On the Complexity of mCP-nets

Thomas Lukasiewicz and **Enrico Malizia**
Department of Computer Science,
University of Oxford, UK
firstname.lastname@cs.ox.ac.uk

Abstract

mCP-nets are an expressive and intuitive formalism based on CP-nets to reason about preferences of groups of agents. The dominance semantics of mCP-nets is based on the concept of voting, and different voting schemes give rise to different dominance semantics for the group. Unlike CP-nets, which received an extensive complexity analysis, mCP-nets, as reported multiple times in the literature, lack a precise study of the voting tasks' complexity. Prior to this work, only a complexity analysis of brute-force algorithms for these tasks was available, and this analysis only gave EXPTIME upper bounds for most of those problems. In this paper, we start to fill this gap by carrying out a precise computational complexity analysis of voting tasks on acyclic binary polynomially connected mCP-nets whose constituents are standard CP-nets. Interestingly, all these problems actually belong to various levels of the polynomial hierarchy, and some of them even belong to PTIME or LOGSPACE. Furthermore, for most of these problems, we provide completeness results, which show tight lower bounds for problems that (up to date) did not have any explicit non-obvious lower bound.

Introduction

Modeling and reasoning about user preferences is a vast area in AI with an extensive variety of applications. Among them, there is the one of query answering under (group) preferences. For example, in a Web that is moving forward from a document-centric paradigm to a user-centric one, it will be more and more important to present Web search results ranked according user preferences. In fact, in this example, the problem is twofold: How to present a query answer to a user who is known to belong to a group, and hence how to rank results according to group preferences, and, on the other hand, how to rank the answer when the specific user is known along with her own preferences.

In the literature, various models for representing user preferences were proposed, and among them one of the most studied is that of CP-nets (Boutilier et al. 2004), as a vast literature on them demonstrates. This model has proven itself to be useful and intuitive. In CP-nets, user preferences are encoded via a directed graph whose vertices represent the various features of the domain at hand. In the graph of a CP-net, there is an edge from vertex A to vertex B if the value of the feature A influences the choice of the value of feature B. Intuitively, this model captures preferences of the type “given that all the other characteristics are equal, I prefer this value for feature A to that value of feature A”.

CP-nets were used to model preferences of *groups*, obtaining mCP-nets (Rossi, Venable, and Walsh 2004). This multi-agent model is essentially a set of CP-nets, one for each user. Preferences for groups of agents in mCP-nets are defined through voting schemes. In fact, through their own individual CP-nets, each agent votes whether an outcome is preferred to another, and different ways of collecting votes (i.e., different voting schemes) give rise to different dominance semantics for mCP-nets. Various voting schemes were proposed for mCP-nets (Rossi, Venable, and Walsh 2004; Li, Vo, and Kowalczyk 2015). Unlike CP-nets, which received an extensive complexity analysis, a precise complexity analysis of voting tasks on mCP-nets is still missing, as explicitly mentioned multiple times in the literature (Lang 2007; Li, Vo, and Kowalczyk 2010a; 2010b; 2011; 2015). In the original paper (Rossi, Venable, and Walsh 2004), an initial investigation of these complexities was carried out by studying the complexity of brute-force algorithms to solve these problems. Our aim in this paper is to settle these problems in their exact complexity classes, showing, if possible, completeness results.

Contributions. In this paper, we focus on acyclic binary polynomially connected mCP-nets with standard CP-nets, i.e., the constituent CP-nets of an mCP-net rank all the features, and they are not partial CP-nets (see preliminaries). Our contributions are briefly as follows:

- ▷ We carry out a thorough complexity analysis, for the (a) Pareto, (b) Majority, (c) Max, and (d) Ranking voting schemes, of deciding (1) dominance, (2) optimal and (3) optimum outcomes, (4) the existence of optimal and (5) optimum outcomes (see Fig. 1). In most cases, we show completeness, i.e., 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.
- ▷ In the complexity analysis, we distinguish between optimal and optimum outcomes for each voting scheme (based on the idea of distinguishing weak and strong Con-

dorcet winners). An outcome is optimal if it is not dominated by another, and optimum if it dominates all others.

Many of our results are intractability results, where the problems are put at various levels of the polynomial hierarchy. However, although intractability is usually a "bad" news, these results are quite interesting, because (for most of these tasks) in the original study of Rossi, Venable, and Walsh (2004), only EXPTIME upper-bounds were given. Even more interestingly, some of these problems are actually tractable, since they belong to PTIME or even LOGSPACE. Note that, although our hardness results are given for a subclass of mCP-nets, they are lower bounds also over the broader class of mCP-nets with partial CP-nets.

As side products of our work, we furthermore provide two interesting results on complexity theory more in general:

- ▷ We are the first, to our knowledge, to define the Θ_2^{P}-complete COMP-SAT problem of deciding whether a set contains more satisfiable Boolean formulas than those belonging to another set. COMP-SAT is very intuitive and useful as a Θ_2^{P}-hard problem for a reduction.
- ▷ One of the problems analyzed here is shown to be among the very few known natural $\mathrm{D}_2^{\mathrm{P}}$-complete problems.

Organization of the paper. After preliminaries on CP-nets and mCP-nets, we give an overview of the results. Then, we show some of the nets that are used in our hardness proofs, next we report some basic results on CP-nets, and subsequently each further section is devoted to the complexity analysis of a specific voting scheme. For most of the results, we give only proof sketches and intuitions. Details will be provided in a forthcoming extended paper.

