Preference Aggregation with Incomplete CP-nets

Adrian Haret[1], **Arianna Novaro**[2] and **Umberto Grandi**[2]
[1]Institute of Logic and Computation, TU Wien
[2]IRIT, University of Toulouse

Abstract

Generalized CP-nets (gCP-nets) extend standard CP-nets by allowing conditional preference tables to be incomplete. Such generality is desirable, as in practice users may want to express preferences over the values of a variable that depend only on partial assignments for other variables. In this paper we study aggregation of gCP-nets, under the name of *multiple gCP-nets* (*mgCP-nets*). Inspired by existing research on mCP-nets, we define different semantics for mgCP-nets and study the complexity of prominent reasoning tasks such as dominance, consistency and various notions of optimality.

Introduction

What committees, modern technology and holiday destinations have in common is that (i) they come in a variety of options, all customizable into many possible configurations, thus generating a combinatorial domain; and (ii) they can be the object of complex, often conflicting preferences. Hence, one challenge in processing batches of preferences over large combinatorial domains is that complex relations over such domains have to be collectively aggregated. Apart from having to provide an intuitive and compact way for expressing preferences over outcomes, a system that aims to offer nuanced suggestions to *groups*, rather than just individuals, must possess means to reason based on information coming from multiple sources.

In this work we study the aggregation of preferences articulated in the language of generalized CP-nets (gCP-nets). Formulas in this language allow users to express preferences over the values of a variable depending on the satisfaction of a pre-condition, e.g., the formula $a \vee b : c \rhd \overline{c}$ expresses the fact that an outcome satisfying $a \vee b$ and c is preferred to an outcome where c is flipped to $\overline{c}$, but is otherwise the same. This allows users to express complex preferences over a large space of outcomes in a compact manner. The language of gCP-nets extends the established framework of CP-nets (Boutilier et al. 2004) by requiring less precision in the specification of the pre-conditions: whereas a CP-net expects a preference statement over a variable to be specified for all possible assignments to the remaining variables, gCP-nets relax this condition. Though this weakening can lead to fewer comparisons in the space of outcomes, it gives users the freedom to be as precise as they wish in their declarations of preference (with the understanding that having to supply complete preference tables can often be burdensome), and it allows them to further refine their preference statements later on.

In the multi-agent scenario envisioned here, a profile of gCP-nets, called *an mgCP-net*, corresponds to the stated preferences of distinct agents. The main task we consider is that of ordering the outcomes in a way that reflects the wishes of the individual agents. To this end we propose four semantics for mgCP-nets that work by aggregating the information contained in the individual gCP-nets. Our semantics are inspired by similar notions defined for mCP-nets (Rossi, Venable, and Walsh 2004; Lukasiewicz and Malizia 2016), though in our context they apply to a broader class of orders and in certain cases need to be adapted.

Since the number of outcomes is typically exponential in the number of variables, we do not consider generating an entire order on outcomes, but rather study the computational complexity of certain key reasoning tasks such as dominance, consistency and various notions of optimality. Allowing incomplete specifications of conditions under which preferences hold (i.e., working with gCP-nets rather than CP-nets) leads to finer-grained notions of optimality, of a type that do not arise in other contexts (e.g., mCP-nets). We identify nine reasoning tasks and study their complexity with respect to the four semantics, using complexity results for individual gCP-nets, some of which are known (Goldsmith et al. 2008), while others are obtained here.

There is a long line of research that uses logic to model preferences, originating with early contributions in philosophical logic (Halldén 1957; von Wright 1963). It is in this tradition that the *ceteribus paribus* interpretation of typical preferences statements is emphasized (Hansson 1996). This interpretation is at the heart of CP-nets (Boutilier et al. 2004) and other formalisms in the same family, e.g., TCP-nets (Brafman and Domshlak 2002; Brafman, Domshlak, and Shimony 2006), conditional preference theories (Wilson 2004), and conditional importance networks (Bouveret, Endriss, and Lang 2009). Other logical frameworks include the general language PL (Bienvenu, Lang, and Wilson 2010) and various modal logics (Boutilier 1994; van Benthem, Girard, and Roy 2009). Generalized CP-nets have been considered before, both under the standard *ceteris paribus* semantics (Domshlak et al. 2009; Boutilier et al. 2004), as well as other semantics (Ciaccia 2007). Significantly for our present purposes, the complexity of most reasoning tasks with respect to a single gCP-net has already been studied (Goldsmith et al. 2008). With respect to the multi-agent case, there is a wealth of research on aggregation of CP-nets (Rossi, Venable, and Walsh 2004; Grandi et al. 2014; Li, Vo, and Kowalczyk 2014; 2015; Lukasiewicz and Malizia 2016), including work on probabilistic CP-nets for handling preferences coming from a group of agents (Cornelio et al. 2013; 2015). Despite the importance of incompleteness in the aggregation of preferences (Rossi, Venable, and Walsh 2011), to the best of our knowledge, aggregation tasks for gCP-nets have not been considered.

The paper is structured as follows. We first introduce the syntax and semantics of gCP-nets. Then we define the computational problems of checking consistency of a gCP-net, as well as several notions of optimality. Finally, we move to the multi-agent setting, where we aggregate a profile of gCP-nets representing preferences of multiple agents, according to different semantics. We conclude by restating some open problems and by pointing to a number of directions for future research.

Preliminaries

Syntax Let $\mathcal{V} = \{X, Y, Z, \dots\}$ be a finite set of *variables*. Each variable $X \in \mathcal{V}$ has a finite domain $D(X) = \{x_1, \dots, x_k\}$ of possible *values*. Slightly abusing notation, we write x_i to indicate an assignment of value $x_i \in D(X)$ to variable X.

In our preference language, presented below, the user can express preferences over the values of one variable depending on what obtains with respect to other variables. First, if $X \in \mathcal{V}$ is a variable and $x_i, x_j \in D(X)$ are two possible values for X, then $x_i \rhd x_j$ is a *preference over* X, with the intuitive meaning that value x_i is preferred to x_j for X. Preferences are allowed only over values of a single variable at a time. If $\mathcal{W} \subseteq \mathcal{V}$ is a set of variables, then *a propositional formula* ψ *over* $\mathcal{W}$ is defined over $\bigcup_{X\in\mathcal{W}} D(X)$ using the familiar propositional connectives, and is interpreted in the intuitive way. Observe that, since variables in $\mathcal{V}$ may have non-binary domains, the formula $\neg x_i$, for $x_i \in D(X)$, is equivalent to $\bigvee_{x_j \in D(X), x_j \neq x_i} x_j$. Formulas such as $x_i \wedge x_j$, for x_i and x_j different values of variable X, are inconsistent.

Example 1. If $D(X) = \{x_1, x_2, x_3\}$, the formula $\neg x_1$ means that either x_2 or x_3 is assigned to X, and hence $\neg x_1$ is equivalent to $x_2 \vee x_3$. Moreover, $x_2 \wedge x_3$ is inconsistent.

If ψ is a propositional formula over variables in $\mathcal{W}$, X is a variable such that $X \notin \mathcal{W}$ and $\pi = x_i \rhd x_j$ is a preference over X, then the formula $\varphi = \psi : \pi$ is a *conditional preference statement*, with the intuitive meaning that if ψ is true, then preferences behave according to π. We will typically call ψ the *pre-condition of* φ. A *gCP-net* $N = \{\varphi_1, \dots, \varphi_n\}$ is a finite set of conditional preference statements. We will sometimes write $\psi : (x_i \rhd x_j \rhd x_k)$ as shorthand for $\psi : (x_i \rhd x_j)$ and $\psi : (x_j \rhd x_k)$. As with standard CP-nets, we can represent a gCP-net with a dependency graph, where nodes are variables and there is an edge from X to Y if values of X occur in the pre-condition of a preference over values of Y. We say that a gCP-net N is *acyclic* if its dependency graph is acyclic.

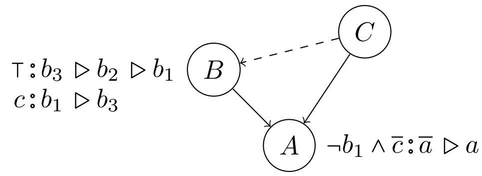

Figure 1: Dependency graphs and conditional preference statements for the gCP-nets N_1 and N_2 of Example 2. The additional dependency brought about by statement φ_3 is shown with a dashed line.

