Collective Decision Making Using Conditional Preference Networks Without Considering All Users' Preferences Over All Attributes

by

Mina Joroughi

Bachelor of Information Technology Engineering, University of Tabriz , 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Computer Science

In the Graduate Academic Unit of Computer Science

Supervisor(s):	Michael Fleming, PhD, Computer Science
Examining Board:	Huajie Zhang, PhD, Computer Science, Chair
	Scott Buffett, PhD, Computer Science
	Donglei Du, PhD, Business Administration

This thesis is accepted

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

May, 2017

Abstract

Due to the great increase in e-marketing and high competition to attract more customers and keep them satisfied, collective decision making, the process of making recommendations to a group of people, has become an active research area.

CP-nets (Conditional Preference Networks) are a widely used tool to represent users' preferences, but the problem is that in real world situations, we have a large number of users conveying their preferences over a large number of attributes; therefore, comparing the exponential number of outcomes for all users in a collective decision making process is infeasible.

In this research, we have looked at reducing the number of outcomes to be considered in the process of collective decision making by examining the question of whether we need to consider all users' preferences over all attributes. We propose a novel procedure for collective decision making by clustering users and considering users' preferences only over the most important attributes for that cluster. The use of attribute-weighting techniques and clustering methods allows for searching in a much smaller subspace of attributes

and as a consequence requires a smaller number of comparisons between the outcomes, which makes our method more practical for real world problems. In our approach to make users as satisfied as possible, our methods produce two different kinds of outcomes: a global recommended outcome and cluster-specific outcomes, which can be offered in different situations. The results of our experiments demonstrate that the methods can produce high-quality recommendations despite the fact that users' preferences over many attributes are ignored.

Dedication

To my parents and to my love, Masoud ...

Acknowledgments

I would like to thank my advisor, Dr. Michael Fleming for guiding and supporting me over the years. You have set an example of excellence as a researcher, mentor, instructor, and role model. I could not have imagined having a better advisor and mentor for my master's study.

I would like to thank my thesis committee members, Dr. Huajie Zhang, Dr. Scott Buffett and Dr. Donglei Du for all of their guidance through this process; your discussion, ideas, and feedback have been absolutely invaluable.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Overview

In this chapter we will provide an overview of collective decision making and the motivation for our approach to this challenging problem.

1.1 The Importance of Collective Decision Making

Due to the widespread use of the internet, some advancements in electronic commerce and the large number of online customers, collective decision making has become an active research area. Collective decision making involves considering the preferences of a group of users and recommending an outcome that is favourable for the group as a whole[2].

Collective decision making plays an increasingly important role in this competitive world. In e-marketing, one of the best policies to attract more customers is providing a product based on each user's preferences, but the problem is that sometimes some industries cannot customize their product based on each user's preferences. As an example, if a car company wants to attract more customers in the competitive automotive market, product customization based on each customer's preferences would not be a reasonable idea. Therefore, these kinds of industries are looking for a method to offer one or more products that will attract more customers. This approach can be done by collective decision making, which helps to get the preferences of each customer and then offer one or more products that are preferable for a group of them.

1.2 The Need for Conditional Preference Networks (CP-nets)

Quantitative assessments of classical methods have made utility functions more complicated to apply in real world collective decision making problems; therefore, in diverse areas like recommender systems, collective decision making, and preference elicitation, we are faced with the problem of representing users' preferences[17]. Consider the following statements.

- Users typically feel more comfortable conveying their preferences in a qualitative way instead of a quantitative one.
- Some preferences are conditional, and representing these conditional

preferences in a quantitative way is hard.

- Users prefer to express their preferences over individual attributes rather than whole outcomes, which consist of values assigned to every attribute.

For these reasons, there is a critical need for a kind of graphical model that has good readability and is close to the way in which users state their preferences in natural language. That is why conditional preference networks (CP-nets) were introduced by the AI research community as a graphical model for representing qualitative and conditional preferences of users [5, 6]. CP-nets are presented in more detail in Chapter 2

1.3 The Use of CP-nets in Collective Decision Making

By retrieving preference information from each user and constructing a good model of their preferences, an effective recommender system should recommend the most favourable outcome to users in a reasonable time. Recommending a good product for users requires a good understanding of users' preferences. As discussed in Section 1.2, CP-nets can be used to represent preferences in a natural way. There is a need for a practical method to search the entire set of outcomes, aggregate the preferences of all users and find one or more outcomes that are desirable for a majority of users.

1.4 Our Approach in Collective Decision Making

In the real world, we have a large number of users expressing their preferences over a large number of attributes; comparing the preferences of these users can have complexity that is exponential in the number of attributes. That is why we are going to use an approach to make the use of CP-nets more practical in collective decision making problems.
The overall goal of this thesis research is to develop a few different techniques for making recommendations to a group of users by focusing only on the preferences over each user's most important attributes, and to evaluate these techniques against each other, to get a sense of which ones would be the most promising for future research. The measures of success used in Chapter 4 include the methods' performance against a set of random recommendations and their ability to outperform a simple baseline method.

1.4.1 Reducing the Number of Attributes

If we have a large number of attributes and we want users to give their preferences over them, comparing all the outcomes with each other is NP-hard [6]. We will propose to use a weighting method to decrease the number of attributes considered for each user and, as a consequence, make the use of CP-nets with a large number of attributes more practical.

1.4.2 Using Clustering to Decrease the Number of Comparisons Between Users

In recent collective decision making methods for deciding on recommended outcomes, there is a need to compare all users' preferences over all attributes [19, 20, 22, 23], but comparing this large number of users' preferences on a large number of attributes is infeasible in the real world. Therefore, we are going to cluster the users in smaller groups and instead of comparing the preferences of all users over all attributes, we will focus on a smaller number of preferences over the attributes that are important for that cluster. This leads to further reduction in the number of comparisons between outcomes. Clustering is performed by looking not only at the structure of users' preference networks, but also at their actual preferences over the most important attributes.

In this research, our input is a medium to large number of CP-nets with a medium to large number of attributes, and our goal is to overcome the exponential number of comparisons between outcomes for each user by reducing the number of attributes and using clustering to decrease the number of comparisons over attributes to get a final outcome that is desirable for a majority of users or for the set of users in each cluster.

