind and according to the reduction shown above is polynomially connected. The following result shows that $\phi(X)$ is satisfiable if and only if a particular outcome of $\mathcal{M}_{\mathrm{ipo}}(\phi)$ is not Pareto optimal.

Lemma 4.5. *Let $\phi(X)$ be a 3CNF Boolean formula, and let α be the outcome of $\mathcal{M}_{\mathrm{ipo}}(\phi)$ assigning non-overlined values to all features. Then, $\phi(X)$ is satisfiable if and only if α is not Pareto optimal in $\mathcal{M}_{\mathrm{ipo}}(\phi)$.*

Proof (sketch). The key point of the proof is that the features of the formula net downline of the interconnecting net can be flipped to their overlined values if and only if the formula ϕ is satisfiable. Details of the proof are at page 132. □

The above property implies that, in mCP-nets, deciding the Pareto optimality of an outcome is co-NP-hard.

Theorem 4.6. *Let $\mathcal{M}$ be an mCP-net, and let $\alpha \in \mathcal{O}_{\mathcal{M}}$ be an outcome. Deciding whether α is Pareto optimal is co-NP-hard. Hardness holds even on polynomially connected classes of acyclic binary mCP-nets with indegree at most three and at most two agents.*

Proof. We prove that Is-Pareto-Optimal is co-NP-hard by showing a reduction from Unsat. Let $\phi(X)$ be a 3CNF formula, and consider the 2CP-net $\mathcal{M}_{\mathrm{ipo}}(\phi) = \langle N_1^{\mathrm{ipo}}, N_2^{\mathrm{ipo}} \rangle$. Consider the outcome $\alpha \in \mathcal{O}_{\mathcal{M}_{\mathrm{ipo}}(\phi)}$ in which the values of all features are non-overlined. By Lemma 4.5, $\phi(X)$ is unsatisfiable (and hence a “yes”-instance of Unsat) if and only if α is Pareto optimal in $\mathcal{M}_{\mathrm{ipo}}(\phi)$. □

Notice here that the presence of at least two agents in an mCP-net is an essential source of complexity for the problem Is-Pareto-Optimal. In fact, if there were only one agent, then deciding whether an outcome α is Pareto optimal would be tantamount to checking whether α is optimal for that only agent, and we have seen already that this task can be carried out in LOGSPACE (see Theorem 3.7).

By combining the two above results, we immediately conclude that deciding the Pareto optimality of an outcome over polynomially connected classes of acyclic mCP-nets is co-NP-complete.

Corollary 4.7. *Let $\mathcal{C}$ be a polynomially connected class of acyclic mCP-nets. Let $\mathcal{M} \in \mathcal{C}$ be an mCP-net, and let $\alpha \in \mathcal{O}_{\mathcal{M}}$ be an outcome. Deciding whether α is Pareto optimal is co-NP-complete.*

We now focus on the problem of deciding the existence of Pareto optimal outcomes in mCP-nets. We show that every mCP-nets has a Pareto optimal outcome, which implies that the problem is trivial (i.e., can be answered in constant time). More formally, consider the following problem.

Problem: Exists-Pareto-Optimal
Instance: An mCP-net $\mathcal{M}$.
Question: Does $\mathcal{M}$ have a Pareto optimal outcome?

The following lemma shows that an acyclic CP-net has always a Pareto optimal outcome.

Lemma 4.8. *Let $\mathcal{M}$ be an acyclic mCP-net. Then, $\mathcal{M}$ has (always) a Pareto optimal outcome.*

Proof. Let $\mathcal{M} = \langle N_1, \ldots, N_m \rangle$ be an mCP-net, and assume by contradiction that $\mathcal{M}$ has no Pareto optimal outcome. Consider any net N_i of $\mathcal{M}$. By the fact that N_i is acyclic, it follows that there is a unique optimum outcome o_{N_i} in N_i. Since we assume that in $\mathcal{M}$ there are no Pareto optimal outcomes, there must be an outcome $\beta \neq o_{N_i}$ such that $\beta \succ^{p}_{\mathcal{M}} o_{N_i}$, i.e., β is preferred to o_{N_i} by *all* agents of $\mathcal{M}$. However, there is no outcome β such that $\beta \succ_{N_i} o_{N_i}$, because o_{N_i} is the optimum outcome in N_i: a contradiction. Therefore, there must be a Pareto optimal outcome in $\mathcal{M}$. □

4.4. Complexity of Pareto optimums on mCP-nets

We now focus on Pareto optimum outcomes. In particular, the problems analyzed are deciding whether an outcome is Pareto optimum, and deciding whether an mCP-net has a Pareto optimum outcome. Details of the proofs in this section are given starting at page 133.

To study the complexity of these problems, we first point out the following property. An mCP-net has a Pareto optimum outcome if and only if all its individual CP-nets have the very same optimum outcome. Indeed, to be Pareto optimum, an outcome has to dominate all other outcomes in all the CP-nets of the mCP-net: this property is satisfied only by an outcome that is the optimum in all the CP-nets of the mCP-net. We state this property in the following lemma.

Lemma 4.9. *Let $\mathcal{M}$ be an acyclic mCP-net. Then, $\mathcal{M}$ has a Pareto optimum outcome if and only if all the individual CP-nets of $\mathcal{M}$ have the very same optimum outcome (that, in which case, is also the Pareto optimum outcome of $\mathcal{M}$).*

Based on the property above, we characterize the complexity of problems on Pareto optimum outcomes. We first focus on deciding whether outcomes are Pareto optimum in mCP-nets. We show that this problem is solvable in LOGSPACE. More formally, consider the following problem.

Problem: Is-Pareto-Optimum
Instance: An mCP-net $\mathcal{M}$ and an outcome α.
Question: Is α Pareto optimum in $\mathcal{M}$?

Observe that, by Lemma 4.9, to verify whether an outcome α is Pareto optimum, it suffices to check, for each individual CP-net in turn, whether α is the individual optimal outcome for that agent. Recall that checking the individual optimality of α is in LOGSPACE (see Theorem 3.7). We state this result in the following theorem.

Theorem 4.10. *Let $\mathcal{M}$ be an acyclic mCP-net, and let $\alpha \in \mathcal{O}_{\mathcal{M}}$ be an outcome. Deciding whether α is Pareto optimum in $\mathcal{M}$ is in LOGSPACE.*

To conclude, we study the complexity of deciding whether an mCP-net has a Pareto optimum outcome. We show that this problem is in P. More formally, consider the following problem.

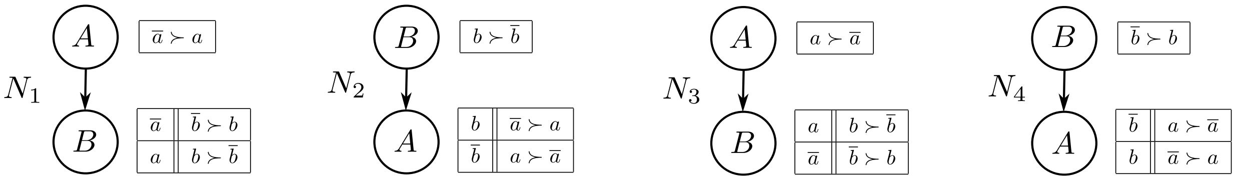


Fig. 9. The 4CP-net $\mathcal{M}_{NoWin}$ of Theorem 5.1.

Problem: EXISTS-PARETO-OPTIMUM
Instance: An mCP-net $\mathcal{M}$.
Question: Does $\mathcal{M}$ have a Pareto optimum outcome?

Since it is possible to check in LOGSPACE whether an outcome α is Pareto optimum, we can conclude that deciding whether there exists a Pareto optimum in an mCP-net can be done in polynomial time. Indeed, to answer this question, we can compute the optimum outcome α of one of the CP-nets of the mCP-net (this can be carried out in polynomial time [11]), and then check in LOGSPACE that α is Pareto optimum (see above).

Theorem 4.11. *Let $\mathcal{M}$ be an acyclic mCP-net. Deciding whether $\mathcal{M}$ has a Pareto optimum outcome is in P.*

5. Complexity of majority voting on *m*CP-nets

In this section, we characterize the complexity of majority voting tasks on mCP-nets. We start by showing that there are mCP-nets without majority optimal and optimum outcomes, which implies that deciding the existence of majority optimal and optimum outcomes is not a trivial problem. In the preliminary section, we also define some CP-nets that will be used in the reductions exhibited in this section. More specifically, we will introduce direct nets, which are CP-nets having a designated outcome as optimal. Moreover, we introduce summarized formula nets, which, similarly to formula nets, are CP-nets encoding Boolean formulas, but they associate the satisfiability of formulas with outcomes having specific values on just two features. Then, we analyze the complexity of deciding majority dominance in mCP-nets, which is shown NP-complete. Subsequently, we devote our analysis to the problems related to majority optimal outcomes, namely, deciding whether an outcome is majority optimal, and deciding whether an mCP-net has a majority optimal outcome. We prove the former co-NP-complete and the latter Σ_2^{P}-complete. To conclude, we study the complexity of problems on majority optimum outcomes, namely, deciding whether an outcome is majority optimum, and deciding whether an mCP-net has a majority optimum outcome. We prove the former Π_2^{P}-complete and the latter is shown Π_2^{P}-hard and belonging to $\mathrm{D}_2^{\mathrm{P}}$.

Recall that, given an mCP-net $\mathcal{M}$ and two different outcomes $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$, it holds that $\beta \succ^{maj}_{\mathcal{M}} \alpha$, if the majority of agents prefer β to α, i.e., $|S^{\succ}_{\mathcal{M}}(\beta,\alpha)| > |S^{\prec}_{\mathcal{M}}(\beta,\alpha)| + |S^{\bowtie}_{\mathcal{M}}(\beta,\alpha)|$. An outcome α is majority optimal in an mCP-net if α is not majority dominated by any other outcome, while α is majority optimum if α majority dominates all other outcomes. What we call majority optimal and majority optimum outcomes, in some works (see, e.g., the works by Li et al. [58] and Felsenthal and Tideman [31]) are named weak and (strong) Condorcet winners, respectively. However, in the literature (see, e.g., the works by Brandt et al. [20,21] and Baumeister and Rothe [7]), the nomenclature of weak/strong Condorcet winner has also been used with a slightly different meaning. In particular, with this different meaning, the strong Condorcet winner is the majority optimum outcome when the majority dominance relation requires that more than half of the agents prefer an outcome to another, while the weak Condorcet winner is the majority optimum outcome when the majority dominance relation requires that at least half of the agents prefer an outcome to another. To avoid any confusion, in this paper, we prefer to stick to the concepts of majority optimal and majority optimum outcomes, which we introduced in Section 2.1.

5.1. Preliminaries

We first show that there are mCP-nets that do not have any majority optimal outcome, and hence neither a majority optimum outcome. Thus, deciding whether an mCP-net has majority optimal or optimum outcomes is a non-trivial problem.

Theorem 5.1. *There are acyclic binary singly-connected mCP-nets not having majority optimal and majority optimum outcomes.*

Proof. Consider the acyclic binary singly connected 4CP-net $\mathcal{M}_{NoWin} = \langle N_1, N_2, N_3, N_4 \rangle$ defined in Fig. 9.

The preferences encoded in the four nets are: $\bar{a}\bar{b} \succ_{N_1} \bar{a}b \succ_{N_1} ab \succ_{N_1} a\bar{b}$; $\bar{a}b \succ_{N_2} ab \succ_{N_2} a\bar{b} \succ_{N_2} \bar{a}\bar{b}$; $ab \succ_{N_3} a\bar{b} \succ_{N_3} \bar{a}\bar{b} \succ_{N_3} \bar{a}b$; and $a\bar{b} \succ_{N_4} \bar{a}\bar{b} \succ_{N_4} \bar{a}b \succ_{N_4} ab$. Observe that: ab is not majority optimal, because $\bar{a}b \succ^{maj}_{\mathcal{M}_{NoWin}} ab$; $\bar{a}b$ is not majority optimal, because $\bar{a}\bar{b} \succ^{maj}_{\mathcal{M}_{NoWin}} \bar{a}b$; $a\bar{b}$ is not majority optimal, because $ab \succ^{maj}_{\mathcal{M}_{NoWin}} a\bar{b}$; and $\bar{a}\bar{b}$ is not majority optimal, because $a\bar{b} \succ^{maj}_{\mathcal{M}_{NoWin}} \bar{a}\bar{b}$. This implies that $\mathcal{M}_{NoWin}$ does not have any majority optimal outcome, and hence also either a majority optimum outcome. □

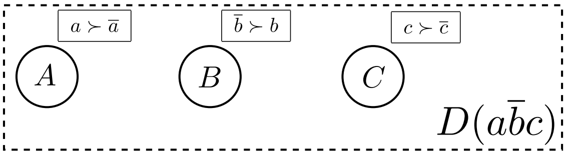


Fig. 10. The direct net $D(\alpha)$, with $\alpha = a\bar{b}c$.

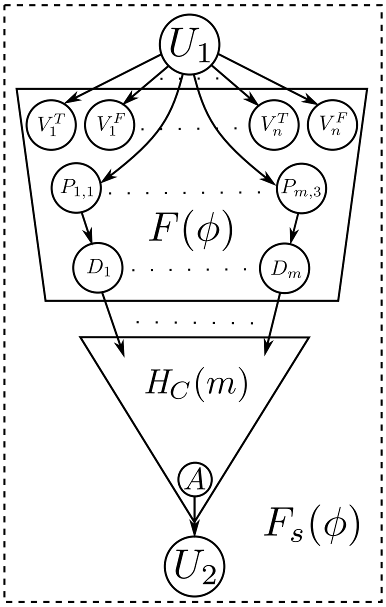


Fig. 11. A schematic representation of CP-net $F_s(\phi)$.

We now define two CP-nets that will be used in the reductions of this section on majority voting. First, we define *direct nets*, which intuitively are CP-nets that have a specific desired optimum outcome (recall that an acyclic CP-net has a unique optimal outcome, which is also optimum). Let $\mathcal{S}$ be a set of binary features defined over the usual values, and let $\alpha \in Dom(\mathcal{S})$ be an outcome over $\mathcal{S}$. The *direct net* $D(\alpha) = \langle \mathcal{G}_{D(\alpha)}, Dom_{D(\alpha)}, CPT_{D(\alpha)} \rangle$ is the CP-net such that $\mathcal{F}_{D(\alpha)} = \mathcal{S}$, $\mathcal{E}_{D(\alpha)} = \varnothing$, the domain of each feature F of $D(\alpha)$ is the same as the domain of F in $\mathcal{S}$, and the CP tables of $D(\alpha)$ are such that, given a feature F, if $\alpha[F] = f$, then the CP table for F is $\boxed{f \succ \overline{f}}$, otherwise (i.e., $\alpha[F] = \overline{f}$) the CP table for F is $\boxed{\overline{f} \succ f}$. See Fig. 10 for an example. Given any outcome $\beta \in \mathcal{O}_{D(\alpha)}$ such that $\beta \neq \alpha$, it holds that $\alpha \succ_{D(\alpha)} \beta$. Moreover, let β and γ be two different outcomes of $D(\alpha)$ such that, for all features F for which $\gamma[F] \neq \beta[F]$, it holds that $\gamma[F] = \alpha[F]$ (and hence $\beta[F] \neq \alpha[F]$). Then, $\gamma \succ_{D(\alpha)} \beta$. Note that $D(\alpha)$ is binary, acyclic, singly connected, its indegree is zero, and the net can be built in polynomial time from α.

We finally introduce *summarized formula nets*, which are similar to formula nets (of Section 3.1), with the advantage that these new nets put in relationship the satisfiability of Boolean formulas with the flip of only two features, instead of with the flip of all variable and clause features. This advantage comes at the cost of losing the single connectedness property of the nets, which, instead, is satisfied in non-summarized formula nets. Formally, let $\phi(X)$ be a Boolean formula in 3CNF defined over the set of Boolean variables $X = \{x_1, \ldots, x_n\}$, and whose set of clauses is $C = \{c_1, \ldots, c_m\}$. From $\phi(X)$, we build the CP-net $F_s(\phi)$ in the following way (see Fig. 11).

The CP-net $F_s(\phi)$ embeds a formula net $F(\phi)$ (defined in Section 3.1) with its features and links. Moreover, in $F_s(\phi)$, there is an interconnecting net $H_C(m)$ (defined in Section 4.1), which is attached to all clause features of $F(\phi)$. We denote by $\mathcal{A}$ the set of features belonging to the net $H_C(m)$ embedded in $F_s(\phi)$. To conclude with the features of $F_s(\phi)$, there are two more features: U_1 and U_2, where U_1 has no parents and is linked to all variable and literal features of $F(\phi)$, while U_2 is not the parent of any feature and its unique parent is the apex of the interconnecting net.

The CP tables of $F_s(\phi)$ are as follows:

- feature U_1 has the CP table $\boxed{\overline{u_1} \succ u_1}$;
- for each variable $x_i \in X$, features V_i^T and V_i^F have the CP tables

u_1	$\overline{v_i^T} \succ v_i^T$
$\overline{u_1}$	$v_i^T \succ \overline{v_i^T}$

and

u_1	$\overline{v_i^F} \succ v_i^F$
$\overline{u_1}$	$v_i^F \succ \overline{v_i^F}$

, respectively;

- for each literal $\ell_{j,k}$, if $\ell_{j,k} = x_i$, then feature $P_{j,k}$ has the CP table

$u_1 \wedge \overline{v_i^T} \wedge v_i^F$	$\overline{p_{j,k}} \succ p_{j,k}$
else	$p_{j,k} \succ \overline{p_{j,k}}$

;

otherwise, if $\ell_{j,k} = \neg x_i$, then $P_{j,k}$ has the CP table

$u_1 \wedge v_i^T \wedge \overline{v_i^F}$	$\overline{p_{j,k}} \succ p_{j,k}$
else	$p_{j,k} \succ \overline{p_{j,k}}$

;

- clause features have the same CP table as in $F(\phi)$;
- features of the conjunctive interconnecting net $H_C(m)$ have the usual CP tables;
- feature U_2, if feature A is the apex of the interconnecting net, has the CP table

$\overline{a}$	$\overline{u_2} \succ u_2$
a	$u_2 \succ \overline{u_2}$

.

Note that $F_s(\phi)$ is binary, acyclic, its indegree is three, and the net can be built in polynomial time in the size of ϕ. Moreover, the class of mCP-nets $\{\mathcal{M}_{\text{ipo}}(\phi)\}_\phi$ derived from formulas ϕ of the specified kind and according to the reduction shown above is polynomially connected.

Now we give an equivalent of Lemma 3.1 for $F_s(\phi)$. In particular, the following lemma and corollary show that ϕ is satisfiable if and only if a particular outcome dominates others in $F_s(\phi)$.

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 $\overline{\beta}[U_1U_2] = \overline{u_1 u_2}$, 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 $\phi(X)$ if and only if $\overline{\beta} \succ_{F_s(\phi)} \alpha_{\sigma_X}$;*
(2) *There is no extension of σ_X to X satisfying $\phi(X)$ if and only if $\overline{\beta} \bowtie_{F_s(\phi)} \alpha_{\sigma_X}$.*

Proof (sketch). The intuition at the base of the proof of this property is that, by linking U_1 to variable and literal features, these cannot be flipped to their overlined values once U_1 is $\overline{u_1}$. Therefore, distinct literal features, attached to the same features V_i^T and V_i^F, cannot be flipped to their overlined values according to contrasting values of V_i^T and V_i^F. Details of the proof are at page 133. □

Corollary 5.3. *Let $\phi(X)$ be a Boolean formula in 3CNF, and let α and $\overline{\beta}$ be two outcomes of $\mathcal{O}_{F_s(\phi)}$ such that in α the values of all features are non-overlined, and in $\overline{\beta}$ the values of only U_1 and of U_2 are overlined. Then:*

- *$\phi(X)$ is satisfiable if and only if $\overline{\beta} \succ_{F_s(\phi)} \alpha$, and*
- *$\phi(X)$ is unsatisfiable if and only if $\overline{\beta} \bowtie_{F_s(\phi)} \alpha$.*

Proof. Observe that, when the empty assignment σ_X is considered, outcome α_{σ_X} of the statement of Lemma 5.2 coincides with outcome α of the statement of this corollary. Moreover, any assignment over X is an extension of the empty one, and hence Lemma 5.2 applies. □

5.2. Complexity of majority dominance on mCP-nets

First, we analyze the problem of deciding majority dominance on CP-nets, which is shown NP-complete. More formally, consider the following problem.