Example 2. On an online booking service for the city of New York, homes are characterized by three features: the type of accommodation one can book (A), which can be either an entire apartment (a) or a room in a shared apartment ($\overline{a}$); the borough where the home is located (B), which can be either Manhattan (b_1), Brooklyn (b_2) or Queens (b_3);[1] the cost (C), which can be high (c) or normal ($\overline{c}$). Thus, the set of variables is $\mathcal{V} = \{A, B, C\}$ and the domains of the variables are $D(A) = \{a, \overline{a}\}$, $D(B) = \{b_1, b_2, b_3\}$ and $D(C) = \{c, \overline{c}\}$.

Alice wishes to use this service to find a place to stay in New York, and submits the gCP net $N_1 = \{\varphi_1, \varphi_2\}$, where the statements are as follows:

$$\begin{array}{lrl} (\varphi_1) & \top : & b_3 \rhd b_2 \rhd b_1, \\ (\varphi_2) & \neg b_1 \wedge \overline{c} : & \overline{a} \rhd a. \end{array}$$

Later on, Alice updates her gCP-net by adding statement φ_3:

$$(\varphi_3) \quad c : \quad b_1 \rhd b_3.$$

Statement φ_1 says that Alice has an unconditional preference of living in Queens over Brooklyn over Manhattan. Statement φ_2 says that if the apartment is not in Manhattan (i.e., it is either in Brooklyn or Queens) and is reasonably priced, then a shared place is better. Statement φ_3 says that if the apartment is costly, then presumably one is going for the royal treatment and so Manhattan is now better than Queens. The dependency graphs for Alice's gCP-nets N_1 and $N_2 = N_1 \cup \{\varphi_3\}$ are depicted in Figure 1.

Semantics An *outcome* assigns to each variable $X \in \mathcal{V}$ a value in their domain $D(X)$. Outcomes (think a particular house, or movie) encode full specifications of objects in terms of their features. We write $x_1 y_2 z_3 \dots$ for the outcome where X is assigned x_1, Y is assigned y_2, Z is assigned z_3, and so on. We denote by $\mathcal{O} = D(X) \times D(Y) \times D(Z) \times \dots$ the set of all outcomes. If $X \in \mathcal{V}$ is a variable, we denote by $o[X]$ the value of outcome o on X; if $\mathcal{W} \subseteq \mathcal{V}$ is a set of variables, we write $o[\mathcal{W}]$ for the values of outcome o on $\mathcal{W}$. We write $o \models \psi$ to say that outcome o satisfies propositional formula ψ in the familiar propositional sense.

Preferences over outcomes are represented using a binary relation $>$ over $\mathcal{O}$. If $o_1 > o_2$ we say that o_1 *dominates* o_2 *with respect to* $>$, the intuitive meaning of which is that o_1

[1] For simplicity, The Bronx and Staten Island are omitted.

is preferred to o_2 in $>$. If $o_1 > o_2$ and $o_2 \not> o_1$, we say that o_1 *strictly dominates* o_2 *with respect to* $>$. If two outcomes o_1 and o_2 are incomparable with respect to $>$, i.e., $o_1 \not> o_2$ and $o_2 \not> o_1$, we write $o_1 \bowtie o_2$. The semantics of conditional preference statements is defined over binary relations on $\mathcal{O}$ using the notion of worsening flips. as follows.

Definition 1. If $x_i, x_j \in D(X)$, $\varphi = \psi : x_i \rhd x_j$ is a conditional preference statement with respect to X, and o_1 and o_2 are two outcomes, then there is a *worsening flip from* o_1 *to* o_2 *sanctioned by* φ if $o_1 \vDash \psi$, $o_2 \vDash \psi$, $o_1[Y] = o_2[Y]$, for any $Y \in \mathcal{V} \setminus \{X\}$ and $o_1[X] = x_i$, $o_2[X] = x_j$.

Intuitively, there is a worsening flip from o_1 to o_2 sanctioned by $\varphi = \psi : (x_i \rhd x_j)$ if outcomes o_1 and o_2 both satisfy condition ψ, and they are identical except for the fact that o_1 assigns x_i to X and o_2 assigns x_j to X. In other words, the preference statement φ is interpreted as saying that if condition ψ is true, then, all else being equal, making x_i true is better than making x_j true. Note that the *ceteris paribus* clause holds for models of ψ as well.

We say that there is an *improving flip from* o_2 *to* o_1 *sanctioned by* φ if there is a worsening flip from o_1 to o_2 sanctioned by φ. If N is a gCP-net, a worsening (respectively, improving) flip from o_2 to o_1 sanctioned by N is a worsening (respectively, improving) flip from o_2 to o_1 sanctioned by some $\varphi \in N$.

Definition 2. If N is a gCP-net and o, o' are two outcomes, then o *dominates* o' *with respect to* N, written $o >_N o'$, if there exists a sequence of outcomes $o_1, \dots, o_k$ such that $o_1 = o$, $o_k = o'$ and, for every $i \in \{1, \dots, k-1\}$, there exists a worsening flip from o_i to o_{i+1} sanctioned by N.

We call $>_N$ the *induced model of* N and often write $>_i$ instead of $>_{N_i}$ when clear from context. There may be a chain of worsening flips starting with an outcome o and ending back on o: a gCP-net is *consistent* if there is no such a chain. In particular, since $>_N$ is transitive, this is equivalent to saying that there is no outcome o such that $o >_N o$.

Example 3. For the scenario described in Example 2 there are 12 possible outcomes. The outcome $o = ab_2\overline{c}$ refers to a reasonably priced and private apartment in Brooklyn. For the variables A and B, $o[\{A, B\}] = ab_2$ refers to the values of outcome o on variables A and B. For the gCP-net $N_1 = \{\varphi_1, \varphi_2\}$ provided by Alice, the induced model $>_1$, as well as the worsening flips induced by adding φ_3, is depicted in Figure 2. Adding statement φ_3 to N_1 results in an inconsistent gCP-net, as the induced model $>_2$ of $N_2 = N_1 \cup \{\varphi_3\}$ contains the sequence of worsening flips ab_3c, ab_2c, ab_1c, ab_3c, which implies that $ab_3c >_2 ab_3c$.

As mentioned, in statements of the form $\psi : \pi$ the *ceteris paribus* assumption holds even for the models of ψ. In Example 2, this means in particular that when interpreting a statement such as $\varphi_2 = \neg b_1 \wedge \overline{c} : \overline{a} \rhd a$, we induce the rankings $\overline{a}b_2\overline{c} > ab_2\overline{c}$ and $\overline{a}b_3\overline{c} > ab_3\overline{c}$, but not the ranking $\overline{a}b_2\overline{c} > ab_3\overline{c}$, though both $\overline{a}b_2\overline{c}$ and $ab_3\overline{c}$ satisfy condition $\neg b_1 \wedge \overline{c}$. While it has been argued that this *ceteris paribus* interpretation of preference statements induces insufficiently many comparisons on outcomes (Ciaccia 2007), we believe it to be justified here, since (i) it does not infer more than what is strictly warranted by the agent's statements, and (ii) it gives the agents more freedom to refine their orders without thereby creating inconsistencies, as the following example illustrates.

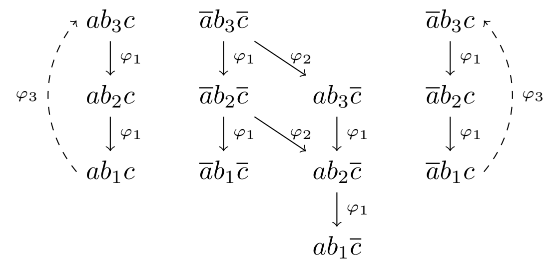

Figure 2: The induced model from Example 3. An arrow from o_1 to o_2 indicates a worsening flip sanctioned by N_1, and arrows are labeled with the preference statement inducing them. Arrows induced by transitivity are omitted.