1.5 Thesis Structure

In the next chapter, we will talk about background information and related work on collective decision making. Chapter 3 includes a detailed explanation of our method in collective decision making and its implementation. The results and analysis of experiments that we have done on some random CP-nets are described in Chapter 4. Finally, in Chapter 5, we will talk about the conclusions and future work.

Chapter 2

Background

In this chapter, background information on decision making involving users' preferences in combinatorial domains will be explained.

2.1 Combinatorial Domains

If $V = \{X_1, ..., X_m\}$ is a set of m combinatorial attributes and $D(X_i)$ represents the domain of each attribute, we can define an outcome by concatenating the values of each attribute. Hence, the total number of outcomes can be defined by multiplying the domain sizes of all m attributes, which is $D(X_1) \times ... \times D(X_m)$ [23]. For example, if we define three binary attributes $V = \{X_1, X_2, X_3\}$ and $D(X_1) = \{x_1, \overline{x}_1\}$, $D(X_2) = \{x_2, \overline{x}_2\}$ and $D(X_3) = \{x_3, \overline{x}_3\}$, then the assignment $x_1\overline{x}_2x_3$ assigns x_1 to attribute X_1, $\overline{x}_2$ to attribute X_2 and x_3 to attribute X_3.

If, in an outcome, all the attributes are assigned a value in their domain, we can call it a complete assignment; otherwise, this outcome is called a partial assignment. If we have two subsets of V called X and Y where $X \cap Y = \emptyset$ and $X \cup Y = V$ and if we denote the concatenation of values of X and Y by xy, we can call y a completion of assignment to x which can be represented by $comp(x)$. For example, for $V = \{A, B, C, D, E\}$, if $X = \{A, B, C\}$ and $Y = \{D, E\}$ and if we assign $x = a\bar{b}\bar{c}$ and $y = d\bar{e}$, then xy denotes a complete assignment of attributes: the outcome $a\bar{b}\bar{c}d\bar{e}$ [25].

2.2 Preferences in Combinatorial Domains

Preference indicates the order over the satisfaction of a user for different outcomes. Based on each user's preference relation we can order outcomes; if o and o' are two possible outcomes, we can define three different relations between them [24].

- $o \succeq o'$ which means that o is at least as preferable as o' to the user.
- $o \succ o'$ which means o is strictly preferred over o' by the user.
- $o \sim o'$ which means that the user is indifferent between o and o'. The relations $o \succeq o'$ and $o' \succeq o$ both hold; therefore, o and o' are incomparable.

Because the number of outcomes is exponential in the number of attributes, finding direct relations between all pairs of outcomes is infeasible in the real world.

2.3 Representation of CP-nets

Conditional preference networks (CP-nets) are a graphical method for representing users' preferences in a well structured manner [6]. CP-nets, are a kind of directed acyclic graph (DAG) $G = (V, E)$, where V is a set of nodes representing attributes $V = \{X_1, ..., X_m\}$ with discrete domains $D(X_1), ..., D(X_m)$, and E is a set of directed arcs showing the dependency between the attributes. Each one of the attributes can be defined as a parent of others; therefore, each feature X_i can have a set of parents $Pa(X_i)$ that affect the preference over the values of X_i. The preference over the values of each attribute can be shown by the use of a conditional preference table ($CPT(X_i)$) showing the preference over the values of $Dom(X_i)$ for each combination of values for its parents $Pa(X_i)$.

For example, if we have two binary variables $V = \{X_1, X_2\}$, then the directed arc $E = \{(X_1, X_2)\}$ means that the user's preference over the value of X_2 is dependent on the value of X_1. The notation $x_1 : x_2 \succ \overline{x}_2$ in a conditional preference table for this feature means that if the value of X_1 is equal to x_1, then the user prefers $X_2 = x_2$ to $X_2 = \overline{x}_2$.

2.3.1 Example of CP-nets

A typical CP-net is illustrated in Figure 2.1 where the attributes are X_1, X_2, X_3 and X_4 with binary domains ($x_1, \overline{x}_1$ are values for attribute $X_1, x_2, \overline{x}_2$ are values for attribute $X_2, x_3, \overline{x}_3$ are values for attribute $X_3, x_4, \overline{x}_4$ are values for attribute X_4). As shown in the CP-net:

- For attribute X_1

  X_1 has no parents, and $CPT(X_1)$ specifies that x_1 is unconditionally preferable to $\overline{x}_1$, $x_1 \succ \overline{x}_1$.

- For attribute X_2

  X_1 is the parent of X_2, and $CPT(X_2)$ shows a preference order on X_2's values under the condition of X_1's values, $x_1 : x_2 \succ \overline{x}_2$, $\overline{x}_1 : \overline{x}_2 \succ x_2$.

- For attribute X_3

  X_1 is the parent of X_3, and $CPT(X_3)$ shows a preference order on X_3's values under the condition of X_1's values, $x_1 : \overline{x}_3 \succ x_3$, $\overline{x}_1 : x_3 \succ \overline{x}_3$.

- For attribute X_4

  X_2, X_3 are the parents of X_4, and $CPT(X_4)$ shows a preference order on X_4's values under the condition of X_2 and X_3's values, $(\overline{x}_2 \wedge \overline{x}_3) \vee (x_2 \wedge x_3) : \overline{x}_4 \succ x_4$, $(\overline{x}_2 \wedge x_3) \vee (x_2 \wedge \overline{x}_3) : x_4 \succ \overline{x}_4$

