

An Efficient Procedure for Collective Decision-making with CP-nets

Minyi Li and Quoc Bao Vo and Ryszard Kowalczyk¹

Abstract. This paper studies the problem of collective decision-making in the case where the agents' preferences are represented by CP-nets (conditional preference networks). In many real-world decision-making problems, the number of possible outcomes is exponential in the number of domain variables. Most related works either do not consider computational concerns, or depend on a strong assumption that all the agents' CP-nets share a common preferential-independence structure. To this end, we introduce a novel procedure for collective decision-making with CP-nets. Our proposed approach allows the agents to have different preferential-independence structures and guarantees Pareto-optimality. Our experimental results demonstrate that our proposed procedure is computationally efficient and produces the results that are close to the fair Minimax solution.

1 Introduction

Classical decision theory usually represents agents' preferences mathematically by utility functions. However, utility-based preference elicitation is complicated and typical users may not be able to provide much more than qualitative rankings of outcomes [8]. This makes the utility-based decision-making models particularly hard to apply in most real world applications. Moreover, since many decision-making problems involve a *combinatorial domain*², it is practically impossible to represent preference relations explicitly. The AI community has been developing languages for representing preferences on such domains in a succinct way, while preserving a good readability that similar to the way users express their preferences in natural languages [1, 8, 2, 3, 5].

In this paper, we investigate the theory of CP-net (conditional preference networks) as a formal model for representing and reasoning with the agents' preferences. The motivation of applying CP-nets in collective decision-making scenarios is that, CP-nets specifies individual preference relations in a relatively compact, intuitive, and structured manner, and thus enables us to easily encode human preferences and highly supports real world applications.

However, once the users' preferences have been elicited and represented by CP-nets in the agent system, the following problem then arise: what are the desirable properties of a chosen outcome?

Optimality and fairness are two typical design goals of collective decision-making mechanisms [4]. By optimal solution, we refer to a solution that is *Pareto-optimal*. An outcome is Pareto-optimal if there exists no other outcome which is at least as preferred as this outcome for all the agents and strictly better for at least one agent [4]. Another

desirable property considering *fairness* would be to ask for an outcome maximizing the well-being of the worst-off agent, called the *egalitarian social welfare*. However, this cannot be directly used in the context of CP-nets, because CP-nets do not provide any quantitative measurement (i.e. utilities) of the agents' relative satisfactions [12]. Rossi *et al.* [15] present a Ranking semantics to computationally measure the quality of an outcome for a given agent using the length of the path between the optimal outcome and that outcome. In order to preserve all the strict preferences induced by a CP-net, in this paper we consider a similar ranking semantics based on the longest path. Given an individual CP-net, the rank of an outcome is the *longest* path between the optimal outcome and that outcome in the corresponding induced preference graph of the given CP-net. Based on ranking semantics, we argue that the goal of the collective decision-making procedure should be to produce *Pareto optimal* outcomes that *minimize the maximum rank of the agents*. Accordingly, an outcome is called a *Minimax* solution if it minimizes the maximum rank of the agents among all possible outcomes. Note that finding an accurate Minimax solution over the outcome space requires each agent to rank each possible outcome (called *Ranking Query* in the rest of the paper), compute the global rank of each outcome by choosing the maximum rank among the agents, and check that whether the rank of the given outcome satisfies the above condition. When the number of all possible outcomes is large, it is unpractical to search for an accurate Minimax solution.

Our objective is to design an efficient procedure for collective decision-making with CP-nets, which can lead rational agents to *optimal* and *fair* outcomes. Decision-making with a collection of CP-nets have been studied in the literature, i.e. voting and preference aggregation [11, 15, 13, 9, 16, 6, 17, 10]. However, most existing works either do not consider computational concerns, or depend on a strong assumption of common preferential-independence structure. For instance, Rossi *et al.* [15] propose various voting semantics for aggregating multiple agents' preferences represented by CP-nets; but they do not consider computational concerns. Dominance testing for an arbitrary CP-net is PSPACE-complete [7]. While outcome optimization with multiple agents furthermore requires dominance testing on each pair of possible outcomes on each individual CP-net. This problem is likely not a member of NP or coNP, but is in fact, even harder. Thus, their approach may not be applicable when the number of possible outcomes from which to choose is large, especially when the set of outcomes has a combinatorial structure. For example, having 10 binary variables each involved agent would need to compare $\binom{2^{10}}{2} = 523776$ pairs of outcomes. Instead of applying standard voting rules directly, Lang [9] consider decomposition with voting rules assuming that the agents' preferences have a common preferential-independence structure. However, such a demanding as-

¹ Swinburne University of Technology, Australia, email: myli,BVo,RKowalczyk@swin.edu.au

² The number of possible outcomes is exponential in the number of domain variables.

sumption is unlikely to be met in most real world applications. Xia [16] extend this setting to cope with the case where agents do not share a common preferential-independence structure. However, the formulation in [16] does not come with any practical algorithm for computing the winners of a particular voting rule.

Our proposed procedure makes a distinct contribution from other existing voting protocols. It enables agents with different preferential-independencies to trade off between variables based on their CP-nets. It is computationally efficient, guarantees Pareto-optimality and produces the results that are close to the fair Minimax solution. The paper is structured as follows. We first provide some background knowledge in Section 2. We then present the proposed approach in Section 3 and analyse its general properties in Section 4. In Section 5 we present the experimental results, and finally we provide the concluding remarks and discuss about the future work in Section 6.