Preliminaries

CP-nets. A CP-net N is a triple $\langle \mathcal{G}_N, Dom_N, CPT_N \rangle$, where $\mathcal{G}_N = \langle \mathcal{F}_N, \mathcal{E}_N \rangle$ is a directed graph whose vertices in $\mathcal{F}_N$ (also called *features*) are labelled through two functions Dom_N and CPT_N, which associate with every feature $F \in \mathcal{F}_N$ a domain $Dom_N(F)$ and a CP table $CPT_N(F)$, respectively. The domain of F is the set of all the values that F may assume in the possible outcomes. CP tables encode preferences over feature values. The CP table of F is a two column table with a row for *any* possible values' combination of *all* the parent features of F in $\mathcal{G}_N$, and in each row there is a total order over $Dom_N(F)$. This order encodes the agent preferences for F's values when specific values of F's parents are considered: $\overline{f} \succ f$ denotes that value $\overline{f}$ is preferred to value f. If F has no parents, its CP table has only one row with a total order over $Dom_N(F)$.

A CP-net N is *acyclic* if $\mathcal{G}_N$ is acyclic; N is *singly connected*, or *polynomially connected*, if, given any two distinct features G and F, in $\mathcal{G}_N$ there is only one path, or there are at most polynomially many distinct paths, respectively, connecting G to F; N is *binary* if each feature of N has two domain values. If N is a binary CP-net, and A is one of its features, then we usually assume that $Dom_N(A) = \{a, \overline{a}\}$.

CP-nets' semantics is based on the concepts of improving and worsening flip, which are symmetric concepts leading to the same definition: In this paper, we consider improving flips. An outcome is an object in which all the features of N have a specific value. For a feature F and an outcome α, $\alpha[F]$ is the value of F in α. Let F be a feature, and let α, β be two outcomes differing *only* on the value of F. Flipping variable F from $\alpha[F]$ to $\beta[F]$ is an *improving flip* (of F in N) iff, in the row of F's CP table associated with the specific values of the parents of F in α, $\beta[F] \succ \alpha[F]$. For a CP-net N, we define the (*extended*) *preference graph* $G_N = \langle V_N, E_N \rangle$ of N, where the nodes V_N are all the possible outcomes of N, and given two nodes (outcomes) $\alpha, \beta \in V_N$ the directed edge $\langle \alpha, \beta \rangle$ is in E_N iff there is an improving flipping (of any single feature) from α to β. To conclude, an agent prefers outcome β to α, and β *dominates* α (in N), denoted $\beta \succ_N \alpha$, iff there is path from α to β in G_N. If there is not such a path in G_N, then β does not dominate α, and it is denoted by $\beta \not\succ_N \alpha$. If $\beta \not\succ_N \alpha$ and $\alpha \not\succ_N \beta$, then β and α are *incomparable*, and we denote it by $\beta \bowtie_N \alpha$.

For an acyclic CP-net N, G_N is acyclic, and the preferences encoded by N are consistent (Boutilier et al. 2004), i.e., there is no outcome α with $\alpha \succ_N \alpha$. Furthermore, in G_N there is only one outcome o_N, called the optimal or optimum, which is not dominated by any other outcome and dominates all the others (Boutilier et al. 2004). $Rank_N(\alpha)$, denoting the rank of α in N, is the length of the shortest path in G_N from α to o_N (Rossi, Venable, and Walsh 2004).

In this paper, unless stated otherwise, all the CP-nets considered are acyclic, binary, and polynomially connected.

***m*CP-nets.** An mCP-net $\mathcal{M}$ is a collection $\langle N_1, \ldots, N_m \rangle$ of m CP-nets defined over the same set of features which, in turn, have the same domain. The "m" of an mCP-net stands for "multiple" agents and also indicates that the preferences of m agents are modeled in the net, so a 3CP-net is an mCP-net with $m = 3$. In the original definition of mCP-nets, *partial* CP-nets are allowed to be constituent of mCP-nets. In this paper, we restrict our attention to mCP-nets in which agents' individual nets are standard CP-nets. The difference is that we do not allow for non-ranked features in agents' CP-nets, and hence for us there is no distinction between private, shared, and visible features (see Rossi, Venable, and Walsh 2004 for a definition of them). Note that, although the features of the individual CP-nets are the same, the graph structure of the individual nets may be different, i.e., the links between the features in the various individual CP-nets may vary.

The semantics of mCP-nets is based on voting. Let $\mathcal{M} = \langle N_1, \ldots, N_m \rangle$ be an mCP-net, and let α, β be two outcomes. We define the sets $S^{\succ}_{\mathcal{M}}(\alpha, \beta) = \{i \mid \alpha \succ_{N_i} \beta\}$, $S^{\prec}_{\mathcal{M}}(\alpha, \beta) = \{i \mid \alpha \prec_{N_i} \beta\}$, and $S^{\bowtie}_{\mathcal{M}}(\alpha, \beta) = \{i \mid \alpha \bowtie_{N_i} \beta\}$. $Rank_{\mathcal{M}}(\alpha) = \sum_{1 \le i \le m} Rank_{N_i}(\alpha)$ is the rank of α in $\mathcal{M}$ (Rossi, Venable, and Walsh 2004).

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

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

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

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

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

An mCP-net is acyclic, binary, singly connected, or polynomially connected, if all its CP-nets are acyclic, binary, singly connected, or polynomially connected, respectively. Also for mCP-nets, unless stated otherwise, the considered mCP-nets are acyclic, binary, and polynomially connected.

Complexity classes. We assume the reader to be familiar with basic concepts of computational complexity and of the polynomial hierarchy (PH); see (Johnson 1990). We next recall some less usual classes that we encounter in our results. Θ_2^{P} is the class of languages recognizable by deterministic Turing machines in polynomial time querying at most logarithmic-many times an NP oracle. $\mathrm{D}_2^{\mathrm{P}}$ and $\mathrm{D}_3^{\mathrm{P}}$ (see, e.g., Wooldridge and Dunne 2004) generalize the class D^{P} (Papadimitriou and Yannakakis 1984): more generally, the class $\mathrm{D}_k^{\mathrm{P}} = \{L \mid L = L' \cap L'', L' \in \Sigma_k^{\mathrm{P}}, L'' \in \Pi_k^{\mathrm{P}}\}$, $k \geq 1$, is the "conjunction" of Σ_k^{P} and Π_k^{P}; in particular, $\mathrm{D}_1^{\mathrm{P}} = \mathrm{D}^{\mathrm{P}}$.