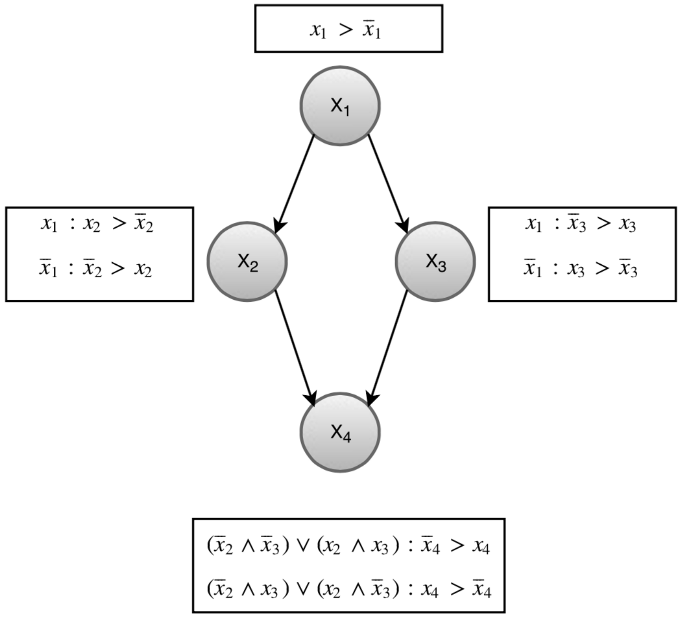

Figure 2.1: Example of a CP-net

An outcome of the CP-net is a concatenation of the domain values of each attribute. For example, $x_1\overline{x}_2\overline{x}_3x_4$ is one of the outcomes of the CP-net presented in Figure 2.1.

2.3.2 Dominance Checking

Given a CP-net, how can we use it for reasoning about outcomes? In any preference representation formalism, we are looking for a method that helps us to answer the question of whether one outcome α dominates the other β, which is called dominance testing. We can find three different relations between two outcomes α and β: $\alpha \succ \beta$ means α is strictly preferred over

β, $\beta \succ \alpha$ means β is strictly preferred over α, and $\alpha \bowtie \beta$ means α is incomparable to β. In the third situation, the CP-net structure does not have enough information to find out which one of the outcomes is preferred over the other.

2.3.2.1 Worsening Flipping Sequence

Boutilier et al. [4] present a dominance checking algorithm to find out if one outcome is preferable over the other one. This dominance checking algorithm uses steps called worsening flips, which help us to find the relation between outcomes. Worsening flips use CPTs to find a change in the value of the attribute of an outcome to the value that is less preferred. For example, in the CP-net in Figure 2.1, passing from $x_1 x_2 \overline{x}_3 x_4$ to $x_1 \overline{x}_2 \overline{x}_3 x_4$ is a worsening flip since x_2 is better than $\overline{x}_2$ given x_1 .

Definition 2.3.1. *(WORSENING FLIPPING SEQUENCE, based on the definition of an improving flipping sequence[21]) A sequence of outcomes* $\alpha = \gamma_1, \gamma_2, ..., \gamma_{m-1}, \gamma_m = \beta$ *such that*

$$\alpha = \gamma_1 \succ \gamma_2 \succ ... \succ \gamma_{m-1} \succ \gamma_m = \beta$$

is a worsening flipping sequence if and only if, $\forall 1 \leqslant i \leqslant m-1$*, outcome* γ_i *is different from the outcome* γ_{i+1} *in the value of exactly one variable* X*, and* $\gamma_i[X] \succ \gamma_{i+1}[X]$ *with respect to an acyclic CP-net.*

Here, the notation $\gamma_i[X]$ means the value assigned to variable X in outcome γ_i.

Therefore, we can establish a dominance relation between two outcomes that differ on more than one attribute, and an outcome α can dominate another outcome β if there is a sequence of worsening flips from α to β. Unfortunately, finding a worsening flipping sequence is NP-hard because, even for acyclic CP-nets, sometimes there are exponentially long chains of flips between outcomes [13].

2.3.2.2 Induced Preference Graph

An induced preference graph helps us to find the relation of dominance between outcomes; for example, if one outcome α is better than the other β, there should be a chain of worsening flips from α to β. In this graph, a direct arrow from one outcome X_i to the other X_j expresses that X_i and X_j are comparable, and X_j is preferable over X_i. If there is no directed arrow between two outcomes, this means that they are not comparable by their CPTs [6]. The induced preference graph of the CP-net in Figure 2.1 is represented by Figure 2.2, and the process for worsening flips to find out if outcome $x_1x_2\overline{x}_3x_4$ is preferred over outcome $x_1\overline{x}_2x_3\overline{x}_4$ by the use of the CPT of each attribute is as follows:

$x_1x_2\overline{x}_3x_4 \succ x_1\overline{x}_2\overline{x}_3x_4$ (since $x_1 : x_2 \succ \overline{x}_2$, x_2 flipped to $\overline{x}_2$)

$x_1\overline{x}_2\overline{x}_3x_4 \succ x_1\overline{x}_2x_3x_4$ (since $x_1 : \overline{x}_3 \succ x_3$, $\overline{x}_3$ is flipped to x_3)

$x_1\overline{x}_2x_3x_4 \succ x_1\overline{x}_2x_3\overline{x}_4$ (since $\overline{x}_2 \wedge x_3 : x_4 \succ \overline{x}_4$, x_4 is flipped to $\overline{x}_4$)

Then $x_1x_2\overline{x}_3x_4 \succ x_1\overline{x}_2\overline{x}_3x_4 \succ x_1\overline{x}_2x_3x_4 \succ x_1\overline{x}_2x_3\overline{x}_4$. Thus outcome $x_1x_2\overline{x}_3x_4$ is preferred over outcome $x_1\overline{x}_2x_3\overline{x}_4$, because there is a worsening flipping sequence from outcome $x_1x_2\overline{x}_3x_4$ to outcome $x_1\overline{x}_2x_3\overline{x}_4$. Note that it can be determined from Figure 2.2 that the outcome $x_1x_2\overline{x}_3x_4$ dominates all other outcomes.