Example 4. Consider $\mathcal{V}$ as in Example 2. Anna submits the gCP-net $N = \{\overline{a} \vee \overline{c} : b_3 \rhd b_2\}$. If the semantics was not limited to the *ceteris paribus* comparisons, we could derive that $\overline{a}b_3c >_N \overline{a}b_2\overline{c}$, meaning that Anna prefers an expensive shared apartment in Queens to a cheap shared apartment in Brooklyn. Suppose Anna wants to be more precise about her preferences and adds some statements to N, leading to $N' = \{\overline{a} \vee \overline{c} : b_3 \rhd b_2, b_2 \wedge \overline{c} : \overline{a} \rhd a, b_3 \wedge c : a \rhd \overline{a}, a \wedge c : b_2 \rhd b_3, a \wedge b_2 : \overline{c} \rhd c\}$. From N' we now derive $\overline{a}b_2\overline{c} >_{N'} \overline{a}b_3c$, i.e., N' is now inconsistent, which would not have happened under the *ceteris paribus* assumption.

Reasoning with a single gCP-net Generating the entire order on outcomes induced by a gCP-net might be too costly and, in most cases, pointless. For the particular applications we have in mind we do better to focus on some reasoning tasks of interest: these usually concern consistency, dominance relations between specific outcomes, and various notions of optimality. To formally define them, we must first introduce some preliminary notation.

If $>$ is a binary relation on $\mathcal{O}$ and o is an outcome, then o is *weakly non-dominated* if, for any outcome o', it holds that $o' > o$ implies $o > o'$. If there is no outcome o' such that $o' > o$, including $o' = o$, then we say that o is simply *non-dominated*. If $o > o'$ for all outcomes o', then o is a *dominating* outcome. If o is dominating as well as non-dominated, then it is *strongly dominating*. Two quick observations are in order: if o is weakly non-dominated, then it is possible that o is part of a cycle in $>$, as long as the cycle is not dominated by an outcome outside it. Likewise, if o is dominating, then it can be involved in a cycle in $>$.

Given a gCP-net N and outcomes o, o_1 and o_2, the reasoning tasks of interest are summarized in Table 1. The complexity of these reasoning tasks with respect to a single gCP-net has been established in previous work (Goldsmith et al. 2008), with the results presented in Table 2. DOMINANCE, CONSISTENCY, wNON-DOM'ED, DOM'ING, STR-DOM'ING, ∃DOM'ING and ∃STR-DOM'ING have been shown to be PSPACE-

DOMINANCE:	$o_1 >_N o_2$.
CONSISTENCY:	N is consistent.
wNON-DOM'ED:	o is weakly non-dominated in $>_N$.
NON-DOM'ED:	o is non-dominated in $>_N$.
DOM'ING:	o is dominating in $>_N$.
STR-DOM'ING:	o is strongly dominating in $>_N$.
$\exists$NON-DOM'ED:	there is a non-dominated outcome in $>_N$.
$\exists$DOM'ING:	there is a dominating outcome in $>_N$.
$\exists$STR-DOM'ING:	there is a strongly dominating outcome in $>_N$.

Table 1: Reasoning tasks with respect to a single gCP-net

complete in the general case, with the result for DOMINANCE holding even when the gCP-net N is consistent. The NON-DOM'ED problem is in P, while $\exists$NON-DOM'ED is NP-complete. If N is consistent, then the DOM'ING and $\exists$DOM'ING problems are in coNP. We also mention here, as it will prove useful later on, that the SELF-DOMINANCE problem, i.e., the problem of determining for a given gCP-net N and outcome o whether $o >_N o$, is also PSPACE-complete.

Recall that for regular CP-nets acyclicity of the dependency graph is enough to guarantee consistency (Boutilier et al. 2004). This is not true of gCP-nets.

Example 5. Take a gCP-net N over $\mathcal{V} = \{A, B, C, D\}$, where $N = \{a : d_1 \rhd d_2,\ b : d_2 \rhd d_3,\ c : d_3 \rhd d_1\}$. The dependency graph of N is easily seen to be acyclic. Nonetheless, N sanctions the sequence of worsening flips $abcd_1$, $abcd_2$, $abcd_3$, $abcd_1$, hence $abcd_1 >_N abcd_1$.[2]

The problem revealed by Example 5 is that a gCP-net can be acyclic, yet still contain a set of preference statements which are triggered by the same partial assignment and for which a cycle is derived. Thus, it is relevant to ask about the complexity of checking consistency for acyclic gCP-nets, i.e., given an acyclic gCP-net N for which we can assume, without loss of generality, that all preferences π in the statements are over the same variable, to determine whether:

aCONSISTENCY: N is consistent.

Intuitively, checking aCONSISTENCY revolves around the question of whether there exists $N' \subseteq N$, for which the conjunction of all pre-conditions is satisfiable and the preference statements lead to a cycle over the outcomes. Observe that checking consistency for a single conditional preference statement $\varphi = \psi : \pi$ amounts to checking whether there are cycles in the preferences occurring in π, which can be done in polynomial time, as the problem is essentially that of finding the connected components in a directed graph.

Proposition 1. aCONSISTENCY is coNP-complete.

Proof. For membership, we show that the complement of aCONSISTENCY, i.e., deciding whether an acyclic gCP-net $N = \{\psi_1 : \pi_1, \ldots, \psi_n : \pi_n\}$ is inconsistent, is in NP. We guess an assignment for the ψ_i's: then, we check in polynomial time whether the π_i's induce a cycle.

For hardness we reduce from UNSAT. Consider an instance of UNSAT, i.e., a propositional formula φ whose unsatisfiability we want to check. Construct now a gCP-net $N = \{\varphi : a \rhd a\}$ for A a fresh variable whose values $D(A) = \{a, \overline{a}\}$ do not occur in φ. If φ is unsatisfiable, then $\varphi : a \rhd a$ is discarded when constructing $>_N$, and thus N is consistent, since it has no cycles. On the other hand, suppose φ is satisfiable. Then $\varphi : a \rhd a$ leads to a cycle in $>_N$. $\square$

An issue related to the one just mentioned is checking whether a gCP-net is a CP-net, which has been also shown to be coNP-complete (Goldsmith et al. 2008). The two problems are distinct, since an acyclic gCP-net is not necessarily an (acyclic) CP-net—though the converse statement holds, since gCP-nets are more general than CP-nets.

mgCP-nets

For the multi-agent case we define mgCP-nets, introduce four semantics, nine reasoning tasks related to dominance and optimality, and analyze the computational complexity of these reasoning task with respect to the defined semantics.

Definitions and Semantics

An *mgCP-net M* is a multi-set $M = \langle N_1, \ldots, N_m \rangle$ of gCP-nets over the set $\mathcal{V}$ of variables. The m in 'mgCP' does double duty: once as a reminder that we are dealing with many gCP-nets, and then as a variable for the number of gCP-nets in a profile. We think of the semantics for mgCP-nets, essentially, as a binary relation over outcomes, reflecting the domination relationships induced by the individual gCP-nets in M. More concretely, for every mgCP-net M we define a binary relation $>_M$ on outcomes, called the *induced collective model of M*, which is obtained by aggregating the induced models of the gCP-nets in M, with notions such as dominance and consistency analogous to the ones for single gCP-nets (taking into account that transitivity is not guaranteed).

Before presenting the semantics, we need some preliminary notions. Given an mgCP-net $M = \langle N_1, \ldots, N_m \rangle$ and two outcomes o_1, o_2, we define the following sets:

$$s_M^{o_1 > o_2} = \{N_i \in M \mid o_1 >_i o_2\},$$
$$s_M^{o_1 \bowtie o_2} = \{N_i \in M \mid o_1 \not>_i o_2 \text{ and } o_2 \not>_i o_1\}.$$

A *ranking function r with respect to a gCP-net N (mgCP-net M, respectively)* assigns to every outcome o a nonnegative number $r_N(o)$ ($r_M(o)$, respectively). If there is no danger of ambiguity, we write $r_i(o)$ instead of $r_{N_i}(o)$.