Problem:	Majority-Dominance
Instance:	An mCP-net $\mathcal{M}$, and two outcomes $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$.
Question:	Is $\beta \succ_{\mathcal{M}}^{maj} \alpha$?

The following result shows that, for preference profiles represented via an NP-representation scheme, deciding majority dominance is in NP.

Theorem 5.4. *Let $\mathscr{P}$ be a preference profile defined over the same combinatorial domain and represented via an NP-representation scheme, and let α and β be two outcomes. Then, deciding whether $\beta \succ^{maj}_{\mathscr{P}} \alpha$ is in NP.*

Proof. Let $\mathscr{P} = \langle \mathcal{R}_1, \dots, \mathcal{R}_m \rangle$. Observe first that, since $|S^{\succ}_{\mathscr{P}}(\beta,\alpha)| + |S^{\prec}_{\mathscr{P}}(\beta,\alpha)| + |S^{\bowtie}_{\mathscr{P}}(\beta,\alpha)| = m$, it holds that $|S^{\succ}_{\mathscr{P}}(\beta,\alpha)| > |S^{\prec}_{\mathscr{P}}(\beta,\alpha)| + |S^{\bowtie}_{\mathscr{P}}(\beta,\alpha)|$ if and only if $|S^{\succ}_{\mathscr{P}}(\beta,\alpha)| > \lfloor \frac{m}{2} \rfloor$. If $\beta \succ^{maj}_{\mathscr{P}} \alpha$, then, for more than half of the agents i, $\beta \succ_{\mathcal{R}_i} \alpha$. For such agents, since the preferences are represented via an NP-representation scheme, there is a polynomial witness that they prefer β to α. Therefore, to show that $\beta \succ^{maj}_{\mathscr{P}} \alpha$, it suffices to guess a set S of players preferring β to α, along with the polynomial witness of their preference, and then check that $|S| > \lfloor \frac{m}{2} \rfloor$ and that the witnesses are valid. The overall guess requires only polynomial space, and it can be checked in polynomial time. □

Observe that, on 1CP-nets, majority dominance is equivalent to dominance on (simple) CP-nets. Therefore, the following result, which follows directly from Theorem 3.3, states that on *m*CP-nets deciding majority dominance is NP-hard.

Theorem 5.5. *Let $\mathcal{M}$ be an mCP-net, and let $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$ be two outcomes. Then, deciding whether $\beta \succ^{maj}_{\mathcal{M}} \alpha$ is NP-hard. Hardness holds even on classes of singly connected acyclic binary mCP-nets with indegree at most three and at most one agent.*

By combining the two above results, we immediately obtain that deciding majority dominance over polynomially connected classes of acyclic *m*CP-nets is NP-complete.

Corollary 5.6. *Let $\mathcal{C}$ be a polynomially connected class of acyclic mCP-nets. Let $\mathcal{M} \in \mathcal{C}$ be an mCP-net, and let $\alpha, \beta \in \mathcal{O}_{\mathcal{M}}$ be two outcomes. Then, deciding whether $\beta \succ^{maj}_{\mathcal{M}} \alpha$ is NP-complete.*

5.3. Complexity of majority optimality on mCP-nets

Here, we analyze the problems on majority optimal outcomes. In particular, the problems considered are deciding whether an outcome is majority optimal, and deciding whether an *m*CP-net has a majority optimal outcome. We first focus on deciding majority optimality of outcomes in *m*CP-nets. We show that this problem is co-NP-complete. More formally, consider the following problem.

Problem:	Is-Majority-Optimal
Instance:	An *m*CP-net $\mathcal{M}$, and an outcome $\alpha \in \mathcal{O}_{\mathcal{M}}$.
Question:	Is α majority optimal in $\mathcal{M}$?

The following theorem shows that, on preference profiles represented via an NP-representation scheme, deciding whether an outcome is majority optimal is in co-NP.

Theorem 5.7. *Let $\mathscr{P}$ be a preference profile defined over the same combinatorial domain and represented via an NP-representation scheme, and let α be an outcome. Then, deciding whether α is majority optimal in $\mathscr{P}$ is in co-NP.*

Proof. We show that disproving α being majority optimal is in NP. If α is not majority optimal in $\mathscr{P}$, then there is an outcome β such that $\beta \succ^{maj}_{\mathscr{P}} \alpha$. Therefore, we can guess such an outcome β along with the witness that $\beta \succ^{maj}_{\mathscr{P}} \alpha$ (i.e., the set of agents preferring β to α). This guess requires only polynomial space, and can be checked in polynomial time (see the proof of Theorem 5.4). □

Observe that, on 2CP-nets, majority dominance and Pareto dominance are equivalent. Therefore, the following result, which follows directly from Theorem 4.6, shows that on *m*CP-nets deciding majority optimality is co-NP-hard.

Theorem 5.8. *Let $\mathcal{M}$ be an mCP-net, and let $\alpha \in \mathcal{O}_{\mathcal{M}}$ be an outcome. Then, deciding whether α is majority optimal is co-NP-hard. Hardness holds even on polynomially connected classes of acyclic binary mCP-nets with indegree at most three and at most two agents.*

By combining the two previous results, it follows immediately that deciding the majority optimality of an outcome over polynomially connected classes of acyclic *m*CP-nets is co-NP-complete.

Corollary 5.9. *Let $\mathcal{C}$ be a polynomially connected class of acyclic mCP-nets. Let $\mathcal{M} \in \mathcal{C}$ be an mCP-net, and let $\alpha \in \mathcal{O}_{\mathcal{M}}$ be an outcome. Then, deciding whether α is majority optimal is co-NP-complete.*

We now focus on the problem of deciding the existence of majority optimal outcomes in *m*CP-nets. We show that this problem is Σ^{P}_2-complete. More formally, consider the following problem.

Problem: Exists-Majority-Optimal
Instance: An mCP-net $\mathcal{M}$.
Question: Does $\mathcal{M}$ have a majority optimal outcome?

The following theorem shows that deciding whether a preference profile represented via an NP-representation scheme has a majority optimal outcome is in Σ_2^{P}.

Theorem 5.10. *Let $\mathscr{P}$ be a preference profile defined over the same combinatorial domain and represented via an NP-representation scheme. Then, deciding whether $\mathscr{P}$ has a majority optimal outcome is in Σ_2^{P}.*

Proof. To show that $\mathscr{P}$ has a majority optimal outcome, it suffices to guess an outcome α and then check that α is actually majority optimal. Observe that guessing α requires an NP machine, and the final check can be carried out in co-NP (see Theorem 5.7), which can be carried out by an oracle. Therefore, the overall procedure is in Σ_2^{P}. □

To prove the Σ_2^{P}-hardness of Exists-Majority-Optimal, we use a reduction from $\mathrm{QBF}_{2,\forall,\neg}^{CNF}$. Consider the following construction. Let $\Phi = (\exists X)(\forall Y)\neg\phi(X,Y)$ be a quantified formula where $\phi(X,Y)$ is a 3CNF Boolean formula defined over two disjoint sets $X = \{x_1, \ldots, x_{n_X}\}$ and $Y = \{y_1, \ldots, y_{n_Y}\}$ of Boolean variables, and whose set of clauses is $C = \{c_1, \ldots, c_m\}$. From Φ, we define the 6CP-net $\mathcal{M}_{\mathrm{eml}}(\Phi) = \langle N_1^{\mathrm{eml}}, \ldots, N_6^{\mathrm{eml}} \rangle$ as follows (intuitions on the aims of the constructions of the CP-nets of $\mathcal{M}_{\mathrm{eml}}(\Phi)$ will be given in the proof sketch of Lemma 5.11 below).

The features of $\mathcal{M}_{\mathrm{eml}}(\Phi)$ are:

- all the features of a net $F_s(\phi)$ (defined in Section 5.1) 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 $F_s(\phi)$); for further reference, we call A the apex of the interconnecting net;
- all the features of set $\mathcal{V}' = \{V_i' \mid x_i \in X\}$;
- all the features of the set $\mathcal{B}$ which are the features B_i of a disjunctive interconnecting net $H_{\mathrm{D}}(|\mathcal{V}'| + |\mathcal{W}| + |\mathcal{P}| + |\mathcal{D}| + |\mathcal{A}|)$ and its apex is feature B (observe that 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 $\mathcal{M}_{\mathrm{eml}}(\Phi)$ are: $\mathcal{V} \cup \mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B} \cup \{U_1, U_2\}$.
The CP-nets of $\mathcal{M}_{\mathrm{eml}}(\Phi)$ are:

- N_1^{eml} is composed by a net $F_s(\phi)$ with its features, links, CP tables, and a direct net $D(\gamma)$ (see Section 5.1), where γ is defined over the set of features $\mathcal{V}' \cup \mathcal{B}$ and assigns non-overlined values to all of them.
- N_2^{eml} is similar to net N_1^{eml}, with the only differences that features U_1 and U_2 are exchanged, and the CP tables of N_2^{eml} are adjusted to reflect this change.
- $N_3^{\mathrm{eml}} = \langle \mathcal{F}_{N_3^{\mathrm{eml}}}, \mathcal{E}_{N_3^{\mathrm{eml}}} \rangle$ is as follows (see Fig. 12 for a schematic representation of the links). Links of N_3^{eml} are the following:
  - for each $x_i \in X$, $\{\{V_i^T, V_i'\}, \{V_i^F, V_i'\}\} \subset \mathcal{E}_{N_3^{\mathrm{eml}}}$;
  - a disjunctive interconnecting $H_{\mathrm{C}}(|\mathcal{V}'| + |\mathcal{W}| + |\mathcal{P}| + |\mathcal{D}| + |\mathcal{A}|)$ over feature set $\mathcal{B}$ which is connected to the features in $\mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A}$;
  - $\{\{B, U_1\}, \{U_1, U_2\}\} \subset \mathcal{E}_{N_3^{\mathrm{eml}}}$.

  CP tables of N_3^{eml} are the following:
  - features $F \in (\mathcal{V} \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A})$ have the CP tables $\boxed{f \succ \overline{f}}$;
  - for each variable $x_i \in X$, feature $V_i' \in \mathcal{V}'$ has the CP table

$\overline{v_i^T} \wedge \overline{v_i^F}$	$\overline{v_i'} \succ v_i'$
else	$v_i' \succ \overline{v_i'}$

    ;
  - features in $\mathcal{B}$ of the interconnecting net have the usual CP tables;
  - U_1 has the CP table

$\overline{b}$	$\overline{u_1} \succ u_1$
b	$u_1 \succ \overline{u_1}$

    ;
  - U_2 has the CP table

$\overline{u_1}$	$\overline{u_2} \succ u_2$
u_1	$u_2 \succ \overline{u_2}$

    .

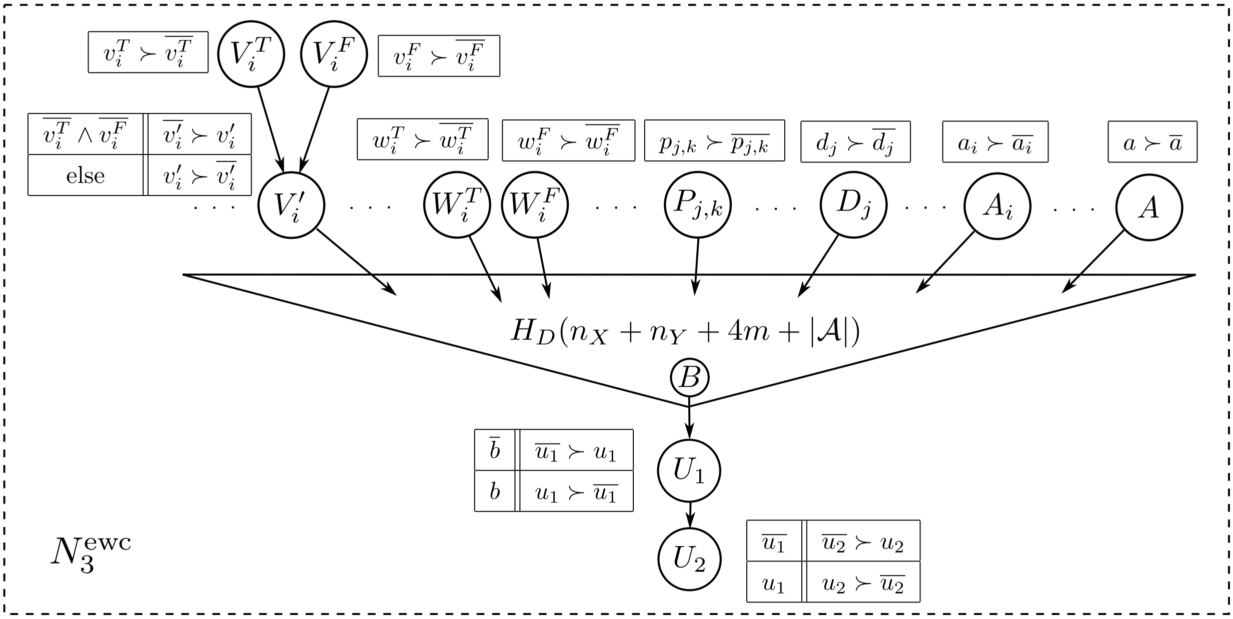

$v_i^T \succ \overline{v_i^T}$
V_i^T
V_i^F
$v_i^F \succ \overline{v_i^F}$
$\overline{v_i^T} \wedge \overline{v_i^F}$ | $\overline{v_i'} \succ v_i'$
else | $v_i' \succ \overline{v_i'}$
$w_i^T \succ \overline{w_i^T}$
$w_i^F \succ \overline{w_i^F}$
$p_{j,k} \succ \overline{p_{j,k}}$
$d_j \succ \overline{d_j}$
$a_i \succ \overline{a_i}$
$a \succ \overline{a}$
V_i'
W_i^T
W_i^F
$P_{j,k}$
D_j
A_i
A
$H_D(n_X + n_Y + 4m + |\mathcal{A}|)$
B
$\overline{b}$ | $\overline{u_1} \succ u_1$
b | $u_1 \succ \overline{u_1}$
U_1
U_2
$\overline{u_1}$ | $\overline{u_2} \succ u_2$
u_1 | $u_2 \succ \overline{u_2}$
N_3^{ewc}

Fig. 12. A schematic illustration of net N_3^{eml} of $\mathcal{M}_{\mathrm{eml}}(\Phi)$. The direct net embedded in N_3^{eml} is not reported in the figure.

- N_4^{eml} is equal to net N_3^{eml};
- N_5^{eml} is composed by a link from U_1 to U_2, and a direct net $D(\gamma)$, where γ is defined over all features but U_1 and U_2, and assigns non-overlined values to all of them. The other CP tables of N_5^{eml} are:
  - feature U_1 has the CP table $\boxed{\overline{u_1} \succ u_1}$;
  - feature U_2 has the CP table

$\overline{u_1}$	$\overline{u_2} \succ u_2$
u_1	$u_2 \succ \overline{u_2}$

.

- N_6^{eml} is characterized by having a link from feature U_2 to any other feature. The CP tables of N_6^{eml} are:
  - feature U_2 has the CP table $\boxed{\overline{u_2} \succ u_2}$;
  - features F different from U_2 have the CP table

$\overline{u_2}$	$f \succ \overline{f}$
u_2	$\overline{f} \succ f$

.

Observe that $\mathcal{M}_{\mathrm{eml}}(\Phi)$ is acyclic, binary, its indegree is three, and can be computed in polynomial time from ϕ. Moreover, the class of mCP-nets $\{\mathcal{M}_{\mathrm{eml}}(\Phi)\}_\Phi$ derived from formulas Φ of the specified kind and according to the reduction shown above is polynomially connected. It is possible to show that $\Phi = (\exists X)(\forall Y)\neg\phi(X, Y)$ is valid if and only if $\mathcal{M}_{\mathrm{eml}}(\Phi)$ has a majority optimal outcome.

Lemma 5.11. *Let $\Phi = (\exists X)(\forall Y)\neg\phi(X, Y)$ be a quantified Boolean formula, where $\phi(X, Y)$ is a 3CNF Boolean formula, defined over two disjoint sets X and Y of Boolean variables. Then, Φ is valid if and only if $\mathcal{M}_{\mathrm{eml}}(\Phi)$ has a majority optimal outcome.*

Proof (sketch). The intuition at the base of the proof is to put in relationship truth assignments over the variable set X with outcomes of $\mathcal{M}_{\mathrm{eml}}(\Phi)$. In particular, given an assignment σ_X over X, the associated outcome is β_{σ_X}, where σ_X is encoded over the feature set $\mathcal{V}$ in the usual way, and all other features have non-overlined values. We show first that outcomes β not in the form of a β_{σ_X} are not majority optimal. This is achieved by the specific construction of nets N_3^{eml}, N_4^{eml}, and N_5^{eml}, which support such outcomes β not in the form of a β_{σ_X} to be dominated by outcomes in which features of $\mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B}$ are flipped to their non-overlined values. To have majority dominance in a 6CP-net, we need four agents preferring an outcome to another. Hence, to achieve the above goal, the preferences of nets N_3^{eml}, N_4^{eml}, and N_5^{eml}, are assisted by the preferences of nets N_1^{eml}, N_2^{eml}, and N_6^{eml}, depending on the specific values of features in $\mathcal{V} \cup \{U_1, U_2\}$ in the aforementioned outcome β. Next, we show that if all assignments σ_X are such $(\forall Y)\neg\phi(X/\sigma_X, Y)$ is not valid (i.e., it holds that $(\exists Y)\phi(X/\sigma_X, Y)$, which means that $\phi(X/\sigma_X, Y)$ is satisfiable), then each outcome of the kind β_{σ_X} is not majority optimal. This is obtained by showing that such outcomes β_{σ_X} are majority dominated by the outcome $\overline{\alpha}$, which assigns overlined values only to U_1 and U_2. The fact that $\overline{\alpha}$ majority dominates β_{σ_X} is proven by exploiting the preference structure of nets N_1^{eml} and N_2^{eml}, that encodes the satisfiability of $\phi(X,Y)$, and the preference structure of nets N_5^{eml} and N_6^{eml}, that supports the preference of $\overline{\alpha}$ over many different outcomes. To conclude, we show that if there is an assignment σ_X such that $(\forall Y)\neg\phi(X/\sigma_X,Y)$ is valid (i.e., $\phi(X/\sigma_X,Y)$ is not satisfiable), then β_{σ_X} is majority optimal, i.e., β_{σ_X} is not be majority dominated. We achieve this by showing that:

- β_{σ_X} is not dominated by any outcome differing from β_{σ_X} on features in $\mathcal{W}\cup\mathcal{P}\cup\mathcal{D}\cup\mathcal{A}$, because nets N_3^{eml}, N_4^{eml}, and N_5^{eml}, vote against it, because they prefer outcomes like β_{σ_X} with non-overlined values for features in $\mathcal{W}\cup\mathcal{P}\cup\mathcal{D}\cup\mathcal{A}$;
- β_{σ_X} is not dominated by any outcome differing from β_{σ_X} on features in $\mathcal{V}'\cup\mathcal{B}$, because nets N_1^{eml}, N_2^{eml}, and N_5^{eml}, vote against it, because they prefer outcomes like β_{σ_X} with non-overlined values for features in $\mathcal{V}'\cup\mathcal{B}$;
- β_{σ_X} is not dominated by any outcome differing from β_{σ_X} on features U_1 and U_2, because nets N_3^{eml}, N_4^{eml}, and N_1^{eml} or N_2^{eml}, vote against it. In particular, N_3^{eml} and N_4^{eml} prefer outcomes like β_{σ_X} with non-overlined values for U_1 and U_2, when the other features have non-overlined values. Nets N_1^{eml} or N_2^{eml}, when $\phi(X/\sigma_X,Y)$ is not satisfiable, do not prefer to β_{σ_X} any other outcome with values for U_1 and U_2 different from u_1u_2;
- β_{σ_X} is not dominated by any outcome differing from β_{σ_X} on features in $\mathcal{V}$, because either nets N_3^{eml}, N_4^{eml}, and N_5^{eml}, or nets N_1^{eml}, N_2^{eml}, and N_6^{eml}, depending on the specific case considered, vote against it. In particular, nets N_3^{eml}, N_4^{eml}, and N_5^{eml} are against the flipping of features $\mathcal{V}$ from their non-overlined value to their overlined value. Nets N_1^{eml}, N_2^{eml}, and N_6^{eml}, when features U_1 and U_2 have values u_1u_2 like in β_{σ_X}, are against the flipping of features $\mathcal{V}$ from their overlined value to their non-overlined value.