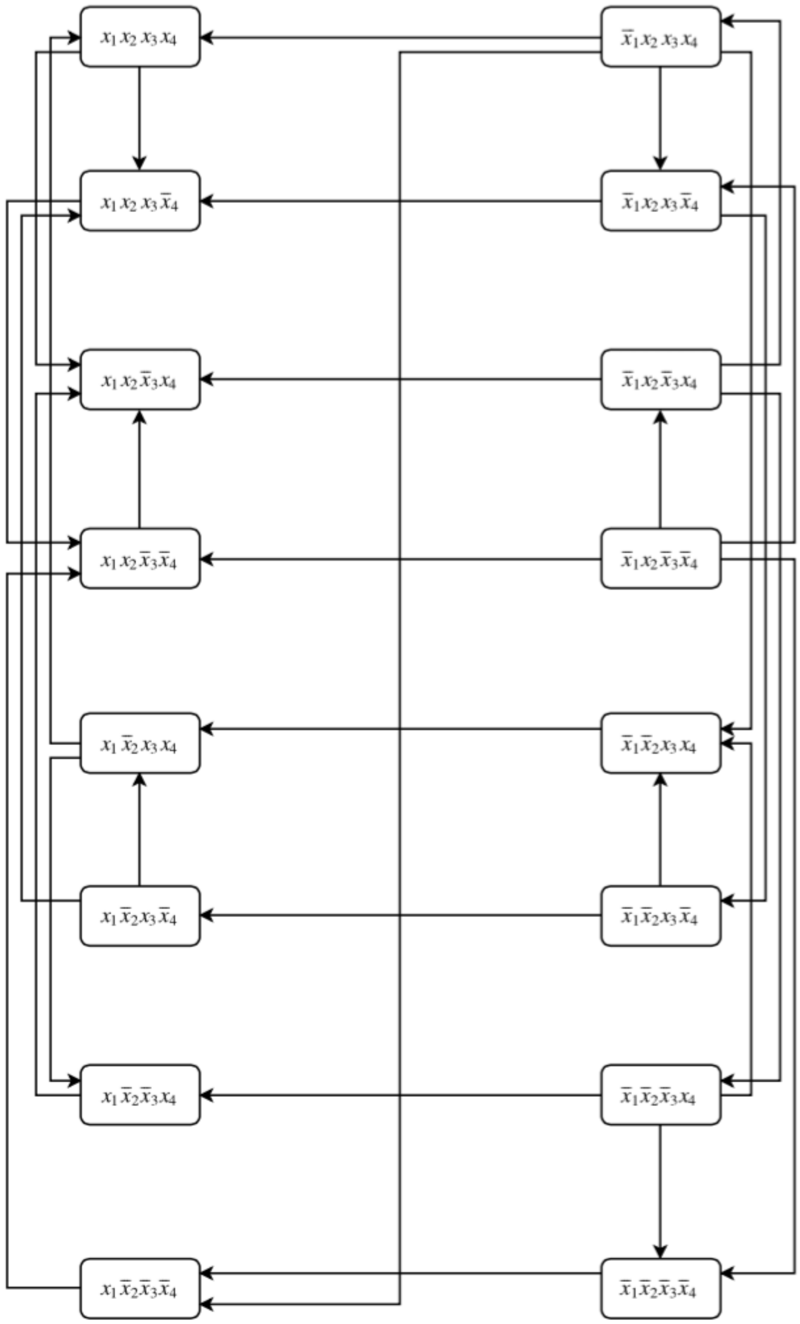

Figure 2.2: Induced Preference Graph

2.3.3 Dominance Testing in Our Approach

In this research, for dominance checking, we are going to use an approach by Santhanam et al. [29]. In this approach, they provide an encoding of CP-nets into a Kripke structure[11], which encodes the problem of dominance into a graph reachability analysis, and then use a model checker named $NuSMV$[12] for computing dominance testing. Therefore, the code that they have provided takes a CP-net structure and two outcomes as input and determines whether there is a dominance relation between the two outcomes. The output of the code is the worsening flipping sequence from the best outcome to the worst outcome, if there is any dominance between the two outcomes, and “no dominance”, if there is no worsening flipping sequence from one outcome to the other.

2.4 Weighting Method

The structure of a CP-net helps us to assign a weight to each one of the attributes such that attributes in higher levels are more important than ones in lower levels [5, 6, 7].

Definition 2.4.1. *(IMPORTANCE WEIGHT) In a given acyclic CP-net the importance weight of attribute X is defined by this formula [21]:*

$$w_X = 1 + \sum_{Y \in Ch(X)} w_Y.(|D(Y)| - 1)$$

where $Ch(X)$ is the set of children of X and $|D(Y)|$ is the domain size of attribute Y. For example, for a CP-net with binary attributes, $|D(X)| = 2$ for all attributes X. The process of weighting for the CP-net depicted in Figure 2.3, where each attribute is binary, can be calculated as follows:

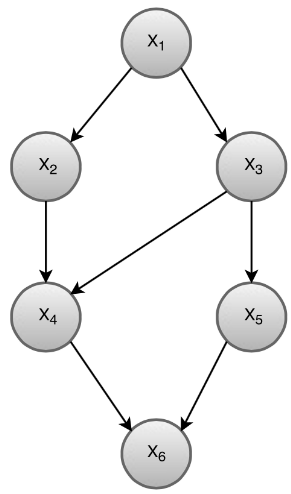

Figure 2.3: Weighting Example

$W_{X_6} = 1$

$W_{X_5} = 1 + W_{X_6}.(2-1) = 2$

$W_{X_4} = 1 + W_{X_6}.(2-1) = 2$

$W_{X_3} = 1 + (W_{X_5}.(2-1) + W_{X_4}.(2-1)) = 5$

$W_{X_2} = 1 + W_{X_4}.(2-1) = 3$

$W_{X_1} = 1 + (W_{X_2}.(2-1) + W_{X_3}.(2-1)) = 9$

2.5 Penalty Scoring Function

In this section, we are going to explain a penalty scoring function based on the research done by Li et al. [21] and Domshlak et al. [14]. If we have an attribute X with the domain size of $|D(X)|$, we can define $|D(X)|$ degrees of penalties for attribute X, which is $d_1, d_2, ..., d_{|D(X)|}$. We assign a penalty 0 for the most preferred value of attribute X, while the least preferred value of this attribute would have a penalty of $|D(X)|-1$. For instance, suppose that attribute X has three different values. The domain of this attribute would be $D(X) = \{x, x', x''\}$ and if $x \succ x' \succ x''$ are the user's preferences over this attribute, we can convey that $d_1 = 0$ can be assigned to the value of x, $d_2 = 1$ can be assigned to the value of x', and finally, $d_3 = 2$ can be assigned to the value of x''. If we have an outcome γ, the penalty associated with the attribute X in that outcome can be defined by multiplying the degree of penalty for the value of attribute X by the weight of that attribute. The penalty for the whole outcome can then be defined by this formula [14, 21].

$$\forall \gamma \in O, \quad pen(\gamma) = \sum_{x \in V} w_x.(d_x)^\gamma$$

where V is the set of variables, w_x is the importance weight of attribute X and $(d_x)^\gamma$ is the degree of penalty of X in γ . For example, consider the CP-net represented in Figure 2.1. If we have an outcome $\gamma = \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4$, the penalty score for this outcome would be calculated as follows:

For attribute X_1 as this user prefers x_1 unconditionally over $\overline{x}_1$, $(d_{x_1})^\gamma = 1$.

