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 allow 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 importan
t 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 consid-

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
We write \( \psi \) statements. We will sometimes write \( \psi = N \). We will typically call that if \( \psi \) non-binary domains, the formula \( V \) familiar propositional connectives, and is interpreted in the time. If \( V \) intuitions meaning that value \( x \) is assigned to \( X \) for the outcome \( x \). Preferences are allowed only over values of a single variable at a time. If \( W \subseteq V \) is a set of variables, then a propositional formula \( \psi \) over \( W \) is defined over \( \cup_{x \in V} D(x) \) using the familiar propositional connectives, and is interpreted in the intuitive way. Observe that, since variables in \( V \) may have non-binary domains, the formula \( \neg x_i \), for \( x_i \in D(X) \), is equivalent to \( \neg_{D(x_i)}(x_i) \). Formulas such as \( x_i \land x_j \), 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 \( x_2 \lor x_3 \). Moreover, \( x_2 \land x_3 \) is inconsistent.

If \( \psi \) is a propositional formula over variables in \( W \), \( X \) is a variable such that \( X \notin W \) and \( \pi = x_i \vartriangleright x_j \) is a preference over \( X \), then the formula \( \varphi = \psi ; \pi \) is a conditional preference statement, with the intuitive meaning that if \( \psi \) is true, then preferences behave according to \( \pi \). We will typically call \( \psi \) the pre-condition of \( \varphi \). A gCP-net \( N = \{ \varphi_1, \ldots, \varphi_n \} \) is a finite set of conditional preference statements. We will sometimes write \( \psi(x_i \vartriangleright x_j) \) as shorthand for \( \psi(x_i \vartriangleright x_j) \) and \( \psi(x_i \vartriangleright x_j) \). 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.

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 (\( a \)); the borough where the home is located (B), which can be either Manhattan (b1), Brooklyn (b2) or Queens (b3); the cost (C), which can be high (c) or normal (\( c \)). Thus, the set of variables is \( V = \{ A, B, C \} \) and the domains of the variables are \( D(A) = \{ a, c \} \) and \( D(B) = \{ b_1, b_2, b_3 \} \) and \( D(C) = \{ c, c \} \).

Alice wishes to use this service to find a place to stay in New York, and submits the gCP net \( N_1 \) and \( N_2 \) of Example 2, where the statements are as follows:

\[
\begin{align*}
(\varphi_1) &\quad \tau : b_3 \vartriangleright b_2 \vartriangleright b_1, \\
(\varphi_2) &\quad \neg b_1 \land \neg b_2 \land \neg b_3.
\end{align*}
\]

Later on, Alice updates her gCP-net by adding statement \( \varphi_3 \):

\[
(\varphi_3) \quad c : b_1 \vartriangleright b_3.
\]

Statement \( \varphi_1 \) says that Alice has an unconditional preference of living in Queens over Brooklyn over Manhattan. Statement \( \varphi_2 \) 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 \( \varphi_3 \) 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 \( N_2 \) are depicted in Figure 1.

**Semantics** An outcome assigns to each variable \( X \in V \) 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 \( x_1y_2z_3 \ldots \) 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 \( O = D(X) \times D(Y) \times D(Z) \times \ldots \) the set of all outcomes. If \( X \in V \) is a variable, we denote by \( o[X] \) the value of outcome \( o \) on \( X \); if \( W \subseteq V \) is a set of variables, we write \( o[W] \) for the values of outcome \( o \) on \( W \). We write \( o \models \psi \) to say that outcome \( o \) satisfies propositional formula \( \psi \) in the familiar propositional sense.

Preferences over outcomes are represented using a binary relation \( \succ \) over \( O \). If \( o_1 \succ o_2 \) we say that \( o_1 \) dominates \( o_2 \) with respect to \( \succ \), the intuitive meaning of which is that \( o_1 \)

---

\( ^3 \)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 \( o_1 \equiv o_2 \). 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 \( x_i, x_j \in D(X) \), \( \varphi = \psi; x_i \triangleright x_j \) 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 \( \varphi \) if \( o_1 = \psi, o_2 = \psi, o_1[Y] = o_2[Y] \), for any \( Y \in V \setminus \{X\} \) 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 \( \varphi = \psi; (x_i \triangleright x_j) \) if outcomes \( o_1 \) and \( o_2 \) both satisfy condition \( \psi \), 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 \( \varphi \) is interpreted as saying that if condition \( \psi \) is true, then, all else being equal, making \( x_j \) true is better than making \( x_i \) true. Note that the ceteris paribus clause holds for models of \( \psi \) as well.