Given a gCP-net N, *the dominance equivalence relation* $>_N^d$ is defined by saying that $o_1 >_N^d o_2$ if $o_1 >_N o_2$ and $o_2 \not>_N o_1$, and $o_1 \approx_N^d o_2$ if $o_1 = o_2$, or $o_1 >_N o_2$ and $o_2 >_N o_1$ (Goldsmith et al. 2008). The dominance equivalence relation is, indeed, an equivalence relation, and it therefore partitions the set of outcomes into equivalence classes. If o is an outcome, its equivalence class with respect to $>_N^d$ (i.e., the set of outcomes that includes o and, if they exist, all outcomes with which o forms a cycle in $>_N$) is called *the dominance class of o with respect to N* and is denoted by $[o]_N$, with the subscript duly omitted when clear from context. The dominance classes themselves form

[2] An analogous example appears in (Wilson 2004).

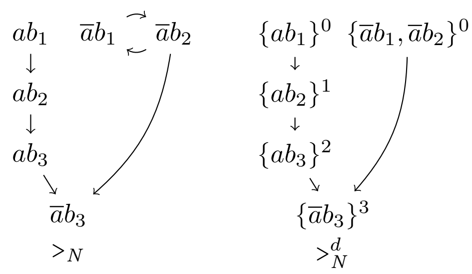

Figure 3: Dominance classes and longest path ranks.

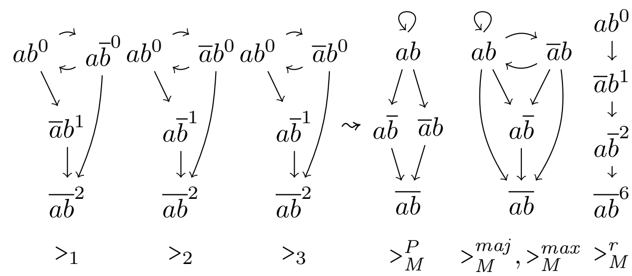

Figure 4: Individual and collective semantics for the mgCP-net M of Example 7. Edges in the induced models $>_{1-3}$ indicate worsening flips sanctioned by the gCP-nets N_{1-3}, respectively, whereas edges in the induced collective models indicate domination relations obtained through aggregation. Arrows inferred through transitivity in $>^P_M$ and $>^r_M$ are omitted; since $>^{maj}_M$ and $>^{max}_M$ are not guaranteed to be transitive, every domination relation in them is made explicit. Longest path ranks assigned by $r^{\texttt{lp}}_i$ are shown as a superscript.

a strict partial order, which we denote, overloading notation, by $>^d_N$. We say that $[o_1]$ *dominates* $[o_2]$ *with respect to* $>^d_N$, written $[o_1] >^d_N [o_2]$, if $o_1 >^d_N o_2$. A dominance class $[o]$ is *non-dominated with respect to* $>^d_N$ if there is no dominance class which dominates it with respect to $>^d_N$.

Finally, we can now define the rank of an outcome in $>_N$. The *longest path rank function* $r^{\texttt{lp}}_N$ assigns to an outcome o the length of the longest path from $[o]$ to a non-dominated dominance class in $>^d_N$.[3]

Example 6. Consider $\mathcal{V} = \{A, B\}$ with domains as in Example 2, and a gCP-net $N = \{\top : b_1 \rhd b_2 \rhd b_3, b_3 : a \rhd \overline{a}, \overline{a} : b_2 \rhd b_1\}$. Figure 3 shows the induced model $>_N$, the dominance classes, the strict partial order $>^d_N$ on dominance classes, and the longest path ranks assigned by $r^{\texttt{lp}}$.

We can now move on to defining the semantics of mgCP-nets, using the semantics defined for mCP-nets (Rossi, Venable, and Walsh 2004; Lukasiewicz and Malizia 2016) as a starting point.

Definition 3. If $M = \langle N_1, \ldots, N_m\rangle$ is an mgCP-net, the *Pareto relation* $>^P_M$, *majority relation* $>^{maj}_M$, *maximality relation* $>^{max}_M$ and *rank relation* $>^r_M$ *with respect to* M are defined, for any o_1 and o_2, as follows:

$$\begin{array}{ll} o_1 >^P_M o_2 & \text{if } o_1 >_i o_2, \text{for every } N_i \in M; \\ o_1 >^{maj}_M o_2 & \text{if } o_1 >_i o_2, \text{for } \left\lceil \frac{m+1}{2} \right\rceil N_i \in M; \\ o_1 >^{max}_M o_2 & \text{if } |s^{o_1 > o_2}_M| > max\{|s^{o_2 > o_1}_M|, |s^{o_1 \bowtie o_2}_M|\}; \\ o_1 >^r_M o_2 & \text{if } r_M(o_1) \leq r_M(o_2). \end{array}$$

If S is a semantics, we call $>^S_M$ the *S-induced collective model of M*, or, more briefly, the *S-induced model of M*. Given an mgCP-net M and a semantics S, if $o_1 >^S_M o_2$, we say that o_1 *S-dominates* o_2 *with respect to* M. We say that M is *S-consistent* (or simply *consistent*), if there is no set of outcomes $o_0, \ldots, o_k$ such that $o_0 >_M \ldots >_M o_k >_M o_0$. S-non-dominated, S-weakly non-dominated, S-dominating and S-strongly dominating outcomes for an mgCP-net M are defined analogously as for individual gCP-nets.

For the rank relation r_M, we focus here on a particular function, reminiscent of Borda's rule for aggregating total linear orders (Wilson 2004), obtained by summing up the $r^{\texttt{lp}}$ score for all agents in M. Thus, for an mgCP-net $M = \langle N_1, \ldots, N_m\rangle$, we will take the rank of o with respect to M to be $r_M(o) = \sum_{N_i \in M} r^{\texttt{lp}}_i(o)$.

Example 7. Alice, Bob and Carol want to go on holiday to New York together, and are looking for a shared apartment. Luckily, the booking service of Example 2 can handle preferences submitted by different agents. The variables are $\mathcal{V} = \{A, B\}$ as in Example 2, though for simplicity we now assume each variable is binary. Alice submits $N_1 = \{\top : a \rhd \overline{a}, \top : b \rhd \overline{b}, a : \overline{b} \rhd b\}$, while Bob and Carol submit $N_2 = N_3 = \{\top : a \rhd \overline{a}, \top : b \rhd \overline{b}, b : \overline{a} \rhd a\}$, with the corresponding 3gCP-net being $M = \langle N_1, N_2, N_3\rangle$. The induced models $>_1$, $>_2$ and $>_3$ together with the induced collective models $>^P_M$, $>^{maj}_M$, $>^{max}_M$ and $>^r_M$, are shown in Figure 4. None of the induced individual models has a strongly dominating outcome, though ab is *weakly* non-dominated, as well as dominating, in each, and thus a prime candidate for being at the top of the list of suggested outcomes. Since ab self-dominates in each of the individual induced models, this domination relation carries over to the induced collective models. The rank of an outcome in $>^r_M$ is computed by summing up its ranks in the individual induced models. Thus, $r_M(\overline{ab}) = \sum_{i=1}^3 r^{\texttt{lp}}_i(\overline{ab}) = 6$.

A few observations with respect to the semantics and their motivation are in order at this point. First, for different outcomes o_1 and o_2, if $o_1 >^P_M o_2$, then $o_1 >^{maj}_M o_2$, and if $o_1 >^{maj}_M o_2$, then $o_1 >^{max}_M o_2$. Second, the maj- and max-induced models $>^{maj}_M$ and $>^{max}_M$, respectively, are not guaranteed to be transitive. However, the Pareto-induced model $>^P_M$ is transitive, since if $o_1 >^P_M o_2$ and $o_2 >^P_M o_3$, then $o_1 >^P_i o_3$, for every $N_i \in M$, and thus $o_1 >^P_M o_3$. Hence, if there is a set of outcomes such that $o_0 >^P_M \ldots >^P_M o_k >^P_M o_0$, then we can contract this chain to $o_0 >^P_M o_0$. It follows that the condition for Pareto-consistency of mgCP-nets coincides with consistency for individual gCP-nets, i.e., M is

[3] The rank semantics has been used before to aggregate acyclic CP-nets (Rossi, Venable, and Walsh 2004; Lukasiewicz and Malizia 2016). However, since acyclic CP-nets feature no cycles between outcomes, the rank function is defined more easily than here.

