mCP nets: representing and reasoning with preferences of multiple agents

F. Rossi and **K. B. Venable**
Dept. of Mathematics
University of Padova
Padova, Italy
{frossi,kvenable}@math.unipd.it

T. Walsh
Cork Constraint Computation Centre
University College Cork
Cork, Ireland
tw@4c.ucc.ie

Abstract

We introduce mCP nets, an extension of the CP net formalism to model and handle the qualitative and conditional preferences of multiple agents. We give a number of different semantics for reasoning with mCP nets. The semantics are all based on the idea of individual agents voting. We describe how to test optimality and preference ordering within a mCP net, and we give complexity results for such tasks. We also discuss whether the voting schemes fairly combine together the preferences of the individual agents.

Introduction and Motivation

In many situations, we need to represent and reason about the simultaneous preferences of several agents, and to aggregate such preferences (see for example (Yager 2001)). As a motivating example, suppose you invite three friends round for dinner. Alice prefers fish to beef. Bob, on the other hand, prefers beef to fish. Finally, Carol is like Alice and prefers fish to beef. What do you cook? Both choices are Pareto optimal. If you cook fish then changing to beef will be more preferred by Bob, but less preferred by Carol and Alice. Similarly, changing from beef to fish will be more preferred by Alice and Carol, but less preferred by Bob. However, fish is perhaps the "best" choice according to a majority ordering as it is the preferred choice for both Alice and Carol, whilst beef is the preferred choice for only Bob.

Which wine do you serve with the main course? Alice, Bob and Carol are fortunately more consistent here. If it is fish then they all prefer white wine to red. However, if it is beef, then they all prefer red wine to white. Finally, do you serve cheese or dessert? Even though you are happy to serve everyone their own choice of cheese or dessert, you must still determine and reason about your friends' preferences. For example, if his girlfriend Alice has cheese, Bob prefers cheese to dessert. However, if his girlfriend Alice has dessert, he will not be able to resist so he prefers dessert to cheese.

This example demonstrates that multiple agents may have some features in common but not all (e.g. the main dish is common to all but the choice of cheese or dessert is not), that there may no longer be a single optimal solution, that there can be several definitions of optimality (e.g. Pareto optimal *versus* a majority dominance), that preferences may be conditional (e.g. the preference for wine depends on the choice of the main course) and dependent on the features of other agents (e.g. Bob's preference for dessert or cheese depends on Alice's choice for dessert or cheese). We therefore propose a framework for representing and reasoning with such preferences.

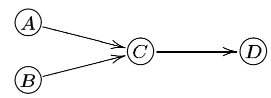

Figure 1: The dependency graph of the example CP net.

The paper is structured as follows. We first describe CP nets, a general purpose framework for representing and reasoning with qualitative and conditional preferences (Boutilier *et al.* 1999). We then introduce partial CP nets, an extension of the formalism which we use to represent the preferences of a single agent. By combining together several partial CP nets, we obtain a mCP net which can represent the preferences of multiple agents. We give a semantics for mCP nets based on the idea of the individual agents voting. Different voting schemes yield different preference orderings. We then describe how to test optimality and preference dominance, and we compute the complexity of such tasks. We also discuss whether the voting schemes fairly combine together the preferences of the individual agents.

Background: CP nets

In many applications, it is natural to express preferences via generic qualitative (usually partial) preference relations over variable assignments. For example, it is often more intuitive to say "I prefer red wine to white wine", rather than "Red wine has preference 0.7 and white wine has preference 0.4". The former statement provides less information, but does not require careful selection of preference values. Moreover, we often wish to represent conditional preferences, as in "If it is meat, then I prefer red wine to white". Qualitative and conditional preference statements are thus useful components of many applications.