Given $X_1 = \overline{x}_1$, this user's preference over X_2 is $\overline{x}_2 \succ x_2$ and therefore $(d_{x_2})^\gamma = 0$. Also, this user's preference for attribute X_3, given $X_1 = \overline{x}_1$, is $x_3 \succ \overline{x}_3$, and so $(d_{x_3})^\gamma = 1$. Finally, as the value of X_4 is dependent on x_2 and x_3, and the preference of this user given $X_2 = \overline{x}_2$ and $X_3 = \overline{x}_3$ is $\overline{x}_4 \succ x_4$, the value assigned to $(d_{x_4})^\gamma = 0$ and then the penalty of outcome $\gamma = \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4$ will be:

$pen(\gamma) = w_{x_1} \cdot 1 + w_{x_2} \cdot 0 + w_{x_3} \cdot 1 + w_{x_4} \cdot 0 = 5 \times 1 + 2 \times 0 + 2 \times 1 + 1 \times 0 = 7$

2.6 Clustering

In clustering, we partition existing objects into different classes whose members are similar in some way, helping us to predict unknown structures, models and information. Based on the objects in each cluster and their similar attributes, we could decide about unknown attributes of each member in that specific cluster [32]. In many research areas, clustering helps us to improve efficiency and it can be divided into two different groups, hierarchical and partitioning clustering [27].

2.6.1 Hierarchical Methods

This method could be useful when we want to merge smaller groups into big ones or vice versa; it is similar to a tree-like clustering process, and for merging or splitting the clusters, the distance between them can be defined in three ways: the shortest distance from any member of one cluster to any member of another cluster, the longest distance from any member of one cluster to any member of another cluster, and the average distance from any member of one cluster to any member of another cluster.

2.6.2 Partitioning Clustering

This method partitions the whole set of data into small groups; the most commonly used algorithm for partitioning clustering is k-means clustering [27], which in the first step randomly defines k points and then in the second step assigns each data item to the closest point. In the third step, this clustering method recalculates each cluster by defining a new center based on the objects in the current cluster and then repeats the last two steps until the center does not change.

2.6.3 Clustering CP-nets

One of the important challenges in clustering is finding a method that helps us to measure the similarity between objects. Furthermore, the similarity measurement method should be easy to compute and easily explainable. One of the most important challenges in the research for this thesis is that in the field of clustering CP-nets, we have a limited amount of research[30, 31] and the approach behind their method of clustering is based on creating induced preference graphs. The process of constructing an induced preference graph for each user is NP-hard [6], so this method of clustering would not be feasible in our approach. In this specific situation, we are going to consider different ways of representing CP-nets as vectors. We can then use cosine similarity to measure the similarity between CP-nets, with no need to construct the induced preference graph.

2.6.4 Finding The Distance Between CP-nets by The Use of Cosine Similarity

If we think of each user's preferences as points in n-dimensional space, we can say two users are similar if they are close to each other in that n-dimensional space. The idea behind cosine similarity is based on the angle between two lines from the origin to each of these points; when it is relatively small, we can say that those two points are similar to each other. The input for finding the cosine similarity is a pair of vectors showing the weights for each attribute for each user, and the output is a number in the range $[0, 1]$. If we use $\overrightarrow{d}_A$ and $\overrightarrow{d}_B$ to represent two vectors for the weights of attributes for CP-net A and CP-net B, the cosine similarity between these two CP-nets can be found using Equation 2.1, where $w(t_i, d_A)$ is the weight of attribute t_i based on the CP-net of user A, and $|V|$ shows the number of attributes.

$$\cos(\mathbf{d_A}, \mathbf{d_B}) = \frac{\overrightarrow{d_A} \cdot \overrightarrow{d_B}}{\|\overrightarrow{d_A}\|\|\overrightarrow{d_B}\|} = \frac{\sum_{i=1}^{|V|} \mathbf{w}(\mathbf{t_i}, \mathbf{d_A})\mathbf{w}(\mathbf{t_i}, \mathbf{d_B})}{\sqrt{\sum_{i=1}^{|V|} (\mathbf{w}(\mathbf{t_i}, \mathbf{d_A}))^2}\sqrt{\sum_{i=1}^{|V|} (\mathbf{w}(\mathbf{t_i}, \mathbf{d_B}))^2}} \tag{2.1}$$

For example, if Table 2.1 shows the weights of two users for five attributes, the cosine similarity between these two users can be computed as shown below:

Table 2.1: Weights for each Attribute

User	Att 1	Att 2	Att 3	Att 4	Att 5
User 1	5	1	1	3	1
User 2	5	1	1	2	4

$$\cos(\mathbf{d_A}, \mathbf{d_B}) = \frac{\overrightarrow{d_A} \cdot \overrightarrow{d_B}}{\|\overrightarrow{d_A}\| \|\overrightarrow{d_B}\|} = \frac{5 \times 5 + 1 \times 1 + 1 \times 1 + 3 \times 2 + 1 \times 4}{\sqrt{5^2 + 1^2 + 1^2 + 3^2 + 1^2}\sqrt{5^2 + 1^2 + 1^2 + 2^2 + 4^2}} = 0.89$$

After finding the cosine similarity between each pair of users, we can give the resulting similarity matrix as an input to one of the clustering tools and get the clusters and names of the users in each cluster as an output.

