Making Social Choices from Individuals' CP-nets

Keith Purrington and Edmund H. Durfee
Computer Science and Engineering
University of Michigan, Ann Arbor, MI 48109 USA
purringk@umich.edu, durfee@umich.edu

ABSTRACT

CP-nets are an attractive model for representing individual preferences, in part because they allow us to find the best outcome for an agent in time that is proportional to just the number of features in an outcome. In this paper, we investigate whether similar efficiencies can apply to finding the best social outcome for agents whose individual preferences are captured in CP-nets. Because CP-nets provide only qualitative information, we adopt a way to compare outcomes across agents based on each outcome's relative standing in the individuals' spaces of possible outcomes. This in turn guides the search through the outcome preference graphs that are induced by the agents' CP-nets to find the optimal social outcome. Because these induced preference graphs are exponential in the number of features, we examine the conditions under which the agents can search directly using their CP-nets, and show that our approach yields near-optimal social outcomes in exponentially less time.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*multiagent systems*

General Terms

Algorithms

Keywords

Multiagent Systems :: Argumentation, negotiation, and conflict handling
Multiagent Systems :: Societal aspects :: Social and organizational structures

1. INTRODUCTION

When multiple agents interact, there is a need to make social choices—to find outcomes that are, in some way, best for a group. We assume that the agents we are dealing with have encoded their preferences in CP-nets [1]. CP-nets are attractive for the natural way in which they allow agents to express preferences and because they allow very efficient preferential optimization. The question we seek to answer is whether such efficiencies apply in the multi-agent setting, where we take maximin as the standard of optimality, echoing John Rawls [2].

The most closely related work [3] combines agents' individual CP-nets into what the authors call an mCP-net. This structure can be queried to find outcomes satisfying a variety of voting criteria. While our work lacks much of the flexibility of that approach, our focus has been to find efficient algorithms by exploiting the underlying CP-net structure. In contrast, the inference in [3] requires searching in the entire outcome space—producing algorithms that are exponential in the number of feature variables.

We begin by defining a metric that gives us one way (of possibly many) to assess the quality of outcomes from a social perspective, despite the limitations of the representation. From this, a naïve algorithm follows naturally, allowing us to find optimal outcomes, albeit with an algorithm that is combinatorial in the number of feature variables. Finally, by introducing an additional semantic assumption, we can directly construct socially optimal outcomes, as in the single-agent case.

2. CP-NETS OVERVIEW

In [4], Boutilier *et al*. present a model (called a CP-net) that describes an agent's conditional *ceteris paribus* preferences. Briefly, a CP-net is a directed graph, in which each node is annotated with a conditional preference (CP) table. Each node's CP table indicates the preferred value for that node's variable, conditioned on the values of its parents (the nodes that point to it).

For our work, the two important features of CP-nets are that the representation induces a partial order on outcomes (complete variable assignments), and that preferential optimization can be done very efficiently.

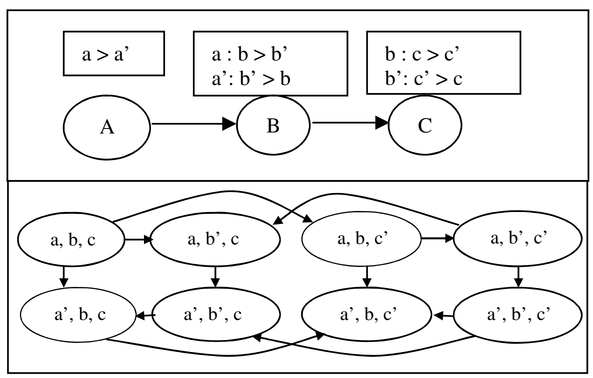

Figure 1: CP-net and its induced preference graph

Under the CP-net semantics, outcomes are only directly comparable if they disagree on the value of exactly one variable; in such a case, the outcome that assigns a more preferred value to that variable is better (it "dominates" the less preferred outcome). These relations can be captured in the induced preference graph of a CP-net, which defines a partial order on outcomes. Figure 1 shows a simple CP-net and its induced preference graph.

AAMAS'07, May 14-18, 2007, Honolulu, Hawaii, USA.

978-81-904262-7-5 (RPS) ©2007 IFAAMAS

The authors of [4] show that, given an assignment to some (possibly empty) subset of the variables (called "evidence"), the most preferred outcome consistent with the evidence can be found via a simple greedy assignment. Proceeding in topological order, if each variable is set to its most preferred value, given its parents' values, the resulting outcome will be better than any other possible assignment. The efficiency of this optimization algorithm is one of the chief attractions of the CP-nets representation.

3. MULTI-AGENT OUTCOME COMPARISON

Consider the case of two agents with the following partial orders: agent A_1, whose preferences are shown in Figure 1, and agent A_2, shown in Figure 2, which has three unconditional preferences (for a', b', and c'). For the sake of illustration, assume a naïve social choice function that considers only each agent's first choice, restricting our choice to *(a,b,c)* or *(a',b',c')*. Deciding between these competing outcomes requires that we have a way to assess the quality of each, for each agent.