Pareto-consistent if and only if there is no outcome o for which $o >_M^P o$. Observe that by definition we cannot have $o >_M^{max} o$, though other forms of inconsistency are possible. Third, the longest-path induced $>_M^r$ turns out to be a total pre-order on outcomes, since every outcome gets a rank in $>_M^r$ and two outcomes can get the same rank. Therefore, $>_M^r$ is transitive by design.

Why focus on these semantics in particular? The main reason is that they represent clear and intuitive aggregation principles, combining strands from both Social Choice and Knowledge Representation. Pareto semantics, for instance, recalls the notion of unanimity (a mainstay of Social Choice functions), as well as skeptical acceptance (used in many Knowledge Representation formalisms). Of course, in most cases it will be too stringent and more relaxed procedures will have to be invoked, e.g., majoritarian or rank-based. The fact that the semantics are not mutually exclusive suggests that they can be used alongside each other, e.g., to deliver results when the Pareto semantics is undecided.

Reasoning tasks for mgCP-nets

Given an mgCP-net M and a semantics S, the reasoning tasks we look at are the same as the ones for individual gCP-nets (see Table 1), the only difference being that the reasoning tasks for mgCP-nets are parameterized by the semantics S. A general overview of the results, alongside existing results for single gCP-nets, is given in Table 2. This is also a good place to recall that PSPACE = coPSPACE and that, by Savitch's Theorem, NPSPACE = PSPACE (Arora and Barak 2009). We will make use of both facts in our proofs.

Pareto Semantics The Pareto semantics is one of consensus, i.e., all agents have to agree that o_1 dominates o_2 in order for this to be reflected in the aggregated result. This makes it possible to use existing complexity results for individual gCP-nets (Goldsmith et al. 2008), but it does not also mean that easy tasks stay easy. In particular, we cannot leverage the P result for checking whether an outcome o_1 is non-dominated in individual gCP-nets: if o_1 is found to be dominated by o_2 in some gCP-net N_i from M, this is of no help in deciding whether o_1 is dominated in $>_M^P$. The outcome o_2 would have to dominate o_1 in *every* $>_i$ for this to hold. The task actually turns out to be PSPACE-complete, and a similar thing holds for checking existence of a Pareto non-dominated outcome. In fact, all the tasks considered turn out to be PSPACE-complete.

Theorem 1. The P–NON-DOM'ED problem for mgCP-nets is PSPACE-complete.

Proof. We show that the complement, i.e., checking whether o is dominated in $>_M^P$, is PSPACE-complete. For membership, guess an outcome o' and check whether $o' >_M^P o$, which amounts to at most m PSPACE tasks. For hardness, we do a reduction from the SELF-DOMINANCE problem for single gCP-nets. Thus, given a gCP-net N and an outcome $o = xyz\cdots$, take the 2gCP-net $M = \langle N_1, N_2\rangle$, where $N_1 = N$ and $N_2 = \{(y \wedge z \wedge \dots) : x \rhd x\}$: in other words, N_2 is such that the only induced comparison in $>_2$ is one in which o self-dominates. The claim, then, is that o is self-dominating in $>_N$ iff o is dominated in $>_M^P$. To see this, assume first that $o >_N o$, hence $o >_1 o$. Since $o >_2 o$ by design, it follows that $o >_M^P o$. Conversely, if o is dominated in $>_M^P$, then, since o is not dominated by any other outcome in $>_2$, this can only be because it is dominated by itself in $>_M^P$, and hence it self-dominates in $>_N$. □

Theorem 2. The P–∃NON-DOM'ED problem for mgCP-nets is PSPACE-complete.

Proof. For membership, it suffices to guess an outcome o and ask the PSPACE–complete problem P–NON-DOM'ED for M and o, where M is the given mgCP-net. This is in NPSPACE, and thus in PSPACE (recall that NPSPACE = PSPACE). For hardness, we reduce from P–NON-DOM'ED. Consider an instance of this problem, i.e., an mgCP-net $M = \langle N_1, \dots, N_m\rangle$ and some outcome $o = v_1 \dots v_k$ for $\mathcal{V} = \{V_1, \dots, V_k\}$ and $v_i \in D(V_i)$ for $i \in \{1, \dots, k\}$. We now construct a slightly different mgCP-net $M' = \langle N_1', \dots, N_m'\rangle$, where $N_i' = N_i \cup \{\top : v_i' \rhd v_i' \mid v_i' \in D(V_i), v_i' \neq v_i\}$. The intuitive idea is that any outcome $o' \neq o$ is now self-dominating in $>_{N_i'}$, and hence self-dominating in $>_{M'}^P$. Thus, if there is a non-dominated outcome at all in $>_{M'}^P$, then it must be o: and this only happens if o is non-dominated in M. In other words, o is non-dominated in M if and only if there is a non-dominated outcome in M', which concludes the proof. □

Proposition 2. The P–DOMINANCE, P–CONSISTENCY, P–WNON-DOM'ED, P–DOM'ING, P–STR-DOM'ING, P–∃DOM'ING, P–∃STR-DOM'ING problems for mgCP-nets are PSPACE-complete.

Proof. Hardness here is inherited from the single gCP-case, so we focus on membership. We assume an mgCP-net $M = \langle N_1, \dots, N_m\rangle$. For P–DOMINANCE, we have to check whether $o_1 >_i o_2$, for every $N_i \in M$. This amounts to solving m PSPACE tasks, which is also in PSPACE. For P–CONSISTENCY, recall that M being Pareto-consistent is equivalent to $o \not>_M^P o$, for any outcome o, i.e., $o \not>_i o$, for some $N_i \in M$. To verify this we ask of every outcome whether $o >_i o$, for every $N_i \in M$, which amounts to a (potentially exponential) number of PSPACE tasks. A similar algorithm works for P–WNON-DOM'ED, where we need to take every outcome o' and check whether we have that $o' >_i o$ and $o \not>_i o'$, for every $N_i \in M$. Existence of such an outcome o' implies that o is not weakly non-dominated in $>_M$, while lack of existence implies the contrary. For S–DOM'ING, we have that o is a dominating outcome iff $o >_M^P o'$, for any outcome o'. This is equivalent to o being dominating in every induced model $>_i$, for $N_i \in M$. Determining this involves solving m PSPACE tasks. For P–STR-DOM'ING we have to check that o is dominating in every $>_i$, for $N_i \in M$ and, in addition, that o is strongly dominating in at least one $>_i$. To see why this is sufficient to settle the question, suppose o were dominating in every $>_i$, but strongly dominating in neither of them: then we have that $o >_i o$ for every $N_i \in M$, and thus $o >_M^P o$, which means that o is not strongly dominating in $>_M^P$. Checking whether o is strongly-dominating in some $>_i$ is in PSPACE, thus our task

	single gCP-nets	mgCP-nets			
		Pareto	*maj*	*max*	*r*
S–DOMINANCE	PSPACE-c	PSPACE-c	PSPACE-c	PSPACE-c	PSPACE-h
S–CONSISTENCY	PSPACE-c	PSPACE-c	PSPACE-h	PSPACE-h	—
S–wNON-DOM'ED	PSPACE-c	PSPACE-c	PSPACE-h	PSPACE-h	PSPACE-h
S–NON-DOM'ED	in P	PSPACE-c	PSPACE-c	in PSPACE	—
S–DOM'ING	PSPACE-c	PSPACE-c	PSPACE-c	PSPACE-c	PSPACE-h
S–STR-DOM'ING	PSPACE-c	PSPACE-c	PSPACE-c	PSPACE-c	—
S–∃NON-DOM'ED	NP-c	PSPACE-c	NP-h	NP-h	—
S–∃DOM'ING	PSPACE-c	PSPACE-c	PSPACE-c	PSPACE-c	—
S–∃STR-DOM'ING	PSPACE-c	PSPACE-c	PSPACE-c	PSPACE-c	—

Table 2: Complexity results for single gCP-nets and mgCP-nets; entries for single gCP-nets are the result of previous work (Goldsmith et al. 2008), and are presented here for comparison—they are to be interpreted under the standard gCP-net semantics (i.e., S plays no role here); '-c' and '-h' are short for -complete and -hard, respectively, for a given class; the line '—' means that the answer is trivial (consult the corresponding sections for more details).

is in PSPACE. For P–∃DOM'ING and P–∃STR-DOM'ING, respectively, we can go through every outcome and ask whether it is dominating and strongly dominating, respectively, in $>_M^P$. This consists entirely of PSPACE tasks. □

Majority Semantics We now turn to the majority semantics maj, inspired by the well known majority rule in preference aggregation. By definition, when the number of agents is even we get a *strict* version of majority. The results can be however easily adapted to a *weak* version of majority.