CP nets (Boutilier *et al.* 1999) are a graphical model for compactly representing conditional and qualitative preference relations. They exploit conditional preferential independence by structuring an agent's preferences under the *ceteris paribus* assumption (e.g. under the "ceteris paribus" or "all other things being equal" assumption). Informally, CP nets are sets of *conditional ceteris paribus (CP)* preference statements. For instance, the statement *"I prefer red wine to white wine if meat is served."* asserts that, given two meals that differ *only* in the kind of wine served *and* both containing meat, the meal with a red wine is preferable to the meal with a white wine. Many philosophers and AI researchers (Doyle & Wellman 1994) have argued that many of our preferences are of this type.

CP nets bear some similarity to Bayesian networks. Both utilize directed graphs where each node stands for a domain variable, and assume a set of features $\mathbf{F} = \{X_1, \ldots, X_n\}$ with finite, discrete domains $\mathcal{D}(X_1), \ldots, \mathcal{D}(X_n)$. For each feature X_i, each user specifies a set of *parent* features $Pa(X_i)$ that can affect her preferences over the values of X_i. This defines a dependency graph in which each node X_i has $Pa(X_i)$ as its immediate predecessors. Given this structural information, the user explicitly specifies her preference over the values of X_i for *each complete outcome* on $Pa(X_i)$. This preference is assumed to take the form of total (Boutilier *et al.* 1999) or partial order over $\mathcal{D}(X)$.

For example, consider a CP net with the dependency graph given in Figure 1, whose features are A, B, C, and D, with binary domains containing f and $\overline{f}$ if F is the name of the feature, and with the preference statements as follows: $a \succ \overline{a}$, $b \succ \overline{b}$, $(a \wedge b) \vee (\overline{a} \wedge \overline{b}) : c \succ \overline{c}$, $(a \wedge \overline{b}) \vee (\overline{a} \wedge b) : \overline{c} \succ c$, $c : d \succ \overline{d}$, $\overline{c} : \overline{d} \succ d$. Here, statement $a \succ \overline{a}$ represents the unconditional preference for $A = a$ over $A = \overline{a}$, while statement $c : d \succ \overline{d}$ represents $D = d$ is preferred to $D = \overline{d}$, given that $C = c$.

How do we reason with such CP statements? The semantics of CP nets depends on the notion of a worsening flip. A worsening flip is a change in the value of a variable to a value which is less preferred by the CP statement for that variable. For example, in the CP net of Fig. 1, passing from $abcd$ to $ab\overline{c}d$ is a worsening flip since c is better than $\overline{c}$ given a and b. We say that one outcome α is better than another outcome β (written $\alpha \succ \beta$) iff there is a chain of worsening flips from α to β. This definition induces a strict partial order over the outcomes. In general, there may be many optimal outcomes. However, in acyclic CP nets, there is only one.

Several types of queries can be asked about CP nets. First, given a CP net, what are the optimal outcomes? For acyclic CP nets, such a query is answerable in linear time (Boutilier *et al.* 1999): we forward sweep through the CP net, starting with the unconditional variables, following the arrows in the dependency graph and assigning at each step the most preferred value in the preference table. For instance, in the CP net of Figure 1 above, we would choose $A = a$ and $B = b$, then $C = c$ and then $D = d$. The optimal outcome is therefore $abcd$. The same complexity also holds for testing whether an outcome is optimal since an acyclic CP net has only one optimal outcome. We can find this optimal outcome (in linear time) and then compare it to the given one (again in linear time).

The second type of query is a dominance query. Given two outcomes, is one better than the other? Unfortunately, this query is NP-hard even for acyclic CP nets (Domshlak & Brafman 2002), Whilst tractable special cases exist, there are also acyclic CP nets in which there are exponentially long chains of worsening flips between two outcomes. In the CP net of Figure 1, $\overline{a}b\overline{c}\overline{d}$ is worse than $abcd$.

Partial CP nets

We first introduce partial CP nets. These will be used to represent the preferences of the individual agents in a mCP net. A partial CP net is one in which certain features may not be ranked. Intuitively, this means that the agent is indifferent to the values of such features. The presence of non ranked features is needed since an agent's preferences may depend on another agent's preferences (e.g. Bob's preference for dessert or cheese depends on Alice's choice).