In the given example, either outcome ensures that one agent gets its first choice, so our concern is, in each case, for the agent that fails to get its first choice. Outcome *(a,b,c)* is best for A_1, but is the worst possible outcome for A_2. On the other hand, *(a',b',c')* is only 5th or 6th of 8 for A_1, while being the best of all for A_2, and thus is intuitively the better social choice (of these 2 outcomes).

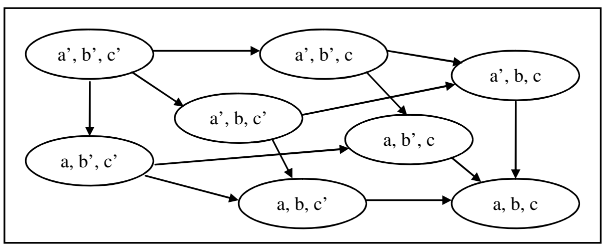

Figure 2: Preference graph for three unconditional preferences. A negative value is preferred for each variable.

By generalizing this idea of the *degree* to which an outcome is good for an agent, we can define one way to assess competing outcomes in a multi-agent setting. We make no claims that this is the "right" way, since clearly if the agents had different strengths of preference then alternative outcomes might be optimal. The CP-net representation provides too little information to make a definitive selection; the algorithm we develop, however, is applicable to other ways of assessing competing outcomes, so long as they can be grouped into better and worse tiers.

Because the presence of a long path from the root to an outcome indicates that the outcome is relatively bad, based on the long chain of successively worse outcomes, we use the length of this chain to divide outcomes into tiers, as follows.

Definition 1: *An outcome belongs to tier N* ***IFF*** *it is dominated by some outcome in tier N-1, and is dominated by no outcome in tier K, for all K ≥ N. We say that outcome x in tier X is better than any outcome y in tier Y>X, worse than any outcome w in tier W<X, and incomparable with any other outcome x' also in tier X.*

We must also account for the fact that the "shape" of the outcome space may be very different for different agents. In our example, the second agent has a wider, shallower graph with a larger branching factor. Thus, to compare outcomes across agents, we need to normalize the information provided by the tiers. For each outcome, we define a *satisfaction interval,* or just *interval,* based on its position in the tier structure. These intervals are written as subintervals of [1, 0], where the endpoints encode the proportion of outcomes that are better and that are worse, respectively.

Outcomes can now be evaluated in a multi-agent setting using these satisfactions intervals. Because we are looking for maximin outcomes, our algorithm can compare intervals based on their low endpoints, using the maxes only to break ties.

4. FINDING OPTIMAL OUTCOMES

If efficiency is of no concern, we can use the induced preference graph for each agent to find the best social outcome. Generating, and even representing the induced preference graph for a CP-net turns out to be combinatorial in the number of variables, which makes it generally impractical as the basis of an algorithm. Although this algorithm is inefficient, it provides a good baseline for comparing the results of more sophisticated algorithms.

The algorithm operates by simultaneously searching agents' outcome graphs from the top down, using the satisfaction interval associated with each tier to control the rate at which the search proceeds for each agent. For each agent, a set of candidate outcomes is maintained that contains all the outcomes that an agent is willing to consider. For each agent, the algorithm examines the highest tier of outcomes that are not currently in its candidate set, and for the agent(s) with the highest min for this next tier, adds those outcomes to the candidate set(s). Once the intersection of the agents' candidate sets is non-empty, one of the outcomes in the intersection must be maximin. Checking the max values for every agent for the outcomes in the intersection will find one that is maximin. (This paper is necessarily scant on details. Please contact the first author for a paper providing a more detailed description of the algorithms, proofs, etc.)

Optimality Proof Sketch: Showing maximin optimality is straightforward by contradiction. If the outcome *O* returned by the algorithm is not optimal, some other outcome O^* must be, instead. By definition, O^* must either have the same min as *O* and a higher max, or it must have a larger min. In each case, the algorithm would have returned O^* instead, either by finding it in the intersection earlier, or in the final step that compares maxes. ■

In principle, a similar approach can be used to find an optimal outcome for a single agent. In this case, however, the structure of the CP-net can be exploited to directly construct an optimal outcome, rather than performing the costly search through the induced preference graph. We now examine an approach for the multi-agent setting that similarly works directly on CP-nets.

5. MULTI-AGENT FORWARD SWEEP

In moving from searching through the outcome graphs to constructing an outcome by directly assigning variables, we must adapt the way in which we handle agents' satisfaction intervals. They were originally defined on full outcomes, using the outcome graph. Now, however, we will never have the full outcome graphs at our disposal; furthermore, we are working primarily with partial assignments to the variables. As a result, we must introduce an additional assumption to allow the computation of the necessary intervals.