2.7 Related Work

As explained in Section 1.2, CP-nets are one of the best studied approaches for representing and reasoning with users' preferences. There is a need for an efficient algorithm for dominance testing to find out if one outcome is preferred to another one. Current techniques for dominance testing search for an improving flipping sequence from one outcome to the other one, which is PSPACE-complete [16]. Santhanam et al. [29] have explored an approach to dominance testing via model checking, by encoding CP-nets into a Kripke structure [11] and checking for dominance using the widely used model checker called $NuSMV$ [12].

Based on the work by Boutilier et al. [5, 6, 7], we can claim that the rich structure of CP-nets allows us to define different levels of importance for attributes; hence, attributes in higher levels are considered to be more important than ones in lower levels. This idea helped Li et al. [21] to define a penalty scoring function based on the work by Domshlak et al. [14], which translates each qualitative outcome of CP-nets into a quantitative one and therefore helps us to compare the desirability of outcomes in a numeric way. Collective decision making with CP-nets has been studied in the literature [18, 28]. One approach to collective decision making is issue-by-issue sequential election, but because all the attributes are not preferentially independent, this method can lead to a sub-optimal choice [10]. Moreover, in earlier research on collective decision making, users' preferences over attributes should follow a linear order referred to as o-legality [26, 33]; by this assumption the users' preference order over each feature should be independent of the following attributes in some order O, which helps to decide on one issue after another and then voting rules can be applied to select the winning outcome. However, in the real world, as we have a conditional structure on the attributes, all users' preferences do not follow a linear order. Therefore, social choice and collective decision making are more complex and challenging.

In the next steps over collective decision making, there is some research in which procedures are presented for collective decision making, assuming that all agents' CP-nets should have a common preferential structure [10, 33]. Li et al. [19, 22] address the problem of having the same preference structure for all the users and outcomes by offering novel approaches for collective decision making and considering computational concerns and preferential-independence structure.

Another problem of collective decision making is the number of pairwise comparisons for dominance testing. In some papers [20, 23, 25], they addressed this problem by introducing a distributed protocol, which reduces the number of comparisons by finding a quantitative way of comparing outcomes and then defining a procedure to get the optimal outcome with multiple users.

Li et al. [23] have structured the procedure of collective decision making in combinatorial domains. In this paper, each user's preferences have been presented with Tradeoffs-enhanced Conditional Preference Networks (TCP-nets)[8] and then they have translated each TCP-net into a penalty scoring function. In the next step, they have defined a collective penalty scoring function by introducing a heuristic algorithm to overcome the problem of the large number of possible outcomes.

In another paper, Li et al. [20] have introduced two different phases; in the first one they are generating an order over the outcomes for each user and then in the second phase, they have a procedure to prune the outcome space to get the most desirable outcome from each user's set of ordered outcomes.

Wang et al. [30] have proposed a qualitative approach to complete incomplete information of users. In this approach, incomplete information of users is completed based on similar users' preferences, but the idea behind finding similar agents is based on generating the entire induced preference graph, which is not feasible with a large number of users and attributes.

However, two major questions about all of this research in collective decision making are the following:

- Do we need to compare each user's preferences over *all* of the attributes?
- Do we need to compare *all* users' preferences when deciding on the value of each attribute for a group of users?

Chapter 3

A New Approach to Collective Decision Making

In this chapter, we will explain all the steps involved in our approach to collective decision making, given a set of CP-nets representing the preferences of a group of users.

- Select the most important attributes for each user.
- Use different proposed weighting techniques to create vectors of attribute weights for each user.
- Use cosine similarity to place users in clusters.
- Combine the preferences of users in each cluster to generate recommended outcomes for the group of users.

3.1 Reducing The Number of Attributes

The process of dominance testing for CP-nets is PSPACE-complete [16]. If we have a large number of attributes, the number of outcomes increases exponentially. To get dominance testing results with a reasonable number of comparisons, we need a way to decrease the number of outcomes that have to be considered. We propose to do this by assigning an importance weight to each attribute for each user and then focusing only on the most important attributes. In the following sections we are going to explain the steps taken to reduce the number of attributes.

3.1.1 Weighting Method

The weighting method plays a crucial role in decreasing the number of comparisons for dominance testing. Giving weights to attributes, based on each user's CP-net, and choosing the most important ones helps us not to compare all the outcomes with each other, but to compare only *partial assignments* (assignments of values to the attributes that are important for that user).

In each CP-net, based on the dependency of each node, we can define a child list for that node. In a recursive way, and by the use of the formula defined in Section 2.4, we can give a weight for each one of the nodes. The process of weighting will be started from the leaf nodes and then we will assign a weight to attributes for which all of the child nodes have a weight. Then we will continue this process until all nodes get a weight. Figure 3.1 and the steps below show the process of weighting attributes in a CP-net.