Partiality requires us to relax the semantics slightly. A *worsening flip* in a partial CP net is the change in the value of a variable A such that:

- if A is ranked, the flip is worsening in the CP table of A;
- if A is not ranked, it is worsening in the CP tables of all features that depend on A.

For example, if A is not ranked, and for B we have $a : b \succ \overline{b}$ and $\overline{a} : \overline{b} \succ b$, then passing from ab to $\overline{a}b$ is a worsening flip as we go from the best ranked position to the worst.

In addition to worsening flips, we now also have *indifferent* and *incomparable* flips. Indifferent flips are all those flips of a non-ranked variable such that, for each CP table of the features that depend on this feature, they are neither improving nor worsening. Suppose A is not ranked, B has three values, and we have the CP statement $a : b_1 \succ b_2 \succ b_3$ and $\overline{a} : b_3 \succ b_2 \succ b_1$. Then, passing from ab_2 to $\overline{a}b_2$ is an indifferent flip. Incomparable flips are all those flips which are neither worsening, nor improving, nor indifferent. For example, if A is not ranked, for B we have $a : b \succ \overline{b}$ and $\overline{a} : \overline{b} \succ b$, and for C we have $\overline{a} : c \succ \overline{c}$ and $a : \overline{c} \succ c$, passing from abc to $\overline{a}bc$ is an incomparable flip.

Each agent represents their preferences by a partial CP net. An agent prefers outcome α to outcome β (written $\alpha \succ \beta$) iff, in their partial CP net, there is a chain of flips from α to β where each flip is worsening or indifferent, and there is at least one worsening flip. This ordering is again a strict partial ordering. However, outcomes can now also be incomparable and indifferent. An agent is indifferent between outcomes α and β (written $\alpha \approx \beta$) iff at least one chain of flips between them consists only of indifferent flips. This requires the outcomes to differ only for the values of non-ranked features (e.g. $ab_2 \approx \bar{a}b_2$). Similarly one outcome α is incomparable to another β (written $\alpha \bowtie \beta$) iff it is not the case that $\alpha \succ \beta$, $\beta \succ \alpha$ or $\alpha \approx \beta$ (e.g. $ab_3 \bowtie ab_1$).

Partial CP nets can have more than one *optimal outcome* even if their dependency graphs are acyclic: there is one for each possible outcome of the non-ranked features. Finding one of the optimal outcomes in an acyclic partial CP net is linear: it is enough to choose any value for the non-ranked features, and to perform the forward sweep for the remaining features. Optimality testing in acyclic partial CP nets is also linear since, given an outcome, it is enough to set the non-ranked features to the values given by the outcome, and then perform the forward sweep from here. If the resulting outcome is the same as the given one, it is optimal, otherwise it is not.

Given a partial CP net and two outcomes, checking if one dominates another is as difficult as for CP nets. Checking indifference is instead $O(kn)$, where n is the size of the net and k is the number of non-ranked features, assuming a constant time to access a CP table. For each non-ranked feature, we need to check whether flipping it is indifferent for all features depending on it (there may be $O(n)$ of them).

mCP nets

We now put together several partial CP nets to represent the preferences of multiple agents. A **mCP net** is a set of m partial CP nets which may share some features, such that every feature is ranked by at least one of the partial CP nets. Graphically, an mCP net is obtained by combining the graphs of the partial CP nets so we have one occurrence of each shared feature. For simplicity, we assume that the partial CP nets in a mCP net are acyclic. We leave it as future work to see what happens when we relax this assumption. The "m" in mCP net stands for both multiple agents and the number of agents. Thus, a 3CP net is a mCP net in which $m = 3$. Note that a CP net is a 1CP net. Hence, mCP nets (with $m = 1$) immediately inherit all the complexity results for regular CP nets.