Theorem 3. The problems maj–DOMINANCE, maj–DOM'ING, maj–STR-DOM'ING, maj–∃DOM'ING and maj–∃STR-DOM'ING are PSPACE-complete.

Proof. For all problems, PSPACE-hardness is inherited from the corresponding single agent problems, by considering a mgCP-net with $m = 1$. We now establish membership.

For maj–DOMINANCE, consider an algorithm counting whether there are more than $\lceil \frac{m+1}{2} \rceil$ agents in M such that for each i it holds that $o_1 >_i o_2$. If this is the case answer 'yes', otherwise answer 'no'. This algorithm need to keep track of the yes/no answer of at most m PSPACE problems, and hence it is in PSPACE.

For maj–DOM'ING, consider an algorithm checking for each $o' \in \mathcal{O}$ whether there is a set S of agents, such that $|S| \geq \lceil \frac{m+1}{2} \rceil$, where each agent $i \in S$ has $o >_i o'$. Hence, if for some o' such a set S is found, the algorithm answers 'yes', otherwise it answers 'no'. Thus we need to repeat at most $|\Pi_{X \in \mathcal{V}} D(X)|$ times (for all possible outcomes), at most m (for all agents) dominance PSPACE tasks.

For maj–STR-DOM'ING, consider an algorithm which solves the problems maj–DOM'ING and maj–NON-DOM'ED, which are both in PSPACE, and answers 'yes' if and only if for both problems it gets a positive answer.

For maj–∃DOM'ING and maj–∃STR-DOM'ING, consider an algorithm solving the problems maj–DOM'ING and maj–STR-DOM'ING, respectively, for all outcomes $o \in \mathcal{O}$, and which says 'yes' if at least for one instance the answer is positive. This amounts to solving a (possibly exponential) number of PSPACE tasks. □

Proposition 3. The maj–CONSISTENCY and maj–wNON-DOM'ED problems are PSPACE-hard, while maj–∃NON-DOM'ED is NP-hard.

Proof. In all cases, reduce from the corresponding single-agent complete problems where $m = 1$. □

Proposition 4. The maj–NON-DOM'ED problem for mgCP-nets is PSPACE–complete.

Proof. Let $\overline{maj\text{–NON-DOM'ED}}$ be the complement of maj–NON-DOM'ED, namely the problem asking whether there is some outcome o' such that $o' >_M^{maj} o$, for given o and M. This amounts to checking whether for at least $\lceil \frac{m+1}{2} \rceil$ agents i in M it is the case that $o' >_i o$.

We now show that $\overline{maj\text{–NON-DOM'ED}}$ is in NPSPACE. Consider an algorithm guessing an outcome o' and then checking if there are more than $\lceil \frac{m+1}{2} \rceil$ agents i such that $o' >_i o$ and in this case it outputs 'yes'. This algorithm solves at most m PSPACE problems. Since $\overline{maj\text{–NON-DOM'ED}}$ is in NPSPACE, we have that $\overline{maj\text{–NON-DOM'ED}}$ is in PSPACE by Savitch's Theorem, and thus its complement maj–NON-DOM'ED is in coPSPACE, which means that maj–NON-DOM'ED is in PSPACE.

Proof of hardness is identical to that of Theorem 1, since maj semantics for $m = 2$ the majority corresponds to the total number of agents. □

Max Semantics Max semantics refines maj semantics by taking into account also incomparabilities. This semantics does not admit cycles of length at most 2. In fact, for $>_M^{max}$ to be inconsistent there would need to be two outcomes o_1 and o_2 such that $o_1 >_M^{max} o_2$ and $o_2 >_M^{max} o_1$, implying a contradiction between $|s_M^{o_1 > o_2}| > |s_M^{o_2 > o_1}|$ and $|s_M^{o_2 > o_1}| > |s_M^{o_1 > o_2}|$.

Theorem 4. The max–DOMINANCE, max–DOM'ING, max–STR-DOM'ING, max–∃DOM'ING and max–∃STR-DOM'ING problems for mgCP-nets are PSPACE-complete.

Proof. PSPACE-hardness is inherited from the corresponding single agent problems, by considering a mgCP-net with $m = 1$. Thus, we focus here on PSPACE-membership.

For max–DOMINANCE, consider an algorithm that stores $|s_M^{o_1 > o_2}|$ as supp, i.e., the number of agents i in M for whom $o_1 >_i o_2$. Observe that supp $\leq m$. Then, the algorithm stores $|s_M^{o_2 > o_1}|$ as opp, i.e., the number of agents in M such that $o_2 >_i o_1$. Again, opp $\leq m$. Then, it stores m − supp − opp as inc; i.e., the number of agents in M for whom o_1 and o_2 are incomparable. Finally, if inc ≥ opp and supp ≥ inc, or if inc ≤ opp and supp ≥ opp, the algorithm answers 'yes', and 'no' otherwise.

For max–DOM'ING, consider an algorithm that does the following procedure for any $o' \in \mathcal{O}$: similarly to the previous argument for max–DOMINANCE, the algorithm stores as supp the number of agents i in M such that $o' >_i o$, then the number of agents k in M such that $o >_k o'$ as opp, and the number of agents considering o and o' as incomparable in inc = m − supp − opp. Analogously to the case for max–DOMINANCE, the algorithm checks if inc ≥ opp and supp ≥ inc, or if inc ≤ opp and supp ≥ opp, in which cases it answers 'yes', and 'no' otherwise. The algorithm solves a (potentially exponential) number of PSPACE tasks, which is itself in PSPACE.

For max–STR-DOM'ING, it suffices to design an algorithm which runs the algorithms for max–DOM'ING and max–NON-DOM'ED (both in PSPACE), and which answers 'yes' if and only if both tasks return a 'yes'.

For max–∃DOM'ING and max–∃STR-DOM'ING, consider an algorithm asking the problems max–DOM'ING and max–STR-DOM'ING for all outcomes $o \in \mathcal{O}$, and answering 'yes' if for at least one of the outcomes the answer is positive. This amounts to solving a (possibly exponential) number of PSPACE tasks. □

Proposition 5. The max–CONSISTENCY and max–wNON-DOM'ED problems for mgCP-nets are PSPACE-hard, while max–∃NON-DOM'ED is NP-hard.

Proof. In all cases, reduce from the corresponding single-agent complete problems where $m = 1$. □

Proposition 6. The max–NON-DOM'ED problem for mgCP-nets is in PSPACE.

Proof. Consider an algorithm that checks for all $o' \in \mathcal{O}$ whether $o' >_M^{max} o$, which is a PSPACE problem according to Theorem 4, and it answers 'yes' if every one of these tasks gives a negative answer. □

Observe that the hardness result of Theorem 1 cannot be adapted to max–NON-DOM'ED since $>_M^{max}$ has no self-dominating outcomes by definition.

Rank semantics Since the r-induced model of an mgCP-net M is a total preorder, the notions of strongly dominating outcome and non-dominated outcome are vacuous, since $o >_M^r o$ for every outcome o. Furthermore, weakly non-dominated outcomes always exist, and they coincide with dominating outcomes. Thus, the r–CONSISTENCY, r–NON-DOM'ED, r–STR-DOM'ING, r–∃NON-DOM'ED, r–∃DOM'ING, and r–∃STR-DOM'ING problems have trivial answers. We will focus, in the following, on r–DOMINANCE and r–wNON-DOM'ED, with the understanding that a solution to the latter problem doubles as a solution to the r–DOM'ING problem. We first show, using results on single gCP-nets (Goldsmith et al. 2008) and some intermediary results that finding the rank of an outcome and comparing two outcomes with respect to their rank is PSPACE-hard even in the single gCP-case (Proposition 7). This then carries over to the multi-agent case (Theorem 5).