The correctness of the single-agent forward sweep relies on the fact that any variable is at least as important as all of its descendents, combined. We extend this semantic assumption, taking the inequality to be strict. With this assumption, we can compute the satisfaction interval for any partial assignment to the variables by iterative refinement at each level of the CP-network.

Beginning with the interval [1,0], intervals are refined based on an agent's purely local satisfaction with just the variables in a single level. This, in turn is found by counting the number of ways the variables in the level could be assigned strictly more and less preferably than the current assignment.

Then, the additional assumption allows us to "project" the new local interval into the overall interval found after the previous level to generate the updated interval. Thus, a local interval of [1, .5] indicates that the updated partial assignment puts the agent in the top half of its previous interval. Likewise, and interval of [.66, .33] places the agent in the middle third of its previous interval.

The following theorem is very important in showing the correctness of the later algorithms. It follows directly from the definitions of the interval updates.

Theorem 1: *If every variable in the first j levels of a CP-net takes on its preferred value, the interval for the partial assignment to those n variables is [1, $1-1/2^n$], independent of the network topology for those variables.*

5.1 Forward Sweep in Simple Networks

We first begin by restricting our attention to networks that are identical in topology. This allows us (for the moment) to consider variables one level (of the CP-net) at a time, and to be sure that agents' satisfaction intervals will shrink at the same rate. The algorithm can be divided into two fundamentally different stages. In the first stage, the algorithm allows both agents to assign the variables in the next level of their respective CP-nets, relying on the identical topologies to equalize the rate at which agents increase their satisfaction. This stage persists until an agent finds that a variable that it would like to assign has already been set to a non-preferred value by the other agent. Theorem 1 allows us to show that during this stage of the algorithm, this greedy assignment does, indeed, ensure a maximin outcome.

The second stage of the algorithm begins once one agent is more disappointed than the other with the assignment to some level. Again using Theorem 1, we can show that at this point that agent *must* come out worse than the other in the final outcome, and thus should be given all of its remaining preferences via a simple single-agent forward sweep.

To identify the point at which the algorithm switches from the first stage to the second, the algorithm performs a small amount of look ahead, to figure out what variables in the next level are in conflict. For the variables that both agents would like to assign in the next level, care must be taken in deciding how they are to be divided. Where possible, they should be split so that each agent is disappointed about the assignments to the same number of variables. When this is not possible, the algorithm must test the competing assignments to the conflicting variables by comparing the results of single-agent forward sweeps, in order to determine which of the ways of dividing the variables ultimately makes the worst-off agent happiest.

5.2 Networks of Arbitrary Topology

When we move from identical to arbitrary topologies for agents' CP-nets, the same techniques allow us to show that a greedy assignment is optimal, but there are many more details involved in specifying the full algorithm. The earlier proof relies on the fact that a variable occurs strictly earlier in any topological order for one agent than the other. By tracking how many variables each agent has assigned and the size of the next level, we can decide when an agent can assign the variables in that next level.

This approach works up until the point that an agent looks ahead and sees that assigning to the variables in its next level will bring it into conflict with variables in the other agent's network. When this happens, it is unclear at what point the agent should be allowed to begin assigning variables in that next level.

In particular, the order in which variables are considered becomes very important. If an agent A_1 has its choice of four variables in a level to next assign, it should first assign the variable that is at the lowest level in A_2's CP-net. The reason for this is that A_2 may suddenly find that it is worse off under any possible outcome (as in the simpler cases) and needs to assign the rest of the variables. In this case, A_1 should have created potential conflicts with the lowest-level variables in A_2's network that it could. We are still working out the final details of the general case

6. CONCLUDING REMARKS

By defining one metric (of several reasonable ones) to compare outcomes across agents in a way that is consistent with the CP-net semantics, we developed an algorithm for finding maximin optimal social outcomes by searching through agents' preference graphs. To avoid searching the outcome space that is exponential in the number of feature variables, we introduced an additional semantic assumption that has enabled us to exploit the structure of the underlying CP-nets to construct a maximin outcome very efficiently.

ACKNOWLEDGEMENTS

The work reported in this paper was supported, in part, by the National Science Foundation under grant IIS-0534280. We would like to thank Martha Pollack and Erin Rhode for valuable feedback on this work.

7. REFERENCES

[1] Boutilier, C., Brafman, R.I., Domshlak, C., Hoos, H.H., and Poole, D. "CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements." *Journal of Artificial Intelligence Research (JAIR)*, 2003.

[2] Rawls, J. *A Theory of Justice*. Cambridge, Massachusetts, Harvard University Press, 1971

[3] Rossi, F., Venable, K.B., Walsh, T. "mCP nets: Representing and Reasoning with Preferences of Multiple Agents." In: *Procs. of the 19th Conf. on Artificial Intelligence,* LNCS 749, AAAI Press 729—734, 2004