Details of the proof are given at page 135. □

We now prove that deciding whether an mCP-net has a majority optimal outcome is Σ_2^{P}-hard.

Theorem 5.12. *Let $\mathcal{M}$ be an mCP-net. Then, deciding whether there is a majority optimal outcome in $\mathcal{M}$ is Σ_2^{P}-hard. Hardness holds even on polynomially connected classes of acyclic binary mCP-nets with indegree at most three and at most six agents.*

Proof. We prove the hardness of Exists-Majority-Optimal by showing a reduction from QBF$^{CNF}_{2,\forall,\neg}$. Let $\Phi=(\exists X)(\forall Y)\neg\phi(X,Y)$ be an instance of QBF$^{CNF}_{2,\forall,\neg}$, and consider the 6CP-net $\mathcal{M}_{\mathrm{eml}}(\Phi)$. By Lemma 5.11, Φ is a "yes"-instance of QBF$^{CNF}_{2,\forall,\neg}$ if and only if there is majority optimal outcome in $\mathcal{M}_{\mathrm{eml}}(\Phi)$. □

By combining the two above results, we immediately conclude that deciding the existence of majority optimal outcomes over polynomially connected classes of acyclic mCP-nets is Σ_2^{P}-complete.

Corollary 5.13. *Let $\mathcal{C}$ be a polynomially connected class of acyclic mCP-nets. Let $\mathcal{M}\in\mathcal{C}$ be an mCP-net. Then, deciding whether there is a majority optimal outcome in $\mathcal{M}$ is Σ_2^{P}-complete.*

5.4. Complexity of majority optimums on mCP-nets

We now focus on majority optimum outcomes. In particular, the problems analyzed are deciding whether an outcome is majority optimum, and deciding whether an mCP-net has a majority optimum outcome. We first consider the problem of deciding whether an outcome is majority optimum in an mCP-net. We prove that this problem is Π_2^{P}-complete. More formally, consider the following problem.

Problem: Is-Majority-Optimum
Instance: An mCP-net $\mathcal{M}$, and an outcome $\alpha\in\mathcal{O}_{\mathcal{M}}$.
Question: Is α majority optimum in $\mathcal{M}$?

The following result shows that, on preference profiles represented via an NP-representations scheme, deciding whether an outcome is majority optimum is in Π_2^{P}.

Theorem 5.14. *Let $\mathscr{P}$ be a preference profile defined over the same combinatorial domain and represented via an NP-representation scheme, and let α be an outcome. Deciding whether α is majority optimum in $\mathscr{P}$ is in Π_2^{P}.*

Proof. We prove the statement by showing that deciding whether α is *not* majority optimum in $\mathscr{P}$ is in Σ_2^{P}. If α is majority optimum, then there is an outcome β such that $\alpha\not\succ^{maj}_{\mathscr{P}}\beta$. Therefore, in order to prove that α is not majority optimum, it suffices to guess β, and then check that $\alpha\not\succ^{maj}_{\mathscr{P}}\beta$. Observe that guessing β requires an NP machine, and then checking $\alpha\not\succ^{maj}_{\mathscr{P}}\beta$ is in co-NP (see Theorem 5.4), which can be carried out by an oracle. Therefore, the overall procedure is in Σ_2^{P}. □

To prove the Π_2^{P}-hardness of Is-Majority-Optimum, we use a reduction from $\mathrm{QBF}^{CNF}_{2,\forall,\neg}$. Consider the following construction. Let $\Phi = (\exists X)(\forall Y)\neg\phi(X,Y)$ 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 $\phi(X,Y)$, we define the 3CP-net $\mathcal{M}_{\mathrm{imm}}(\Phi) = \langle N_1^{\mathrm{imm}}, N_2^{\mathrm{imm}}, N_3^{\mathrm{imm}}\rangle$ in the following way (intuitions on the aims of the constructions of the CP-nets of $\mathcal{M}_{\mathrm{imm}}(\Phi)$ are given in the proof sketch of Lemma 5.15 below).

The features of $\mathcal{M}_{\mathrm{imm}}(\Phi)$ are:

- all the features of a net $F_s(\phi)$ (defined in Section 5.1) 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 set $\mathcal{V}' = \{V_i' \mid x_i \in X\}$;
- all the features of the set $\mathcal{B}$ which are the features B_i of a disjunctive interconnecting net (defined in Section 4.1) $H_{\mathrm{D}}(|\mathcal{V}'| + |\mathcal{W}| + |\mathcal{P}| + |\mathcal{D}| + |\mathcal{A}|)$ and its apex is feature B (observe that 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 $\mathcal{M}_{\mathrm{imm}}(\Phi)$ are: $\mathcal{V} \cup \mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B} \cup \{U_1, U_2\}$.

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

- N_1^{imm} is composed by a net $F_s(\phi)$ with its features, links, and CP tables, and a direct net $D(\gamma)$ (see Section 5.1), where γ is defined over features in $\mathcal{V}' \cup \mathcal{B}$ and assigns non-overlined values to all of them.
- $N_2^{\mathrm{imm}} = D(\overline{\alpha})$, with $\overline{\alpha}$ defined over all the features of $\mathcal{M}_{\mathrm{imm}}(\Phi)$, and having overlined values only for features U_1 and U_2.
- $N_3^{\mathrm{imm}} = \langle \mathcal{F}_{N_3^{\mathrm{imm}}}, \mathcal{E}_{N_3^{\mathrm{imm}}}\rangle$ is as follows (see Fig. 12 for a schematic representation of the links of N_3^{eml}, which is equivalent to N_3^{imm}). Links of N_3^{imm} are the following:
  - for each $x_i \in X$, $\{\{V_i^T, V_i'\}, \{V_i^F, V_i'\}\} \subset \mathcal{E}_{N_3^{\mathrm{imm}}}$;
  - a disjunctive interconnecting $H_{\mathrm{C}}(|\mathcal{V}'| + |\mathcal{W}| + |\mathcal{P}| + |\mathcal{D}| + |\mathcal{A}|)$ over feature set $\mathcal{B}$ which is connected to the features in $\mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A}$;
  - $\{\{B, U_1\}, \{U_1, U_2\}\} \subset \mathcal{E}_{N_3^{\mathrm{imm}}}$.

  CP tables of N_3^{imm} are the following:
  - features $F \in (\mathcal{V} \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A})$ have the CP tables $\boxed{f \succ \overline{f}}$;
  - for each variable $x_i \in X$, feature $V_i' \in \mathcal{V}'$ has the CP table

$\overline{v_i^T} \wedge \overline{v_i^F}$	$\overline{v_i'} \succ v_i'$
else	$v_i' \succ \overline{v_i'}$

;

  - features in $\mathcal{B}$ of the interconnecting net have the usual CP tables;
  - U_1 has the CP table

$\overline{b}$	$\overline{u_1} \succ u_1$
b	$u_1 \succ \overline{u_1}$

;

  - U_2 has the CP table

$\overline{u_1}$	$\overline{u_2} \succ u_2$
u_1	$u_2 \succ \overline{u_2}$

.

Observe that $\mathcal{M}_{\mathrm{imm}}(\Phi)$ is acyclic, binary, its indegree is three, and can be computed in polynomial time from ϕ. Moreover, the class of mCP-nets $\{\mathcal{M}_{\mathrm{imm}}(\Phi)\}_\Phi$ derived from formulas Φ of the specified kind and according to the reduction shown above is polynomially connected. The following result shows that $\Phi = (\exists X)(\forall Y)\neg\phi(X,Y)$ is valid if and only if a particular outcome of $\mathcal{M}_{\mathrm{imm}}(\Phi)$ is majority optimum.

Lemma 5.15. *Let $\Phi = (\exists X)(\forall Y)\neg\phi(X,Y)$ be a quantified Boolean formula, where $\phi(X,Y)$ is a 3CNF Boolean formula, defined over two disjoint sets X and Y of Boolean variables. Then, Φ is valid if and only if $\mathcal{M}_{\mathrm{imm}}(\Phi)$ does not have a majority optimum outcome. In particular, when $\mathcal{M}_{\mathrm{imm}}(\Phi)$ has a majority optimum outcome it is the outcome $\overline{\alpha} \in \mathcal{O}_{\mathcal{M}_{\mathrm{imm}}(\Phi)}$ assigning overlined values only to features U_1 and U_2.*

Proof (sketch). The intuition at the base of the proof is to put in relationship truth assignments over the variable set X with outcomes of $\mathcal{M}_{\mathrm{imm}}(\Phi)$. In particular, given an assignment σ_X over X, the associated outcome is β_{σ_X}, where σ_X is encoded over the feature set $\mathcal{V}$ in the usual way, and all other features have non-overlined values. We show first that $\overline{\alpha}$ majority dominates any other outcome β that is not in the form of a β_{σ_X} outcome, and hence none of them is majority optimum. Intuitively, we achieve by the specific preference structure of nets N_2^{imm} and N_3^{imm}. Indeed, N_2^{imm} always prefers

$\overline{\alpha}$, because N_2^{imm} is a direct net purposefully designed in this way. Net N_3^{imm} prefers outcomes having non-overlined values for features $\mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B} \cup$. In some cases, we need the preference structure of N_1^{imm} along with N_2^{imm} to show that $\overline{\alpha}$ is majority preferred to outcomes that are not in the form of a β_{σ_X}. Next, we partition the outcomes of the kind β_{σ_X} in those with σ_X such that $(\forall Y)\neg\phi(X/\sigma_X, Y)$ is not valid (i.e., it holds that $(\exists Y)\phi(X/\sigma_X, Y)$, which means that $\phi(X/\sigma_X, Y)$ is satisfiable), and those with σ_X such that $(\forall Y)\neg\phi(X/\sigma_X, Y)$ is valid (i.e., $\phi(X/\sigma_X, Y)$ is not satisfiable). For the outcomes β_{σ_X} with σ_X such that $\phi(X/\sigma_X, Y)$ is satisfiable, we show that $\overline{\alpha}$ majority dominates them. We prove this by exploiting the fact that N_1^{imm} encodes the satisfiability of $\phi(X, Y)$, and hence, according to N_1^{imm}, $\overline{\alpha}$ is preferred to β_{σ_X}, and the fact that N_2^{imm} prefers $\overline{\alpha}$. For the outcomes β_{σ_X} with σ_X such that $\phi(X/\sigma_X, Y)$ is not satisfiable, we show that $\overline{\alpha}$ does not majority dominate them and is not majority dominated by them. We achieve this again by exploiting the preference structure of N_1^{imm}, because, when $\phi(X/\sigma_X, Y)$ is not satisfiable, $\overline{\alpha} \bowtie_{N_1^{\text{imm}}} \beta_{\sigma_X}$. Therefore, we obtain that $\overline{\alpha}$ is majority optimum if and only if there are no assignments σ_X such that $(\forall Y)\neg\phi(X/\sigma_X, Y)$ is valid. Details of the proof are given at page 138. □

The next result shows that deciding whether an outcome is majority optimum in an mCP-net is Π_2^{P}-hard.

Theorem 5.16. *Let $\mathcal{M}$ be an mCP-net, and let $\alpha \in \mathcal{O}_{\mathcal{M}}$ be an outcome. Then, deciding whether α is majority optimum in $\mathcal{M}$ is Π_2^{P}-hard. Hardness holds even on polynomially connected classes of acyclic binary mCP-nets with indegree at most three and at most three agents.*

Proof. We prove the statement by showing that deciding whether α is *not* majority optimum is Σ_2^{P}-hard, and we do this by exhibiting a reduction from $\text{QBF}_{2,\forall,\neg}^{CNF}$ to the complement problem to Is-Majority-Optimum. Let $\Phi = (\exists X)(\forall Y)\neg\phi(X, Y)$ be an instance of $\text{QBF}_{2,\forall,\neg}^{CNF}$, and consider the 3CP-net $\mathcal{M}_{\text{imm}}(\Phi)$, and the outcome $\overline{\alpha}$ in which only the values of features U_1 and U_2 are overlined. By Lemma 5.15, Φ is a "yes"-instance of $\text{QBF}_{2,\forall,\neg}^{CNF}$ if and only if α is *not* majority optimum in $\mathcal{M}_{\text{imm}}(\Phi)$. □

Note that, with respect to the number of agents, the above result is optimal. Indeed, since majority dominance and Pareto dominance are equivalent on mCP-nets with $m \le 2$, it is not possible to show the Π_2^{P}-hardness of Is-Majority-Optimum on mCP-nets with $m < 3$.

By combining the two previous results, we immediately conclude that deciding whether an outcome is majority optimum over polynomially connected classes of acyclic mCP-nets is Π_2^{P}-complete.

Corollary 5.17. *Let $\mathcal{C}$ be a polynomially connected class of acyclic mCP-nets. Let $\mathcal{M} \in \mathcal{C}$ be an mCP-net, and let $\alpha \in \mathcal{O}_{\mathcal{M}}$ be an outcome. Then, deciding whether α is majority optimum in $\mathcal{M}$ is Π_2^{P}-complete.*

To conclude, we study the complexity of deciding whether an mCP-net has a majority optimum outcome. We show that the problem is Π_2^{P}-hard and belongs to D_2^{P}. More formally, consider the following problem.

Problem:	Exists-Majority-Optimum
Instance:	An mCP-net $\mathcal{M}$.
Question:	Does $\mathcal{M}$ have a majority optimum outcome?

We prove that deciding whether a preference profile represented via an NP-representation scheme has a majority optimum outcome is in D_2^{P}.

To achieve this result, we carry out the following considerations. First, observe that an outcome to be majority optimum must be also majority optimal. Hence, to individuate majority optimum outcomes, we just need to look for them among the majority optimal ones. To do so, we can select the majority optimal outcomes and exclude those that are not also majority optimum. Notice that, when the majority optimum exists, then it is the only majority optimal outcome. For this reason, a preference profile $\mathscr{P}$ has a majority optimum outcome if and only if $\mathscr{P}$ has majority optimal outcomes and it is *not* true that $\mathscr{P}$ has majority optimal outcomes that are not also majority optimum. Hence, there are two tasks needed to be solved in order to decide whether a profile $\mathscr{P}$ has a majority optimum outcome: (1) decide whether $\mathscr{P}$ has majority optimal outcomes; and (2) decided whether $\mathscr{P}$ does not have majority optimal outcomes that are not also optimum. By combining the complexity of these two tasks, we show that deciding whether a profile $\mathscr{P}$ has a majority optimum outcome is in D_2^{P}.

We already know that the complexity of task (1) is in Σ_2^{P} (see Theorem 5.10). The following lemma shows that the *complement* of task (2) is in Σ_2^{P}.

Lemma 5.18. *Let $\mathscr{P}$ be a preference profile defined over the same combinatorial domain and represented via an NP-representation scheme. Then, deciding whether $\mathscr{P}$ has majority optimal outcomes that are not also majority optimum is in Σ_2^{P}.*

Proof. To solve the task of the lemma, it is sufficient to guess two different outcomes α and β, which can be carried out in NP. Then, through a co-NP oracle call, we check that α is actually majority optimal (see Theorem 5.7), then through another oracle call in co-NP, we check that $\alpha \not\succ^{maj}_{\mathscr{P}} \beta$ (see Theorem 5.4). If the answer to the latter oracle call is "yes", then it means that α is a majority optimal outcome that is not majority optimum (because α does not majority dominate β). To conclude, observe that the overall procedure is in Σ^{P}_2. □

We are now ready to prove that deciding the existence of majority optimum outcomes is in $\mathrm{D}^{\mathrm{P}}_2$.

Theorem 5.19. *Let $\mathscr{P}$ be a preference profile defined over the same combinatorial domain and represented via an NP-representation scheme. Then, deciding whether $\mathscr{P}$ has a majority optimum outcome is in $\mathrm{D}^{\mathrm{P}}_2$.*

Proof. Let $\mathcal{O}$ be a combinatorial domain. Let A be the set of all the preference profiles over $\mathcal{O}$ having at least a majority optimal outcome. Let B be the set of all the preference profiles over $\mathcal{O}$ *not* having majority optimal outcomes that are not also majority optimum (observe that in B there might be profiles with no majority optimal outcome at all, and profiles having a majority optimal outcome that is also majority optimum). The intersection $A \cap B$ is the set of all the preference profiles over $\mathcal{O}$ having a majority optimum outcome: deciding A is the task (1) mentioned above, and deciding B is the task (2) mentioned above.

If we focus only on NP-representations schemes, then set A can be decided in Σ^{P}_2 (see Theorem 5.10), and set B can be decided in Π^{P}_2 (recognizing the complement of B is in Σ^{P}_2, see Lemma 5.18). Therefore, deciding whether $\mathscr{P}$ has a majority optimum outcome is in $\mathrm{D}^{\mathrm{P}}_2$. □

For a lower bound of the problem Exists-Majority-Optimum we can exploit the construction used to show the lower bound of Is-Majority-Optimum, and obtain the following theorem, which proves that on mCP-nets deciding the existence of a majority optimum outcome is Π^{P}_2-hard.

Theorem 5.20. *Let $\mathcal{M}$ be an mCP-net. Then, deciding whether $\mathcal{M}$ has a majority optimum outcome is Π^{P}_2-hard. Hardness holds even on polynomially connected classes of acyclic binary mCP-nets with indegree at most three and at most three agents.*

Proof. We prove the statement by showing a reduction from $\mathrm{QBF}^{CNF}_{2,\forall,\neg}$ to the complement problem to Exists-Majority-Optimum. Let $\Phi = (\exists X)(\forall Y)\neg\phi(X,Y)$ be an instance of $\mathrm{QBF}^{CNF}_{2,\forall,\neg}$, and consider the 3CP-net $\mathcal{M}_{\mathrm{imm}}(\Phi)$. By Lemma 5.15, Φ is a "yes"-instance of $\mathrm{QBF}^{CNF}_{2,\forall,\neg}$ if and only if $\mathcal{M}_{\mathrm{imm}}(\Phi)$ has not a majority optimum outcome. □

By combining the two above results, we immediately conclude that deciding the existence of majority optimum outcomes over polynomially connected classes of acyclic mCP-nets is Π^{P}_2-hard and in $\mathrm{D}^{\mathrm{P}}_2$.