Overview of the Results

Given an mCP-net $\mathcal{M}$, for a voting scheme s, we analyze five tasks: Given two outcomes α, β, decide wether $\alpha \succ^s_{\mathcal{M}} \beta$ (s-DOMINANCE); decide whether an outcome α is s optimal or s optimum (IS-s-OPTIMAL and IS-s-OPTIMUM, respectively); and decide whether $\mathcal{M}$ has an s optimal or an s optimum outcome at all (EXISTS-s-OPTIMAL and EXISTS-s-OPTIMUM, respectively). For the majority voting, the problems' names reflect the fact that a majority optimal outcome and a majority optimum outcome are also called a weak Condorcet winner and a (strong) Condorcet winner, respectively. A summary of our results is in Figure 1.

We observe that Rank voting is the easiest semantics to evaluate. This is due to the fact that rank dominance is feasible in polynomial time, and hence all the other tasks are tractable. Pareto voting is the second least difficult semantics to compute. Pareto dominance is NP-complete, however, this complexity does not carry over to the other tasks and cause a substantial increase of their complexity, because compared to majority and max voting, Pareto voting is structurally simpler. In fact, since Pareto voting is based on unanimity, 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 the Pareto voting makes the other tasks not more difficult than the dominance test, or even tractable. Max voting turns out to be more complex than majority voting, as the dominance test of the former is more complex than the one of the latter. In fact, if to show that an outcome majority dominates another, it suffices to guess a big enough set of agents (indeed, the majority) voting for that dominance (feasible in NP), for max voting this is not enough. In fact, to decide whether an outcome α max dominates an outcome β, we have to carry out a precise counting of the agents preferring α to β, and vice-versa, and this is more complex than plain NP.

	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 PTIME
MAJORITY	MAJORITY-DOMINANCE	NP-complete
	IS-WEAK-CONDORCET	co-NP-complete
	EXISTS-WEAK-CONDORCET	Σ_2^{P}-complete
	IS-STRONG-CONDORCET	Π_2^{P}-complete
	EXISTS-STRONG-CONDORCET	$\mathrm{D}_2^{\mathrm{P}}$-complete
MAX	MAX-DOMINANCE	Θ_2^{P}-complete
	IS-MAX-OPTIMAL	Π_2^{P}-complete
	EXISTS-MAX-OPTIMAL	in Σ_3^{P}
	IS-MAX-OPTIMUM	Π_2^{P}-complete
	EXISTS-MAX-OPTIMUM	in $\mathrm{D}_3^{\mathrm{P}}$
RANK	RANK-DOMINANCE	in PTIME
	IS-RANK-OPTIMAL	in PTIME
	EXISTS-RANK-OPTIMAL	$\Theta(1)^*$
	IS-RANK-OPTIMUM	in PTIME
	EXISTS-RANK-OPTIMUM	in PTIME

Figure 1: Summary of the results on acyclic binary polynomially connected mCP-Nets. *Descends from results in (Rossi, Venable, and Walsh 2004).

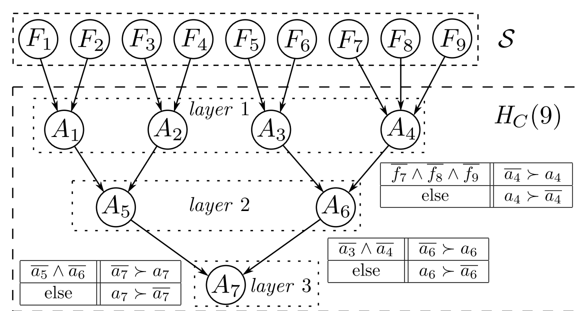

Figure 2: An interconnecting net $H_{\mathrm{C}}(9)$. Not all the CP tables are reported in the figure.

Building Blocks

We introduce some CP-nets used as parts of bigger nets.

The "interconnecting" CP-net $H_{\mathrm{C}}(m)$ aims at "propagating the information" that all the features of a set $\mathcal{S}$ have been flipped to their overlined value. See Figure 2 for an $H_{\mathrm{C}}(9)$ interconnecting net. $H_{\mathrm{C}}(m)$ is an acyclic DAG partitioned into layers. In $H_{\mathrm{C}}(m)$, 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 of the same layer have no common parents, and in every layer at most one feature has three parents. In the first layer, these connection properties hold w.r.t. the features of $\mathcal{S}$. The layer with a unique feature, called the apex, is the last layer of the net. For each feature A of $H_{\mathrm{C}}(m)$, $\overline{a} \succ a$ iff the value of all A's parents is overlined. It is easy to see that, when all the features in $\mathcal{S}$ have overlined values in an outcome α, there is an improving flipping sequence from α to an outcome β in which the values of all the features of $H_{\mathrm{C}}(m)$ are overlined.

Many of our complexity results exploit two CP-nets encoding satisfiability of 3CNF Boolean formulas. The first

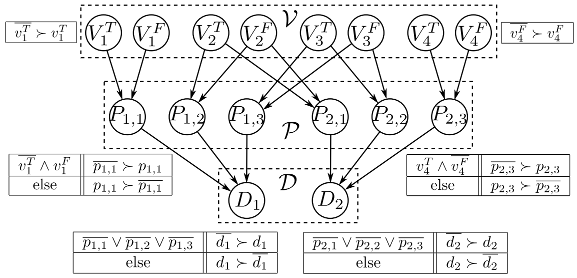

Figure 3: The CP-net $F(\phi)$, where $\phi(x_1, x_2, x_3, x_4) = (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_2 \vee x_3 \vee \neg x_4)$. Not all the CP tables are reported in the figure.

net is inspired by a similar net in (Boutilier et al. 2004). Ours has a smaller indegree (three instead of six) allowing us to show a stronger result for dominance hardness on CP-nets.