2 Background

Let $\mathbf{V} = \{X_1, \dots, X_m\}$ be a set of m variables. For each $X_k \in \mathbf{V}$, $D(X_k)$ is the *value domain* of X_k . A variable X_k is *binary* if $D(X_k) = \{x_k, \bar{x}_k\}$. If $\mathbf{X} = \{X_{i_1}, \dots, X_{i_p}\} \subseteq \mathbf{V}$, with $i_1 < \dots < i_p$ then $D(\mathbf{X})$ denotes $D(X_{i_1}) \times \dots \times D(X_{i_p})$ and $\mathbf{x}$ denotes an assignment of variable values to $\mathbf{X}$ ($\mathbf{x} \in D(\mathbf{X})$). If $\mathbf{X} = \mathbf{V}$, $\mathbf{x}$ is a *complete assignment*; otherwise $\mathbf{x}$ is called a *partial assignment*. We denote by $\mathbf{x}[X_k]$ the value $x_k \in D(X_k)$ assigned to variable X_k by that assignment; and $\mathbf{x}[\mathbf{W}]$ denotes the assignment of variable values $\mathbf{w} \in D(\mathbf{W})$ assigned to the set of variables $\mathbf{W} \subseteq \mathbf{X}$ by that assignment.

Let $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ be nonempty sets that partition $\mathbf{V}$ and $>$ a preference relation over $D(\mathbf{V})$. $\mathbf{X}$ is (*conditionally*) *preferentially independent* of $\mathbf{Y}$ given $\mathbf{Z}$ iff for all $\mathbf{x}, \mathbf{x}' \in D(\mathbf{X})$, $\mathbf{y}, \mathbf{y}' \in D(\mathbf{Y})$, $\mathbf{z} \in D(\mathbf{Z})$:

$$\mathbf{xyz} > \mathbf{x'y'z} \text{ iff } \mathbf{xy'z} > \mathbf{x'y'z}$$

A CP-net $\mathcal{N}$ [8] over $\mathbf{V}$ is an annotated directed graph $\mathcal{G}$ over $X_1, \dots, X_m$, in which nodes stand for the problem variables. Each node X_k is annotated with a conditional preference table $CPT(X_k)$, which associates a total order $>_{X_k|\mathbf{p}}$ with each instantiation $\mathbf{p}$ of X_k 's parents $Pa(X_k)$, i.e. $\mathbf{p} \in D(Pa(X_k))$. For instance, let $\mathbf{V} = \{X_1, X_2, X_3\}$, all three being binary, and assume that the preference of a given agent over all possible outcomes can be defined by a CP-net whose structural part is the directed acyclic graph $\mathcal{G} = \{(X_1, X_2), (X_1, X_3), (X_2, X_3)\}$; this means that the agent's preference over the values of X_1 is unconditional, preference over the values of X_2 (resp. X_3) is fully determined given the values of X_1 (resp. the values of X_1 and X_2). The preference statements contained in the conditional preference tables are written with the usual notation, that is, $x_1\bar{x}_2 : x_3 > \bar{x}_3$ means that when x_1 is true and x_2 is false then $X_3 = x_3$ is preferred to $X_3 = \bar{x}_3$.

In this paper, we make the classical assumption that each agent A_i 's preference is captured by an acyclic binary-valued CP-net $\mathcal{N}_i$, and the ordering $>_{A_i|\mathbf{p}}$, $\mathbf{p} \in D(Pa_i(X_k))$, expressed in the conditional preference table is total. As such, conditional expressions of indifference are not allowed, and thus an agent will not be indifferent between two outcomes. For each agent A_i , $\mathcal{N}_i$ induces a preference ranking over the outcome space: o_1 is strictly preferred to o_2 for agent A_i if $\mathcal{N}_i$ entails $o_1 > o_2$ (denoted by $\mathcal{N}_i \models o_1 > o_2$). And $\mathcal{N}_i \models o_1 > o_2$ iff there exists a sequence of *improving flips* from o_2 to o_1 respecting the conditional preference tables of $\mathcal{N}_i$; otherwise, $\mathcal{N}_i \not\models o_1 > o_2$. An *improving flip* is the flip of the value of a single variable X_k within an outcome to directly compute a preferred

(better) outcome for agent A_i based on $CPT_i(X_k)$. *Note that the preference relation induced from a CP-net is generally not complete* [8]. Two outcomes o_1 and o_2 may also be incomparable for an agent A_i : $\mathcal{N}_i \not\models o_1 > o_2$ and $\mathcal{N}_i \not\models o_2 > o_1$.

3 Our proposed procedure

In this section we present our proposed procedure CDCP: Procedure for Collective Decision-making with CP-nets. In this paper, we consider binary-valued variables. Assume a set of n agents $\mathbf{A} = \{A_1, \dots, A_n\}$ who make decision over a set of m binary variables $\mathbf{V} = \{X_1, \dots, X_m\}$. The preference of each agent A_i is captured by an acyclic CP-net $\mathcal{N}_i$.

First phase of CDCP: The first phase of CDCP involves a recursive procedure CDCP.P1, and each recursive call accepts two arguments:

1. $\mathbf{z}$: A reference point, which is a partial assignment to the problem variables $\mathbf{Z} \subseteq \mathbf{V}$.
2. $\mathbf{S}$: A set of agents who are involved in the current call.