Corollary 5.21. *Let $\mathcal{C}$ be a polynomially connected class of acyclic mCP-nets. Let $\mathcal{M} \in \mathcal{C}$ be an mCP-net. Then, deciding whether $\mathcal{M}$ has a majority optimum outcome is Π^{P}_2-hard and in $\mathrm{D}^{\mathrm{P}}_2$.*

6. Related work

In this section, we describe the larger context of this work in the literature, especially its relationship to previous work on compact representations of preferences, global and sequential voting, and the notion of $\mathcal{O}$-legality.

6.1. Compact representations of preferences

The preferences of agents can be represented in different ways, and preference (representation) models can essentially be divided into *quantitative* and *qualitative* ones [11,29,52]. The former preference models associate with each outcome a numerical value, which is the value of a utility function, and preferences between outcomes are evaluated by comparing the utility values. The latter preference models provide a (not necessarily complete) order over the outcomes, which can be represented in multiple ways, e.g., via a plain sequence of outcomes, or, more generally, via a binary relation over the outcomes, i.e., a set of ordered pairs of outcomes, in which the first outcome of the pair is considered preferred to the second. The key point of qualitative preferences is that there is no precise quantification of the utility associated with the outcomes. For this reason, it has been argued in the literature that qualitative preference are easier to be stated by humans [3,11,16,43], as it is not necessary to estimate the utility values, which can be quite challenging. In this paper, we focus only on qualitative preference models, encoding preference relations.

A preference model is *extensive*, if for any preference relation $\mathcal{R}$ that it is capable to represent, each outcome of the domain of $\mathcal{R}$ appears explicitly at least once in its representation of $\mathcal{R}$. This implies that the size of its representation of $\mathcal{R}$ is at least linear in the number of outcomes of $\mathcal{R}$. So, for example, a sequence of all the outcomes, or listing all the pairs of a preference relation, are both extensive representations. By definition, the number of possible outcomes in combinatorial domains is exponential in the number of features. Hence, the extensive representation of agents' preferences over combinatorial domains becomes quickly infeasible and unrealistic when the number of features is more than just a few. For this reason, compact formalisms to represent combinatorial preferences are needed [51,52,55]. An ideal compact representation scheme for combinatorial preferences would be one such that the space required for the representation is polynomial in the number of the features (and not in the number of the outcomes, like for extensive representations). For information-theoretic reasons, it is not possible to have a preference model to compactly represent *any* possible preference relation over a combinatorial domain. Indeed, for a combinatorial domain characterized by n features of two values each, i.e., *binary features*, the number of all possible (complete) relations is $(2^n)!$, which is $O(2^{2^n})$.[5] Hence, no preference model could ever represent all the possible preference relations using only polynomial space in n [55]. Nonetheless, if we want to represent preference relations showing particular structures or patterns, then it is possible to take advantage of these patterns and "decompose" the relation to summarize it into a concise representation. In the literature, there has been considerable work on exploiting the structure of preferences to appropriately decompose them, and, e.g., Lang [52] gives a survey of different logical languages for compact preference representation (see also the work of Boutilier et al. [11] for more references).

Among the preferences' structural patterns exploited to achieve compact representations, the (conditional) preference independence of the features is one of the most studied (see the works by Boutilier et al. [11], Chevaleyre et al. [22], Keeney and Raiffa [49], Lang [52,53], Lang and Xia [54,55], Xia et al. [84], and references therein). Intuitively, the (conditional) preferential independence of the features implies that the preference relation between outcomes varying on specific features can be influenced by the values of some features and can be totally independent from the values of some other features. When features are (conditionally) preferentially independent, it roughly means that portions of the structure of the preference relation are replicated throughout the preference relation. Therefore, these patterns in the preferences can be "factorised" to save space in the representation.

In the literature, *(conditional) ceteris paribus preference statements* have been proposed several times to compactly represent preferences with (conditional) preferential independencies among (sets of) features [11,52]. Moreover, a preference representation scheme should capture statements that are natural for agents to assess, and (conditional) ceteris paribus preference statements have several times been argued to be intuitive for users, as they resemble the way in which humans express their preferences and act upon them [11,13,44,45].

Ceteris paribus means "all else being equal", and ceteris paribus statements were classically defined to be non-conditional [44]. Given a set of features $\mathcal{V} \subseteq \mathcal{F}$, a non-conditional ceteris paribus preference statement sounds like: "Outcomes varying over $\mathcal{V}$, and all else being equal, are ranked according to the following preference relation restricted over $\mathcal{V}$". For example, a non-conditional ceteris paribus preference statement is "I prefer a round table to a square one, all else being equal" [44]. In this example, the set $\mathcal{V}$ contains the feature "shape" (of the table). Observe that, here, the "all else being equal" does not refer to the fact that the values of the other features determine the preference relation restricted over $\mathcal{V}$. Instead, it means that the given ceteris paribus statement allows to compare outcomes varying only on $\mathcal{V}$: outcomes varying over $\mathcal{V}$ *and* other features cannot be compared via the given statement.

Boutilier et al. [11,13] extended the idea of ceteris paribus statements to conditional ceteris paribus preference statements. In this case, given two disjoint sets of features $\mathcal{V}, \mathcal{Z} \subseteq \mathcal{F}$, a conditional ceteris paribus preference statement sounds like: "Given the specific instantiation γ of values for the features in $\mathcal{Z}$, outcomes varying over $\mathcal{V}$, and all else being equal, are ranked according to the following preference relation restricted over $\mathcal{V}$". For example, a conditional ceteris paribus preference statement could be "Given that the main course of the dinner is meat, I prefer a red wine to a white one, all else being equal". Here, $\mathcal{Z}$ contains the feature "main course", while $\mathcal{V}$ contains the feature "wine".

Among the representation schemes proposed in the literature based on (conditional) ceteris paribus statements, one using propositional logics was proposed by Lang [52], while Boutilier et al. [11,13] proposed CP-nets; for other representations based on (conditional) ceteris paribus statements, see references in the works by Lang [52] and Boutilier et al. [11]. More details on CP-nets are given in the introduction and in the preliminaries.

6.2. Global and sequential voting

Dependencies are certainly a critical characteristic to model, especially to attain compact representations, however, they can become troublesome when combinatorial preferences are aggregated. Whether dependencies are actually problematic or not depends on the specific ways in which agents' votes are collected. These specific ways in which votes are collected are called *voting protocols* [24]. A voting protocol characterizes how a voting rule is implemented, i.e., it determines which information is elicited from the agents, and when this is done. In the literature, a number of different voting protocols for combinatorial vote have been considered (see, e.g., the works by Chevaleyre et al. [22], Lacy and Niou [50], Lang and Xia [55], and references therein). Two of them are global and sequential voting [22,50,53–55,84].

In *global voting*, agents' preferences over the *entire combinatorial domain* are collected at the *same time*, and then a voting rule is applied to the entire preference profile to select the winner(s). In case of (m)CP-nets, global voting consists in the agents communicating their whole CP-nets for vote aggregation. In *sequential voting*, votes are collected feature-by-feature. In particular, agents express at the *same time* their preferences over the *individual* features, and votes for the different features

[5] Allen et al. [4] provide an interesting study on the number of different CP-nets that may exist when various constraints on their structure are considered.

are collected *sequentially* in consecutive steps. In this case, a voting rule determines the winning value v for a feature, and, most importantly, agents are informed about v before the next feature is considered for voting in the protocol. Note that sometimes, in sequential voting, instead of voting for a sequence of single features, agents may be asked in a step of the protocol to express their preferences over the combined values (i.e., the value vectors) of a set of features (see, e.g., the works by Xia et al. [84] and Airiau et al. [1]). Global voting is the protocol of the voting semantics in mCP-nets.

Dependencies are not an issue in global voting, because agents communicate their preferences over the entire (exponential) combinatorial domain (or the entire CP-nets, if CP-nets are used to represent the agents' preferences), and hence all the information needed for the aggregation is available. However, global voting can be expensive to implement and evaluate, especially if extensive representations are adopted or preference relations are extensively unfolded from the compact representation before any further processing. Strictly speaking, from a theoretical perspective the computational complexity of aggregating preferences represented via extensive schemes can be lower than the complexity of performing the same task over compactly represented preferences. However, this computational simplification is artificial, because the computational complexity of a problem is evaluated relative to the size of the input. For extensive representations, the input is huge, and hence the computational complexity of problems over this kind of input can be low. Nevertheless, processing input of huge size can be computationally challenging. The computational burden of global voting can be limited by adopting sequential voting. A benefit of sequential voting over global voting is lowering the communication complexity, i.e., the amount of information needed to be exchanged by the agents to implement the protocol [22,24,55]. In sequential voting, agents are enquired in consecutive steps about their preferences for individual features, therefore the information required to be exchanged is only the preferences "projected" over individual features. However, feature dependencies can be quite detrimental for sequential voting, to the point that sub-optimal outcomes are selected—examples of this phenomenon are called *multiple election paradoxes*—or agents can experience regret after voting [22,50,53–55,84]. Intuitively, paradoxes may occur in sequential voting, because votes over the different features are collected separately, and this can clash with the different individual preferential dependencies that agents may have between features—more specifically, it might be the case that agents have to vote on a feature whose preferences depend on features for which it has not been voted yet [19].

Lacy and Niou [50] showed that these paradoxes in sequential voting can be (partly) avoided if the considered preferences are separable, i.e., they do not have dependencies among features. Intuitively, when represented via CP-nets, combinatorial preferences without dependencies among features do not have any edge between vertices. This is a very strong assumption, and it is unlikely to be met in practice [53,54,84,85]. In fact, in many real situations, preferences exhibit feature preference dependencies, and having the ability to deal with them is important, also because separable preferences are a quite small fraction of combinatorial preferences, and hence their expressivity is quite limited.

6.3. *Overcoming multiple election paradoxes via $\mathcal{O}$-legality*

To overcome the strong limitation imposed by preference separability, Lang [53] proposed and investigated a weaker structural restriction of preferences, called $\mathcal{O}$-legality, which preserves the nice properties of sequential voting (when evaluated over separable preferences) on a wider class of combinatorial preferences. Intuitively, a preference profile is $\mathcal{O}$-legal, if the dependencies among the features for all the agents comply with a common sequence. More formally, if $\mathcal{O} = (F_1, \dots, F_m)$ is a sequence of all the features of the combinatorial domain, a preference profile $\mathcal{P}$ is *$\mathcal{O}$-legal*, if for any agent A and any two features F_i and F_j, if F_i precedes F_j in $\mathcal{O}$, then A's preferences for F_i do not depend on F_j's value [85]. The concept of $\mathcal{O}$-legality has an immediate translation over CP-nets. Indeed, a profile of (preferences represented via) CP-nets is $\mathcal{O}$-legal, if $\mathcal{O}$ is a topological order for all the CP-nets' graphs of the profile. Observe that the existence of a topological order for the features of a CP-net imposes that the graph of the CP-net is acyclic.

The intuitiveness of CP-nets to model combinatorial preferences, together with the convenient characterization of $\mathcal{O}$-legality in CP-nets, has largely encouraged the study of sequential voting in $\mathcal{O}$-legal (and acyclic) CP-nets. In a first group of works, the sequential composition of the voting rule assumed that a feature order was given beforehand (and the voting rule was defined upon the given order) [53,54,84]. Next, this idea was generalized by assuming that the sequential voting rule was not defined given the specific order [85], however, the existence of a shared topological order among the features was anyway required. A study on how well solutions computed via sequential voting approximate the winning outcomes obtained via global voting was presented by Conitzer and Xia [25]. Also various other works considered $\mathcal{O}$-legal CP-nets (see, e.g., the works by Cornelio et al. [27], Grandi et al. [36], Loreggia et al. [60], Maran et al. [65], and Mattei et al. [66]). Among them, an interesting approach to preference aggregation over $\mathcal{O}$-legal CP-nets was proposed by Cornelio et al. [27], where "probabilistic" CP-nets were used to represent the result of the aggregation (see the works by Bigot et al. [8] and Cornelio et al. [26] for more on probabilistic CP-nets).

6.4. *Going beyond $\mathcal{O}$-legality*

However, also $\mathcal{O}$-legality is somewhat demanding, because it imposes that there are no "inversions" in the preference dependencies. For example, if in a profile of CP-nets encoding preferences for a dinner there were an agent whose choice of the main dish influences the choice of the wine and an other agent whose choice of the wine influences the choice of the main dish, then those CP-nets would not be $\mathcal{O}$-legal. (Observe that the profile of CP-nets in Fig. 2 is not $\mathcal{O}$-legal). Hence, also assuming $\mathcal{O}$-legality is in the end quite restrictive [58,75,82]. To go beyond the restrictions imposed by $\mathcal{O}$-legality, different approaches were proposed.

One of these approaches is generalizing the idea of sequential voting. A voting agenda specifies the order in which the features have to be considered in sequential voting. If the voting agenda does not clash with any of the feature dependencies of all the agents, then it is possible to avoid multiple election paradoxes. In particular, if $\mathcal{O}$ is a shared topological order of the features of the CP-nets in a profile, then $\mathcal{O}$ is a voting agenda compatible with the feature dependencies of all the agents. A generalization of this idea is to have a sequence of elections for sets of joint features that cannot be decomposed due to preferential dependencies of some of the agents. In this way, the problem can be shifted to deciding suitable generalized voting agendas that do not clash with the agents' feature dependencies [1].

Another approach to overcome the limitations imposed by $\mathcal{O}$-legality is hypercubewise preference aggregation [82]. This family of voting rules decomposes the preference aggregation task into two phases: first, a (hypercubewise) dominance graph is built by applying local voting rules to set of outcomes differing only for the value of a single specific feature (these outcomes are the neighboring vertices of the induced preference graph of CP-nets, which, in case of binary features, constitute a hypercube—this is where the name of the voting rule comes from); and then the winners are chosen from the hypercubewise dominance graph via choice sets functions, which may select dominating or undominated outcomes in the graph. The idea of the hypercubewise aggregation is at the base of the definition of the hypercubewise Condorcet winner (i.e., the hypercubewise majority winner, which is the outcome majority dominating all its neighbors) [23,82], also called the local Condercet winner, in which the hypercubewise dominance graph is obtained via majority voting. The definition of the hypercubewise Condorcet winner is different from the definition of the standard, or global, Condorcet winner (the majority optimum outcome, in this paper) obtained from global voting [23,82]. Also the dominance relation inferred from the hypercubewise dominance graph is different from the standard majority dominance relation obtained via global voting [58]. Li et al. [58] investigated also other relations between local and global Condorcet winners. It was shown that deciding the existence of hypercubewise Condorcet winners is NP-complete [23,58].

A third approach proposes, similarly to the previous, to define new voting rules that can also be applied over profiles of non-$\mathcal{O}$-legal CP-nets. These new rules select the outcomes minimizing the total value of a loss function evaluated over the profile of CP-nets [74,75].

6.5. Analysis of global voting over CP-nets

Although the approaches mentioned in Section 6.4 can deal with CP-net profiles that are not $\mathcal{O}$-legal, they do not address the complexity analysis of global voting over (not necessarily $\mathcal{O}$-legal) acyclic CP-nets. In fact, global voting over non-$\mathcal{O}$-legal acyclic CP-nets has not received as much attention as sequential voting, although it was explicitly stated in the literature that a theoretical comparison between global and sequential voting would have been highly promising [53].

Rossi et al. [71] carried out the first work studying global voting over (not necessarily $\mathcal{O}$-legal) acyclic CP-nets, when mCP-nets were defined. Recall that the group dominance semantics of mCP-nets is global voting over a profile of CP-nets. Voting schemes over mCP-nets were considered from an algorithmic perspective by Rossi et al. [71], which gave a computational insight for global voting over CP-nets. However, most of the algorithms considered by Rossi et al. [71] were brute-force. Therefore, these algorithms gave only EXPTIME upper bounds for most of the global voting tasks over CP-nets, and no hardness result was provided by Rossi et al. [71].

Algorithms exploiting SAT solvers to compute Pareto and majority optimal outcomes according to global voting over profiles of (not necessarily $\mathcal{O}$-legal) acyclic CP-nets were proposed by Li et al. in [57] and [56], respectively. A SAT solver is used by Li et al. [58] to compute, over profiles of even cyclic (and therefore also non-$\mathcal{O}$-legal) CP-nets, majority optimal and majority optimum outcomes according to global voting, starting from hypercubewise weak Condorcet winners. The approach adopted by Li et al. [58] was subsequently extended by Li et al. [59] to also consider the possibility of multi-valued and incomplete CP-nets.

Despite the mentioned works advanced the study of global voting over (not necessarily $\mathcal{O}$-legal) acyclic CP-nets, they still did not provide precise complexity results. As mentioned in the introduction, the precise complexity of these problems was actually reported as an open problem multiple times in the literature [53,56–59,74]. Our work is the first in the literature tackling directly the complexity analysis of dominance in mCP-nets (and hence the complexity of global voting over CP-nets).

7. Conclusion

In this paper, we have carried out a thorough complexity analysis of the Pareto and majority semantics in mCP-nets. Given the specific definitions of group dominance in mCP-nets, these results also characterize the complexity of Pareto and majority global voting over CP-nets, which was missing and asked for in the literature various times. Unlike what is often assumed in the literature, in this work, we have not restricted the profiles of CP-nets to be $\mathcal{O}$-legal, which makes the results achieved here more general. For the Pareto and the majority voting schemes, we have analyzed the problems of deciding dominance, optimal and optimum outcomes, and the existence of optimal and optimum outcomes. We have shown completeness results for most cases, which means that we have provided tight lower bounds 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. Our hardness results are given for polynomially connected classes of binary acyclic (m)CP-nets. This means that our hardness results extend to classes of (m)CP-nets encompassing the CP-nets here considered, and in particular also to general mCP-nets with partial CP-nets or multi-valued features. The various problems analyzed here have been put at various levels of the polynomial hierarchy, and some of them are even tractable (in P or LOGSPACE), which is quite interesting given that for most of these tasks only EXPTIME upper-bounds were known in the literature.

There are various possible directions for further research. The lower bound for the problem of deciding the existence of majority optimum outcomes does not match the upper bound (Π_2^P-hardness and membership in D_2^P, respectively). Hence, it would be interesting to close this gap and find the precise complexity of the problem. Furthermore, characterizing the complexity of preference aggregation when partial CP-nets are allowed to be part of mCP-nets would be interesting, since with (standard) CP-nets, indifference between outcomes are not allowed. Having constraints on the feasibility of outcomes is another interesting direction of investigation. Without any constraint, CP-nets model agents' preferences when it is assumed that all outcomes are attainable. However, this is not always the case. During the aggregation process, we should take into account what outcomes are feasible. For example, to decide whether an outcome is majority dominated by another, we should check that the latter is actually feasible. It will be interesting studying the case in which constraints are issued over the outcome domain prior the preference aggregation and the case in which constraints are considered after the aggregation. A similar idea characterized the solution concepts in NTU cooperative games defined via constraints [37,39]. 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 [12,69]. Finally, it will also be interesting investigating structural restrictions on the structure of CP-nets, in the spirit of what was done in the works by Brafman et al. [17], Greco et al. [38,40,41], and Ieong and Shoham [46], to identify broader classes of CP-nets where the dominance test is tractable, whereas, in general, over acyclic CP-nets the dominance test is NP-hard.

Acknowledgements

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

Appendix A. Detailed proofs

A.1. Proofs for Section 3

Lemma 3.1. *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 $F(\phi)$ encoding σ_X on the feature set $\mathcal{V}$, and assigning non-overlined values to all other features, and let $\overline{\beta}$ be the outcome assigning overlined values to all and only variable and clause features. Then:*

(1) There is an extension of σ_X to X satisfying $\phi(X)$ if and only if $\overline{\beta} \succ_{F(\phi)} \alpha_{\sigma_X}$.
(2) There is no extension of σ_X to X satisfying $\phi(X)$ if and only if $\overline{\beta} \bowtie_{F(\phi)} \alpha_{\sigma_X}$.