We say that there is an improving flip from \( o_2 \) to \( o_1 \) sanctioned by \( \varphi \) if there is a worsening flip from \( o_1 \) to \( o_2 \) sanctioned by \( \varphi \). 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 \( \varphi \in \mathcal{N} \).

**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 \( o_1, \ldots, o_k \) such that \( o = o_1, o_k = o' \), and, for every \( i \in \{1, \ldots, k-1\} \), 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 \( >_N \) instead of \( >_N o \), 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 \( o = ab\bar{c} \) refers to a reasonably priced and private apartment in Brooklyn. For the variables \( A \) and \( B \), of \( \{A, B\} \) is \( ab \) 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 \( \varphi_3 \), is depicted in Figure 2. Adding statement \( \varphi_3 \) 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 \( ab\bar{c}, ab\bar{c}, ab\bar{c}, ab\bar{c} \), which implies that \( ab\bar{c} >_2 ab\bar{c} \).

As mentioned, in statements of the form \( \psi \land \pi \) the ceteris paribus assumption holds even for the models of \( \psi \). In Example 2, this means in particular that when interpreting a statement such as \( \varphi_2 = \neg b_1 \land \pi; \pi \triangleright a \), we induce the rankings \( \neg b_1 \land \pi > ab\bar{c} \) and \( \neg b_1 \land \pi > ab\bar{c} \), but not the ranking \( \neg b_1 \land \pi > ab\bar{c} \), though both \( \neg b_1 \land \pi > ab\bar{c} \) and \( \neg b_1 \land \pi > ab\bar{c} \) satisfy condition \( \neg b_1 \land \pi \triangleright \).

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.

**Example 4.** Consider \( \mathcal{V} \) as in Example 2. Anna submits the gCP-net \( N = \{\pi \land \tau; b_2 \triangleright b_3\} \). If the semantics was not limited to the ceteris paribus comparisons, we could derive that \( \neg b_3 \triangleright_N \neg b_2 \triangleright \pi \), 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 \( N' = \{\pi \land \tau; b_3 \triangleright b_2 \land b_3; \pi \triangleright a, b_3 \land c; a \land c \land b_2 \triangleright b_3, a \land b_2 \triangleright c \} \). From \( N' \) we now derive \( \neg b_2 \triangleright_N \neg b_3 \), 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 \( > \).

![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.](image-url)
Table 1: Reasoning tasks with respect to a single gCP-net

<table>
<thead>
<tr>
<th>Reasoning Task</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DOMINANCE</strong></td>
<td>( a_1 \succ N a_2 )</td>
</tr>
<tr>
<td><strong>CONSISTENCY</strong></td>
<td>( N ) is consistent.</td>
</tr>
<tr>
<td><strong>wNON-DOM’ED</strong></td>
<td>( o ) is weakly non-dominated in ( \succ N ).</td>
</tr>
<tr>
<td><strong>NON-DOM’ED</strong></td>
<td>( o ) is non-dominated in ( \succ N ).</td>
</tr>
<tr>
<td><strong>DOM’ING</strong></td>
<td>( o ) is dominating in ( \succ N ).</td>
</tr>
<tr>
<td><strong>STR-DOM’ING</strong></td>
<td>( o ) is strongly dominating in ( \succ N ).</td>
</tr>
<tr>
<td><strong>( \exists )Non-DOM’ED</strong></td>
<td>there is a non-dominated outcome in ( \succ N ).</td>
</tr>
<tr>
<td><strong>( \exists )DOM’ING</strong></td>
<td>there is a dominating outcome in ( \succ N ).</td>
</tr>
<tr>
<td><strong>( \exists )STR-DOM’ING</strong></td>
<td>there is a strongly dominating outcome in ( \succ N ).</td>
</tr>
</tbody>
</table>