The initial call to CDCP.P1 is done with $\mathbf{z} = \langle \rangle$ (all the variable values are Null) and the whole set of agents $\mathbf{S} = \mathbf{A}$. CDCP.P1 works as follows. It starts with an empty set of outcomes, and continuously extends it by adding new Pareto-optimal outcomes. Each call to CDCP.P1 performs the following three steps:

Step 1: If only one agent is involved in the current call (line 2), the agent then optimizes the values of the rest of the variables ($\mathbf{V} - \mathbf{Z}$) based on the reference point $\mathbf{z}$. This optimized outcome will be added to *outSet* as one of the candidate outcomes (line 3). The current recursive call ends and returns *outSet* to the previous call (line 27). Otherwise, each involved agent $A_i \in \mathbf{S}$ elicits their preferences to a set of remaining variables $Var_{\mathbf{z}}^{A_i} = \{X \in (\mathbf{V} - \mathbf{Z}) \mid Pa_i(X) \subseteq \mathbf{Z}\}$ (line 5–8). Given $\mathbf{z}$, agent A_i will elicit the preference to a remaining variable $X \in (\mathbf{V} - \mathbf{Z})$ iff $Pa_i(X) \subseteq \mathbf{Z}$, i.e. either A_i has unconditional preference over X , or the parent context of X is given by $\mathbf{z}$. Note that different agents may elicit their preferences to different sets of variables, because different agents may have different preferential-independence structures. Let $Var_{\mathbf{z}}^{\mathbf{S}}$ denotes the union of each $Var_{\mathbf{z}}^{A_i}$ ($A_i \in \mathbf{S}$) (line 9).

Step 2: We choose a *fair smallest* set of variables $\mathbf{Y} \subseteq Var_{\mathbf{z}}^{\mathbf{S}}$ to be considered in the current call (line 10). The chosen set $\mathbf{Y}$ first has to satisfy the following condition: $\mathbf{Y}$ is the *smallest* set of variables that *involves all the agents in the current call*. That means, for each agent $A_i \in \mathbf{S}$, at least one of the variables it elicits should be included in $\mathbf{Y}$ ($\mathbf{Y} \cap Var_{\mathbf{z}}^{A_i} \neq \emptyset$). When there are several eligible candidate sets, we give *higher* priority to the *consistent* preferences. Here we divide the variables which the agents elicit their preferences to into two categories:

Definition 1 (Consistent variables) *In the current call to CDCP.P1, given a reference point $\mathbf{z}$, if the preferred values to the variable $X \in Var_{\mathbf{z}}^{\mathbf{S}}$ are consistent among the agents who elicit the preference to X , then X is called a consistent variable: $\forall A_i \in \mathbf{S}$, if $X \in Var_{\mathbf{z}}^{A_i}$, then $x >_{A_i|\mathbf{z}} \bar{x}$.*³

Definition 2 (Conflicting variables) *In the current call to CDCP.P1, given a reference point $\mathbf{z}$, if there are more than one*

³ x and $\bar{x}$ are the notations of the binary domain of X , $x = \bar{\bar{x}}$. For instance, when $X = X_1$, if $x = x_1$, then $\bar{x}$ denotes $\bar{x}_1$; or if $x = \bar{x}_1$, then $\bar{x}$ denotes x_1 .

agent elicit their preferences to the variable $X \in Var_z^S$, and the preferred values of the agents to X are conflicting: $\exists A_i, A_j \in S$, $X \in Var_z^{A_i} \cap Var_z^{A_j}$ and $x \succ_{A_i}^{X|z} \bar{x}$ but $\bar{x} \succ_{A_j}^{X|z} x$. Then X is called a conflicting variable.

For instance, assume that there are three agents A_1, A_2 and A_3 involved in the current call. Agent A_1 elicits the preferred values $X_1 = x_1, X_3 = x_3$; Agent A_2 elicits $X_1 = x_1$ and $X_2 = x_2$; Agent A_3 elicits $X_2 = x_2$ and $X_3 = \bar{x}_3$; Thus $Var_z^S = \{X_1, X_2, X_3\}$, in which X_1 and X_2 are consistent variables; and X_3 is conflicting variables. There are three smallest sets that involved all agents: $\{X_1, X_2\}, \{X_1, X_3\}, \{X_2, X_3\}$. In this case, $Y = \{X_1, X_2\}$ will be chosen in the current call, because the preferred values of the agents to the variables in $\{X_1, X_2\}$ are consistent.

Step 3: Let q denotes the number of conflicting variables in Y . Then the agents' preferred values to the variables in Y forms a set of 2^q combinations of variable values to Y , denoted by $Com_z^S(Y)$. If Y is the complement of Z (line 11), each combination $y \in Com_z^S(Y)$ forms a complete assignment $zy \in D(V)$ with the reference point z . All these complete assignments will be added to $outSet$ (line 12–14). The current recursive call ends and returns $outSet$ to the previous call (line 27).

Otherwise, the reference point z will be extended with each $y \in Com_z^S(Y)$ and forms the corresponding groups of agents respectively (line 16–24). Each group $group$ with zy as the reference point $group.z$ (line 17) has the following involved participants as $group.S$ (line 18–22): *i*) the agents who only elicit preference to consistent variables in Y ; *ii*) the agents who prefer x to a conflicting variable $X \in Y$, but $group.z[X] = \bar{x}$. As such, the agents are able to trade-off between variables: *the agent with conflicting preference to a variable X participates in the group with a less preferred value to X , while continuing to optimize the values of the rest of the variables.*