Proof. We first prove (1).

($\Rightarrow$) Assume that there is an extension σ'_X of σ_X to X satisfying $\phi(X)$. To prove that $\overline{\beta} \succ_{F(\phi)} \alpha_{\sigma_X}$ we show that there is an improving flipping sequence from α_{σ_X} to $\overline{\beta}$.
Recall that σ'_X is a complete assignment over X, and that if σ_X is complete, then $\sigma'_X = \sigma_X$. For each variable x_i not defined in σ_X, if $\sigma'_X[x_i] = \texttt{true}$, then we flip feature V_i^T from v_i^T to $\overline{v_i^T}$, analogously if $\sigma'_X[x_i] = \texttt{false}$, then we flip feature V_i^F from v_i^F to $\overline{v_i^F}$.
For all literals $\ell_{j,k}$ evaluating to $\texttt{true}$ in σ'_X, we flip the corresponding literal features $P_{j,k}$ from $p_{j,k}$ to $\overline{p_{j,k}}$.
Since σ'_X is a satisfying assignment, for each clause c_j of ϕ, there is at least a literal $\ell_{j,k}$ evaluating to $\texttt{true}$ in σ'_X. For this reason, given any clause feature D_j, at this point of the flipping sequence, there is at least one literal feature $P_{j,k}$ with value $\overline{p_{j,k}}$, and hence we can flip D_j from d_j to $\overline{d_j}$. We can do this for every clause feature.
Then, we flip to their overlined value all variable features that have not been flipped until now. By the definition of the CP tables of literal features, we can flip all features $P_{j,k}$ having value $\overline{p_{j,k}}$ to $p_{j,k}$, because in the outcome having been built so far through the flips shown above, for all pairs of features (V_i^T, V_i^F), their values are $\overline{v_i^T v_i^F}$.
To conclude, observe that the obtained outcome is exactly $\overline{\beta}$, and hence $\overline{\beta} \succ_{F(\phi)} \alpha_{\sigma_X}$.

($\Leftarrow$) Assume that $\overline{\beta} \succ_{F(\phi)} \alpha_{\sigma_X}$. We show that there is an extension σ'_X of σ_X to X satisfying $\phi(X)$. Since $\overline{\beta} \succ_{F(\phi)} \alpha_{\sigma_X}$, there is an improving flipping sequence $\rho\colon \gamma_0 \to \ldots \to \gamma_z$ from $\gamma_0 = \alpha_{\sigma_X}$ to $\gamma_z = \overline{\beta}$. Consider the truth assignment σ'_X built as follows: If there is an index q such that $\gamma_q[V_i^T V_i^F] = \overline{v_i^T} v_i^F$, then $\sigma'_X[x_i] = \texttt{true}$; and if there is an index q such that $\gamma_q[V_i^T V_i^F] = v_i^T \overline{v_i^F}$, then $\sigma'_X[x_i] = \texttt{false}$.
We first show that σ'_X is consistent, complete, and an extension of σ_X to X. Observe that, by the definition of the CP tables of variable features, and the fact that those features have no parents, once a variable feature is flipped to its

overlined value, it cannot be flipped back. Therefore, it cannot be the case that there are indices q and r for which there are variable features V_i^T and V_i^F such that $\gamma_q[V_i^T V_i^F] = \overline{v_i^T} v_i^F$ and $\gamma_r[V_i^T V_i^F] = v_i^T \overline{v_i^F}$. Hence, σ'_X is consistent. Moreover, we claim that, for any variable x_i, there is always an index q such that $\gamma_q[V_i^T V_i^F] = \overline{v_i^T} v_i^F$ or $\gamma_q[V_i^T V_i^F] = v_i^T \overline{v_i^F}$, which implies that σ'_X is complete. Indeed, if x_i has a value in σ_X, then the index q that we are looking for is $q = 0$ (because either $\gamma_0[V_i^T V_i^F] = \overline{v_i^T} v_i^F$ or $\gamma_0[V_i^T V_i^F] = v_i^T \overline{v_i^F}$, by the definition of $\gamma_0 = \alpha_{\sigma_X}$). Observe that this, along with the consistency of σ'_X proven above, implies that $\sigma'_X[x_i] = \sigma_X[x_i]$ for each variable x_i having a truth value in σ_X. On the other hand, if x_i has not a value in σ_X, then, since $\gamma_0[V_i^T V_i^F] = v_i^T v_i^F$ and $\gamma_z[V_i^T V_i^F] = \overline{v_i^T v_i^F}$, it must be the case that there is an index $q > 0$ such that $\gamma_q[V_i^T V_i^F] = \overline{v_i^T} v_i^F$ or $\gamma_q[V_i^T V_i^F] = v_i^T \overline{v_i^F}$. Hence, σ'_X is complete. To conclude, by all the properties above, σ'_X is also an extension of σ_X to X.
We now prove that σ'_X satisfies $\phi(X)$ by showing that σ'_X satisfies all clauses of $\phi(X)$.
Let c_j be a clause of $\phi(X)$, and consider clause feature D_j. Because $\gamma_0[D_j] = d_j$ and $\gamma_z[D_j] = \overline{d_j}$, there is an index t such that $\gamma_t[D_j] = d_j$, $\gamma_{t+1}[D_j] = \overline{d_j}$, and $\gamma_t \rightarrow_{D_j} \gamma_{t+1}$. Since flipping D_j has to be an improving flip, it must be the case that there is a literal feature $P_{j,k}$ such that $\gamma_t[P_{j,k}] = \overline{p_{j,k}}$. Therefore, since $\gamma_0[P_{j,k}] = p_{j,k}$, there is an index $s < t$ such that $\gamma_s[P_{j,k}] = p_{j,k}$, $\gamma_{s+1}[P_{j,k}] = \overline{p_{j,k}}$, and $\gamma_s \rightarrow_{P_{j,k}} \gamma_{s+1}$. Now there are two cases: either (a) $\ell_{j,k} = x_i$, or (b) $\ell_{j,k} = \neg x_i$. For (a), since flipping $P_{j,k}$ has to be an improving flip, it must be the case that $\gamma_s[V_i^T V_i^F] = \overline{v_i^T} v_i^F$, therefore $\sigma'_X[x_i] = \texttt{true}$, and hence σ'_X satisfies c_j. For (b), again since flipping $P_{j,k}$ has to be an improving flip, it must be the case that $\gamma_s[V_i^T V_i^F] = v_i^T \overline{v_i^F}$, therefore $\sigma'_X[x_i] = \texttt{false}$ and hence σ'_X satisfies c_j.
Therefore, σ'_X satisfies all clauses of ϕ, and hence σ'_X is an extension of σ_X to X satisfying $\phi(X)$.

We now prove (2). We know that $\overline{\beta} \bowtie_{F(\phi)} \alpha_{\sigma_X}$ if and only if $\alpha_{\sigma_X} \not\succ_{F(\phi)} \overline{\beta}$ *and* $\overline{\beta} \not\succ_{F(\phi)} \alpha_{\sigma_X}$. First, observe that $\alpha_{\sigma_X} \not\succ_{F(\phi)} \overline{\beta}$ is always true, as there is no improving flipping sequence from $\overline{\beta}$ to α_{σ_X}. Indeed, the values of variable features in $\overline{\beta}$ cannot be flipped to their non-overlined values according to their CP tables in $F(\phi)$, because they are the most preferred values and variable features do not have parents. So, $\overline{\beta} \bowtie_{F(\phi)} \alpha_{\sigma_X}$ if and only if $\overline{\beta} \not\succ_{F(\phi)} \alpha_{\sigma_X}$. Hence, showing that there is *no* extension of σ_X to X satisfying $\phi(X)$ if and only if $\overline{\beta} \bowtie_{F(\phi)} \alpha_{\sigma_X}$ is equivalent to showing that there is *no* extension of σ_X to X satisfying $\phi(X)$ if and only if $\overline{\beta} \not\succ_{F(\phi)} \alpha_{\sigma_X}$. However, we have already shown this in (1). □

A.2. Proofs for Section 4

Lemma 4.5. *Let $\phi(X)$ be a 3CNF Boolean formula, and let α be the outcome of $\mathcal{M}_{\text{ipo}}(\phi)$ assigning non-overlined values to all features. Then, $\phi(X)$ is satisfiable if and only if α is not Pareto optimal in $\mathcal{M}_{\text{ipo}}(\phi)$.*

Proof. To prove the statement of the lemma, we first show the two following properties.

Property 4.5.(1). *If $\phi(X)$ is unsatisfiable, and $\beta \in \mathcal{O}_{\mathcal{M}_{\text{ipo}}(\phi)}$ is an outcome such that $\beta \neq \alpha$, then $\beta \succ_{N_1^{\text{ipo}}} \alpha$ implies that $\beta \not\succ_{N_2^{\text{ipo}}} \alpha$, and $\beta \succ_{N_2^{\text{ipo}}} \alpha$ implies that $\beta \not\succ_{N_1^{\text{ipo}}} \alpha$.*

Proof. By inspection of the proof of Lemma 3.1, since ϕ is unsatisfiable, there is no improving flipping sequence in N_1^{ipo} that from α arrives to an outcome in which the values of *all* the clause features of the net $F(\phi)^a$ are overlined. For this reason, by the definition of the interconnecting net $H_C(m)$, in N_1^{ipo} there is no improving flipping sequence that from α arrives to an outcome in which the value of the apex of $H_C(m)$ is overlined. This implies, moreover, that in N_1^{ipo} there is no improving flipping sequence that from α arrives to an outcome in which the values of some of the features of the net $F(\phi)^b$ are overlined. So, any improving flipping sequence in N_1^{ipo} from α arrives to outcomes in which values of features of $F(\phi)^a$ are overlined, while values of feature of $F(\phi)^b$ are *non*-overlined.

Symmetrically, since ϕ is unsatisfiable, any improving flipping sequence in N_2^{ipo} from α arrives to outcomes in which values of features of $F(\phi)^a$ are *non*-overlined, while values of feature of $F(\phi)^b$ are overlined.

Now, assume that β is such that $\beta \succ_{N_1^{\text{ipo}}} \alpha$, hence in N_1^{ipo} there is an improving flipping sequence from α to β. From what we have said, β is such that values of features of $F(\phi)^a$ are overlined, while values of feature of $F(\phi)^b$ are *non*-overlined. Therefore, β cannot be reached through an improving flipping sequence in N_2^{ipo}. Thus, $\beta \not\succ_{N_2^{\text{ipo}}} \alpha$.

Symmetrically, it can be shown that if $\beta \succ_{N_2^{\text{ipo}}} \alpha$, then $\beta \not\succ_{N_1^{\text{ipo}}} \alpha$. □

Property 4.5.(2). *If $\phi(X)$ is satisfiable, then outcome β assigning overlined values to all variable and clause features of $\mathcal{M}_{\text{ipo}}(\phi)$ and to all features of $H_C(m)$ is such that $\beta \succ_{N_1^{\text{ipo}}} \alpha$ and $\beta \succ_{N_2^{\text{ipo}}} \alpha$.*

Proof. Since ϕ is satisfiable, by inspection of the proof of Lemma 3.1, there is an improving flipping sequence in N_1^{ipo} that from α arrives to an outcome α_1 in which the values of *all* variable and clause features of $F(\phi)^a$ are overlined. By definition of the interconnecting net $H_C(m)$, in N_1^{ipo} there is also an improving flipping sequence that from α_1 arrives to an outcome α_2 in which the values of all features (and also of the apex) of $H_C(m)$ are overlined. This allows α_2 to be further improved by a flipping sequence to an outcome α_3 in which the values of *all* variable and clause features of $F(\phi)^b$ are overlined, because ϕ is satisfiable. Observe that $\alpha_3 = \beta$.

Symmetrically, it can be shown that in N_2^{ipo} there exists an improving flipping sequence from α to the very same β. Therefore, $\beta \succ_{N_1^{\text{ipo}}} \alpha$ and $\beta \succ_{N_2^{\text{ipo}}} \alpha$. □

The combination of the two properties above entails that $\phi(X)$ is satisfiable if and only if α is not a Pareto optimal outcome of $\mathcal{M}_{\text{ipo}}(\phi)$. □

Lemma 4.9. *Let $\mathcal{M}$ be an acyclic mCP-net. Then, $\mathcal{M}$ has a Pareto optimum outcome if and only if all the individual CP-nets of $\mathcal{M}$ have the very same optimum outcome (that, in which case, is also the Pareto optimum outcome of $\mathcal{M}$).*

Proof. We show first that if $\mathcal{M} = \langle N_1, \dots, N_m \rangle$ has a Pareto optimum outcome α, then all the individual CP-nets have the very same optimum outcome, which is α. By definition, a Pareto optimum outcome is unique and Pareto dominates all other outcomes. This means that, for any outcome $\beta \neq \alpha$, and for all i, $\alpha \succ_{N_i} \beta$. We know that in each individual CP-net the only outcome dominating all the others is the individual optimum. Therefore, α equals all the individual optimum outcomes. For the other direction of the proof, observe that, since all the CP-nets of $\mathcal{M}$ have the very same optimum outcome α, outcome α Pareto dominates all other outcomes, implying that α is Pareto optimum in $\mathcal{M}$. □

Theorem 4.10. *Let $\mathcal{M}$ be an acyclic mCP-net, and let $\alpha \in \mathcal{O}_{\mathcal{M}}$ be an outcome. Deciding whether α is Pareto optimum in $\mathcal{M}$ is in LOGSPACE.*

Proof. From Lemma 4.9, we know that α is Pareto optimum in $\mathcal{M}$ if and only if α equals all the individual optimum outcomes of the individual CP-nets of $\mathcal{M}$. Therefore, in order to check whether α is actually Pareto optimum it suffices to check, for each individual CP-net in turn, whether α is the individual optimal outcome for that agent. If α is different from even one of the individual optimum outcomes, then α is not Pareto optimum. Remember that checking the individual optimality of α is in LOGSPACE (see Theorem 3.7), and hence, by reuse of work space, we can check in LOGSPACE whether α equals all the individual optimum outcomes. □

Theorem 4.10. *Let $\mathcal{M}$ be an acyclic mCP-net. Deciding whether $\mathcal{M}$ has a Pareto optimum outcome is in P.*

Proof. From Lemma 4.9, we know that $\mathcal{M}$ has a Pareto optimum outcome if and only if all the individual CP-nets have the very same individual optimum outcome. Hence, in order to decide whether $\mathcal{M}$ has a Pareto optimum outcome, it suffices to compute the individual optimum outcome of the first agent. We can do this in polynomial time [11]. After this, we compare the just computed individual optimum outcome with all the other individual optimum outcomes. This can be carried out, by reuse of work space, in logarithmic space (see Theorem 4.10), and hence in polynomial time (by the inclusion LOGSPACE $\subseteq$ P). If all the individual optimum outcomes are equal, then we answer yes, otherwise no. Observe that the overall procedure is in P. □

A.3. Proofs for Section 5

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 $\overline{\beta}[U_1 U_2] = \overline{u_1 u_2}$, 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 $\phi(X)$ if and only if $\overline{\beta} \succ_{F_s(\phi)} \alpha_{\sigma_X}$;
(2) There is no extension of σ_X to X satisfying $\phi(X)$ if and only if $\overline{\beta} \bowtie_{F_s(\phi)} \alpha_{\sigma_X}$.

Proof. We first prove (1).

($\Rightarrow$) Assume that there is an extension σ'_X of σ_X to X satisfying $\phi(X)$. We show that $\overline{\beta} \succ_{F_s(\phi)} \alpha_{\sigma_X}$ by exhibiting an improving flipping sequence from α_{σ_X} to $\overline{\beta}$.

We claim that, by Lemma 3.1, outcome β (which is different from $\overline{\beta}$) assigning overlined values to all variable and clause features is such that $\beta \succ_{F_s(\phi)} \alpha_{\sigma_X}$. Indeed, features in $\mathcal{V} \cup \mathcal{P} \cup \mathcal{D}$ are linked in $F_s(\phi)$ through the very same links of a net $F(\phi)$. Moreover, the value u_1 assigned to feature U_1 in α_{σ_X} selects in the CP tables of features $\mathcal{V} \cup \mathcal{P} \cup \mathcal{D}$ specific preference rankings that are equivalent to those in the CP tables of features $\mathcal{V} \cup \mathcal{P} \cup \mathcal{D}$ in $F(\phi)$. Now, since $\beta \succ_{F_s(\phi)} \alpha_{\sigma_X}$, there is an improving flipping sequence from α_{σ_X} to β. Then, we can flip the values of all features of the interconnecting net $H_C(m)$ (including the apex) and of U_2. Now we flip the value of feature U_1 from u_1 to $\overline{u_1}$. Recall that in β, all variable features have overlined values, and their values have not been flipped after outcome β was reached in the improving flipping sequence. Therefore, since the value of U_1 is $\overline{u_1}$, given the CP table of variable features, we can flip features in $\mathcal{V}$ to any configuration of values (even leaving everything as it is), and in particular we can flip them to match the values of features of $\mathcal{V}$ in $\overline{\beta}$. Next, all literal features can be flipped to their non-overlined values (recall that U_1 has value $\overline{u_1}$ now). Then, we flip all clause features to their non-overlined values, and after this, in the proper order, we can flip the features of the interconnecting net to their non-overlined values. Observe that the obtained outcome is precisely $\overline{\beta}$, and hence $\overline{\beta} \succ_{F_s(\phi)} \alpha_{\sigma_X}$.

($\Leftarrow$) Assume that $\overline{\beta} \succ_{F_s(\phi)} \alpha_{\sigma_X}$. We show that there is an extension σ'_X of σ_X to X satisfying $\phi(X)$. Since $\overline{\beta} \succ_{F_s(\phi)} \alpha_{\sigma_X}$, there is an improving flipping sequence $\rho\colon \gamma_0 \to \ldots \to \gamma_z$ from $\gamma_0 = \alpha_{\sigma_X}$ to $\gamma_z = \overline{\beta}$.

Since $\gamma_0[U_1] = u_1$ and $\gamma_z[U_1] = \overline{u_1}$, there must be an index s in which $\gamma_s[U_1] = u_1$, $\gamma_{s+1}[U_1] = \overline{u_1}$, and $\gamma_s \to_{U_1} \gamma_{s+1}$. Moreover, because U_1 has no parents, in the sequence ρ, feature U_1 can be flipped only once. Therefore, for all $p < s$, $\alpha_p[U_1] = u_1$, and for all $p > s$, $\alpha_p[U_1] = \overline{u_1}$.

We claim that we can assume w.l.o.g. that all variable features have overlined values in γ_s. Indeed, if this is not the case, we can always modify as follows an improving flipping sequence ρ, from α_{σ_X} to $\overline{\beta}$, to obtain an improving flipping sequence ρ', from α_{σ_X} to $\overline{\beta}$, satisfying the required assumption. In particular, consider all variable features F having non-overlined values just before feature U_1 is flipped. We can flip all of them before flipping U_1 to $\overline{u_1}$, and, after having flipped U_1, we can flip them all back to the values they had before. The new sequence satisfies the required assumption, and, moreover, it is still improving, and it is still a sequence from α_{σ_X} to $\overline{\beta}$.

Consider the truth assignment σ'_X built as follows: If there is an index $p < s$ such that $\gamma_p[V_i^T V_i^F] = \overline{v_i^T} v_i^F$, then $\sigma'_X[x_i] = \texttt{true}$; if there is an index $p < s$ such that $\gamma_p[V_i^T V_i^F] = v_i^T \overline{v_i^F}$, then $\sigma'_X[x_i] = \texttt{false}$.