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

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 \( M = \{ N_1, \ldots, N_m \} \) of gCP-nets over the set \( V \) 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 dominance relationships induced by the individual gCP-nets in \( M \). More concretely, for every mgCP-net \( M \) we define a binary relation \( \succ M \) 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 \( M = \{ N_1, \ldots, N_m \} \) and two outcomes \( o_1, o_2 \), we define the following sets:

\[
\begin{align*}
S_M^{o_1 \succ o_2} &= \{ N_i \in M \mid o_1 \succ N o_2 \}, \\
S_M^{o_1 \equiv o_2} &= \{ N_i \in M \mid o_1 \equiv N o_2 \}.
\end{align*}
\]

A ranking function \( r \) with respect to a gCP-net \( N \) (mgCP-net \( M \), respectively) assigns to every outcome \( o \) a non-negative number \( r_N(o) \) (\( r_M(o) \), respectively). If there is no danger of ambiguity, we write \( r(o) \) instead of \( r_N(o) \).

Given a gCP-net \( N \), the dominance equivalence relation \( \succ_N^d \) is defined by saying that \( o_1 \succ_N^d o_2 \) if \( o_1 \succ N o_2 \) and \( o_2 \not\equiv_N o_1 \), and \( o_1 \equiv_N^d o_2 \) if \( o_1 = o_2 \) or \( o_1 \succ N o_2 \) and \( o_2 \succ N o_1 \) (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 \( \succ_N^d \) (i.e., the set of outcomes that includes \( o \) and, if they exist, all outcomes with which \( o \) forms a cycle in \( \succ_N^d \)) 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
a strict partial order, which we denote, overloading notation, by \( >_N \). We say that \([o_1] \) dominates \([o_2] \) with respect to \( >_N \), written \([o_1] > N [o_2] \), if \( o_1 > N o_2 \). A dominance class \([o] \) is non-dominated with respect to \( >_N \) if there is no dominance class which dominates it with respect to \( >_N \).

Finally, we can now define the rank of an outcome in \( >_N \).

The longest path rank function \( r_{i}^{lp} \) assigns to an outcome \( o \) the length of the longest path from \([o] \) to a non-dominated dominance class in \( >_N \).

**Example 6.** Consider \( \mathcal{V} = \{A, B\} \) with domains as in Example 2, and a gCP-net \( N = \{\varnothing; b_1, b_2, b_3; a > \pi, \pi; b_2, b_1\} \). Figure 3 shows the induced model \( >_N \), the dominance classes, the strict partial order \( >_N \) on dominance classes, and the longest path ranks assigned by \( r_{i}^{lp} \).

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 \( M = \{N_1, \ldots, N_m\} \) is an mgCP-net, the Pareto relation \( >_M \), majority relation \( >_M^{maj} \), maximality relation \( >_M^{max} \), and rank relation \( >_M^{r} \) with respect to \( M \) are defined, for any \( o_1 \) and \( o_2 \), as follows:

\[
\begin{align*}
& o_1 >_M o_2 & \text{if } o_1 > o_2, \text{ for every } N_i \in M; \\
& o_1 >_M^{maj} o_2 & \text{if } o_1 >_M o_2, \text{ for } [o_1] \neq [o_2]; \\
& o_1 >_M^{max} o_2 & \text{if } [o_1] = [o_2] = [o_3] \text{ and } \max ([o_1] \cup [o_2] \cup [o_3]); \\
& o_1 >_M^{r} o_2 & \text{if } [o_1] \leq [o_2].
\end{align*}
\]

If \( S \) is a semantics, we call \( >_M \) 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 \( o_1 >_M o_2 \), 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

\[
\text{r}_{i}^{lp} \text{ score for all agents in } M. \text{ Thus, for an mgCP-net } M = \{N_1, \ldots, N_m\}, \text{ we will take the rank of } o \text{ with respect to } M \text{ to be } r_M(o) = \sum_{n_i \in M} r_{i}^{lp}(o).
\]

**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 \( N_1 = \{\varnothing; a > \pi, \pi; b, a > \pi, b\} \), while Bob and Carol submit \( N_2 = N_3 = \{\varnothing; a > \pi, a > \pi, a > \pi, a > \pi, b, a > \pi, b, a > \pi, b, a > \pi, b, a > \pi, b, a > \pi, b\} \), with the corresponding 3gCP-net being \( M = \{N_1, N_2, N_3\} \). The induced models \( >_1, >_2 \) and \( >_3 \) together with the induced collective models \( >_M, >_M^{maj}, >_M^{max} \), and \( >_M^{r} \), 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 \( >_M^{r} \) is computed by summing up its ranks in the individual induced models. Thus, \( r_M(ab) = \sum_{i=1}^{3} r_{i}^{lp}(ab) = 6 \).