For instance, suppose that there are four agents involved in the current call to CDCP.P1. The agents' CP-nets are depicted in Fig. 1. Now assume that a reference point $z = x_1$ is given. According to the agents' CP-nets, then agent A_1 elicits $X_2 = x_2$; agent A_2 elicits $X_3 = x_3$; agent A_3 elicits $X_4 = x_4$; agent A_4 elicits $X_3 = \bar{x}_3$ and $X_4 = \bar{x}_4$. Note that agent A_4 is able to elicit the preference to X_3 in the current call because the parent context of X_3 is already given by z (See Fig. 1(d), $Pa_4(X_3) = \{X_1\}$ and $z[X_1] = x_1$). There is only one eligible set of variables $Y = \{X_2, X_3, X_4\}$, in which X_2 is a consistent variable while X_3 and X_4 are conflicting variables. Thus $Com_z^S(Y) = \{x_2x_3x_4, x_2x_3\bar{x}_4, x_2\bar{x}_3x_4, x_2\bar{x}_3\bar{x}_4\}$. As $Y \cup Z \neq V$ (the value of variable X_5 hasn't been determined), we extend z with each $y \in Com_z^S(Y)$ and forms the corresponding four groups of agents. For instance, the agent group with $group.z = x_1x_2x_3x_4$ has A_1 (A_1 only elicits the preference to the consistent variables in Y) and A_4 (A_4 elicits the preferred value $\bar{x}_3$ to X_3 but $group.z[X_3] = x_3$) as its participants. In essence, both of the conflicting preferences to X_3 and X_4 result in A_4 participating in this group. Similarly, the group with reference point $group.z = x_1x_2x_3\bar{x}_4$ has a set of involved agents $group.S = \{A_1, A_3, A_4\}$; the group with $group.z = x_1x_2\bar{x}_3x_4$ has a set of involved agents $group.S = \{A_1, A_2, A_4\}$; and the group with $group.z = x_1x_2\bar{x}_3\bar{x}_4$ has a set of involved agents $group.S = \{A_1, A_2, A_3\}$. Each group then performs another call to CDCP.P1 (line 23).

Second phase of CDCP: At the end of the recursive CDCP.P1 procedure, we obtain a set of candidate outcomes, which is the basis for

Procedure 1: CDCP.P1(z, S)

Input: z , a reference point; S , a set of involved agents

Output: $outSet$, a fair set of Pareto-optimal outcomes

```

1   $outSet \leftarrow \emptyset$ 
2  if  $\|S\| = 1$  then
3    add S.OPTIMIZE( $z$ ) to  $outSet$ ;
4  else
5    forall  $A_i \in S$  do
6      elicits preferred values to a set of variables  $Var_z^{A_i}$ , where
         $Var_z^{A_i} = \{X \in (V - Z) \mid Pa_i(X) \subseteq Z\}$ ;
7    end
8     $Var_z^S \leftarrow \bigcup_{A_i \in S} Var_z^{A_i}$ ;
9    Choose a fair smallest subset  $Y \subseteq Var_z^S$ ;
10   if  $Y \cup Z = V$  then
11     forall  $y \in Com_z^S(Y)$  do
12       add  $zy$  to  $outSet$ ;
13     end
14   else
15     forall  $y \in Com_z^S(Y)$  do
16        $group.z \leftarrow zy$ ;  $group.S \leftarrow \emptyset$ ;
17       forall  $A_i \in S$  do
18         if  $(Y \cap Var_z^{A_i} \subseteq consistentVar)$  or
            $(\exists X \in Y \cap Var_z^{A_i}, x \succ_{A_i}^{X|z} \bar{x} \text{ and } group.z[X] = \bar{x})$ 
19         then
20           add  $A_i$  to  $group.S$ ;
21         end
22       end
23       add CDCP.P1( $group.z, group.S$ ) to  $outSet$ ;
24     end
25   end
26 return  $outSet$ ;

```

choosing the final outcome preferred by multiple agents. We then ask each agent to rank each outcome (called *Ranking query* in the rest of the paper) in this set [15]:

Definition 3 (Rank) Given a CP-net N_i of an individual agent A_i , the rank of an outcome is zero if the outcome is optimal for A_i , otherwise it is the length of the longest sequence of improving flips between that outcome and the optimal outcome in the corresponding induced preference graph of N_i .

Based on the ranks of the agents, we choose a final outcome from this set which minimizes the maximum rank of the agents. Note that searching for an accurate Minimax solution among all possible outcomes requires each agent to rank each possible outcome, compute the global rank of each outcome by choosing the maximum rank among the agents, and check whether the rank of the given outcome satisfies the Minimax criterion. Consequently, when the number of all possible outcomes is large, it is practically impossible to compute an accurate Minimax solution. Instead of searching for an accurate Minimax solution, our proposed approach heuristically reduces the size of the search space, i.e. the set of outcomes generated by CDCP.P1 can be much smaller than the entire outcome space, and thus is computationally efficient. Moreover, as we will show in the experiment section, the features of the final outcomes chosen by the proposed approach are close to the accurate Minimax solution.

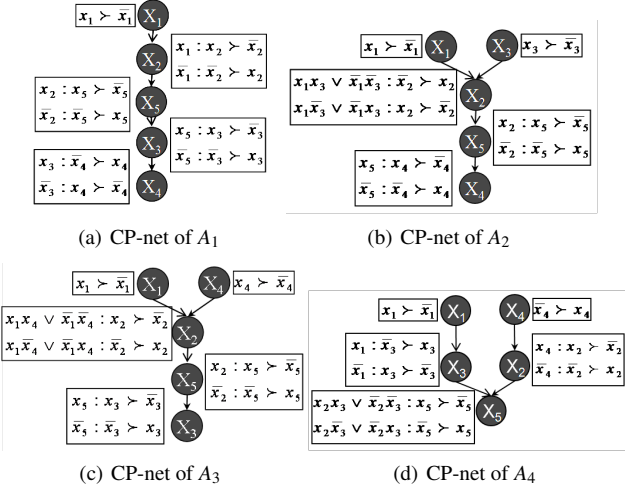


Figure 1. CP-nets of the five agents

An example

We now demonstrate the execution of the proposed approach with an example. Consider the problem of four agents making decision over a set of five binary variables $\mathbf{V} = \{X_1, X_2, X_3, X_4, X_5\}$. The CPTs and relation graphs of each agent are depicted in Fig. 1.