We first show that σ'_X is consistent, complete, and an extension of σ_X to X. Observe that, by the definition of the CP tables of variable features, before the s-th step of the sequence, once a variable feature is flipped to its overlined value, it cannot be flipped back (this may happen only after the s-th step). Therefore, it cannot be the case that there are indices $p < s$ and $q < s$ for which there are variable features V_i^T and V_i^F such that $\gamma_p[V_i^T V_i^F] = \overline{v_i^T} v_i^F$ and $\gamma_q[V_i^T V_i^F] = v_i^T \overline{v_i^F}$. Hence, σ'_X is consistent. Moreover, we claim that, for any variable x_i, there is always an index $p < s$ such that $\gamma_p[V_i^T V_i^F] = \overline{v_i^T} v_i^F$ or $\gamma_p[V_i^T V_i^F] = v_i^T \overline{v_i^F}$, which implies that σ'_X is complete. Indeed, if x_i has a value in σ_X, then the index p that we are looking for is $p = 0$ (because either $\gamma_0[V_i^T V_i^F] = \overline{v_i^T} v_i^F$ or $\gamma_0[V_i^T V_i^F] = v_i^T \overline{v_i^F}$, by the definition of $\gamma_0 = \alpha_{\sigma_X}$). Observe that this and the consistency of σ'_X proven above imply that $\sigma'_X[x_i] = \sigma_x[x_i]$ for each variable x_i having a truth value in σ_X. On the other hand, if x_i has not a value in σ_X, then, since $\gamma_0[V_i^T V_i^F] = v_i^T v_i^F$, and we are assuming that $\gamma_s[V_i^T V_i^F] = \overline{v_i^T v_i^F}$, it must be the case that there is an index $p < s$ such that $\gamma_p[V_i^T V_i^F] = \overline{v_i^T} v_i^F$ or $\gamma_p[V_i^T V_i^F] = v_i^T \overline{v_i^F}$. Hence, σ'_X is complete. To conclude, by all the properties above, σ'_X is an extension of σ_X to X.

We now show that σ'_X satisfies $\phi(X)$ by showing that σ'_X satisfies all the clauses of $\phi(X)$.

Because $\gamma_0[U_2] = u_2$ and $\gamma_z[U_2] = \overline{u_2}$, there must be an index t such that $\gamma_t[U_2] = u_2$, $\gamma_{t+1}[U_2] = \overline{u_2}$, and $\gamma_t \to_{U_2} \gamma_{t+1}$. (Indices s and t are not in any particular relationship; it could be $s < t$ but also $t < s$.) By the definition of the CP tables, for $\gamma_t \to_{U_2} \gamma_{t+1}$ to be an improving flip, it must be the case that, for the apex A of the interconnecting net, $\gamma_t[A] = \overline{a}$. This requires that there is an index $r < t$, for which in γ_r all clause features have their overlined values (because the conjunctive interconnecting net is linked to the set of clause features).

Let c_j be any clause of $\phi(X)$, and consider feature D_j. Because $\gamma_0[D_j] = d_j$ and $\gamma_r[D_j] = \overline{d_j}$, there is an index $q < r$ such that $\gamma_q[D_j] = d_j$, $\gamma_{q+1}[D_j] = \overline{d_j}$, and $\gamma_q \to_{D_j} \gamma_{q+1}$. Since flipping D_j has to be an improving flip, it must be the case that there is a literal feature $P_{j,k}$ such that $\gamma_q[P_{j,k}] = \overline{p_{j,k}}$. Therefore, since $\gamma_0[P_{j,k}] = p_{j,k}$ and $\gamma_q[P_{j,k}] = \overline{p_{j,k}}$, there is an index $p < q$ such that $\gamma_p[P_{j,k}] = p_{j,k}$, $\gamma_{p+1}[P_{j,k}] = \overline{p_{j,k}}$, and $\gamma_p \to_{P_{j,k}} \gamma_{p+1}$. By the definition of the CP table of $P_{j,k}$, it must also be the case that $p < s$. This means that, for all the literal features that in ρ change their value from non-overlined to overlined, their flipping happens before the s-th step. Hence, if two different literal features linked to the same variable features V_i^T and V_i^F flip before the s-th step, then their flipping is based on consistent values assigned to V_i^T and V_i^F.

Now there are two cases: either (a) $\ell_{j,k} = x_i$, or (b) $\ell_{j,k} = \neg x_i$. For (a), since flipping $P_{j,k}$ has to be an improving flip, it must be the case that $\gamma_p[V_i^T V_i^F] = \overline{v_i^T} v_i^F$, therefore $\sigma'_X[x_i] = \texttt{true}$, and hence σ'_X satisfies c_j. For (b), again since flipping $P_{j,k}$ has to be an improving flip, it must be the case that $\gamma_p[V_i^T V_i^F] = v_i^T \overline{v_i^F}$, therefore $\sigma'_X[x_i] = \texttt{false}$, and hence σ'_X satisfies c_j.

Therefore, if $\overline{\beta} \succ_{F_s(\phi)} \alpha_{\sigma_X}$, then there is an extension σ'_X of σ_X to X satisfying $\phi(X)$.

We now prove (2). We know that $\overline{\beta} \bowtie_{F_s(\phi)} \alpha_{\sigma_X}$ if and only if $\alpha_{\sigma_X} \not\succ_{F_s(\phi)} \overline{\beta}$ *and* $\overline{\beta} \not\succ_{F_s(\phi)} \alpha_{\sigma_X}$. First, observe that $\alpha_{\sigma_X} \not\succ_{F_s(\phi)} \overline{\beta}$ is always true, because there is no improving flipping sequence from $\overline{\beta}$ to α_{σ_X}, as the value $\overline{u_1}$ of feature U_1 in $\overline{\beta}$ cannot be flipped in $F_s(\phi)$, because it is the most preferred value of U_1, and U_1 does not have parents. So, $\overline{\beta} \bowtie_{F_s(\phi)} \alpha_{\sigma_X}$ if and only if $\overline{\beta} \not\succ_{F_s(\phi)} \alpha_{\sigma_X}$. Hence, showing that there is *no* extension of σ_X to X satisfying $\phi(X)$ if and only if $\overline{\beta} \bowtie_{F_s(\phi)} \alpha_{\sigma_X}$ is equivalent to showing that there is *no* extension of σ_X to X satisfying $\phi(X)$ if and only if $\overline{\beta} \not\succ_{F_s(\phi)} \alpha_{\sigma_X}$. However, we have already shown this in (1). □

Lemma 5.11. *Let $\Phi = (\exists X)(\forall Y)\neg\phi(X, Y)$ be a quantified Boolean formula, where $\phi(X, Y)$ is a 3CNF Boolean formula, defined over two disjoint sets X and Y of Boolean variables. Then, Φ is valid if and only if $\mathcal{M}_{\text{eml}}(\Phi)$ has a majority optimal outcome.*

Proof. To prove the statement of the lemma, we have to analyze the majority dominance relationships between outcome pairs. To organize this task, we define the following sets of outcomes:

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

The sets O_d and O_c constitute a partition of $\mathcal{O}_{\mathcal{M}_{\text{eml}}(\Phi)}$. On the contrary, $O_d{}'$, $O_d{}''$, and $O_d{}'''$, do not constitute a partition of O_d, because they are not disjoint. We show that only outcomes of a subset S (whose detailed characterization will be given toward the end of the proof) of O_c might be majority optimal. We do so by showing that (1) all outcomes in $(O_d \cup O_c) \setminus S$ are majority dominated by some other outcome, and hence they are not majority optimal; and that (2) all outcomes in S, which might be empty, are not majority dominated, and hence they are majority optimal. Therefore, $\mathcal{M}_{\text{eml}}(\Phi)$ has a majority optimal outcome if and only if S is non-empty.

We recall that, since $\mathcal{M}_{\text{eml}}(\Phi)$ is a 6CP-net, if α and β are two outcomes, $|S^{\succ}_{\mathcal{M}_{\text{eml}}(\Phi)}(\beta, \alpha)| \geq 4$ implies that $\beta \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \alpha$.

Property 5.11.(1). *Let $\beta' \in O_d{}'$ be an outcome. Then, β' is not majority optimal in $\mathcal{M}_{\text{eml}}(\Phi)$.*

Proof. There are two cases: either (1) $\beta'[U_1U_2] \neq u_1u_2$, or (2) $\beta'[U_1U_2] = u_1u_2$:

(1) Let γ be the outcome assigning non-overlined values to all features in $(\mathcal{V} \cup \mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B})$, and such that $\gamma[U_1U_2] = \beta'[U_1U_2]$. We prove that $\gamma \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta'$. Consider net N_3^{eml}. The following is an improving flipping sequence from β' to γ, showing that $\gamma \succ_{N_3^{\text{eml}}} \beta'$. We flip to their non-overlined value, in the following order, all features in $\mathcal{V}$, $\mathcal{V}'$, $\mathcal{W}$, $\mathcal{P}$, $\mathcal{D}$, and $\mathcal{A}$, having an overlined value in β'. Next, we flip in the proper order all features in $\mathcal{B}$ having an overlined value in β' to their non-overlined value. The outcome obtained is precisely γ. Moreover, since $N_4^{\text{eml}} = N_3^{\text{eml}}$, $\gamma \succ_{N_4^{\text{eml}}} \beta'$, as well. Now, observe that by the definition of N_5^{eml}, $\gamma \succ_{N_5^{\text{eml}}} \beta'$. We have seen that there are three agents preferring γ to β'.

Now there are two cases: either (a) $\beta'[U_1] = \overline{u_1}$, or (b) $\beta'[U_1] = u_1$:

(a) Let us focus on N_1^{eml}. The following is an improving flipping sequence from β' to γ in N_1^{eml}, showing that $\gamma \succ_{N_1^{\text{eml}}} \beta'$. Since $\beta'[U_1] = \overline{u_1}$, we can flip to their non-overlined value all features in $\mathcal{V}$ and $\mathcal{W}$ having an overlined value in β'. Next, we flip to their non-overlined value all features in $\mathcal{P}$, $\mathcal{D}$, $\mathcal{A}$ (in the proper order), $\mathcal{V}'$, and $\mathcal{B}$, having an overlined value in β'. The outcome reached is exactly γ. Therefore, in (1)(a), there are four agents preferring γ to β', and hence $\gamma \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta'$.

(b) As we are in (1)(b), $\beta'[U_1] = u_1$ and $\beta'[U_1U_2] \neq u_1u_2$, hence $\beta'[U_1U_2] = u_1\overline{u_2}$. Let us focus on N_2^{eml}. The improving flipping sequence from β' to γ in N_1^{eml} in (1)(a) is also an improving flipping sequence from β' to γ in N_2^{eml} (because $\beta'[U_2] = \overline{u_2}$). Therefore, in (1)(b), there are four agents preferring γ to β', and hence $\gamma \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta'$.

(2) Let γ' be the outcome assigning an overlined value only to U_1. We prove that $\gamma' \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta'$. Consider net N_1^{eml}. By the definition of the net, we can flip U_1 from u_1 to $\overline{u_1}$. Once this is done, by performing the improving flipping sequence from β' to γ in N_1^{eml} exhibited in (1)(a), we reach γ' in this case. Hence, $\gamma' \succ_{N_1^{\text{eml}}} \beta'$. Consider now net N_3^{eml}. Since $\beta' \in O_d{}'$, there is a feature $F \in (\mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B})$ such that $\beta'[F] = \overline{f}$. By the definition of the disjunctive interconnecting net embedded in N_3^{eml}, there is an improving flipping sequence from β' to an outcome in which U_1 has an overlined value (we flip the features in the interconnecting net until we flip its apex, and then we flip U_1 to its overlined value). At this point, by performing the improving flipping sequence from β' to γ in N_3^{eml} shown in (1) we reach γ' in this case. Hence, $\gamma' \succ_{N_3^{\text{eml}}} \beta'$. Since $N_4^{\text{eml}} = N_3^{\text{eml}}$, $\gamma' \succ_{N_4^{\text{eml}}} \beta'$, as well. Finally, by the definition of N_5^{eml}, $\gamma' \succ_{N_5^{\text{eml}}} \beta'$. Therefore, $\gamma' \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta'$, and hence β' is not majority optimal. □

Property 5.11.(2). *Let $\beta'' \in O_d{}''$ be an outcome. Then, β'' is not majority optimal in $\mathcal{M}_{\text{eml}}(\Phi)$.*

Proof. By Property 5.11.(1) we can focus on those outcomes β'' assigning non-overlined values to all features in $\mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B}$. There are three cases: (1) $\beta'[U_1U_2] = \overline{u_1}u_2$, (2) $\beta'[U_1U_2] = u_1\overline{u_2}$, or (3) $\beta'[U_1U_2] = \overline{u_1u_2}$:

(1) Let γ be the outcome such that, for all features $F \notin \{U_1, U_2\}$, $\gamma[F] = \beta''[F]$, and $\gamma[U_1U_2] = \overline{u_1u_2} \neq \beta''[U_1U_2]$. By the definition of N_2^{eml}, N_3^{eml}, N_4^{eml}, and N_5^{eml}, $\gamma \succ_{N_2^{\text{eml}}} \beta''$, $\gamma \succ_{N_3^{\text{eml}}} \beta''$, $\gamma \succ_{N_4^{\text{eml}}} \beta''$, $\gamma \succ_{N_5^{\text{eml}}} \beta''$, respectively. Therefore, $\gamma \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta''$, and β'' is not majority optimal.

(2) Let γ be the outcome such that, for all features $F \notin \{U_1, U_2\}$, $\gamma[F] = \beta''[F]$, and $\gamma[U_1U_2] = u_1u_2 \neq \beta''[U_1U_2]$. We show that $\gamma \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta''$. Consider net N_1^{eml}. Since we are assuming that all features in $\mathcal{A}$ have non-overlined value in β'', we can flip U_2 from $\overline{u_2}$ to u_2. Hence, $\gamma \succ_{N_1^{\text{eml}}} \beta''$. By the definition of N_3^{eml}, N_4^{eml}, and N_5^{eml}, $\gamma \succ_{N_3^{\text{eml}}} \beta''$, $\gamma \succ_{N_4^{\text{eml}}} \beta''$, $\gamma \succ_{N_5^{\text{eml}}} \beta''$, respectively. Therefore, $\gamma \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta''$, and β'' is not majority optimal.

(3) Let γ be the outcome such that, for all features $F \notin \{U_1, U_2\}$, $\gamma[F] = \beta''[F]$, and $\gamma[U_1U_2] = u_1\overline{u_2} \neq \beta''[U_1U_2]$. We show that $\gamma \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta''$. Consider net N_2^{eml}. Since we are assuming that all features in $\mathcal{A}$ have non-overlined value in β'', we can flip U_1 from $\overline{u_1}$ to u_1. Hence, $\gamma \succ_{N_2^{\text{eml}}} \beta''$. Moreover, we are also assuming that all features in $\mathcal{B}$ have non-overlined value in β'', which implies that in N_3^{eml} we can flip U_1 from $\overline{u_1}$ to u_1. Therefore, $\gamma \succ_{N_3^{\text{eml}}} \beta''$. Since $N_3^{\text{eml}} = N_4^{\text{eml}}$, $\gamma \succ_{N_4^{\text{eml}}} \beta''$, as well. Finally, by the definition of N_6^{eml}, $\gamma \succ_{N_6^{\text{eml}}} \beta''$ (as we are assuming $\beta''[U_2] = \overline{u_2}$). Thus, $\gamma \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta''$, and β'' is not majority optimal. □

Property 5.11.(3). *Let $\beta''' \in O_d'''$ be an outcome. Then, β''' is not majority optimal in $\mathcal{M}_{\text{eml}}(\Phi)$.*

Proof. By Properties 5.11.(1) and 5.11.(2), we can focus on those outcomes β''' assigning non-overlined values to all features in $\mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B}$, and such that $\beta'''[U_1U_2] = u_1u_2$. Since $\beta''' \in O_d'''$, 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 $F \notin \{V_i^T, V_i^F, U_1, U_2\}$, $\gamma[F] = \beta'''[F]$, $\gamma[V_i^T V_i^F] = v_i^T v_i^F \neq \beta'''[V_i^T V_i^F]$, and $\gamma[U_1U_2] = \overline{u_1}u_2 \neq \beta'''[U_1U_2]$. We show that $\gamma \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta'''$.

Consider net N_1^{eml}. We can flip feature U_1 from u_1 to $\overline{u_1}$. Then, we can flip features V_i^T and V_i^F to their non-overlined value. The reached outcome is precisely γ, and hence $\gamma \succ_{N_1^{\text{eml}}} \beta'''$.

Consider now net N_3^{eml}. Since $\beta'''[V_i^T V_i^F] = \overline{v_i^T v_i^F}$, we can flip V_i' from v_i' to $\overline{v_i'}$. Then, we flip V_i^T and V_i^F to their non-overlined value. At this point, since $\beta'''[U_1U_2] = u_1u_2$, the improving flipping sequence in N_3^{eml} exhibited in Case (2) of Property 5.11.(1), in which the flips of features in $\mathcal{V}$ are ignored, is an improving flipping sequence from β''' to γ in this case. Hence, $\gamma \succ_{N_3^{\text{eml}}} \beta'''$. Since $N_3^{\text{eml}} = N_4^{\text{eml}}$, $\gamma \succ_{N_4^{\text{eml}}} \beta'''$, as well.

Finally, by the definition of N_5^{eml}, $\gamma \succ_{N_5^{\text{eml}}} \beta'''$. Therefore, $\gamma \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta'''$. □

The three properties above show that any outcome $\beta \in O_d$ is not majority optimal. Consider now outcomes in O_c. Observe that, since O_d and O_c are disjoint, all outcomes $\beta_c \in O_c$ are characterized by having all features in $\mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B}$ with non-overlined values, $\beta_c[U_1U_2] = u_1u_2$, and, for all variables $x_i \in X$, $\beta_c[V_i^T V_i^F] \neq \overline{v_i^T v_i^F}$.

Given an assignment σ_X for the variables in X, we define $\beta_{\sigma_X} \in O_c$ as the outcome encoding σ_X over the feature set $\mathcal{V}$ as usual. Denote by $\overline{Witn_c}$ the set of all *complete* assignments σ_X over X such that $(\forall Y)\neg\phi(X/\sigma_X, Y)$ is *not* 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 $\overline{Witn}$. Recall 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, $O_c^{Witn} = \{\beta_{\sigma_X} \in O_c \mid \sigma_X \in Witn\}$ and $O_c^{\overline{Witn}} = \{\beta_{\sigma_X} \in O_c \mid \sigma_X \in \overline{Witn}\}$ constitute a partition of O_c. This implies that, if $\beta \in O_c$ is an outcome, then there is a (partial or complete) assignment σ_X over the variables in X such that $\beta = \beta_{\sigma_X}$.

We show that all outcomes in $O_c^{\overline{Witn}}$ are not majority optimal, and that all outcomes in O_c^{Witn} are majority optimal (i.e., O_c^{Witn} is the set S mentioned earlier).

Property 5.11.(4). *Let $\beta_c \in O_c^{\overline{Witn}}$ be an outcome. Then, β_c is not majority optimal in $\mathcal{M}_{\text{eml}}(\Phi)$.*

Proof. Let $\overline{\alpha}$ be the outcome assigning overlined values only to U_1 and U_2. We show that $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta_c$. Let $\sigma_X \in \overline{Witn}$ be the (partial or complete) assignment over X such that $\beta_c = \beta_{\sigma_X}$.

Consider the net N_1^{eml}. First, note that, for any outcome $\beta_c \in O_c$, β_c assigns non-overlined values to all features in $\mathcal{V}' \cup \mathcal{B}$. Therefore, the part of net N_1^{eml} over feature sets $\mathcal{V}'$ 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^{eml}, features in $\mathcal{V}' \cup \mathcal{B}$ have no parents, and they have already their most preferred values in $\overline{\alpha}$ and β_c.