Lemma 1. If o and o' are two outcomes and the Hamming distance between them is $d_H(o, o') = p$,[4] then there exists a gCP-net $N(o, o')$ such that $|N| = p$ and a sequence, of length p, of worsening flips from o to o' sanctioned by N.

Proof. Since worsening flips exist only between outcomes o_i and o_j which differ by only one variable, we can create a chain of outcomes $o_1, \ldots, o_p$ of length p, where $o = o_1$, $o' = o_p$ and o_i and o_{i+1} differ by only one variable, and a conditional preference statement can be defined for every worsening flip form o_i to o_{i+1}, as shown in Example 8. □

Example 8. If $o = abc$ and $o' = \overline{abc}$, then we can reach $\overline{abc}$ from abc in three steps through the chain abc, $ab\overline{c}$, $a\overline{bc}$, $\overline{abc}$. A worsening flip from abc to $ab\overline{c}$ is sanctioned by the statement $a \wedge b : c \rhd \overline{c}$. The statement $a \wedge \overline{c} : b \rhd \overline{b}$ then sanctions the worsening flip from $ab\overline{c}$ to $a\overline{bc}$, and so on. The gCP-net $N(o, o')$ is simply the set of all these preference statements.

If o is an outcome over a set of variables $\mathcal{V}$ and $X \notin \mathcal{V}$ is a variable such that $D(X) = \{x, \overline{x}\}$, we write ox^* for the outcome o' over $\mathcal{V} \cup \{X\}$ such that $o'[Y] = o[Y]$, for any $Y \in \mathcal{V}$, and $o'[X] = x^*$, for $x^* \in D(X)$. We can think of o' as o concatenated with x^*. We now show that given an outcome o of rank 0 in N, we can construct a new gCP-net N' where (a suitable copy of) o has rank k, for any $k \geq 0$.

Lemma 2. If N_1 is a gCP-net and o is an outcome, over variables in $\mathcal{V}$, and $k \geq 0$, then there exists a gCP-net N_2 over variables $\mathcal{V} \cup \{X_1, \ldots, X_k\}$, where all the variables in $\{X_1, \ldots, X_k\}$ are binary and none of them occurs in $\mathcal{V}$, such that $r_1^{\mathrm{lp}}(o) = 0$ iff $r_2^{\mathrm{lp}}(ox_1 \ldots x_k) = k$.

Proof. Let $N_2 = N_1 \cup \{\bigwedge_{X \in \mathcal{V}} o[X] \wedge \bigwedge_{i=1}^{k-1} x_i : \overline{x_k} \rhd x_k, \bigwedge_{X \in \mathcal{V}} o[X] \wedge \bigwedge_{i=1}^{k-2} x_i \wedge \overline{x_k} : \overline{x_{k-1}} \rhd x_{k-1}, \bigwedge_{X \in \mathcal{V}} o[X] \wedge \bigwedge_{i=1}^{k-3} x_i \wedge \bigwedge_{j=k}^{k-1} \overline{x_j} : \overline{x_{k-2}} \rhd x_{k-2}, \ldots, \bigwedge_{X \in \mathcal{V}} o[X] \wedge \bigwedge_{i=2}^{k} \overline{x_i} : \overline{x_1} \rhd x_1\}$. It holds that $r_1^{\mathrm{lp}}(o) = 0$ iff $r_2^{\mathrm{lp}}(ox_1 \ldots x_k) = k$. See Example 9 for an illustration of the construction. □

Example 9. Take a gCP-net N_1 over a single binary variable A, with $N_1 = \{\top : a \rhd \overline{a}\}$. Suppose we want to construct N_2 such that (a suitable constructed avatar of) outcome a has rank 2. To do this add two new binary variables, X and Y, and define N_2 over variables A, X and Y. As for proof of Lemma 2, we first import all the conditional preference statements from N_1: by the *ceteris paribus* semantics, this creates four copies of the dominance relation from $>_1$, one

[4] The Hamming distance between two outcomes is the number of variables on which they differ.

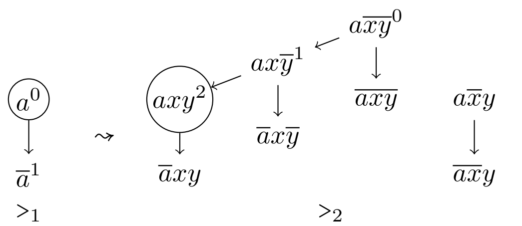

Figure 5: The induced models $>_1$ and $>_2$ from Example 9; the outcome we are interested in relegating from rank 0 to rank 2 is circled.

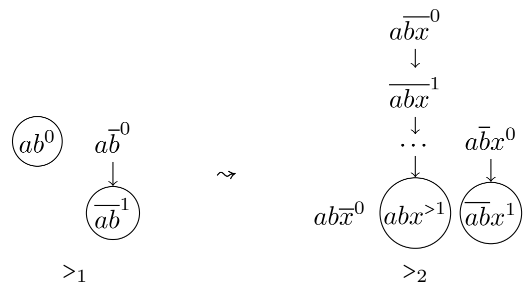

Figure 6: The induced models from Example 10.

for each assignment to variables X and Y. Add, then, statements $a \wedge x : \overline{y} \rhd y$ and $a \wedge \overline{y} : \overline{x} \rhd x$, which create a chain improving flips from axy to $a\overline{x}\overline{y}$ (see Figure 5) of length 2. It is easy to see now that $r_1^{\mathtt{lp}}(a) = 0$ iff $r_2^{\mathtt{lp}}(axy) = 2$.

The following lemma, along with Example 10, shows that, given a gCP-net N_1, we can always create a gCP-net N_2, where a dominance relation in $>_1$ is reflected by a difference between the ranks of two (avatars of the) outcomes in $>_2$.

Lemma 3. If N_1 is a gCP-net and o_1, o_2 are outcomes, over variables in $\mathcal{V}$, then there exists a gCP-net N_2 over variables $\mathcal{V} \cup \{X\}$, where $X \notin \mathcal{V}$ is binary, such that $o_1 >_1 o_2$ iff $r_2^{\mathtt{lp}}(o_1x) < r_2^{\mathtt{lp}}(o_2x)$.

Proof. Note that $d_H(o_1, o_2) = p$ implies $d_H(o_1x, o_2x) = p$. By Lemma 1 we construct the gCP-net $N(o_2\overline{x}, o_1x)$ of size p which sanctions a chain of worsening flips from $o_2\overline{x}$ to o_1x. We now take $N_2 = N_1 \cup N(o_2\overline{x}, o_1x)$. If $o_1 >_1 o_2$, then it is straightforward to see that $r_2^{\mathtt{lp}}(o_1x) < r_2^{\mathtt{lp}}(o_2x)$. Conversely, if $o_1 \not>_1 o_2$, then $r_2^{\mathtt{lp}}(o_1x) \geq r_2^{\mathtt{lp}}(o_2x)$, since o_1 inherits all of o_2's ancestors, and thus its rank is at least as great. See Example 10 for an illustration. $\square$

Example 10. Take $N_1 = \{\overline{b} : a \rhd \overline{a}\}$ and we are interested in outcome ab and $\overline{a}\overline{b}$ (circled in Figure 6), i.e., we want to construct N_2 such that $ab >_1 \overline{a}\overline{b}$ iff $r_2^{\mathtt{lp}}(abx) < r_2^{\mathtt{lp}}(\overline{a}\overline{b}x)$. Taking N_2 as described in the proof of Lemma 3 we get, via the *ceteris paribus* semantics, an extra copy of every dominance relation in $>_1$, one for x and one for $\overline{x}$. What the added preference statements do, now, is take the dominance block $a\overline{b}\overline{x} >_2 \overline{a}\overline{b}\overline{x}$ and place it on top of abx: since this block is a copy of the block $a\overline{b}x >_2 \overline{a}\overline{b}x$, we get that abx inherits all of $\overline{a}\overline{b}x$'s ancestors. Thus, the rank of abx in $>_2$ can be smaller than the rank of $\overline{a}\overline{b}x$ if and only if the path of ancestors of $\overline{a}\overline{b}x$ goes through abx, i.e., only if $abx >_2 \overline{a}\overline{b}x$. But this means that this edge must have been in $>_1$ originally (which is not the case here).