The initial call to CDCP.P1 with an empty reference point ($\mathbf{z} = \langle \rangle$) involves all the agents (numbered as C0; See Fig. 2). The preference elicitations of the agents in the current call are depicted in the corresponding table labelled as C0 in Fig 2: agent A_1 elicits $X_1 = x_1$; agent A_2 elicits $X_1 = x_1$ and $X_3 = x_3$; agent A_3 elicits $X_1 = x_1$ and $X_4 = x_4$; agent A_4 elicits $X_1 = x_1$ and $X_4 = \bar{x}_4$. We then choose a smallest set $\mathbf{Y} = \{X_1\}$ that involved all the agents in the current call. Because X_1 is a consistent variable, there is only one group with $group.\mathbf{z} = x_1$ and $group.\mathbf{S} = \{A_1, A_2, A_3, A_4\}$ generated and it performs the next call (C1) to CDCP.P1.

In C1, with the reference point $\mathbf{z} = x_1$, the agents' preferences elicitation are depicted in the corresponding table labelled as C1 in Fig. 2. In this call there is only one eligible set of variables $\mathbf{Y} = \{X_2, X_3, X_4\}$ and $Com_2^S(\mathbf{Y}) = \{x_2x_3x_4, x_2x_3\bar{x}_4, x_2\bar{x}_3x_4, x_2\bar{x}_3\bar{x}_4\}$. As $\mathbf{Y} \cup \mathbf{Z} \neq \mathbf{V}$, $\mathbf{z} = x_1$ will be extended with each $\mathbf{y} \in Com_2^S(\mathbf{Y})$ and forms the following groups of agents.

Group1: $group.\mathbf{z} = x_1x_2x_3x_4$, $group.\mathbf{S} = \{A_1, A_4\}$;
 Group2: $group.\mathbf{z} = x_1x_2x_3\bar{x}_4$, $group.\mathbf{S} = \{A_1, A_3, A_4\}$;
 Group3: $group.\mathbf{z} = x_1x_2\bar{x}_3x_4$, $group.\mathbf{S} = \{A_1, A_2, A_4\}$;
 Group4: $group.\mathbf{z} = x_1x_2\bar{x}_3\bar{x}_4$, $group.\mathbf{S} = \{A_1, A_2, A_3\}$.
 Each group performs another call to CDCP.P1.

Group1 performs C2 to CDCP.P1 with $\mathbf{z} = x_1x_2x_3x_4$ and two involved agents A_1, A_4 . Both A_1 and A_4 elicit $X_5 = x_5$, and thus $\mathbf{Y} = \{X_5\}$. As $\mathbf{Y}$ complement $\mathbf{Z}$, C2 ends and returns $outSet$ with only one outcome $x_1x_2x_3x_4x_5$ to C1. Similarly, Group2 performs C3 to CDCP.P1 and returns $\{x_1x_2x_3\bar{x}_4x_5\}$ to C1; Group 3 performs C4 and returns a set of two outcomes $\{x_1x_2\bar{x}_3x_4x_5, x_1x_2\bar{x}_3x_4\bar{x}_5\}$ to C1; and Group 4 performs C5 and returns $\{x_1x_2\bar{x}_3\bar{x}_4x_5\}$ to C1.

Now we finished C2-C5 and back to C1. C1 returns a set five of outcomes (i.e. the results from C2-C5) to C0: $x_1x_2x_3x_4x_5$, $x_1x_2x_3\bar{x}_4x_5$, $x_1x_2\bar{x}_3x_4x_5$, $x_1x_2\bar{x}_3x_4\bar{x}_5$ and $x_1x_2\bar{x}_3\bar{x}_4x_5$. C0 adds this set of outcomes to the $outSet$ of C0, CDCP.P1 then ends and returns a set of 5 candidate outcomes.

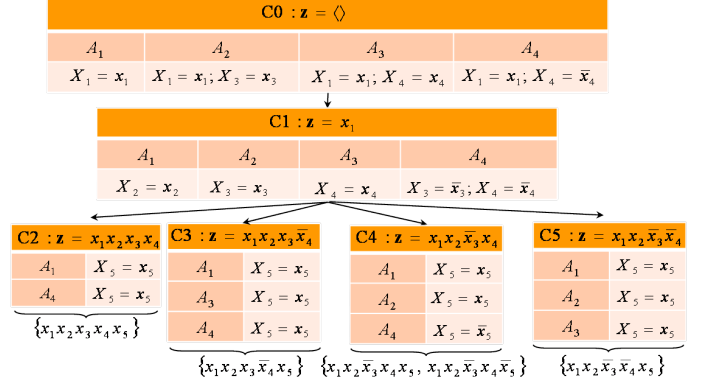


Figure 2. An illustration of the CDCP procedure

Table 1. Ranks of the candidate outcomes

Outcome	A_1	A_2	A_3	A_4	Max
$x_1x_2x_3x_4x_5$	1	3	0	5	5
$x_1x_2x_3\bar{x}_4x_5$	0	4	7	4	7
$x_1x_2\bar{x}_3x_4x_5$	2	4	1	4	4
$x_1x_2\bar{x}_3x_4\bar{x}_5$	3	7	2	3	7
$x_1x_2\bar{x}_3\bar{x}_4x_5$	3	5	8	3	8

Each agent then ranks each outcome returned by CDCP.P1 (See Table 1). The outcome $x_1x_2\bar{x}_3x_4x_5$ will be finally chosen, as the maximum rank of the agents is minimized. In this example, each agent only need to answer 5 ranking queries, which is less than one sixth of the total 32 possible outcomes.