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 $\sigma_X \in \overline{Witn}$, there is an extension σ'_X of σ_X to X such that $(\forall Y)\neg\phi(X/\sigma'_X, Y)$ is not 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 5.2, $\overline{\alpha} \succ_{N_1^{\text{eml}}} \beta_c$. Observe that the differences between N_1^{eml} and N_2^{eml} are only in the roles of features U_1 and U_2, which are exchanged. Hence, by (an adaptation of) Lemma 5.2, $\overline{\alpha} \succ_{N_2^{\text{eml}}} \beta_c$, as well.

Consider now N_5^{eml}. The following is an improving flipping sequence from β_c to $\overline{\alpha}$. We flip U_1 from u_1 to $\overline{u_1}$, and then U_2 from u_2 to $\overline{u_2}$. After this, we flip to their non-overlined value all features in $\mathcal{V}$ having overlined values in β_c. The outcome reached is precisely $\overline{\alpha}$, and hence $\overline{\alpha} \succ_{N_5^{\text{eml}}} \beta_c$.

Finally, consider net N_6^{eml}. The following is an improving flipping sequence from β_c to $\overline{\alpha}$. We flip U_1 from u_1 to $\overline{u_1}$ (remember that $\beta_c[U_2] = u_2$). Next, we flip U_2 from u_2 to $\overline{u_2}$. To conclude, we flip to their non-overlined value all features in $\mathcal{V}$ having overlined values in β_c. The outcome reached is precisely $\overline{\alpha}$, and hence $\overline{\alpha} \succ_{N_6^{\text{eml}}} \beta_c$. Therefore, $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta_c$, and β_c is not majority optimal. □

The goal of the next properties is to show that outcomes in O_c^{Witn} are majority optimal. We prove this by showing that any outcome $\beta_c \in O_c^{Witn}$ is not majority dominated by any other outcome. Note that, since $\mathcal{M}_{\text{eml}}(\Phi)$ is a 6CP-net, for any two outcomes α and β, if $|S^{\succ}_{\mathcal{M}_{\text{eml}}(\Phi)}(\beta,\alpha)| \le 3$, then $\beta \not\succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \alpha$.

Property 5.11.(5). *Let $\beta_c \in O_c^{Witn}$ and γ be two outcomes such that there is a feature $F \in (\mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A})$ for which $\gamma[F] \neq \beta_c[F]$. Then, $\gamma \not\succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta_c$.*

Proof. Since $\gamma[F] \neq \beta_c[F]$, and $\beta_c[F] = f$ because $\beta_c \in O_c^{Witn} \subseteq O_c$, it must be the case that $\gamma[F] = \overline{f}$. Therefore, by the definition of N_3^{eml}, N_4^{eml}, and N_5^{eml}, $\gamma \not\succ_{N_3^{\text{eml}}} \beta_c$, $\gamma \not\succ_{N_4^{\text{eml}}} \beta_c$, and $\gamma \not\succ_{N_5^{\text{eml}}} \beta_c$. Thus, $|S^{\succ}_{\mathcal{M}_{\text{eml}}(\Phi)}(\gamma,\beta_c)| \le 3$, and hence $\gamma \not\succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta_c$. □

Property 5.11.(6). *Let $\beta_c \in O_c^{Witn}$ and γ be two outcomes such that there is a feature $F \in (\mathcal{V}' \cup \mathcal{B})$ for which $\gamma[F] \neq \beta_c[F]$. Then, $\gamma \not\succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta_c$.*

Proof. Since $\gamma[F] \neq \beta_c[F]$, and $\beta_c[F] = f$ because $\beta_c \in O_c^{Witn} \subseteq O_c$, it must be the case that $\gamma[F] = \overline{f}$. Therefore, by the definition of N_1^{eml}, N_2^{eml}, and N_5^{eml}, $\gamma \not\succ_{N_1^{\text{eml}}} \beta_c$, $\gamma \not\succ_{N_2^{\text{eml}}} \beta_c$, and $\gamma \not\succ_{N_5^{\text{eml}}} \beta_c$. Thus, $|S^{\succ}_{\mathcal{M}_{\text{eml}}(\Phi)}(\gamma,\beta_c)| \le 3$, and hence $\gamma \not\succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta_c$. □

Property 5.11.(7). *Let $\beta_c \in O_c^{Witn}$ and γ be two outcomes such that $\gamma[U_1U_2] \neq \beta_c[U_1U_2] = u_1u_2$. Then, $\gamma \not\succ^{maj}_{\mathcal{M}_{\text{eml}}(\Phi)} \beta_c$.*

Proof. Consider first net N_3^{eml}. Since $\beta_c \in O_c^{Witn} \subseteq O_c$, all features in $\mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B}$ have a non-overlined value in β_c, and there is no pair of features (V_i^T, V_i^F) such that $\beta_c[V_i^T V_i^F] = \overline{v_i^T v_i^F}$. Hence, the disjunctive interconnecting net embedded in N_3^{eml} cannot be exploited in any improving flipping sequence that aims at reaching an outcome in which U_1 or U_2 has an overlined value. Hence, for any outcome γ such that $\gamma[U_1] = \overline{u_1}$ or $\gamma[U_2] = \overline{u_2}$, $\gamma \not\succ_{N_3^{\text{eml}}} \beta_c$. By $N_3^{\text{eml}} = N_4^{\text{eml}}$, for any outcome γ such that $\gamma[U_1] = \overline{u_1}$ or $\gamma[U_2] = \overline{u_2}$, $\gamma \not\succ_{N_4^{\text{eml}}} \beta_c$, as well.

Since $\gamma[U_1U_2] \neq \beta_c[U_1U_2] = u_1u_2$, there are the three cases (1) $\gamma[U_1U_2] = u_1\overline{u_2}$, (2) $\gamma[U_1U_2] = \overline{u_1}u_2$, or (3) $\gamma[U_1U_2] = \overline{u_1 u_2}$:

(1) Let us focus on N_1^{eml}. First note that, for any outcome $\beta_c \in O_c$, β_c assigns non-overlined values to all features in $\mathcal{V}' \cup \mathcal{B}$, and by Property 5.11.(7) we can assume that γ assigns non-overlined values to features in $\mathcal{V}' \cup \mathcal{B}$. Therefore, the part of net N_1^{eml} over feature sets $\mathcal{V}'$ and $\mathcal{B}$ does not play an active role in any improving flipping sequence (if exists) either from γ to β_c, or from β_c to γ because, in N_1^{eml}, features in $\mathcal{V}' \cup \mathcal{B}$ have no parents, and they have already their most preferred values in β_c and γ.

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 $(\forall Y)\neg\phi(X/\sigma'_X, Y)$ is not valid, (i.e., such that $\phi(X/\sigma'_X, Y)$ is satisfiable). 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 inspection of the proof of Lemma 5.2, in N_1^{eml} there is no improving flipping sequence from β_c to an outcome in which U_2 has an overlined value, hence $\gamma \not\succ_{N_1^{\text{eml}}} \beta_c$.

From this, and from what we have already shown for nets N_3^{eml} and N_4^{eml}, it follows that $|S^{\succ}_{\mathcal{M}_{\mathrm{eml}}(\Phi)}(\gamma, \beta_c)| \le 3$, and hence $\gamma \not\succ^{maj}_{\mathcal{M}_{\mathrm{eml}}(\Phi)} \beta_c$.

(2) If we focus on N_2^{eml}, a similar argument to the one used in (1) for net N_1^{eml} shows that $\gamma \not\succ_{N_2^{\mathrm{eml}}} \beta_c$ (simply observe that the roles on U_1 and U_2 are exchanged). Again, from this, and from what we have already shown for nets N_3^{eml} and N_4^{eml}, it follows that $|S^{\succ}_{\mathcal{M}_{\mathrm{eml}}(\Phi)}(\gamma, \beta_c)| \le 3$, and hence $\gamma \not\succ^{maj}_{\mathcal{M}_{\mathrm{eml}}(\Phi)} \beta_c$.

(3) By combining the discussions in (1) and (2), it is possible to show that $\gamma \not\succ_{N_1^{\mathrm{eml}}} \beta_c$ and $\gamma \not\succ_{N_2^{\mathrm{eml}}} \beta_c$ (and that $\gamma \not\succ_{N_3^{\mathrm{eml}}} \beta_c$ and $\gamma \not\succ_{N_4^{\mathrm{eml}}} \beta_c$). Thus, $|S^{\succ}_{\mathcal{M}_{\mathrm{eml}}(\Phi)}(\gamma, \beta_c)| \le 2$, and hence $\gamma \not\succ^{maj}_{\mathcal{M}_{\mathrm{eml}}(\Phi)} \beta_c$. □

Property 5.11.(8). *Let $\beta_c \in O_c^{Witn}$ and γ be two outcomes such that there is a feature $F \in \mathcal{V}$ for which $\gamma[F] \neq \beta_c[F]$. Then, $\gamma \not\succ^{maj}_{\mathcal{M}_{\mathrm{eml}}(\Phi)} \beta_c$.*

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 the 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 the 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 assume that $Change \neq \varnothing$, it must be the case that $(Up \cup Down) \neq \varnothing$. There are the two cases (1) $Up \neq \varnothing$, or (2) $Up = \varnothing$:

(1) Since there are variable features in $\mathcal{V}$ changing their value from non-overlined in β_c to overlined in γ, by the definition of N_3^{eml}, N_4^{eml}, and N_5^{eml}, $\gamma \not\succ_{N_3^{\mathrm{eml}}} \beta_c$, $\gamma \not\succ_{N_4^{\mathrm{eml}}} \beta_c$, and $\gamma \not\succ_{N_5^{\mathrm{eml}}} \beta_c$. Therefore, $|S^{\succ}_{\mathcal{M}_{\mathrm{eml}}(\Phi)}(\gamma, \beta_c)| \le 3$, and hence $\gamma \not\succ^{maj}_{\mathcal{M}_{\mathrm{eml}}(\Phi)} \beta_c$.

(2) Since $Up = \varnothing$ and $Change \neq \varnothing$, it must be the case that $Down \neq \varnothing$, and hence there are variable features in $\mathcal{V}$ changing their value from overlined in β_c to non-overlined in γ. Moreover, by Property 5.11.(7), we can assume that $\gamma[U_1U_2] = u_1u_2$. Consider net N_1^{eml}. Observe that feature U_1 in N_1^{eml} has no parents. Hence, once U_1 is flipped from u_1 to $\overline{u_1}$, it cannot be flipped back. Therefore, since $\beta_c[U_1] = \gamma[U_1] = u_1$, in any improving flipping sequence in N_1^{eml} from β_c to γ (if exists), feature U_1 cannot be flipped at all. However, by the definition of the CP tables in N_1^{eml}, when U_1 has value u_1, variable features can be flipped only from non-overlined to overlined. Hence, from $Down \neq \varnothing$ it follows that $\gamma \not\succ_{N_1^{\mathrm{eml}}} \beta_c$. A similar argument (but focused on U_2) shows that $\gamma \not\succ_{N_2^{\mathrm{eml}}} \beta_c$ and $\gamma \not\succ_{N_6^{\mathrm{eml}}} \beta_c$. Therefore, $|S^{\succ}_{\mathcal{M}_{\mathrm{eml}}(\Phi)}(\gamma, \beta_c)| \le 3$, and hence $\gamma \not\succ^{maj}_{\mathcal{M}_{\mathrm{eml}}(\Phi)} \beta_c$. □

We are now ready to prove that $\Phi = (\exists X)(\forall Y)\neg\phi(X, Y)$ is valid if and only if $\mathcal{M}_{\mathrm{eml}}(\Phi)$ has a majority optimal outcome.

($\Rightarrow$) Assume that Φ is valid, hence there is an assignment σ_X for the variables in X such that $\sigma_X \in Witn$. By Properties 5.11.(5), 5.11.(6), 5.11.(7), and 5.11.(8), β_{σ_X} is majority optimal in $\mathcal{M}_{\mathrm{eml}}(\Phi)$, and hence $\mathcal{M}_{\mathrm{eml}}(\Phi)$ has a majority optimal outcome.

($\Leftarrow$) Assume that Φ is not valid, hence there is no assignment in *Witn*, and so O_c^{Witn} is empty. By Properties 5.11.(1), 5.11.(2), 5.11.(3), and 5.11.(4), all outcomes in $O_d \cup O_c^{\overline{Witn}}$ are not majority optimal, and, since O_c^{Witn} is empty, $\mathcal{M}_{\mathrm{eml}}(\Phi)$ does not have a majority optimal outcome. □

Lemma 5.15. *Let $\Phi = (\exists X)(\forall Y)\neg\phi(X, Y)$ be a quantified Boolean formula, where $\phi(X, Y)$ is a 3CNF Boolean formula, defined over two disjoint sets X and Y of Boolean variables. Then, Φ is valid if and only if $\mathcal{M}_{\mathrm{imm}}(\Phi)$ does not have a majority optimum outcome. In particular, when $\mathcal{M}_{\mathrm{imm}}(\Phi)$ has a majority optimum outcome it is the outcome $\overline{\alpha} \in \mathcal{O}_{\mathcal{M}_{\mathrm{imm}}(\Phi)}$ assigning overlined values only to features U_1 and U_2.*

Proof. To prove the statement of the lemma we have to analyze the majority dominance relationship between $\overline{\alpha}$ and the other outcomes. To organize this task, let us define the following sets of outcomes:

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

The sets O_d and O_c constitute a partition of $\mathcal{O}_{\mathcal{M}_{\mathrm{imm}}(\Phi)}$. On the contrary, $O_d{}'$, $O_d{}''$, and $O_d{}'''$, do not constitute a partition of O_d because they are not disjoint. Note that $\overline{\alpha} \in O_d{}''$. We show that $\overline{\alpha}$ is the only outcome that might be majority optimum. We do so by showing that (1) all outcomes different from $\overline{\alpha}$ in $O_d \cup O_c$, but a specific subset S (whose detailed characterization will be given toward the end of the proof) of outcomes of O_c, are majority dominated by $\overline{\alpha}$, which means that all outcomes in $(O_d \cup O_c) \setminus (\{\overline{\alpha}\} \cup S)$ are majority optimum; and that (2) all outcomes in S, which might be empty, neither majority dominate, nor are majority dominated by, $\overline{\alpha}$, and hence they are not majority optimum. Therefore, $\overline{\alpha}$ is majority optimum if and only if S is empty.

Note that, by the definition of N_2^{imm}, $\overline{\alpha} \succ_{N_2^{\mathrm{imm}}} \beta$ for any outcome $\beta \neq \overline{\alpha}$, and hence, since $\mathcal{M}_{\mathrm{imm}}(\Phi)$ is a 3CP-net, if $\overline{\alpha} \succ_{N_1^{\mathrm{imm}}} \beta$ or $\overline{\alpha} \succ_{N_3^{\mathrm{imm}}} \beta$, then $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta$.

Property 5.15.(1). *Let $\beta' \in O_d{}'$ be an outcome. Then, $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta'$.*

Proof. Since $\beta' \in O_d{}'$, let $F \in (\mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B})$ be a feature such that $\beta'[F] = \overline{f}$. Because $\overline{\alpha}[F] = f$, $\beta' \neq \overline{\alpha}$, and hence $\overline{\alpha} \succ_{N_2^{\mathrm{imm}}} \beta'$. Consider net N_3^{imm}. The following is an improving flipping sequence from β' to $\overline{\alpha}$ in N_3^{imm}, showing that $\overline{\alpha} \succ_{N_3^{\mathrm{imm}}} \beta'$, and hence that $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta'$. Since $\beta'[F] = \overline{f}$, by the definition of the disjunctive interconnecting net embedded in N_3^{imm}, we can flip in the proper order some features of the interconnecting net to their overlined values, until we flip its apex B. Once the apex has an overlined value, we flip to their overlined value U_1 and then U_2 (if they do not have an overlined value already). After this, we flip to their non-overlined values all features in $\mathcal{V} \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A}$ having overlined values in β'. Next, we flip to their non-overlined values all features in $\mathcal{V}'$ having overlined values, and subsequently those in $\mathcal{B}$ having overlined values. □

Property 5.15.(2). *Let $\beta'' \in O_d{}''$ be an outcome different from $\overline{\alpha}$. Then, $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\phi)} \beta''$.*

Proof. Since we are assuming $\beta'' \neq \overline{\alpha}$, it holds that $\overline{\alpha} \succ_{N_2^{\mathrm{imm}}} \beta''$. There are the three cases (1) $\beta''[U_1U_2] = \overline{u_1}\overline{u_2}$, or (2) $\beta''[U_1U_2] = \overline{u_1}u_2$, or (3) $\beta''[U_1U_2] = u_1\overline{u_2}$:

(1) Let us focus on net N_3^{imm}. The following is an improving flipping sequence from β'' to $\overline{\alpha}$ in N_3^{imm}, showing that $\overline{\alpha} \succ_{N_3^{\mathrm{imm}}} \beta''$, and hence that $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta''$. We can flip all features, but U_1 and U_2, to their non-overlined values in a proper sequence which could be $\mathcal{V}$, $\mathcal{V}'$, $\mathcal{W}$, $\mathcal{P}$, $\mathcal{D}$, $\mathcal{A}$, and $\mathcal{B}$.

(2) Let us focus again on net N_3^{imm}. The following is an improving flipping sequence from β'' to $\overline{\alpha}$ in N_3^{imm}, showing that $\overline{\alpha} \succ_{N_3^{\mathrm{imm}}} \beta''$, and hence that $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta''$. Since $\beta''[U_1U_2] = \overline{u_1}u_2$, we can flip U_2 from u_2 to $\overline{u_2}$. Now we are again in the case in which $\beta''[U_1U_2] = \overline{u_1}\overline{u_2}$. Hence, there is an improving flipping sequence to $\overline{\alpha}$ as shown in (1).

(3) Let us consider net N_1^{imm}. The following is an improving flipping sequence from β'' to $\overline{\alpha}$ in N_1^{imm}, showing that $\overline{\alpha} \succ_{N_1^{\mathrm{imm}}} \beta''$, and hence that $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta''$. We can flip U_1 from u_1 to $\overline{u_1}$. Then, in the order $\mathcal{V}$, $\mathcal{W}$, $\mathcal{P}$, $\mathcal{D}$, $\mathcal{A}$, $\mathcal{V}'$, and $\mathcal{B}$, we flip features having overlined values to their non-overlined values. □

Property 5.15.(3). *Let $\beta''' \in O_d{}'''$ be an outcome. Then, $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta'''$.*

Proof. Since $\beta''' \in O_d{}''''$, there is a pair of features (V_i^T, V_i^F) such that $\beta'''[V_i^T V_i^F] = \overline{v_i^T v_i^F}$. Because $\overline{\alpha}[V_i^T V_i^F] = v_i^T v_i^F$, $\beta''' \neq \overline{\alpha}$, and hence $\overline{\alpha} \succ_{N_2^{\mathrm{imm}}} \beta'''$. Consider net N_3^{imm}. We show that there is an improving flipping sequence from β''' to $\overline{\alpha}$ in N_3^{imm}, proving that $\overline{\alpha} \succ_{N_3^{\mathrm{imm}}} \beta'''$, and hence that $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta'''$.

Since $\beta'''[V_i^T V_i^F] = \overline{v_i^T v_i^F}$, if $\beta'''[V_i'] = v_i'$, we can flip the value of V_i' from v_i' to $\overline{v_i'}$. This brings us in the case in which there is a feature F in $\mathcal{V}'$ with overlined value. We have already shown in the proof of Property 5.15.(1) that in N_3^{imm} there exists an improving flipping sequence from this outcome to $\overline{\alpha}$. □

The three properties above show that any outcome $\beta \in O_d$ that is different from $\overline{\alpha}$ is majority dominated by $\overline{\alpha}$. This proves that any such outcome β is not majority optimum, and that $\overline{\alpha}$ majority dominates all of them.