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 \( o_1 >_M o_2 \), then \( o_1 >_M^{maj} o_2 \), and if \( o_1 >_M^{max} o_2 \), then \( o_1 >_M^{r} o_2 \). Second, the \( maj \)- and \( max \)-induced models \( >_M^{maj} \) and \( >_M^{max} \), respectively, are not guaranteed to be transitive. However, the Pareto-induced model \( >_M^{p} \) is transitive, since if \( o_1 >_M^{p} o_2 \) and \( o_2 >_M^{p} o_3 \), then \( o_1 >_M^{p} o_3 \), for every \( N_i \in M \), and thus \( o_1 >_M^{p} o_2 \). Hence, if there is a set of outcomes such that \( o_0 >_M \ldots >_M o_k >_M o_0 \), then we can contract this chain to \( o_0 >_M o_0 \). It follows that the condition for Pareto-consistency of mgCP-nets coincides with consistency for individual gCP-nets, i.e., \( M \) is
Theorem 2. The \( P-\exists \text{Non-Dom}' \) problem for mgCP-nets is PSPACE-complete.

Proof. For membership, it suffices to guess an outcome \( o \) and ask the PSPACE-complete problem \( P-\text{Non-Dom}' \) 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 \( P-\text{Non-Dom}' \). Consider an instance of this problem, i.e., an mgCP-net \( M = \{N_1, \ldots, N_m\} \) and some outcome \( o = v_1 \ldots v_k \) for \( \forall = \{V_1, \ldots, V_k\} \) and \( v_i \in D(V_i) \) for \( i \in \{1, \ldots, k\} \). We now construct a slightly different mgCP-net \( M' = \{N'_1, \ldots, N'_m\} \), where \( N'_i = N_i \cup \{\{1; v'_i \triangleright v_i' \mid v'_i \in D(V_i), v'_i \neq v_i\} \}. \) The intuitive idea is that any outcome \( o' \neq o \) is now self-dominating in \( >_{N'} \), and hence self-dominating in \( >_{M'} \). Thus, if there is a non-dominated outcome at all in \( >_{M'} \), 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 \( P-\text{Dominance}, P-\text{Consistency}, P-\text{wNon-Dom}' \), \( P-\text{Dom'ing}, P-\text{Str-Dom'ing}, P-\text{3Dom'ing}, P-\text{3Str-Dom'ing} \) 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 \( M = \{N_1, \ldots, N_m\} \). For \( P-\text{Dominance} \), we have to check whether \( o_1 > o_2 \), for every \( N_i \in M \). This amounts to solving \( m \) PSPACE tasks, which is also in PSPACE. For \( P-\text{Consistency} \), recall that \( M \) being Pareto-consistent is equivalent to \( \exists \text{Dom}' \), for any outcome \( o \), i.e., \( \exists \text{Dom}' \), for some \( N_i \in M \). To verify this we ask of every outcome whether \( o >_1 0 \), for every \( N_i \in M \), which amounts to a (potentially exponential) number of PSPACE tasks. A similar algorithm works for \( P-\text{wNon-Dom}' \), where we need to take every outcome \( o' \) and check whether we have that \( o' > o \) and \( o \not\triangleright_o o' \), 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 \( S-\text{Dom'ing} \), we have that \( o \) is a dominating outcome iff \( o >_{M'} o' \), 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 \( P-\text{Str-Dom'ing} \) 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 \( o > o \) for every \( N_i \in M \), and thus \( o >_{M'} o' \), which means that \( o \) is not strongly dominating in \( >_{M'} \). Checking whether \( o \) is strongly-dominating in some \( >_i \) is in PSPACE, thus our task
and hence it is in \( PSPACE \). For \( P-\exists \text{Dom'ing} \) and \( P-\exists \text{Str-Dom'ing} \), respectively, we can go through every outcome and ask whether it is dominating and strongly dominating, respectively, in \( M^i \). This consists entirely of \( PSPACE \) tasks.