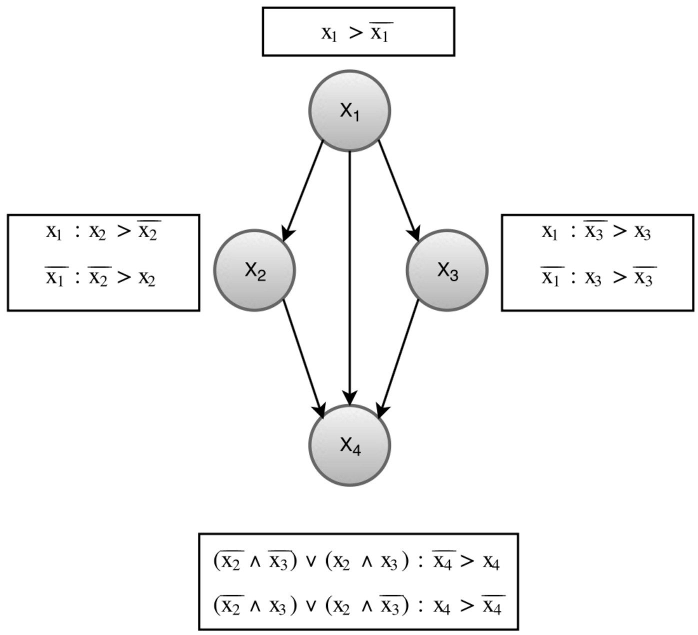

Figure 3.1: Sample CP-net for Demonstrating the Weights of Attributes

$$W_{X_4} = 1$$

$$W_{X_3} = 1 + W_{X_4}.(2 - 1) = 2$$

$$W_{X_2} = 1 + W_{X_4}.(2 - 1) = 2$$

$$W_{X_1} = 1 + (W_{X_2}.(2 - 1) + W_{X_3}.(2 - 1)) + W_{X_4}.(2 - 1)) = 6$$

3.1.2 Choosing the Most Important Attributes

After weighting the attributes of each user's CP-net, we need a procedure to help us in defining a suitable threshold for determining which attributes are considered "important" to the user. By identifying important attributes, we can decrease the number of outcomes to be considered for each CP-net. Hence, if n is the total number of attributes, this weighting method helps us to decrease the number of attributes to m, the number of most important attributes. As a consequence, there is a reduction from a set of 2^n outcomes to a set of 2^m partial assignments (assuming binary attributes), which leads to a good saving of time and space.

The most important attributes for each CP-net are the ones whose weight exceeds a threshold. After finding the most important attributes for each CP-net, we will change the weight of unimportant attributes to zero before beginning the process of clustering CP-nets.

Our approach for choosing the threshold is based on the value of the maximum in-degree. If we know that the maximum in-degree for a CP-net is b and we want to choose important nodes, the formula for finding the threshold and choosing the most important attributes would be as follows:

$$Threshold = MaximumWeight/b$$

where $MaximumWeight$ is the largest weight for any attribute.

3.2 Approaches for Modifying Weights of Attributes

After finding the weight of each attribute and changing the weight of unimportant ones to zero, we need a procedure to find the similarity between each pair of CP-nets. However, one of the problems is that in the process of weighting attributes described in Section 2.4, we only measure the *importance* of the attributes for the users and not their actual *preferences* over these attributes. Suppose that we have two users with the same set of important attributes but their preferences over these attributes are totally the opposite; is it a good idea to put these users in the same cluster? The answer to this question comes from the nature of clustering, that we are partitioning existing objects into different classes whose members are similar. Putting users with very different preferences in the same cluster seems unreasonable; therefore, in the following sections we are going to introduce approaches taken to weight attributes not only based on the structure of CP-nets, but also based on the user's preferences over the attributes.

3.2.1 Modifying Weights Based on Each User's Most Preferred Outcome

In this process, before finding the similarity between each pair of CP-nets, we are going to modify the weights given for each user's most important attributes based on the outcome that is the user's most preferred out of all outcomes. This procedure will help us to differentiate users with the same set of important attributes but different preferences over them.

Suppose that we have two users with their CP-nets and we want to modify the weight given for each one of the attributes based on their most favorable outcome. Figure 3.2 displays the CP-net structure of these two users and Table 3.1 depicts the weights given for attributes. The weights given for these attributes in each one of the CP-nets are very close together, which means that the structure of these CP-nets are similar; however, if we want to compare the similarity of these two CP-nets based on the preferences over attributes we can clearly find out their structure is not as similar as what we were expecting. For example, $User1$ prefers attribute value $\overline{x_1}$ over x_1, while $User2$ prefers value x_1 over $\overline{x_1}$. Therefore, after giving weights for attributes we need a way to modify them based on the preferences of users.

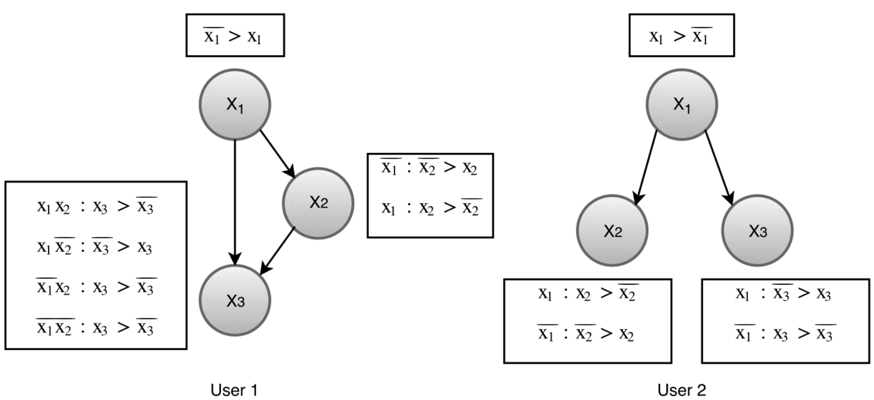

Figure 3.2: Weighting Example

Table 3.1: Primitive Weights for Attributes

	Name of Attributes		
User Name	X_1	X_2	X_3
User 1	4	2	1
User 2	3	1	1

A procedure for modifying the weights of attributes can be achieved by finding the best outcome for each one of the CP-nets and giving positive weight for attributes with a preferred value of x_i and negative weight for the attributes with a preferred value of $\overline{x_i}$. Table 3.2 shows the values for the best outcome for the two CP-nets in Figure 3.2 and their modified weights based on these preferences. The weights of attributes for these two CP-nets show that in finding the similarity of CP-nets, not only the structure of the CP-nets is important, but also users' preferences over attributes.

Table 3.2: Weights of Attributes Based on Best Outcome for Each User

User Name	Name of Attributes	X_1	X_2	X_3
User 1	Value for Att	$\overline{x_1}$	$\overline{x_2}$	x_3
	Weight of Att	-4	-2	1
User 2	Value for Att	x_1	x_2	$\overline{x_3}$
	Weight of Att	3	1	-1

3.2.2 Attribute-Value-Based Weighting

Another approach to the process of weighting the attributes is adjusting the weights of attributes based on the average penalty score associated with outcomes that include each individual attribute value.

In this procedure, after finding the most important attributes for each user, we would be able to define the set of all possible outcomes over those attributes. As we discussed in Section 3.1.2, instead of generating all the outcomes, we are only generating partial assignments involving the most important attributes for the user. The next step in this process is to compare the penalty score described in Section 2.5 for each assignment based on the user's preferences over the important attributes, and then find the average penalty score for assignments that include each attribute value. This step helps us to find out on average which value of each attribute is more preferable for the user. The process below helps us to define a weight for each one of the attributes.