Let $\phi(X)$ be a 3CNF Boolean formula defined over variables $X = \{x_1, \ldots, x_n\}$, with set of clauses $C = \{c_1, \ldots, c_m\}$. $\ell_{j,k}$ denotes the k-th literal of the j-th clause. From ϕ, we build the CP-net $F(\phi)$, called formula net, as follows (see Figure 3 for an illustration). Features of $F(\phi)$ are: For each $x_i \in X$, there are *variable features* V_i^T and V_i^F; for each $c_j \in C$, there is the *clause feature* D_j; for each $\ell_{j,k}$, there is the *literal feature* $P_{j,k}$.

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 $\langle P_{j,k}, D_j\rangle$.

CP tables of $F(\phi)$ are as follows. For each variable feature F, $\overline{f} \succ f$. For each literal $\ell_{j,k}$, if $\ell_{j,k} = x_i$, then, for the literal feature $P_{j,k}$, $\overline{p_{j,k}} \succ p_{j,k}$ iff features V_i^T and V_i^F have values $\overline{v_i^T}$ and v_i^F, respectively; otherwise, if $\ell_{j,k} = \neg x_i$, $\overline{p_{j,k}} \succ p_{j,k}$ iff features V_i^T and V_i^F have values v_i^T and $\overline{v_i^F}$, respectively. For each clause $c_j \in C$, clause feature D_j has CP table $\overline{d_j} \succ d_j$ iff there is at least one among the features $P_{j,1}$, $P_{j,2}$, and $P_{j,3}$ having an overlined value.

Net $\overline{F}(\phi)$ is similar to $F(\phi)$. All the CP tables of $\overline{F}(\phi)$ are similar to those of $F(\phi)$ with the difference that, for all the variable and clause features, non-overlined values are exchanged with overlined values, and vice-versa. $F(\phi)$ and $\overline{F}(\phi)$ are binary, acyclic, singly connected, the indegree of their features is at most three, and they can be built in polynomial time in the size of ϕ. The following can be proven.

Lemma 1. *Let ϕ be a 3CNF Boolean formula, and let α and β be two outcomes such that in α the values of all the features are non-overlined, and in β the values of all and only the variable and clause features are overlined. Then: ϕ is satisfiable iff $\beta \succ_{F(\phi)} \alpha$ (resp., $\alpha \succ_{\overline{F}(\phi)} \beta$); and ϕ is unsatisfiable iff $\alpha \bowtie_{F(\phi)} \beta$ (resp., $\alpha \bowtie_{\overline{F}(\phi)} \beta$).*

Summarized formula nets $F_s(\phi)$ improve on $F(\phi)$. $F_s(\phi)$ relates ϕ's satisfiability with the flip of only two features, U_1 and U_2, instead of with all the variable and clause features. This advantage causes the loss of the single connectedness.

Let $\phi(X)$ be a 3CNF Boolean formula defined over variables $X = \{x_1, \ldots, x_n\}$, with set of clauses $C = \{c_1, \ldots, c_m\}$. From ϕ, we build CP-net $F_s(\phi)$ as follows. Net $F_s(\phi)$ embeds a net $F(\phi)$ with its features and links. In $F_s(\phi)$, there is a net $H_{\mathrm{C}}(m)$ attached to all the clause features of $F(\phi)$, and there are two more features U_1 and U_2. There is a link from U_1 to all the variable and literal features of $F(\phi)$; and a link from the apex of $H_{\mathrm{C}}(m)$ to U_2. CP tables of the features are: For U_1, $\overline{u_1} \succ u_1$. For variable and literal features F, CP tables are as in $F(\phi)$ with the additional requirement that, for $\overline{f} \succ f$ to hold, the value of U_1 must be u_1. CP tables of features in $H_{\mathrm{C}}(m)$ are as usual. For U_2, $\overline{u_2} \succ u_2$ iff the value of the apex of $H_{\mathrm{C}}(m)$ is $\overline{a}$.

The intuition behind the structure of $F_s(\phi)$ is that by linking U_1 to variable and literal features, these ones cannot be flipped to their overlined values once U_1 is $\overline{u_1}$. So, 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. The following can be proven.

Lemma 2. *Let ϕ be a 3CNF Boolean formula, and let α and β be two outcomes such that in α the values of all the features are non-overlined, and in β the values of U_1 and of U_2 are overlined. Then: ϕ is satisfiable iff $\beta \succ_{F_s(\phi)} \alpha$; and ϕ is unsatisfiable iff $\alpha \bowtie_{F_s(\phi)} \beta$.*

Basic Tasks on CP-nets

We now discuss the complexity of tasks on CP-nets, which will be at the base of the complexity analysis of voting tasks.

Given a CP-net N, computing its unique optimal outcome o_N is feasible in linear time through the "forward sweep" procedure (Boutilier et al. 2004). Hence, as pointed out by Rossi, Venable, and Walsh (2004), to decide whether α is optimal, it suffices to compute o_N (in polynomial time), and then compare α to o_N. However, this problem actually belongs to a subclass of PTIME. In fact, if α does not equal o_N, there is a feature for which there is an improving flip.

Theorem 3. *Let N be a CP-net, and let α be an outcome. Deciding whether α is optimal in N is in* LOGSPACE.

Consider the task of computing $Rank_N(\alpha)$. Since N is assumed acyclic, a shortest path in G_N from α to o_N is one in which features are flipped, if necessary, according to a topological order of $\mathcal{G}_N$. So, to evaluate $Rank_N(\alpha)$, we can compute at first o_N, and then count the number of features whose value in α is different from the value in o_N, i.e.,

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

Observe that Equation (1) is computable in polynomial time.

Lemma 4. *Let N be a CP-net, and let α be an outcome. Computing $Rank_N(\alpha)$ is feasible in polynomial time.*

To conclude this section, we recall the following results.

Theorem 5 (Boutilier et al. 2004)**.** *Let N be a CP-net. Given outcomes α, β, deciding whether $\beta \succ_N \alpha$ is in* NP.