4 Formal Properties of CDCP

The objective of an efficient collective decision-making procedure is to lead rational agents to Pareto-optimal outcomes [14]. Rossi et. al [15] provide the definitions of *Pareto dominance* and *Pareto optimality* in the context of CP-nets. Using the assumption that indifference is not allowed, given a pair of outcomes o_1 and o_2 , o_1 Pareto-dominates o_2 (written $o_1 \succ_p o_2$) iff all the agents prefer o_1 to o_2 . An outcome is Pareto-optimal iff no other outcome Pareto-dominates that outcome. Formally, an outcome o is Pareto-optimal iff $\nexists o' \in O$ and $o' \neq o$, s.t. $\forall A_i \in \mathbf{A}$, $N_i \models o' \succ o$. As to proof the Pareto-optimality of the proposed approach, we first extend the following corollary for ordering queries⁴ from [8].

Corollary 1 Let N_i be an acyclic CP-net of agent A_i , and o, o' be a pair of outcomes. If there exists a variable X in N_i , such that

- X is a root variable in N_i , o assigns a more preferred value to X than that assigned by o' : $o[X] \succ_{A_i}^{X|Pa_i(X)} o'[X]$;
- Or, o and o' assign the same values to all ancestors of X in N_i , and given the assignment provided by o (and o') to $Pa_i(X)$, o assigns a more preferred value to X than that assigned by o' : $o[X] \succ_{A_i}^{X|Pa_i(X)} o'[X]$

then $N_i \not\models o' \succ o$

⁴ An ordering query, given a CP-net N_i of an agent A_i and a pair of outcomes o and o' , asks whether $N_i \models o' \succ o$.

Now, we prove that the proposed CDCP procedure guarantees Pareto-optimality.

Theorem 1 *The outcomes generated by CDCP.P1 procedure are Pareto-optimal.*

Proof: From the initial call to the end of the procedure, CDCP.P1 constructively extends the reference point $\mathbf{z}$ with each $\mathbf{y} \in \text{Com}_{\mathbf{z}}^S(\mathbf{Y})$ respectively. In each call to CDCP.P1, each $\mathbf{y} \in \text{Com}_{\mathbf{z}}^S(\mathbf{Y})$ satisfies the following condition. $\forall \mathbf{y}' \in D(\mathbf{Y})$ and $\mathbf{y}' \neq \mathbf{y}$, $\exists A_i \in \mathbf{S}$, s.t. $(\exists X \in \mathbf{Y}, Pa_i(X) = \emptyset \text{ and } \mathbf{y}[X] >_{A_i}^X \mathbf{y}'[X]; \text{ or } Pa_i(X) \subseteq \mathbf{Z} \text{ and } \mathbf{y}[X] >_{A_i}^{X \setminus [Pa_i(X)]} \mathbf{y}'[X])$. Let $\mathbf{U} = \mathbf{Z} \cup \mathbf{Y}$. Consequently, each extension of the reference point $\mathbf{u} = \mathbf{z}\mathbf{y}$ generated by CDCP.P1 satisfies the following condition. $\forall \mathbf{u}' \in D(\mathbf{U})$ and $\mathbf{u}' \neq \mathbf{u}$, $\exists A_i \in \mathbf{A}$, s.t. $(\exists X \in \mathbf{U}, Pa_i(X) = \emptyset \text{ and } \mathbf{u}[X] >_{A_i}^X \mathbf{u}'[X]; \text{ or } Pa_i(X) \subseteq \mathbf{U}, \mathbf{u}[X] >_{A_i}^{X \setminus [Pa_i(X)]} \mathbf{u}'[X])$. When the extension of the reference point $\mathbf{u}$ forms a complete assignment (corresponds to an outcome $\mathbf{u} \in O$), according to Corollary 1, $\forall \mathbf{u}' \in O$ and $\mathbf{u}' \neq \mathbf{u}$, $\exists A_i \in \mathbf{A}$, s.t. $N_i \not\models \mathbf{u}' > \mathbf{u}$. Thus, for each outcome generated by CDCP.P1, there is no other outcome Pareto-dominates that outcome. The outcomes generated by CDCP.P1 are all Pareto-optimal. $\square$ Obviously, the final outcome chosen from the set of outcomes returned by CDCP.P1 is Pareto-optimal.

Below, we analyse the total number of outcomes generated by CDCP.P1 in the worst case scenario.

Theorem 2 *In each call to CDCP.P1 with reference point $\mathbf{z}$ and a set of involved agents $\mathbf{S}$, the number of different combinations of the agents' preferred values to $\mathbf{Y}$ is bounded with $\text{Min}[2^{s-1}, 2^l]$, where s denotes the number of agents in $\mathbf{S}$, and $l = \|\mathbf{V}\| - \|\mathbf{Z}\|$ denotes the number of remaining variables that haven't been assigned by $\mathbf{z}$.*

Proof: In each call to CDCP.P1, the number of different combinations of the agents' preferred values to $\mathbf{Y}$ is determined by the number of conflicting variables in $\mathbf{Y}$. As $\mathbf{Y}$ is the smallest set that involves all the agents, the largest number of conflicting variables in $\mathbf{Y}$ is $\text{Min}[s-1, l]$. Consequently, the largest number of agent groups or combinations of the agents' preferred values to $\mathbf{Y}$ is $\text{Min}[2^{s-1}, 2^l]$. $\square$