Each feature in a partial CP net N of an mCP net is shared, visible or private. Private features are ranked in N only, and are not visible to other partial CP nets. Visible features for N are those used in the conditions of N but not ranked there. Shared features are ranked in N and also in at least one other of the partial CP nets. For example, Bob's conditional preference on cheese or dessert can be represented within Bob's partial CP net as follows: $c_A : c_B \succ \overline{c_B}$ and $\overline{c_A} : \overline{c_B} \succ c_B$ where C_B is the private feature in Bob's partial CP net representing his preference for cheese or desert, and C_A is the visible feature for Bob from Alice's partial CP net representing her preference for cheese or desert.

An outcome for an mCP net is an assignment to all the features in all the partial CP nets to values in their domains. This outcome induces an outcome for each single partial CP nets forming the overall mCp net, by eliminating all the features which do not belong to the partial CP net.

Voting semantics

We will reason about a mCP net by querying each partial CP net in turn and collecting together the results. We can see this as each agent "voting" whether an outcome dominates another. We can obtain different semantics by collecting these votes together in different ways. Many of these semantics may be useful in a distributed setting, or when there are privacy issues. We do not need to know all the preferences of each partial CP net. We may just have to ask if anyone votes against a particular outcome.

The semantics we propose are based on known concepts, like Pareto optimality, lexicographic ordering, and quantitative ranking. Therefore the value of our proposal is more in the embedding of such semantics in the context of mCP nets, where indifference and incomparability coexist, rather than in their concept.

Given a mCP net and two outcomes α and β, let $S_\succ$, $S_\prec$, $S_\approx$ and $S_\bowtie$ be the sets of agents who say, respectively, that $\alpha \succ \beta$, $\alpha \prec \beta$, $\alpha \approx \beta$, and $\alpha \bowtie \beta$. Note that it is not possible for $|S_\succ| = |S_\prec| = |S_\bowtie| = 0$ (i.e. for every agent to be indifferent). Given any two outcomes, some agent must order them or say they are incomparable.

Pareto. This semantics is one of consensus. We say that one outcome α is better than another β (written $\alpha \succ_p \beta$) iff every agent says that $\alpha \succ \beta$ or $\alpha \approx \beta$. That is, $\alpha \succ_p \beta$ iff $|S_\prec| = |S_\bowtie| = 0$. Two outcomes are incomparable iff they are not ordered either way. An outcome is Pareto optimal iff no other outcome is better. Consensus is a stringent test (see (Ephrati & Rosenschein 1996) for the properties of a consensus approach in a multi-agent scenario), and outcomes will often be incomparable as a consequence.

Majority. An alternative criterion is just that a majority of the agents who are not indifferent vote in favor. We say that one outcome α is majority better than another β (written $\alpha \succ_{maj} \beta$) iff $|S_\succ| > |S_\prec| + |S_\bowtie|$. Two outcomes are majority incomparable iff they are not ordered either way. An outcome is majority optimal iff no other outcome is majority better.

Max. A weaker criterion is that more agents vote in favor than against or for incomparability. We say that one outcome α is max better than another β (written $\alpha \succ_{max} \beta$) iff $|S_\succ| > \max(|S_\prec|, |S_\bowtie|)$. Two outcomes are max incomparable iff they are not ordered either way. An outcome is max optimal iff no other outcome is max better.

Lex. The next semantics we consider assumes the agents are ordered in importance. If the first agent orders two outcomes then this is reflected in the final outcome. However, if they are indifferent between two outcomes, we consult the second agent, and so on. We say that one outcome α is lexicographically better than another β (written $\alpha \succ_{lex} \beta$) iff there exists some distinguished agent such that all agents higher in the order say $\alpha \approx \beta$ and the distinguished agent says $\alpha \succ \beta$. Two outcomes are lexicographically incomparable iff there exists some distinguished agent such that all agents higher in the ordered are indifferent between the two outcomes and the outcomes are incomparable to the distinguished agent. Finally, an outcome is lexicographically optimal iff no other outcome is lexicographically better.