We emphasize here that the complexity of dominance, stated in the theorem above, is valid for acyclic polynomially connected CP-nets. The exact complexity for dominance on general acyclic CP-nets is still open, and, in particular, it is currently unknown whether it belongs to NP or not.

The following theorem descends from Lemma 1.

Theorem 6 (improved over Boutilier et al. 2004)**.** *Let N be a CP-net, and let α, β be two outcomes. Deciding whether $\beta \succ_N \alpha$ is* NP-hard. *Hardness holds even if N is singly connected, and the indegree of the features is at most three.*

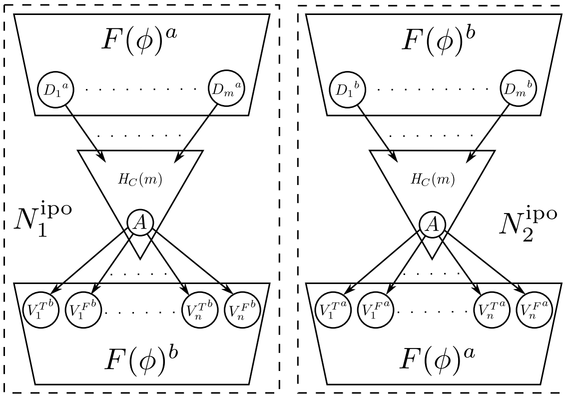

Figure 4: A schematic representation of $\mathcal{M}_{\text{ipo}}(\phi)$.

Pareto Voting

We now focus on Pareto voting, which is based on the concept of unanimity. A witness for an outcome Pareto dominating another is the set of the witnesses of all agents preferring one to the other (see Theorem 5). For the hardness, on 1CP-nets, $\succ^p$ and $\succ$ are equivalent (see Theorem 6).

Theorem 7. *Let $\mathcal{M}$ be an mCP-net, and let α, β be two outcomes. Deciding whether $\beta \succ^p_{\mathcal{M}} \alpha$ is* NP-complete. *Hardness holds even on 1CP-nets.*

Consider the problem IS-PARETO-OPTIMAL: given an mCP-net $\mathcal{M}$ and an outcome α, is α Pareto optimal? We give details of the proofs only for IS-PARETO-OPTIMAL. At first, observe that we can *dis*prove α being Pareto optimal by guessing an outcome β along with the witness that $\beta \succ^p_{\mathcal{M}} \alpha$, and checking the witness (in NP, see Theorem 7).

Theorem 8. *Let $\mathcal{M}$ be an mCP-net, and let α be an outcome. Deciding whether α is Pareto optimal is in* co-NP.

For the hardness we use the following construction (see Figure 4 for a schematic illustration). Let $\phi(X)$ be a 3CNF Boolean formula defined over variables $X = \{x_1, \ldots, x_n\}$, with set of clauses $C = \{c_1, \ldots, c_m\}$. From ϕ we build the 2CP-net $\mathcal{M}_{\text{ipo}}(\phi) = \langle N_1^{\text{ipo}}, N_2^{\text{ipo}} \rangle$.

In N_1^{ipo}, there are two complete copies of net $F(\phi)$, in particular, $F(\phi)^a$ and $F(\phi)^b$. Features of these two nets have the superscript a or b to make them different. Clause features of $F(\phi)^a$, i.e., $D_1{}^a, \ldots, D_m{}^a$, are attached to an interconnecting net $H_{\text{C}}(m)$. Apex A of $H_{\text{C}}(m)$ is attached to *all* the variable features of $F(\phi)^b$. Since variable features of $F(\phi)^b$ now have one parent (the apex A of $H_{\text{C}}(m)$), their CP tables are a bit different from those in $F(\phi)$: For variable features F of $F(\phi)^b$, $\overline{f} \succ f$ iff the apex A has value $\overline{a}$.

N_2^{ipo} is similar to N_1^{ipo}, however $H_{\text{C}}(m)$ attaches clause features of $F(\phi)^b$ to variable features of $F(\phi)^a$, and CP tables are adjusted accordingly. $\mathcal{M}_{\text{ipo}}(\phi)$ is acyclic, binary, polynomially connected, features' indegree is at most three, and $\mathcal{M}_{\text{ipo}}(\phi)$ can be computed in polynomial time from ϕ.

Lemma 9. *Let $\phi(X)$ be a 3CNF Boolean formula, consider the 2CP-net $\mathcal{M}_{\text{ipo}}(\phi)$, and let α be the outcome assigning non-overlined values to all the features. Then, $\phi(X)$ is satisfiable iff α is not Pareto optimal.*

Proof. At first, let us prove the following two claims that will be used to prove the statement of the lemma.

Claim A. *If $\phi(X)$ is unsatisfiable, then, for any outcome β, $\beta \succ_{N_1^{\text{ipo}}} \alpha \Rightarrow \beta \not\succ_{N_2^{\text{ipo}}} \alpha$, and $\beta \succ_{N_2^{\text{ipo}}} \alpha \Rightarrow \beta \not\succ_{N_1^{\text{ipo}}} \alpha$.*

Since ϕ is unsatisfiable, by Lemma 1, in N_1^{ipo} there is no improving flipping sequence from α to outcomes in which the values of *all* the clause features of $F(\phi)^a$ are overlined, or to outcomes in which the value of the apex of $H_{\text{C}}(m)$ is overlined, or to outcomes in which the value of any of the features of $F(\phi)^b$ is overlined. Thus, in N_1^{ipo}, any improving flipping sequence from α leads 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, in N_2^{ipo}, any improving flipping sequence from α leads to outcomes in which values of features of $F(\phi)^a$ are *non*-overlined, while values of feature of $F(\phi)^b$ are overlined.

Therefore, if β is such that $\beta \succ_{N_1^{\text{ipo}}} \alpha$, then $\beta \not\succ_{N_2^{\text{ipo}}} \alpha$. Symmetrically, if $\beta \succ_{N_2^{\text{ipo}}} \alpha$, then $\beta \not\succ_{N_1^{\text{ipo}}} \alpha$.