In essence, in each call to CDCP.P1 when $l \geq s-1$, the only case to generate 2^{s-1} combinations of the agents' preferred values to $\mathbf{Y}$ is as follows. We divide the set of agents $\mathbf{S}$ into two disjoint set $S = S_1 + S_2$, S_1 contains $s-1$ agents eliciting preferences to the disjoint set of variables: $\forall A_i, A_j \in S_1$ and $A_i \neq A_j$, $\text{Var}_{\mathbf{z}}^{A_i} \cap \text{Var}_{\mathbf{z}}^{A_j} = \emptyset$; S_2 contains only one agent and it elicits the conflicting preferences to at least one variable in each $\text{Var}_{\mathbf{z}}^{A_i}$, $A_i \in S_1$. For instance, assume that there are 4 agents in the current call to CDCP, agent A_1, A_2 and A_3 elicit the preferences $X_1 = x_1, X_2 = x_2$ and $X_3 = x_3$, respectively; agent A_4 submit the conflicting preferences $X_1 = \bar{x}_1, X_2 = \bar{x}_2$ and $X_3 = \bar{x}_3$. Then $\mathbf{Y} = \{X_1, X_2, X_3\}$ and there will be totally $2^{4-1} = 8$ combinations of the agents' preferred values to $\mathbf{Y}$. Furthermore, if $l > s-1$, within the corresponding 2^{s-1} groups of agents, there are C_0^{s-1} group with 1 agent, C_1^{s-1} groups (if $s-1 > 1$) with 2 agents, ..., C_{s-2}^{s-1} groups with $s-1$ agents and C_{s-1}^{s-1} group with $s-1$ agents. Thus, to calculate the worst case scenario of the total number of outcomes generated by CDCP.P1 is then recursive. For instance, considering the problem of 4 agents making decision over a set of 10 binary variables. In the worst case, the initial call generates C_0^3 group with 1 agent, C_1^3 groups with 2 agents, C_2^3 groups with 3 agents and C_3^3 group with 3 agents. Each group performs another call to CDCP.P1. Obviously, the group with 1 agent can only generate 1 outcome and the group with two agents can at most generate 2 different outcomes. The group with 3 agents generates C_0^2 group with 1 agent, C_1^2 groups

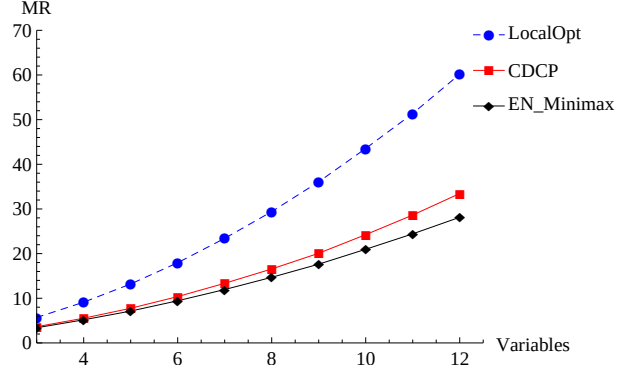


Figure 3. Average maximum rank of the agents

with 2 agents and C_2^2 group with 2 agents. Thus, the total number of outcomes generated by CDCP.P1 in the worst case is $C_0^3 * 1 + C_1^3 * 2 + C_2^3 * (C_0^2 * 1 + C_1^2 * 2 + C_2^2 * 2) + C_3^3 * (C_0^2 * 1 + C_1^2 * 2 + C_2^2 * 2) = 35$, which is less than $1/28$ of the outcome space $2^{10} = 1024$. Note that in this example, in each call to CDCP.P1, $l > s-1$.

5 EXPERIMENTS

In this section, we present the experimental results regarding the fairness and the computational efficiency of our proposed CDCP approach. We have tested CDCP on a large number of scenarios varying the agents' preferences and the number of variables. In those experiments, the variables are binary-valued, the number of agents is 5 and we vary the number of variables m from 3 to 12. For each number of variables, we generate 5,000 random examples of the agents' CP-nets and execute the CDCP procedure. We compare the performance of the proposed CDCP procedure with the enumeration method (EN_Minimax) and another simple method called LocalOpt. The former enumerates the entire outcome space to search for the accurate Minimax solution. The latter apply Minimax criteria only on the set of local optimal outcomes (one for each agent) and chooses a final outcome from this set. In those experiments, we consider the following measurements.

Fairness

Maximum rank of the agents (MR): the average maximum rank of the agents.

Computational efficiency

Excution time : the average execution time.

Number of ranking queries : the average number of ranking queries by each individual agent.

Regarding fairness, Fig. 3 shows the average maximum rank of the agents (MR) of EN_Minimax, LocalOpt and CDCP approaches in those experiments. The average MR of CDCP approach is much smaller than that of LocalOpt approach, and is very close to the accurate Minimax solution computed by EN_Minimax. While we consider in this paper, the length of the longest path to the optimal outcome, as the rank of an outcome, similar results have been obtained when considering the length of shortest path as the rank of the outcome.

Regarding computational efficiency, Fig. 4 shows the average execution time of LocalOpt, EN_Minimax and the proposed CDCP approaches in those experiments. It demonstrates that the average ex-

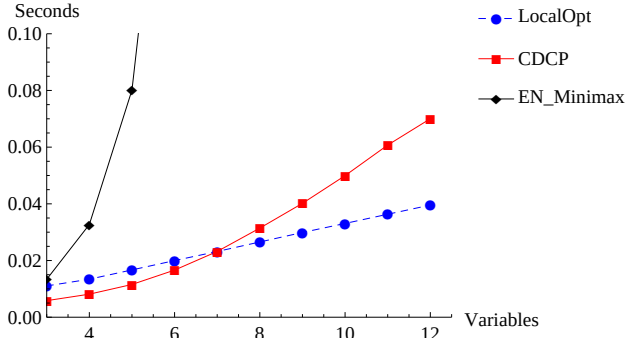


Figure 4. Average execution time. The average execution time of EN_Minimax approach goes further exponentially. When the number of variable is larger than eight, the data of EN_Minimax approach is not printed in this graph.

ecution time of the enumeration method EN_Minimax increases exponentially as the number of variables increases. However, the proposed CDCP approach greatly reduces the execution time, while producing the outcomes that are close to the accurate Minimax solution computed by EN_Minimax method. According to the experimental data, when the number of variables is large (e.g. 12 variables), the execution time is reduced by more than three orders of magnitude compared to that of EN_Minimax method. In addition, we extend the previous experiments with 5-40 agents and 5-40 variables. Table 2 shows the significant reduction in average ranking queries by each individual agent. Especially when the number of variables is large, the number of ranking queries is reduced by many orders of magnitude.