**Majority Semantics** We now turn to the majority semantics \( \text{maj} \), 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 \( \text{maj} - \text{Dominance} \), \( \text{maj} - \text{Dom'ing} \), \( \text{maj} - \text{Str-Dom'ing} \), \( \text{maj} - \exists \text{Dom'ing} \) and \( \text{maj} - \exists \text{Str-Dom'ing} \) 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 \( \text{maj} - \text{Dominance} \), consider an algorithm counting whether there are more than \( \lfloor \frac{m+1}{2} \rfloor \) agents in \( M \) such that for each \( i \) it holds that \( a_i \in S \). 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 \( \text{maj} - \text{Dom'ing} \), consider an algorithm checking for each \( o' \in O \) whether there is a set \( S \) of agents, such that \( |S| \geq \lfloor \frac{m+1}{2} \rfloor \), where each agent \( i \in S \) has \( a_i > o' \). 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 \( |\Pi x \in O \Pi D(x)| \) times (for all possible outcomes), at most \( m \) (for all agents) dominance \( PSPACE \) tasks.

For \( \text{maj} - \text{Str-Dom'ing} \), consider an algorithm which solves the problems \( \text{maj} - \text{Dom'ing} \) and \( \text{maj} - \text{Non-Dom'ed} \), which are both in \( PSPACE \), and answers ‘yes’ if and only if for both problems it gets a positive answer.

For \( \text{maj} - \exists \text{Dom'ing} \) and \( \text{maj} - \exists \text{Str-Dom'ing} \), consider algorithms solving the problems \( \text{maj} - \text{Dom'ing} \) and \( \text{maj} - \text{Str-Dom'ing} \), respectively, for all outcomes \( o \in O \), and which says ‘yes’ if at least one instance the answer is positive. This amounts to solving a (possibly exponential) number of \( PSPACE \) tasks.

**Proposition 3.** The \( \text{maj} - \text{Consistency} \) and \( \text{maj} - \text{Non-Dom'ed} \) problems are \( PSPACE \)-hard, while \( \text{maj} - \exists \text{Non-Dom'ed} \) is \( NP \)-hard.

**Proof.** In all cases, reduce from the corresponding single-agent complete problems where \( m = 1 \).

**Proposition 4.** The \( \text{maj} - \text{Non-Dom'ed} \) problem for \( mgCP \)-nets is \( PSPACE \)-complete.

**Proof.** Let \( \text{maj} - \text{Non-Dom'ed} \) be the complement of \( \text{maj} - \text{Non-Dom'ed} \), namely the problem asking whether there is some outcome \( o' \) such that \( o' > M \). For given \( o \) and \( M \). This amounts to checking whether for at least \( \lfloor \frac{m+1}{2} \rfloor \) agents \( i \) in \( M \) it is the case that \( o' > i \).

We now show that \( \text{maj} - \text{Non-Dom'ed} \) is in \( NPSPACE \). Consider an algorithm guessing an outcome \( o' \) and then checking if there are more than \( \lfloor \frac{m+1}{2} \rfloor \) agents \( i \) such that \( o' > i \) and in this case it outputs ‘yes’. This algorithm solves at most \( m \) \( PSPACE \) problems. Since \( \text{maj} - \text{Non-Dom'ed} \) is in \( NPSPACE \), we have that \( \text{maj} - \text{Non-Dom'ed} \) is in \( PSPACE \) by Savitch’s Theorem, and thus its complement \( \text{maj} - \text{Non-Dom'ed} \) is in \( coPSPACE \), which means that \( \text{maj} - \text{Non-Dom'ed} \) is in \( PSPACE \).

Proof of hardness is identical to that of Theorem 1, since \( \text{maj} \) semantics for \( m = 2 \) the majority corresponds to the total number of agents.

**Max Semantics** Max semantics refines \( \text{maj} \) semantics by taking into account also incomparabilities. This semantics does not admit cycles of length at most 2. In fact, for \( m = \) to be inconsistent there would need to be two outcomes \( o_1 \) and \( o_2 \) such that \( o_1 >_M o_2 \) and \( o_2 >_M o_1 \), implying a contradiction between \( |S_{o_1} > M | > |S_{o_2} > M| \) and \( |S_{o_2} > M| > |S_{o_1} > M| \).

**Theorem 4.** The \( \text{max} - \text{Dominance, max} - \text{Dom'ing}, \text{max} - \text{Str-Dom'ing}, \text{max} - \exists \text{Dom'ing} \) and \( \text{max} - \exists \text{Str-Dom'ing} \) 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.