Claim B. *If $\phi(X)$ is satisfiable, then the outcome β assigning overlined values to all the variable and clause features of $F(\phi)^a$ and $F(\phi)^b$, and to all the features of $H_{\text{C}}(m)$, is such that $\beta \succ_{N_1^{\text{ipo}}} \alpha$ and $\beta \succ_{N_2^{\text{ipo}}} \alpha$.*

Since ϕ is satisfiable, by Lemma 1, in N_1^{ipo} there is an improving flipping sequence from α to an outcome α' assigning overlined values to *all* the variable and clause features of $F(\phi)^a$. By the definition of $H_{\text{C}}(m)$, in N_1^{ipo} there is an improving flipping sequence from α' to an outcome α'' assigning overlined values to all the features of $H_{\text{C}}(m)$. α'' can be further improved by a flipping sequence to an outcome β assigning overlined values to *all* the variable and clause features of $F(\phi)^b$ because ϕ is satisfiable. Symmetrically, in N_2^{ipo}, there is an improving flipping sequence from α to the same β. Therefore, $\beta \succ_{N_1^{\text{ipo}}} \alpha$ and $\beta \succ_{N_2^{\text{ipo}}} \alpha$.

We now show that ϕ is satisfiable iff α is not Pareto optimal. ($\Rightarrow$) Assume that ϕ is satisfiable. By Claim B, there is an outcome β preferred to α by all the agents of $\mathcal{M}_{\text{ipo}}(\phi)$. Thus, α is not Pareto optimal. ($\Leftarrow$) Assume that ϕ is not satisfiable. By Claim A, no outcome β is preferred to α by all the agents of $\mathcal{M}_{\text{ipo}}(\phi)$. Thus, α is Pareto optimal. $\square$

Now, observe that Lemma 9 essentially states that the UNSAT problem can be reduced to IS-PARETO-OPTIMAL.

Theorem 10. *Let $\mathcal{M}$ be an mCP-net, and let α be an outcome. Deciding whether α is Pareto optimal is* co-NP-hard. *Hardness holds even if $\mathcal{M}$ is acyclic, binary, polynomially connected, and is a 2CP-net.*

Deciding whether an mCP-net has a Pareto optimal outcome is trivial, because there is always one (Rossi, Venable, and Walsh 2004). Observe that, due to the unanimity semantics, an mCP-net has a Pareto optimum outcome iff all its CP-nets have the very same individual optimal outcome (that is also Pareto optimum). By combining this with Theorem 3 and Lemma 4, we can state the following.

Theorem 11. *Let $\mathcal{M}$ be an mCP-net. Deciding whether $\mathcal{M}$ has a Pareto optimum outcome is in* PTIME*; deciding whether an outcome α is Pareto optimum is in* LOGSPACE.

Majority Voting

In this section, we concentrate on majority voting, which is related to Condorcet winners. In fact, a majority optimal outcome is a weak Condorcet winner, while a majority optimum outcome is a (strong) Condorcet winner. Observe that it is possible to design four different acyclic binary singly-connected CP-nets with dominance relationships: $\overline{a}\overline{b} \succ_{N_1} \overline{a}b \succ_{N_1} ab \succ_{N_1} a\overline{b}$; $\overline{a}b \succ_{N_2} ab \succ_{N_2} a\overline{b} \succ_{N_2} \overline{a}\overline{b}$; $ab \succ_{N_3} a\overline{b} \succ_{N_3} \overline{a}\overline{b} \succ_{N_3} \overline{a}b$; and $a\overline{b} \succ_{N_4} \overline{a}\overline{b} \succ_{N_4} \overline{a}b \succ_{N_4} ab$. Interestingly, the 4CP-net constituted by the above nets do not have weak and strong Condorcet winners.

Theorem 12. *There are acyclic binary singly-connected mCP-nets not having weak and strong Condorcet winners.*

Let us focus now on the majority dominance. Observe that, $\beta \succ^{maj}_{\mathcal{M}} \alpha$ iff $|S^{\succ}_{\mathcal{M}}(\beta, \alpha)| > \frac{m}{2}$. Hence, a certificate consists in the witnesses of more than $\frac{m}{2}$ agents preferring β to α (see Theorem 5). On the hardness side, on 1CP-nets, $\succ^{maj}$ and $\succ$ are equivalent (see Theorem 6).

Theorem 13. *Let $\mathcal{M}$ be an mCP-net, and let α, β be two outcomes. Deciding whether $\beta \succ^{maj}_{\mathcal{M}} \alpha$ is* NP-complete. *Hardness holds even on 1CP-nets.*

To prove an outcome α *not* to be a weak Condorcet winner, we guess an outcome β and the witness of $\beta \succ^{maj}_{\mathcal{M}} \alpha$ (in NP, see Theorem 13). For the hardness, on 2CP-nets, $\succ^{maj}$ and $\succ^{p}$ are equivalent (see Theorem 10).

Theorem 14. *Let $\mathcal{M}$ be an mCP-net. Deciding whether an outcome α is a weak Condorcet winner is* co-NP-complete. *Hardness holds even on 2CP-nets.*