Rank. The last semantics eliminates incomparability. Each agent ranks each outcome. Given a partial CP net, the rank of an outcome is zero if the outcome is optimal, otherwise it is the length of the shortest chain of worsening flips between one of the optimal outcomes and it. We say that one outcome α is rank better than another β (written $\alpha \succ_r \beta$) iff the sum of the ranks assigned to α is smaller than that assigned to β. Two outcomes are rank indifferent iff the sum of the ranks assigned to them are equal. Either two outcomes are rank indifferent or one must be rank better than the other. Finally, an outcome is rank optimal iff no other outcome is rank better.

Basic properties

The Pareto, Lex and Rank semantics define strict orderings. By comparison, neither the Majority or Max semantics induce a strict ordering. More precisely, the following properties can be proved.

Theorem 1 *$\succ_p$, $\succ_{lex}$ and $\succ_r$ are transitive, irreflexive, and antisymmetric. Thus they induce strict partial orders. $\succ_{maj}$ and $\succ_{max}$ are irreflexive and antisymmetric but may be not transitive.*

The five relations are closely related. We say that one binary relation R subsumes another S iff xRy implies xSy. We say that one binary relation R strictly subsumes another S iff R subsumes S but not vice versa. Finally, we say that two binary relations are incomparable iff neither subsumes the other.

Theorem 2 *$\succ_p$ strictly subsumes $\succ_{maj}$, $\succ_{lex}$ and $\succ_r$. Similarly, $\succ_{maj}$ strictly subsumes $\succ_{max}$. Any other pair of relations is incomparable.*

Figure 2 shows the subsumption schema over the five relations.

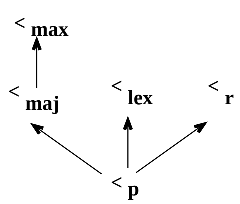

Figure 2: Implications among the six relations corresponding to the voting semantics.

Given two outcomes, we can therefore first try to compare them with $\succ_p$. If they are incomparable, we have three options: either we try to compare them with $\succ_{maj}$ and if they remain incomparable with $\succ_{max}$, or we try to compare them with $\succ_{lex}$, or we try to compare them with $\succ_r$. All options enlarge the set of pairs of outcomes which are comparable. However, only the last option (comparing them with $\succ_r$) is guaranteed to order any two outcomes.

Notice that in all the five relations, except Rank, it is not possible for two outcomes to be indifferent, since we assume that each feature is ranked by at least one of the partial CP nets, while indifference in the qualitative relations (Pareto, Max, Majority, and Lex) means indifference for everybody.

Optimality

From these subsumption results, the following relationships immediately hold between the various optimal outcomes. For example:

- a majority, lexicographical or rank optimal outcome is necessarily Pareto optimal. However, the reverse is not true. For instance, a Pareto optimal outcome is not necessarily rank optimal.
- A max optimal outcome is necessarily majority optimal, but the reverse is not true.

Similarly, from the incomparability results, we can deduce incomparability also for the respective optimal outcomes.

A fundamental question is whether any optimal outcomes exist. While this is obvious for relations which induce a finite strict order like Pareto, it is not immediately obvious for all of the relations. Nevertheless, optimal outcomes always exist for them. The main reason for this result is that all relations are antisymmetric (see Theorem 1), and thus there can be no cycles in the ordering.

Notice that the set of Pareto optimal outcomes does not necessarily coincide with the set of outcomes which are optimal for all the CP nets. In fact, if an outcome is optimal for all, then it is Pareto optimal. However, the converse is not true in general. For example, consider the set of optimal outcomes of a single partial CP net. If they are all incomparable (no matter if they are not optimal for the other CP nets), then they are also Pareto optimal for the mCP net.

We will now give several results on testing optimality and finding optimal outcomes. We assume that there are n binary features in the mCP net, k in the largest partial CP net, and l maximum number of visible features in a partial CP net.