---

**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).
For \( \text{max-Dominance} \), consider an algorithm that stores \( s_{M}^{\geq o} \) as \( \text{supp} \), i.e., the number of agents \( i \in M \) for whom \( o_{1} \succ o_{2} \). Observe that \( \text{supp} \leq m \). Then, the algorithm stores \( s_{M}^{\geq o} \) as \( \text{inc} \), i.e., the number of agents in \( M \) such that \( o_{2} \succ o_{1} \). Again, \( \text{op} \leq m \). Then, it stores \( m - \text{supp} - \text{op} \) as \( \text{inc} \); i.e., the number of agents in \( M \) for whom \( o_{1} \) and \( o_{2} \) are incomparable. Finally, if \( \text{inc} \geq \text{op} \) and \( \text{supp} \geq \text{inc} \), or if \( \text{inc} \leq \text{op} \) and \( \text{supp} \geq \text{op} \), the algorithm answers ‘yes’, and ‘no’ otherwise.

For \( \text{max-Dom'ing} \), consider an algorithm that does the following procedure for any \( o' \in \mathcal{O} \); similarly to the previous argument for \( \text{max-Dominance} \), the algorithm stores as \( \text{supp} \) the number of agents \( i \in M \) such that \( o' \succ i \). Then, the number of agents \( k \in M \) such that \( o \succ k \) \( o' \) as \( \text{op} \), and the number of agents considering \( o \) and \( o' \) as incomparable in \( \text{inc} = m - \text{supp} - \text{op} \). Analogously to the case for \( \text{max-Dominance} \), the algorithm checks if \( \text{inc} \geq \text{op} \) and \( \text{supp} \geq \text{inc} \), or if \( \text{inc} \leq \text{op} \) and \( \text{supp} \geq \text{op} \), in which cases it answers ‘yes’, and ‘no’ otherwise. The algorithm solves a (potentially exponential) number of \( \text{PSPACE} \) tasks, which is itself in \( \text{PSPACE} \).

For \( \text{max-STR-Dom'ing} \), it suffices to design an algorithm which runs the algorithms for \( \text{max-Dom'ing} \) and \( \text{max-Non-Dom'ed} \) (both in \( \text{PSPACE} \)), and which answers ‘yes’ if and only if both tasks return a ‘yes’.

For \( \text{max-3Dom'ing} \) and \( \text{max-3STR-Dom'ing} \), consider an algorithm asking the algorithm \( \text{supp} \) 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 \( \text{PSPACE} \) tasks.

**Proposition 5.** The \( \text{max-Consistency} \) and \( \text{max-wNon-Dom'ed} \) problems for \( \text{mgCP-nets} \) are \( \text{PSPACE} \)-hard, while \( \text{max-3Non-Dom'ed} \) is \( \text{NP} \)-hard.

**Proof.** In all cases, reduce from the corresponding single-agent complete problems where \( m = 1 \).

**Proposition 6.** The \( \text{max-Non-Dom'ed} \) problem for \( \text{mgCP-nets} \) is in \( \text{PSPACE} \).

**Proof.** Consider an algorithm that checks for all \( o' \in \mathcal{O} \) whether \( o' \supseteq_{M} o \), which is a \( \text{PSPACE} \) 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 \( \text{max-Non-Dom'ed} \) since \( \supseteq_{M}^{\text{max}} \) has no self-dominating outcomes by definition.

**Rank semantics** Since the \( r \)-induced model of an \( \text{mgCP-net} \) \( M \) is a total preorder, the notions of strongly dominating outcome and non-dominated outcome are vacuous, since \( o \supseteq_{M} o \) 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, \( r \)-Dom'ing, \( r \)-Non-Dom'ed, \( r \)-3Dom'ing, and \( r \)-3STR-Dom'ing problems have trivial answers. We will focus, in the following, on \( r \)-Dom'ing 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 \( \text{gCP-nets} \) (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 \( \text{PSPACE} \)-hard even in the single \( \text{gCP-case} \) (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 \( d_{H}(o, o') = p \), then there exists a \( \text{gCP-net} N(o, o') \) 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 \( o_{1} \) and \( o_{2} \) 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 \( o_{1} \) and \( o_{1}+1 \) 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 \( abc \) from \( abc \) in three steps through the chain \( abc, ab\bar{b}, ab\bar{c} \). A worsening flip from \( abc \) to \( ab\bar{c} \) is sanctioned by the statement \( a \land b \rightarrow \overline{c} \). The statement \( a \land \overline{c} \rightarrow \overline{b} \) then sanctions the worsening flip from \( ab\overline{b} \) to \( ab\bar{c} \) and so on. The \( \text{gCP-net} N(o, o') \) is simply the set of all these preference statements.