We now focus on deciding the existence of a weak Condorcet winner. We can prove that $\mathcal{M}$ has a weak Condorcet winner by guessing an outcome α (in NP), and checking that α is a weak Condorcet winner (in co-NP, see Theorem 14). To show the hardness, an involved construction is required. We give just an intuition. The reduction is from the problem $\mathrm{QBF}^{CNF}_{2,\exists}$: given a 3CNF formula $\phi(Y, X)$ defined on two disjoint sets of variables X and Y, is $\Phi = (\exists X)(\forall Y)\neg\phi(X, Y)$ valid? $\mathrm{QBF}^{CNF}_{2,\exists}$ is Σ^{P}_2-hard (Stockmeyer 1976). From $\phi(X, Y)$, it is possible to build a 6CP-net $\mathcal{M}_{\mathrm{ewc}}(\phi)$ such that $\phi(X, Y)$ is a 'yes'-instance of $\mathrm{QBF}^{CNF}_{2,\exists}$ iff $\mathcal{M}_{\mathrm{ewc}}(\phi)$ has a weak Condorcet winner. The guiding principle is that of relating assignments on X and outcomes. Two of the nets of $\mathcal{M}_{\mathrm{ewc}}(\phi)$ are $F_s(\phi)$, in which a distinction between variable features from X and from Y is made, and the roles of U_1 and U_2 are exchanged. An assignment σ_X to variables in X is associated with outcome β_{σ_X} in which $\beta_{\sigma_X}[V^T_i] = \overline{v^T_i}$ iff $\sigma_X[x_i] = \mathit{true}$, and $\beta_{\sigma_X}[V^F_i] = \overline{v^F_i}$ iff $\sigma_X[x_i] = \mathit{false}$. This is the core idea: Let α be the outcome assigning overlined values to U_1 and U_2. If σ_X is a *not* a witness for Φ, i.e., $\phi(X/\sigma_X, Y)$ is satisfiable, then, in the nets $F_s(\phi)$, there is an improving flipping sequence from β_{σ_X} to α, and, in the overall mCP-net $\mathcal{M}_{\mathrm{ewc}}(\phi)$, $\alpha \succ^{maj}_{\mathcal{M}_{\mathrm{ewc}}(\phi)} \beta$. If σ_X *is* a witness for Φ, i.e., $\phi(X/\sigma_X, Y)$ is *not* satisfiable, then, in the nets $F_s(\phi)$, there is no improving flipping sequence from β_{σ_X} to α, and, in the overall mCP-net $\mathcal{M}_{\mathrm{ewc}}(\phi)$, β_{σ_X} is not majority dominated by any outcome. To have the reduction working, $\mathcal{M}_{\mathrm{ewc}}(\phi)$ has CP-nets designed such that any outcome not in the form of a β_{σ_X} is majority dominated by some other outcome.

Theorem 15. *Let $\mathcal{M}$ be an mCP-net. Deciding whether $\mathcal{M}$ has a weak Condorcet winner is* Σ^{P}_2-complete. *Hardness holds even on 6CP-nets.*

Let us focus on the problem of deciding whether an outcome is a Condorcet winner. We can show that α is *not* a Condorcet winner by guessing an outcome β (in NP) and checking that $\alpha \not\succ^{maj}_{\mathcal{M}} \beta$ (in co-NP, see Theorem 13). The proof of the hardness uses similar ideas to those for hardness in Theorem 15. However, in this reduction, outcomes not in the form of a β_{σ_X} are all majority dominated by α.

Theorem 16. *Let $\mathcal{M}$ be an mCP-net. Deciding whether an outcome α is a Condorcet winner is* Π^{P}_2-complete. *Hardness holds even on 3CP-nets.*

Deciding the existence of a Condorcet winner is a very interesting problem, since it is one of the very few natural problems known to be $\mathrm{D}^{\mathrm{P}}_2$-complete. Observe that the set S of mCP-nets having a Condorcet winner is such that $S = A \cap B$, where A is the set of mCP-nets having at least a weak Condorcet winner, and $B = C \cap D$, where C is the set of mCP-nets having less than two distinct weak Condorcet winners, and D is the set of mCP-nets having no weak Condorcet winner that is not strong. Deciding A is Σ^{P}_2-complete (see Theorem 15), while deciding B can be shown to be in Π^{P}_2, and it can be shown to be Π^{P}_2-hard by the reduction for hardness in Theorem 15.

Theorem 17. *Let $\mathcal{M}$ be an mCP-net. Deciding whether $\mathcal{M}$ has a Condorcet winner is* $\mathrm{D}^{\mathrm{P}}_2$-complete. *Hardness holds even on 6CP-nets.*

Max Voting

We now analyze the complexity of max voting tasks. At first, observe that Theorem 12 implies also that there are acyclic binary singly-connected mCP-nets without max optimal and optimum outcomes. Let us consider max dominance. Observe that $|S^{\succ}_{\mathcal{M}}(\beta, \alpha)| > \max(|S^{\prec}_{\mathcal{M}}(\beta, \alpha)|, |S^{\bowtie}_{\mathcal{M}}(\beta, \alpha)|)$ iff $|S^{\succ}_{\mathcal{M}}(\beta, \alpha)| > |S^{\prec}_{\mathcal{M}}(\beta, \alpha)|$ *and* $2|S^{\succ}_{\mathcal{M}}(\beta, \alpha)| > m - |S^{\prec}_{\mathcal{M}}(\beta, \alpha)|$. So, to decide whether $\beta \succ^{max}_{\mathcal{M}} \alpha$, it suffices to compute $|S^{\succ}_{\mathcal{M}}(\beta, \alpha)|$ and $|S^{\prec}_{\mathcal{M}}(\beta, \alpha)|$, which can be carried out in polynomial time via binary searches on the range $[0, m]$ by querying an NP oracle. For the hardness, there is a reduction from COMP-SAT: given two sets $\langle A, B\rangle$ of 3CNF formulas, is the number of the satisfiable formulas in A greater than the number of the satisfiable ones in B? COMP-SAT can be shown to be Θ^{P}_2-hard from a problem in (Spakowski and Vogel 2000). The idea of the reduction is to encode each formula ϕ of A in a CP-net $F(\phi)$, and each formula φ of B in a CP-net $\overline{F}(\varphi)$. Outcomes α and β are that assigning non-overlined values to all the features, and that assigning overlined values to all the variable and clause features, respectively. Some more nets are added to have the max semantics working according to the semantics of COMP-SAT. With this construction, it can be shown that $\langle A, B\rangle$ is a 'yes'-instance of COMP-SAT iff β max dominates α.

Theorem 18. *Let $\mathcal{M}$ be an mCP-net, and let α, β be two outcomes. Deciding whether $\beta \succ_{\mathcal{M}}^{max} \alpha$ is* Θ_2^{P}-complete.