Table 2. Average number of ranking queries

Variable Number	Outcome Space	Agent Number			
		5	10	20	40
5	8	3.0	10.0	11.6	13.2
10	1024	7.1	37.1	92.3	122.2
20	1048576	18.7	106.3	1109.9	3036.0
40	1099511627776	29.3	325.3	5021.0	385063.0

6 CONCLUSION AND FUTURE WORK

In this paper, we have applied the practical preference representation model CP-net to represent agents' preferences in collective decision-making and proposed a novel efficient procedure for collective decision-making with CP-nets. There are not many works for aggregating multiple agents' preferences which are represented by CP-nets. However, the existing works either do not address computational issues, or depend on a strong assumption that all the agents share a common preferential-independence structure. Our proposed procedure enables agents with different preferential-independence structures to trade off between variables. It is computationally efficient, guarantees Pareto-optimality and produces fair outcomes.

In this work, we have made simplifying assumptions about the nature of the qualitative preferences such as acyclic CP-nets, binary-valued and having no indifference. In addition, we have not explored more powerful variants such as TCP-nets for representing agents'

preferences in collective decision-making, although they can be similarly applied to support more expressive preferential semantics on preference trade-offs such as relative importance and conditional relative importance. We plan to explore in more detail how best CP-nets can improve the representation of agents' preferences in collective decision-making and produce better outcomes for multiple agents.

REFERENCES

- [1] Fahiem Bacchus and Adam J. Grove, 'Graphical models for preference and utility', in *UAI*, pp. 3–10, (1995).
- [2] Ronen I. Brafman and Carmel Domshlak, 'Introducing variable importance tradeoffs into CP-nets', in *In Proceedings of UAI-02*, pp. 69–76. Morgan Kaufmann.
- [3] Ronen I. Brafman, Carmel Domshlak, and Tanya Kogan, 'Compact value-function representations for qualitative preferences', in *UAI*, pp. 51–59, (2004).
- [4] Steven J. Brams and Alan D. Taylor, *Fair division : from cake-cutting to dispute resolution / Steven J. Brams and Alan D. Taylor*, Cambridge University Press, Cambridge ; New York :, 1996.
- [5] Darius Braziunas and Craig Boutilier, 'Elicitation of factored utilities', *AI Magazine*, **29**(4), (2008).
- [6] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet, 'Preference handling in combinatorial domains: From ai to social choice', *AI Magazine, Special Issue on Preferences*, **29**(4), 37–46, (2008).
- [7] Judy Goldsmith, Jérôme Lang, Mirosław Trzuszczynski, and Nic Wilson, 'The computational complexity of dominance and consistency in CP-nets', *J. Artif. Int. Res.*, **33**(1), 403–432, (2008).
- [8] C. Boutilier R. I. Brafman C. Domshlak H. H. Hoos and D. Poole, 'Cp-nets: A tool for representing and reasoning with conditional *Ceteris Paribus* preference statements', *Journal of Artificial Intelligence Research (JAIR)*, **21**(1), 135–191, (2004).
- [9] Jérôme Lang and Lirong Xia, 'Sequential composition of voting rules in multi-issue domains', *Mathematical Social Sciences*, **57**(3), 304–324, (May 2009).
- [10] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk, 'An efficient majority-rule-based approach for collective decision making with cp-nets', in *Proceedings of the 12th International Conference on the Principles of Knowledge Representation and Reasoning (KR-2010)*, (2010 (to appear)).
- [11] Prasanta K. Pattanaik, *Voting and collective choice; some aspects of the theory of group decision-making [by] Prasanta K. Pattanaik*, University Press, Cambridge [Eng.], 1971.
- [12] M. Pini, F. Rossi, B. Venable, and Toby Walsh, 'Incompleteness and incomparability in preference aggregation', in *Proceedings of 20th IJCAI*, ed., Manuela M. Veloso, pp. 1464–1469, Hyderabad, India, (January 2007). International Joint Conference on Artificial Intelligence.
- [13] Keith Purrington and Edmund H. Durfee, 'Making social choices from individuals' cp-nets', in *AAMAS '07: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pp. 1–3, New York, NY, USA, (2007). ACM.
- [14] Howard Raiffa, *The Art and Science of Negotiation*, Harvard University Press, Cambridge, USA, 1982.
- [15] F. Rossi, K. B. Venable, and T. Walsh, 'mCP nets: representing and reasoning with preferences of multiple agents', in *AAAI'04: Proceedings of the 19th national conference on Artificial intelligence*, pp. 729–734. AAAI Press / The MIT Press, (2004).
- [16] Lirong Xia, Vincent Conitzer, and Jérôme Lang, 'Voting on multiattribute domains with cyclic preferential dependencies', in *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*, pp. 202–207. AAAI Press, (2008).
- [17] Lirong Xia, Jérôme Lang, and Mingsheng Ying, 'Strongly decomposable voting rules on multiattribute domains', in *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pp. 776–781. AAAI Press, (2007).