To test if an outcome is Pareto optimal, we need to compare ($O(2^k)$) the given outcome with all other outcomes ($O(2^n)$) in all partial CP nets (m). Similarly for Max and Majority. For lexicographical optimality, we check if the given outcome is optimal for the first agent ($O(k)$). If it is not, it is not lexicographically optimal. If it is, then, for any optimal outcome of the first agent to which it is indifferent (there may be 2^l of them), we must ask the second agent if the given outcome is better ($O(2^k)$). If also the second agent is always indifferent, we must turn to the next agent and so on (m agents). Finally, for rank optimality, we need to first rank all outcomes for each partial CP net. To do this, we start with the optimal ones, which are $O(2^l)$ and have rank 0, and for each of them we build the flipping sequence to any other of the outcomes (which are 2^k). Then, for each outcome (2^n), we compute the global rank by summing the ranks in each partial CP net (the rank tables have 2^k elements) and we check that the rank of the given outcome is minimal.

Summarizing, testing optimality is easy for Lex if the size of the partial CP nets (k) is bounded, and it is difficult for Pareto, Majority, Max and Rank.

Theorem 3 *Testing optimality has the following complexity:*

- *for Pareto, Majority and Max: $O(m2^{n+k})$ time;*

- *for Lex:* $O(m2^{l+k})$*;*
- *for Rank:* $O(m2^{l+k} + 2^n m2^k) = O(m2^{n+k})$.

We now consider the problem of finding an optimal outcome. For Majority, we need to compare ($O(2^k)$) all outcomes (2^n) to all other outcomes ($O(2^n)$ in all partial CP nets (m). To find a max optimal outcome, we must do the same steps as for Majority. To find a lexicographically optimal outcome, we compute all the optimal outcomes for the first agent ($O(k2^l)$), and then we do the same steps as for testing optimality ($O(m2^l2^k)$) for each one of them (they can be 2^l). For Pareto optimality, since lexicographically optimals are also Pareto optimals, the complexity is the same as for Lex. To find a rank optimal outcome, we need to perform the same steps as for testing optimality, except that we don't have to compare with the given outcome. However, this does not change the overall complexity. To summarize, finding an optimal outcome is easy for Pareto and Lex if the size of the partial CP nets (k) is bounded, and it is difficult for Majority, Max and Rank. Thus, with respect to optimality testing, Pareto now falls into the easy category.

Theorem 4 *Finding an optimal outcome has the following complexity:*

- *for Majority and Max:* $O(m2^{2n+k})$*;*
- *for Lex and Pareto:* $O(m2^{2l+k})$*; the same complexity holds also to find all optimal outcomes, not just one;*
- *for Rank:* $O(m2^{n+k})$ *(for the first optimal outcome, linear for the others).*

Dominance

We again assume that there are n binary features in the mCP net, k in the largest partial CP net, and l visible features.

Theorem 5 *Given two outcomes, testing if one dominates the another has the following complexity:*

- *for Pareto, Majority, Max, and Lex:* $O(m2^k)$*;*
- *for Rank:* $O(m2^{n+k})$.

To determine if $O_1 \succ_p O_2$, we must test whether $O_1 \succ O_2$ or $O_1 \approx O_2$ for all m partial CP nets. Each of these tests takes at most $O(2^k)$ time. To determine if $O_1 \succ_{maj} O_2$, we must also test whether $O_1 \succ O_2$ for all m partial CP nets. Only the last test may give us the required majority. Similarly, to determine if $O_1 \succ_{max} O_2$, we must test whether $O_1 \succ O_2$ for all m partial CP nets. Even for Lex, in the worst case, we must check dominance or indifference in all CP nets, since it may be that they are indifferent for all partial CP nets except the last one. For Rank, a brute force algorithm would rank all outcomes and then compare the ranks of the given two. This takes $O(m2^{n+k})$ as explained above for testing Rank optimality. Thus, although Rank gives us a quantitative ordering, it is easier to test dominance in the qualitative semantics (Pareto, Majority, Max, and Lex). In fact, if the number and size of the partial CP nets are bounded, it takes constant time.