Let us now consider outcomes in O_c. Observe that, since O_d and O_c are disjoint, all outcomes $\beta_c \in O_c$ are characterized by having all features in $\mathcal{V}' \cup \mathcal{W} \cup \mathcal{P} \cup \mathcal{D} \cup \mathcal{A} \cup \mathcal{B}$ with non-overlined values, $\beta_c[U_1U_2] = u_1u_2$, and, for all variables $x_i \in X$, $\beta_c[V_i^T V_i^F] \neq \overline{v_i^T v_i^F}$.

Given an assignment σ_X for the variables in X, we define $\beta_{\sigma_X} \in O_c$ as the outcome encoding σ_X over the feature set $\mathcal{V}$ as usual. Let us denote by $\overline{Witn_c}$ the set of all *complete* assignments σ_X over X such that $(\forall Y)\neg\phi(X/\sigma_X, Y)$ is *not* 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, $O_c^{Witn} = \{\beta_{\sigma_X} \in O_c \mid \sigma_X \in Witn\}$, and $O_c^{\overline{Witn}} = \{\beta_{\sigma_X} \in O_c \mid \sigma_X \in \overline{Witn}\}$ constitute a partition of O_c. This implies that, if $\beta \in O_c$ is an outcome, then there is a (partial or complete) assignment σ_X over the variables in X such that $\beta = \beta_{\sigma_X}$.

We show that all outcomes in $O_c^{\overline{Witn}}$ are majority dominated by $\overline{\alpha}$, and that all outcomes in O_c^{Witn} are not majority dominated by $\overline{\alpha}$ and do not majority dominate $\overline{\alpha}$ (i.e., O_c^{Witn} is the set S mentioned earlier).

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 $\mathcal{V}' \cup \mathcal{B}$, and also $\overline{\alpha}$ assigns non-overlined values to features in $\mathcal{V}' \cup \mathcal{B}$. Therefore, the part of net N_1^{imm} over feature sets $\mathcal{V}'$ 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^{imm}, features in $\mathcal{V}' \cup \mathcal{B}$ have no parents, and they have already their most preferred values in $\overline{\alpha}$ and β_c.

Property 5.15.(4). *Let $\beta_c \in O_c^{\overline{Witn}}$ be an outcome. Then, $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta_c$.*

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^{imm}. 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 $\sigma_X \in \overline{Witn}$, there is an extension σ'_X of σ_X to X such that $(\forall Y)\neg\phi(X/\sigma'_X, Y)$ is not valid (i.e., such that $\phi(X/\sigma'_X, Y)$ is satisfiable), and hence there is an extension of σ_X to $X \cup Y$ satisfying ϕ. Therefore, by Lemma 5.2, $\overline{\alpha} \succ_{N_1^{\mathrm{imm}}} \beta_c$. Since $\overline{\alpha} \in O_d''$ and $\beta_c \in O_c$, $\beta_c \neq \overline{\alpha}$, and hence $\overline{\alpha} \succ_{N_2^{\mathrm{imm}}} \beta_c$. Thus, $\overline{\alpha} \succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta_c$. □

Property 5.15.(5). *Let $\beta_c \in O_c^{Witn}$ be an outcome. Then, $\overline{\alpha} \not\succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta_c$ and $\beta_c \not\succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \overline{\alpha}$.*

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 $(\forall Y)\neg\phi(X/\sigma'_X, Y)$ is not valid, (i.e., such that $\phi(X/\sigma'_X, Y)$ is satisfiable).

Now consider net N_1^{imm}. We claim that $\beta_c \bowtie_{N_1^{\mathrm{imm}}} \overline{\alpha}$. 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 5.2, $\beta_c \bowtie_{N_1^{\mathrm{imm}}} \overline{\alpha}$.

Now, since $\overline{\alpha} \in O_d''$ and $\beta_c \in O_c$, $\beta_c \neq \overline{\alpha}$, and hence $\overline{\alpha} \succ_{N_2^{\mathrm{imm}}} \beta_c$. To conclude, let us now focus on net N_3^{imm}. By $\beta_c \in O_c$, all features in $\mathcal{B}$ have non-overlined values, $\beta_c[U_1 U_2] = u_1 u_2$, and there is no pair of features (V_i^T, V_i^F) such that $\beta_c[V_i^T V_i^F] = \overline{v_i^T}\, \overline{v_i^F}$. Since in N_3^{imm} feature U_1 is attached to the apex B of the interconnecting net, and U_1 can be flipped from u_1 to $\overline{u_1}$ only when the apex B of the interconnecting net has an overlined value, there is no improving flipping sequence from β_c to $\overline{\alpha}$, and hence $\overline{\alpha} \not\succ_{N_3^{\mathrm{imm}}} \beta_c$.

To summarize, we showed that $\beta_c \bowtie_{N_1^{\mathrm{imm}}} \overline{\alpha}$, $\overline{\alpha} \succ_{N_2^{\mathrm{imm}}} \beta_c$, and $\overline{\alpha} \not\succ_{N_3^{\mathrm{imm}}} \beta_c$. Therefore, $\overline{\alpha} \not\succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \beta_c$, because $|S^{\succ}_{\mathcal{M}_{\mathrm{imm}}(\Phi)}(\overline{\alpha}, \beta_c)| < 2$, and $\beta_c \not\succ^{maj}_{\mathcal{M}_{\mathrm{imm}}(\Phi)} \overline{\alpha}$, because $|S^{\succ}_{\mathcal{M}_{\mathrm{imm}}(\Phi)}(\beta_c, \overline{\alpha})| < 2$. □

We are now ready to prove that $\Phi = (\exists X)(\forall Y)\neg\phi(X, Y)$ is valid if and only if $\overline{\alpha}$ is not majority optimum in $\mathcal{M}_{\mathrm{imm}}(\Phi)$.

(⇒) Assume that Φ is valid. By Properties 5.15.(1), 5.15.(2), 5.15.(3), and 5.15.(4), all outcomes in $O_d \cup O_c^{\overline{Witn}}$ different from $\overline{\alpha}$ are not majority optimum. Moreover, from Property 5.15.(5), all outcomes in O_c^{Witn} are not majority optimum, and $\overline{\alpha}$ does not majority dominate outcomes in O_c^{Witn}. Hence, also $\overline{\alpha}$ is not majority optimum. Therefore, in $\mathcal{M}_{\mathrm{imm}}(\Phi)$ there is no majority optimum outcome.

(⇐) Assume that Φ is not valid, hence there is no assignment in *Witn*, and so O_c^{Witn} is empty. By Properties 5.15.(1), 5.15.(2), 5.15.(3), and 5.15.(4), all outcomes in $O_d \cup O_c^{\overline{Witn}}$ different from $\overline{\alpha}$ are majority dominated by $\overline{\alpha}$. Since O_c^{Witn} is empty, $\overline{\alpha}$ majority dominates all other outcomes, which implies that in $\mathcal{M}_{\mathrm{imm}}(\Phi)$ there is a majority optimum outcome, which is $\overline{\alpha}$. □

References

[1] S. Airiau, U. Endriss, U. Grandi, D. Porello, J. Uckelman, Aggregating dependency graphs into voting agendas in multi-issue elections, in: Proc. of IJCAI, 2011, pp. 18–23.

[2] T.E. Allen, CP-nets: From Theory to Practice, PhD Thesis Department of Computer Science, University of Kentucky, 2016.

[3] T.E. Allen, M. Chen, J. Goldsmith, N. Mattei, A. Popova, M. Regenwetter, F. Rossi, C. Zwilling, Beyond theory and data in preference modeling: bringing humans into the loop, in: Proc. of ADT 2015, in: Lect. Notes Comput. Sci., vol. 9346, Springer-Verlag, 2015, pp. 3–18.

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

[5] K.J. Arrow, A.K. Sen, K. Suzumura (Eds.), Handbook of Social Choice and Welfare, vol. 1, Handbooks in Economics, vol. 19, North Holland/Elsevier, Amsterdam, The Netherlands, 2002.

[6] K.J. Arrow, A.K. Sen, K. Suzumura (Eds.), Handbook of Social Choice and Welfare, vol. 2, Handbooks in Economics, vol. 19, North Holland/Elsevier, Amsterdam, The Netherlands, 2011.

[7] D. Baumeister, J. Rothe, Preference aggregation by voting, in: J. Rothe (Ed.), Economics and Computation, Springer-Verlag, Berlin, Heidelberg, Germany, 2016, pp. 197–326.

[8] D. Bigot, H. Fargier, J. Mengin, B. Zanuttini, Probabilistic conditional preference networks, in: Proc. of UAI 2013, AUAI Press, 2013, pp. 72–81.

[9] J.C. Boerkoel Jr., E.H. Durfee, K. Purrington, Generalized solution techniques for preference-based constrained optimization with CP-nets, in: Proc. of AAMAS, 2010, pp. 291–298.

[10] S. Borgwardt, B. Fazzinga, T. Lukasiewicz, A. Shrivastava, O. Tifrea-Marciuska, Preferential query answering over the Semantic Web with possibilistic networks, in: Proc. of IJCAI, 2016, pp. 994–1000.

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

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

[13] C. Boutilier, R.I. Brafman, H.H. Hoos, D. Poole, Reasoning with conditional ceteris paribus preference statements, in: Proc. of UAI, 1999, pp. 71–80.

[14] R.I. Brafman, Y. Chernyavsky, Planning with goal preferences and constraints, in: Proc. of ICAPS, 2005, pp. 182–191.

[15] R.I. Brafman, C. Domshlak, TCP-nets for preference-based product configuration, in: Proc. of the ECAI 2002 Workshop on Configuration, 2002, pp. 101–106.

[16] R.I. Brafman, C. Domshlak, Preference handling — an introductory tutorial, AI Mag. 30 (1) (2009) 58–86.

[17] R.I. Brafman, C. Domshlak, Y. Engel, M. Tennenholtz, Transferable utility planning games, in: Proc. of AAAI, 2010, pp. 709–714.

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

[19] S.J. Brams, D.M. Kilgour, W.S. Zwicker, The paradox of multiple elections, Soc. Choice Welf. 15 (2) (1998) 211–236.

[20] F. Brandt, V. Conitzer, U. Endriss, Computational social choice, in: G. Weiss (Ed.), Multiagent Systems, MIT Press, Cambridge, MA, 2013, pp. 213–283.

[21] F. Brandt, V. Conitzer, U. Endriss, J. Lang, A.D. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, New York, NY, USA, 2016.

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

[23] V. Conitzer, J. Lang, L. Xia, Hypercubewise preference aggregation in multi-issue domains, in: Proc. of IJCAI, 2011, pp. 158–163.

[24] V. Conitzer, T. Sandholm, Communication complexity of common voting rules, in: Proc. of EC, 2005, pp. 78–87.

[25] V. Conitzer, L. Xia, Paradoxes of multiple elections: an approximation approach, in: Proc. of KR, 2012, pp. 179–187.

[26] C. Cornelio, J. Goldsmith, N. Mattei, F. Rossi, K.B. Venable, Updates and uncertainty in CP-nets, in: Proc. of AI 2013, in: Lect. Notes Comput. Sci., vol. 8272, Springer-Verlag, 2013, pp. 301–312.

[27] C. Cornelio, U. Grandi, J. Goldsmith, N. Mattei, F. Rossi, K.B. Venable, Reasoning with PCP-nets in a multi-agent context, in: Proc. of AAMAS, 2015, pp. 969–977.

[28] T. Di Noia, T. Lukasiewicz, M.V. Martinez, G.I. Simari, O. Tifrea-Marciuska, Combining existential rules with the power of CP-theories, in: Proc. of IJCAI, 2015, pp. 2918–2925.

[29] C. Domshlak, E. Hüllermeier, S. Kaci, H. Prade, Preferences in AI: an overview, Artif. Intell. 175 (7) (2011) 1037–1052.

[30] A. Felfernig, L. Hotz, C. Bagley, J. Tiihonen (Eds.), Knowledge-Based Configuration: From Research to Business Cases, Morgan Kaufmann/Elsevier, Waltham, MA, USA, 2014.

[31] D.S. Felsenthal, N. Tideman, Weak condorcet winner(s) revisited, Public Choice 160 (3) (2014) 313–326.

[32] M.R. Garey, D.S. Johnson, Computers and Intractability. A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, New York, NY, USA, 1979.

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

[34] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, in: Proc. of CSL-LICS, 2014, pp. 43:1–43:10.

[35] G. Gottlob, E. Malizia, Achieving new upper bounds for the hypergraph duality problem through logic, SIAM J. Comput. 47 (2) (2018) 456–492.

[36] U. Grandi, H. Luo, N. Maudet, F. Rossi, Aggregating CP-nets with unfeasible outcomes, in: Proc. of CP 2014, in: Lect. Notes Comput. Sci., vol. 8656, Springer, 2014, pp. 366–381.

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

[38] G. Greco, E. Malizia, L. Palopoli, F. Scarcello, On the complexity of compact coalitional games, in: Proc. of IJCAI, 2009, pp. 147–152.

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

[40] G. Greco, E. Malizia, L. Palopoli, F. Scarcello, On the complexity of core, kernel, and bargaining set, Artif. Intell. 175 (12–13) (2011) 1877–1910.

[41] G. Greco, E. Malizia, L. Palopoli, F. Scarcello, On the complexity of the core over coalition structures, in: Proc. of IJCAI, 2011, pp. 216–221.

[42] G. Greco, E. Malizia, L. Palopoli, F. Scarcello, The complexity of the nucleolus in compact games, ACM Trans. Comput. Theory 7 (1) (2014) 3:1–3:52.

[43] J.T. Guerin, T.E. Allen, J. Goldsmith, Learning CP-net preferences online from user queries, in: Proc. of ADT 2013, in: Lect. Notes Comput. Sci., vol. 8176, Springer-Verlag, 2013, pp. 208–220.

[44] S.O. Hansson, What is ceteris paribus preference? J. Philos. Log. 25 (3) (1996) 307–332.

[45] S.O. Hansson, Preference logic, 2nd ed., in: D.M. Gabbay, F. Guenthner (Eds.), Handbook of Philosophical Logic, vol. 4, Springer, 2002, pp. 319–393.

[46] S. Ieong, Y. Shoham, Marginal contribution nets: a compact representation scheme for coalitional games, in: Proc. of EC, 2005, pp. 193–202.

[47] D.S. Johnson, A catalog of complexity classes, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science (Vol. A), Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1990, pp. 67–161.

[48] R.M. Karp, Reducibility among combinatorial problems, in: Proc. of a Symposium on the Complexity of Computer Computations, in: The IBM Research Symposia Series, Springer-Verlag, Berlin Heidelberg, Germany, 1972, pp. 85–103.

[49] R.L. Keeney, H. Raiffa, Decisions with Multiple Objectives: Preferences and Value Tradeoffs, Wiley Ser. Prob. Math. Stat., Prob. Math. Stat., John Wiley & Sons, New York, NY, USA, 1976.

[50] D. Lacy, E.M.S. Niou, A problem with referendums, J. Theor. Polit. 12 (1) (2000) 5–31.

[51] J. Lang, From preference representation to combinatorial vote, in: Proc. of KR, 2002, pp. 277–290.

[52] J. Lang, Logical preference representation and combinatorial vote, Ann. Math. Artif. Intell. 42 (1) (2004) 37–71.

[53] J. Lang, Vote and aggregation in combinatorial domains with structured preferences, in: Proc. of IJCAI, 2007, pp. 1366–1371.

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

[55] J. Lang, L. Xia, Voting in combinatorial domains, in: F. Brandt, V. Conitzer, U. Endriss, J. Lang, A.D. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, New York, NY, USA, 2016, pp. 197–222.

[56] M. Li, Q.B. Vo, R. Kowalczyk, An efficient majority-rule-based approach for collective decision making with CP-nets, in: Proc. of KR, 2010, pp. 578–580.

[57] M. Li, Q.B. Vo, R. Kowalczyk, An efficient procedure for collective decision-making with CP-nets, in: Proc. of ECAI, 2010, pp. 375–380.

[58] M. Li, Q.B. Vo, R. Kowalczyk, Majority-rule-based preference aggregation on multi-attribute domains with CP-nets, in: Proc. of AAMAS, 2011, pp. 659–666.

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

[60] A. Loreggia, N. Mattei, F. Rossi, K.B. Venable, A notion of distance between cp-nets, in: Proc. of AAMAS, 2018, pp. 955–963.

[61] T. Lukasiewicz, E. Malizia, On the complexity of mCP-nets, in: Proc. of AAAI, 2016, pp. 558–564.

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

[63] T. Lukasiewicz, M.V. Martinez, G.I. Simari, Preference-based query answering in Datalog+/– ontologies, in: Proc. of IJCAI, 2013, pp. 1017–1023.

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

[65] A. Maran, M. Nicolas, M.S. Pini, F. Rossi, K.B. Venable, A framework for aggregating influenced CP-nets and its resistance to bribery, in: Proc. of AAAI, 2013, pp. 668–678.

[66] N. Mattei, M.S. Pini, F. Rossi, K.B. Venable, Bribery in voting with CP-nets, Ann. Math. Artif. Intell. 68 (1–3) (2013) 135–160.

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

[68] C.H. Papadimitriou, M. Yannakakis, The complexity of facets (and some facets of complexity), J. Comput. Syst. Sci. 28 (2) (1984) 244–259.

[69] S.D. Prestwich, F. Rossi, K.B. Venable, T. Walsh, Constraint-based preferential optimization, in: Proc. of AAAI, 2005, pp. 461–466.

[70] F. Ricci, L. Rokach, B. Shapira (Eds.), Recommender Systems Handbook, 2nd ed., Springer, New York, NY, USA, 2015.

[71] F. Rossi, K.B. Venable, T. Walsh, mCP Nets: representing and reasoning with preferences of multiple agents, in: Proc. of AAAI, 2004, pp. 729–734.

[72] R.A. Russell, Planning with Preferences Using Maximum Satisfiability, UCAM-CL-TR 822, Computing Laboratory, University of Cambridge, UK, 2012.

[73] D. Shaparau, M. Pistore, P. Traverso, Contingent planning with goal preferences, in: Proc. of AAAI, 2006, pp. 927–935.

[74] S. Sikdar, S. Adali, L. Xia, Optimal decision making with CP-nets and PCP-nets, in: Proc. of EXPLORE, 2017, pp. 32–40.

[75] S. Sikdar, S. Adali, L. Xia, Optimal decision making with CP-nets and PCP-nets (extended abstract), in: Proc. of AAMAS, 2017, pp. 1736–1738.

[76] T.C. Son, E. Pontelli, Planning with preferences using logic programming, Theory Pract. Log. Program. 6 (5) (2006) 559–607.

[77] J. Stein, I. Nunes, E. Cirilo, Preference-based feature model configuration with multiple stakeholders, in: Proc. of SPLC, 2014, pp. 132–141.

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

[79] K.W. Wagner, Bounded query computations, in: Proc. of CoCo, 1988, pp. 260–277.

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

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

[82] L. Xia, V. Conitzer, J. Lang, Voting on multiattribute domains with cyclic preferential dependencies, in: Proc. of AAAI, 2008, pp. 202–207.

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

[84] L. Xia, J. Lang, M. Ying, Sequential voting rules and multiple elections paradoxes, in: Proc. of TARK, 2007, pp. 279–288.

[85] L. Xia, J. Lang, M. Ying, Strongly decomposable voting rules on multiattribute domains, in: Proc. of AAAI, 2007, pp. 776–781.