Let us see how to decide whether an outcome α is max optimal. To *dis*prove it, we can guess an outcome β (in NP), and check that $\beta \succ_{\mathcal{M}}^{max} \alpha$ (in Θ_2^{P}, see Theorem 18). Moreover, for the hardness, we can show a reduction using similar ideas to those for the hardness in Theorem 15, however CP-nets are designed to exploit the max voting semantics.

Theorem 19. *Let $\mathcal{M}$ be an mCP-net, and let α be an outcome. Deciding whether α is max optimal is* Π_2^{P}-complete. *Hardness holds even on* 4*CP-nets.*

To verify that $\mathcal{M}$ has a max optimal outcome, we guess an outcome α (in NP), and then we check that α is max optimal (in Π_2^{P}, see Theorem 19).

Theorem 20. *Let $\mathcal{M}$ be an mCP-net. Deciding whether $\mathcal{M}$ has a max optimal outcome is in Σ_3^{P}.*

To prove an outcome α *not* to be a max optimum, we guess an outcome β (in NP), and check that $\alpha \not\succ_{\mathcal{M}}^{max} \beta$ (in Θ_2^{P}, see Theorem 18). For the hardness, on 3CP-nets, $\succ^{max}$ and $\succ^{maj}$ are equivalent (see Theorem 16).

Theorem 21. *Let $\mathcal{M}$ be an mCP-net, and let α be an outcome. Deciding whether α is a max optimum is* Π_2^{P}-complete. *Hardness holds even on* 3*CP-nets.*

We next focus on deciding whether an mCP-net has a max optimum outcome. Observe that the set S of mCP-nets having a max optimum outcome is such that $S = A \cap B$, where A is the set of mCP-nets having at least a max optimal outcome, and $B = C \cap D$, where C is the set of mCP-nets having less than two distinct max optimal outcomes, and D is the set of mCP-nets having no max optimal outcome that is not optimum. Deciding A is in Σ_3^{P} (see Theorem 20), and deciding B can be shown to be in Π_3^{P}.

Theorem 22. *Let $\mathcal{M}$ be an mCP-net. Deciding whether $\mathcal{M}$ has a max optimum outcome is in $\mathrm{D}_3^{\mathrm{P}}$.*

Rank Voting

In this last section, we focus on rank voting. The rank voting complexity results are based on this observation: Given an mCP-net $\mathcal{M} = \langle N_1, \ldots, N_m \rangle$, $Rank_{\mathcal{M}}(\alpha) = \sum_{1 \leq i \leq m} Rank_{N_i}(\alpha) = \sum_{1 \leq i \leq m} |\{F \mid F \in \mathcal{F}_{\mathcal{M}} \wedge \alpha[F] \neq o_{N_i}[F]\}| = \sum_{F \in \mathcal{F}_{\mathcal{M}}} |\{i \mid 1 \leq i \leq m \wedge \alpha[F] \neq o_{N_i}[F]\}|$. This, together with Lemma 4, implies the following result.

Theorem 23. *Let $\mathcal{M}$ be an mCP-net, and let α, β be two outcomes. Deciding whether $\beta \succ_{\mathcal{M}}^{r} \alpha$ is in* PTIME.

An outcome α is *average optimal* if, for each feature F, $\alpha[F] \in \arg\min_{v \in Dom_{\mathcal{M}}(F)} |\{i \mid 1 \leq i \leq m \wedge v \neq o_{N_i}[F]\}|$. It can be shown that an outcome is rank optimal iff it is average optimal, because only an average optimal outcome minimizes the value of $Rank_{\mathcal{M}}(\alpha)$. Computing an average optimal outcome of an mCP-net is feasible in polynomial time (we just need to compute the individual optimal outcomes to perform the counting operations).

Theorem 24. *Let $\mathcal{M}$ be an mCP-net, and let α be an outcome. Deciding whether α is rank optimal is in* PTIME.

It was shown in (Rossi, Venable, and Walsh 2004) that an mCP-nets has always a rank optimal outcome. Therefore, it is trivial to decide whether an mCP-nets has one.

Since all and only the average optimal outcomes are rank optimal, if there were more than one average optimal outcome, then there would be no rank optimum one. Observe that checking whether an mCP-net has a unique average optimal outcome is feasible in polynomial time.

Theorem 25. *Let $\mathcal{M}$ be an mCP-net, and let α be an outcome. Deciding whether $\mathcal{M}$ has a rank optimum outcome and deciding whether α is rank optimum are in* PTIME.

Acknowledgments

This work has received funding from the EPSRC grants EP/J008346/1, EP/L012138/1, and EP/M025268/1. Many thanks to Jörg Vogel for pointing us out the reduction proving that COMP-SAT is Θ_2^{P}-hard. We thank the anonymous reviewers for helping us to improve the paper.

References

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

Johnson, D. S. 1990. A catalog of complexity classes. In van Leeuwen, J., ed., *Handbook of Theoretical Computer Science (vol. A)*. Elsevier Science Publishers B.V. 67–161.

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

Li, M.; Vo, Q. B.; and Kowalczyk, R. 2010a. An effcient majority-rule-based approach for collective decision making with cp-nets. In *Proc. of KR 2010*, 578–580.

Li, M.; Vo, Q. B.; and Kowalczyk, R. 2010b. An effcient procedure for collective decision-making with CP-nets. In *Proc. of ECAI 2010*, 375–380.

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

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

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

Rossi, F.; Venable, K. B.; and Walsh, T. 2004. mCP nets: representing and reasoning with preferences of multiple agents. In *Proc. of AAAI 2004*, 729–734.

Spakowski, H., and Vogel, J. 2000. Θ_2^p-completeness: A classical approach for new results. In *Proc. of FST&TCS 2000*, LNCS 1974, 348–360.

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

Wooldridge, M., and Dunne, P. E. 2004. On the computational complexity of qualitative coalitional games. *Artif. Intell.* 158(1):27–73.