The fact that Majority and Max may be not transitive implies that for such semantics we cannot optimize any dominance test exploiting the transitive property. Thus, while in the worst case the complexity is the same as for Pareto and Lex, in practice dominance testing can be much easier for such two semantics.

Fairness

Having cast our semantics for mCP nets in terms of voting, it is appropriate to ask if Arrow's theorem (Arrow 1986) about the impossibility of a fair electoral system applies. Are we fairly combining together the preferences of the individual agents? Observe that, a mCP net (and its constituent partial CP nets) can represent preference, indifference and incomparability. By comparison, the votes in an election only express preference and indifference, whilst the election result only expresses society's preference. In addition, partial CP nets can only represent orderings which decompose into independent conditional CP statements, whereas voters in an election can order their votes in any way. As a result, Arrow's theorem does not immediately apply to mCP nets.

In short, Arrow's theorem states that no voting system which totally orders three or more candidates can be fair. That is, no voting system cam be free, transitive, independent to irrelevant alternatives, monotonic and non-dictatorial (Arrow 1986). We will adapt these terms to our scenario. We say that a voting semantics for mCP nets is **free** iff it is possible to represent any possible ordering of the outcomes. Note that 1CP nets cannot directly represent all possible orderings. For instance, no CP net can represent $ab \succ \overline{a}\overline{b} \succ a\overline{b} \succ \overline{a}b$. However, by combining together features, CP nets can represent any ordering. To return to the last example, we would have to replace the two binary features by a single feature with four values. We say that a voting semantics is **transitive** iff the ordering it defines is transitive. Note that the ordering within each partial CP net in a mCP net is transitive by construction. We say that a voting semantics is **independent to irrelevant alternatives** iff the ordering between two outcomes only depends on how the partial CP nets vote on these two outcomes; their votes on other outcomes do not matter. We say that a voting semantics is **monotonic** iff if one agent changes from $\beta \succ \alpha$ or $\beta \approx \alpha$ to $\alpha \succ \beta$ then α cannot become less preferred. Finally, we say that a voting semantics is **non-dictatorial** iff the ordering depends on more than one particular agent. Note that, in assessing if there is a dictator, we ignore incomparability and only consider outcomes which are ordered or indifferent. For example, if one agent says that all outcomes are incomparable, then according to the Pareto semantics, all outcomes are incomparable. However, we do not consider this agent to be a dictator as she does not force any of her *preferences* on the rest. We will say that a semantics for mCP nets is **fair** iff it satisfies all five definitions above.

It is possible for mCP nets to be fair. For example, the Pareto semantics satisfies all five of these properties. The Pareto semantics is transitive, since it is a strict order. Whether one outcome is better, indifferent or incomparable to another only depends on how each agent votes on these two outcomes. Hence, the Pareto semantics is independent to irrelevant alternatives. The Pareto semantics is also monotonic since improving the preference for an outcome in one partial CP net can only move this up the ordering or leave it in the same place. Finally, the Pareto semantics is non-dictatorial as the ordering clearly depends on more than one particular agent.

On the other hand, none of the other semantics are fair. Majority and Max may be not transitive (see Theorem 1). Proofs that the Majority and Max semantics are free, independent to irrelevant alternatives monotonic and non-dictatorial are similar to those for the Pareto semantics. To show that the Lex semantics may be dictatorial, suppose that the first agent orders all her outcomes. Then this agent will dictate the final outcome. Proofs that the Lex semantics are free, transitive, independent to irrelevant alternatives and monotonic are similar to those for the Pareto semantics.

To show that the Rank semantics may not be independent to irrelevant alternatives, suppose $\alpha \succ_r \beta$. That is, the sum of the ranks assigned to α is less than that assigned to β. By introducing irrelevant alternatives, we can increase the rank of α so that its rank is larger than β. Proofs that the Rank semantics are free, transitive, monotonic and non-dictatorial are similar to those for the Pareto semantics.

The following theorem summarizes these results over the five semantics.