If \( o \) is an outcome over a set of variables \( \mathcal{V} \) and \( X \in \mathcal{V} \) is a variable such that \( D(X) = \{x, \overline{x}\} \), we write \( o^x \) 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 \( \text{gCP-net} N' \) where (a suitable copy of) \( o \) has rank \( k \), for any \( k \geq 0 \).

**Lemma 2.** If \( N_{1} \) is a \( \text{gCP-net} \) and \( o \) is an outcome, over variables in \( \mathcal{V} \), and \( k \geq 0 \), then there exists a \( \text{gCP-net} N_{2} \) 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 \( r_{i}^{1}(o) = 0 \) iff \( r_{i}^{2}(o_{x_{1}} \ldots x_{k}) = k \).

**Example 9.** Take a \( \text{gCP-net} N_{1} \) over a single binary variable \( A \), with \( N_{1} = \{T \cup A \rightarrow \pi\} \). Suppose we want to construct \( N_{2} \) such that (a suitable constructed avatar of) outcome \( o \) 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 \( \succ \), one

\[\text{The Hamming distance between two outcomes is the number of variables on which they differ.}\]
for each assignment to variables $X$ and $Y$. Add, then, statements $a \land x \not\succ y$ and $a \land \bar{y} \not\succ x$, which create a chain improving flips from $axy$ to $a\bar{x}y$ (see Figure 5) of length 2. It is easy to see now that $r_1^{lp}(a) = 0$ iff $r_2^{lp}(axy) = 2$.

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 $V$, then there exists a gCP-net $N_2$ over variables $V \cup \{X\}$, where $X \not\in V$ is binary, such that $o_1>_1 o_2$ iff $r_2^{lp}(o_1x) < r_2^{lp}(o_2x)$.

**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 $N(o_2\bar{x}, o_1x)$ of size $p$ which sanctions a chain of worsening flips from $o_2x$ to $o_1x$. We now take $N_2 = N_1 \cup N(o_2\bar{x}, o_1x)$. If $o_1>_1 o_2$, then it is straightforward to see that $r_2^{lp}(o_1x) < r_2^{lp}(o_2x)$.

Conversely, if $o_1 \not>_1 o_2$, then $r_2^{lp}(o_1x) \geq r_2^{lp}(o_2x)$, 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.

**Example 10.** Take $N_1 = \{\bar{a}; a \succ \bar{x}\}$ and we are interested in outcome $ab$ and $\bar{a}b$ (circled in Figure 6), i.e., we want to construct $N_2$ such that $\bar{a}b>_2 ab$ iff $r_2^{lp}(\bar{a}bx) < r_2^{lp}(\bar{a}bx)$. 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 $\bar{x}$. What the added preference statements do, now, is take the dominance block $\bar{a}bx>_2 abx$ and place it on top of $abx$: since this block is a copy of the block $\bar{a}bx>_2 abx$, we get that $abx$ inherits all of $\bar{a}bx$’s ancestors. Thus, the rank of $abx$ in $>_2$ can be smaller than the rank of $\bar{a}bx$ if and only if the path of ancestors of $\bar{a}bx$ goes through $abx$, i.e., only if $abx>_2 \bar{a}bx$. 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 $r_N^{lp}(o_1) = k$;

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

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

**Proof.** For (a), observe that $o_1$ is weakly non-dominated in $>_N$ if and only if it has $r_N^{lp}(o_1) = 0$. 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 $>_N$, and thus $r_N^{lp}(o_1) \leq r_N^{lp}(o_2)$. It follows now that $o_1>_2 o_2$ iff $r_2^{lp}(o_1) = r_2^{lp}(o_2)$. For (c), Lemma 3 describes a reduction from DOMINANCE.

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 $mg$CP-nets are PSPACE-hard.

**Proof.** Inherited from the single gCP-net case.

**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 $mg$CP-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, Venelbre, 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 $mg$CP-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