We can now gather these lemmas into one result.

Proposition 7. For N a gCP-net, o_1 and o_2 two outcomes, and $k \geq 0$, then it is PSPACE-hard to check:

(a) whether $r_N^{\mathtt{lp}}(o_1) = k$;

(b) whether $r_N^{\mathtt{lp}}(o_1) = r_N^{\mathtt{lp}}(o_2)$;

(c) whether $r_N^{\mathtt{lp}}(o_1) < r_N^{\mathtt{lp}}(o_2)$.

Proof. For (a), observe that o_1 is weakly non-dominated in $>_N$ if and only if it has $r_1^{\mathtt{lp}}(o_1) = 0$. Then, Lemma 2 gives us a reduction from wNON-DOM'ED for individual gCP-nets under $>_N$. For (b) we do a reduction from DOMINANCE. Let N_1 be a gCP-net, we define $N_2 = N_1 \cup N(o_2, o_1)$, where $N(o_2, o_1)$ is a gCP-net constructed as in Lemma 1, which induces a dominance relation from o_2 to o_1. Note that if $o_1 >_N o_2$, for some gCP-net N, then o_1 must be on a path from $[o_2]$ to a non-dominated class in $>_N^d$, and thus $r_N^{\mathtt{lp}}(o_1) \leq r_N^{\mathtt{lp}}(o_2)$. It follows now that $o_1 >_1 o_2$ iff $r_2^{\mathtt{lp}}(o_1) = r_2^{\mathtt{lp}}(o_2)$. For (c), Lemma 3 describes a reduction from DOMINANCE. $\square$

It follows from Proposition 7 that for a single gCP-net weak non-dominance (point (a), with $k = 0$) and domination (either (b) or (c)) for rank semantics are both PSPACE-hard.

Theorem 5. The $r-$DOMINANCE and $r-$wNON-DOM'ED problems for mgCP-nets are PSPACE-hard.

Proof. Inherited from the single gCP-net case. $\square$

Conclusions

Starting from the example of an online booking service that ranks holiday houses according to the preferences of multiple users, we were led to the study of mgCP-nets, i.e., profiles of generalized CP-nets (gCP-nets) (Goldsmith et al. 2008), aggregated under various semantics adapted from the literature of mCP-nets (Rossi, Venable, and Walsh 2004; Lukasiewicz and Malizia 2016). Thus, our work bridges two lines of research: that on gCP-nets, on the one hand, which provide a framework for expressing very general types of preferences under the *ceteris paribus* semantics: users are free to provide incomplete, or even cyclic preference statements; and, on the other hand, that on the aggregation of CP-nets coming from different agents. As the main barriers to implementing such frameworks are computational, our focus has been that of studying a variety of complexity problems for mgCP-nets. In particular, we analyzed the consequences of moving from a single- to a multi-agent setting with respect to the complexity of some known problems, such as consistency and optimality. Our findings show that these problems fall uniformly in the PSPACE area of the complexity landscape. On the positive side, for most cases the complexity does not increase when moving to the multi-agent case (PSPACE stays PSPACE), one exception being the problems that deal with non-dominated outcomes.

Though prohibitive, the complexity results obtained point toward possible avenues for future work. It is worth noting that the agents in our setting are not assumed to be consistent: it would be, therefore, interesting to check whether such an assumption lowers the complexity of some of the problems studied here. More generally, the complexity results provide a clear incentive to look for restrictions on gCP-nets that make the reasoning tasks tractable. Empirical studies on the types of preferences people typically express could suggest constraints that, if added to the current framework, might prove useful for modelling real world scenarios. We also wish to close the remaining complexity questions in Table 2 for which upper or lower bounds have not been established yet. Finally, alternative ways to aggregate agents' individual bases could be explored, such as voting directly on formulas in a way similar to judgment aggregation (Endriss 2016).

Acknowledgements

This work has been supported by projects ANR-14-CE24-0007-01 "CoCoRICo-CoDec", FWF P30168-N31, and a Marietta Blau grant (OeAD-GmbH).

References

Arora, S., and Barak, B. 2009. *Computational complexity: a modern approach*. Cambridge University Press.

Bienvenu, M.; Lang, J.; and Wilson, N. 2010. From Preference Logics to Preference Languages, and Back. In *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR-2010)*.

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. *Journal of Artificial Intelligence Research* 21:135–191.

Boutilier, C. 1994. Toward a Logic for Qualitative Decision Theory. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR-1994)*.

Bouveret, S.; Endriss, U.; and Lang, J. 2009. Conditional Importance Networks: A Graphical Language for Representing Ordinal, Monotonic Preferences over Sets of Goods. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-2009)*.

Brafman, R. I., and Domshlak, C. 2002. Introducing Variable Importance Tradeoffs into CP-Nets. In *Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence (UAI-2002)*.

Brafman, R. I.; Domshlak, C.; and Shimony, S. E. 2006. On Graphical Modeling of Preference and Importance. *Journal of Artificial Intelligence Research* 25:389–424.

Ciaccia, P. 2007. Conditional Preferences: A New Semantics for Database Queries. In *Proceedings of the 15th Italian Symposium on Advanced Database Systems (SEBD-2007)*.

Cornelio, C.; Goldsmith, J.; Mattei, N.; Rossi, F.; and Venable, K. B. 2013. Updates and uncertainty in cp-nets. In *Proceedings of the 26th Australasian Joint Conference on Artificial Intelligence (AI-2013)*.

Cornelio, C.; Grandi, U.; Goldsmith, J.; Mattei, N.; Rossi, F.; and Venable, K. B. 2015. Reasoning with pcp-nets in a multi-agent context. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2015)*.

Domshlak, C.; Rossi, F.; Venable, K. B.; and Walsh, T. 2009. Reasoning about soft constraints and conditional preferences: complexity results and approximation techniques. *CoRR* abs/0905.3766.

Endriss, U. 2016. Judgment aggregation. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press. chapter 17.

Goldsmith, J.; Lang, J.; Truszczynski, M.; and Wilson, N. 2008. The Computational Complexity of Dominance and Consistency in CP-Nets. *Journal of Artificial Intelligence Research* 33:403–432.

Grandi, U.; Luo, H.; Maudet, N.; and Rossi, F. 2014. Aggregating CP-nets with Unfeasible Outcomes. In *Proceedings of the 20th International Conference on Principles and Practice of Constraint Programming (CP-2014)*.

Halldén, S. 1957. *On the Logic of "better"*. CWK Gleerup.

Hansson, S. O. 1996. What is ceteris paribus preference? *Journal of Philosophical Logic* 25(3):307–332.

Li, M.; Vo, Q. B.; and Kowalczyk, R. 2014. A distributed social choice protocol for combinatorial domains. *Journal of Heuristics* 20(4):453–481.

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

Lukasiewicz, T., and Malizia, E. 2016. On the Complexity of mCP-nets. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-2016)*.

Rossi, F.; Venable, K. B.; and Walsh, T. 2004. mCP Nets: Representing and Reasoning with Preferences of Multiple Agents. In *Proceedings of the 19th National Conference on Artificial Intelligence, 16th Conference on Innovative Applications of Artificial Intelligence (AAAI-2004)*.

Rossi, F.; Venable, K. B.; and Walsh, T. 2011. A short introduction to preferences: Between artificial intelligence and social choice. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 5(4):1–102.

van Benthem, J.; Girard, P.; and Roy, O. 2009. Everything Else Being Equal: A Modal Logic for *Ceteris Paribus* preferences. *Journal of Philosophical Logic* 38(1):83–125.

von Wright, G. H. 1963. The Logic of Preference. *Philosophy* 40(151):78–79.

Wilson, N. 2004. Extending CP-Nets with Stronger Conditional Preference Statements. In *Proceedings of the 19th National Conference on Artificial Intelligence, 16th Conference on Innovative Applications of Artificial Intelligence (AAAI-2004)*.