Theorem 6 *The Pareto semantics is free, transitive, independent to irrelevant alternatives, monotonic and non-dictatorial. The Majority and Max semantics are free, independent to irrelevant alternatives, monotonic and non-dictatorial, but may not be transitive. The Lex semantics is free, transitive, independent to irrelevant alternatives and monotonic, but may be dictatorial. The Rank semantics is free, transitive, monotonic, and non-dictatorial but may not be independent to irrelevant alternatives.*

Whilst the Pareto semantics is fair, this comes at a price. The Pareto semantics orders fewer outcomes than the Majority, Max, Lex or Rank semantics. If we wish to order two outcomes, we may therefore have to sacrifice some aspect of fairness.

Comparison

The following table summaritzes the main properties of the semantics considered in this paper for mCP nets. In particular, we show whether the semantics yields a strict order or not, we compare the complexity of testing optimality and dominance, and of finding an optimum. Finally, we also indicate whether the semantics is fair or not. This table lets us compare the five semantics easily.

For example, Lex is the semantics to choose if we don't care about fairness and it is reasonable to order the agents. On the other hand, Pareto is the best in terms of complexity (except for optimum testing). However, as it looks for a consensus among all agents, it orders the fewest pairs of outcomes. Max and Majority ordering are less desirable in terms of complexity, since only dominance testing is easy (under certain assumptions). Moreover, they are equal with respect to fairness criteria. Since Majority subsumes Max, Majority has more optimal outcomes and a weaker ordering. Finally, the Rank semantics has bad complexity, and is not fair. This bad complexity seems a result of merging the qualitative nature of CP nets with the quantitative spirit of a rank. Note that this complexity is only seen the first time two outcomes are ordered, and subsequent queries can be answered in linear time.

	Pareto	Maj	Max	Lex	Rank
S.O.	yes	no	no	yes	yes
Test opt.	diff	diff	diff	e-k	diff
Find opt.	e-k	diff	diff	e-k	diff
Dominance	e-k c-m	e-k c-m	e-k c-m	e-k c-m	diff
Fairness	yes	no	no	no	no

Table 1: Comparison among the five semantics. Legenda: S.O. = strict order, diff = difficult, e-k = easy if k bounded, c-m = constant if m bounded).

Conclusions and Future Work

We have introduced mCP nets, an extension of the CP net formalism to represent the qualitative and conditional preferences of multiple agents. We have given a number of different semantics for reasoning with mCP nets. The semantics are all based on the idea of individual agents voting. We then described how to test optimality and preference. We also discussed whether the voting schemes fairly combine together the preferences of the individual agents.

We would like to explore distributed mCP nets. We have assumed so far that decision making is coordinated in some central place. However, in many applications, decision making is more distributed. An additional direction for further research is the extension of our work to CP nets where it is possible to partially order the values of a feature. In this paper we have considered the case in which one either totally orders the values or it does not order them at all.

References

Arrow, K. 1986. Rationality of self and others in an economic system. *Journal of Business* Vol. 59 (4).

Boutilier, C.; Brafman, R.; Hoos, H.; and Poole, D. 1999. Reasoning with conditional ceteris paribus preference statements. In *UAI-99*, 71–80.

Domshlak, C., and Brafman, R. 2002. Cp-nets - reasoning and consistency testing. In *KR-02*, 121–132.

Domshlak, C.; Rossi, F.; Venable, K.; and Walsh, T. 2003. Reasoning about soft constraints and conditional preferences: complexity results and approximation techniques. In *IJCAI-03*.

Doyle, J., and Wellman, M. 1994. Representing preferences as ceteris paribus comparatives. In *AAAI Spring Symposium on Decision-Making Planning*, 69–75.

Ephrati, E., and Rosenschein, J. S. 1996. Deriving consensus in multiagent systems. *Artifi cial Intelligence* 87:21–74.

Yager, R. R. 2001. Fusion of multi-agent preference orderings. *Fuzzy Sets and Systems* 117:1–12.