- For a set of binary attributes $X_1, X_2, \cdots, X_n$, let $V = \{x_1, \overline{x_1}, x_2, \overline{x_2}, \cdots, x_n, \overline{x_n}\}$ be the set of all possible attribute values.

- For each $v \in V$, let p_v be the average penalty score over all outcomes that include value v. For example, p_{x_1} is the average penalty score of outcomes that include the value x_1, while $p_{\overline{x_1}}$ is the average penalty score of outcomes that include the value $\overline{x_1}$.

- Let $\overline{p}$ be the average penalty score over all outcomes.

- For each attribute X_i, define the weight w_i to be the following:

$$w_i = \frac{\overline{p} - p_{x_i}}{\overline{p} - \min\limits_{v \in V} p_v}$$

In this process, the weights of attributes are in the range of $[-1, 1]$ and they will be positive or negative based on each user's preference over the value of those attributes. If the user's preference over the value of an attribute X_i is x_i, it will get a positive weight; on the other hand, if the user's preference is $\overline{x_i}$, it will get a negative value. The magnitude of a weight shows how strong the user's preference is over the value of that attribute. If the weight of an attribute is $+1$, this means that this user strongly prefers x_i to $\overline{x_i}$. When its weight is -1, this means that this user strongly prefers $\overline{x_i}$ to x_i. When its weight is a number close to 0, this means that this user is indifferent over the value of this attribute or that the value x_i is preferred in some conditions, while $\overline{x_i}$ is preferred in other situations.

The example below clarifies the procedure for this approach. If Figure 3.3 depicts the preferences of the user over the three most important attributes, we would be able to find a penalty score for each one of the outcomes. Table 3.3 depicts the penalty score for each one of the outcomes, using the technique described in Section 2.5.

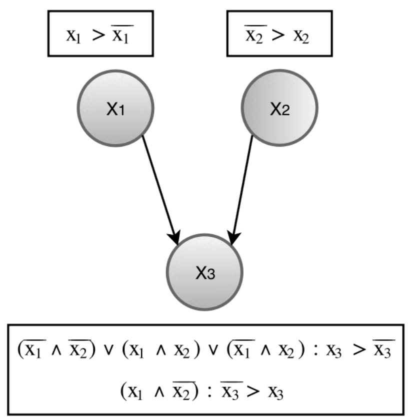

Figure 3.3: Example CP-net for a User

Table 3.3: Penalty Scoring Weight of Each Outcome

Outcomes	Penalty Scoring Weight
$x_1x_2x_3$	2
$x_1x_2\overline{x_3}$	3
$x_1\overline{x_2}x_3$	1
$x_1\overline{x_2x_3}$	0
$\overline{x_1}x_2x_3$	4
$\overline{x_1}x_2\overline{x_3}$	5
$\overline{x_1x_2}x_3$	2
$\overline{x_1x_2x_3}$	3

After finding a penalty score for each one of the outcomes, it is time to find the average penalty score for each attribute value. Table 3.4 shows the average penalty score for the value of each one of the attributes, the overall average and the overall minimum.

Table 3.4: Average Penalty Scoring Weight for the Values of Attributes

Attribute Value	Average Penalty Scoring Weight
x_1	1.5
$\overline{x_1}$	3.5
x_2	3.5
$\overline{x_2}$	1.5
x_3	2.25
$\overline{x_3}$	2.75
Overall Average	2.5
Overall Minimum	1.5

In this step, based on the average penalty scores for each one of the attributes, we can choose the best value for them; in this example, x_1, $\overline{x_2}$ and x_3 are the best values for each attribute. The next step is giving a weight for each one of the attributes based on the values of the attributes for each user. By the use of the formula mentioned above, we would be able to give a weight for each one of the attributes. For example, as the average penalty scoring weight chosen for attribute x_1 is 1.5, the weight of this attribute for that user would be as follows:

$$Weight of Attribute \quad x_1 = (2.5 - 1.5)/(2.5 - 1.5) = 1$$

Table 3.5 shows the weights of attributes. As the preferred value of attribute X_2 is $\overline{x_2}$, it will get a negative weight value, but because both attributes X_1

and X_3 have preferred values of x_1 and x_3, we will assign positive weights for them. The weights over these attributes show that this user strongly prefers x_1 over $\overline{x_1}$ as it is $+1$, strongly prefers $\overline{x_2}$ over x_2 as it is -1, and slightly prefers the value of x_3 over $\overline{x_3}$ as its weight is 0.25 and close to 0.

Table 3.5: Weights of Attributes Based on the Attribute-Value-Based Weighting Method

Attribute	Weight of Attribute
X_1	1
X_2	-1
X_3	0.25

3.3 Finding the Cosine Similarity Between Each Pair of CP-nets

After defining a weight vector for the attributes of each one of the CP-nets, using one of the three weighting methods described in previous sections, we will use cosine similarity, introduced in Section 2.6.4, to find the similarity between each pair of CP-nets. If the number of CP-nets is N, the output of this cosine similarity computation is an $N \times N$ similarity matrix with values in the range $[-1, 1]$. CP-nets that are similar to each other have values close to one and ones that are not similar to each other have values close to negative one; this matrix can then be used as an input for clustering methods. For example, if Table 3.6 shows the weights of important attributes for three users, the cosine similarity between these users and the distance matrix can be computed as shown below:

Table 3.6: Weights for each Attribute

	Name of Attribute				
User	Att 1	Att 2	Att 3	Att 4	Att 5
User 1	5	0	0	-3	-2
User 2	5	0	0	-2	-4
User 3	-3	5	0	0	4

$$\cos(\mathbf{d_{User1}}, \mathbf{d_{User2}}) = \frac{\overrightarrow{d_{User1}} \cdot \overrightarrow{d_{User2}}}{\|\overrightarrow{d_{User1}}\| \|\overrightarrow{d_{User2}}\|}$$

$$= \frac{5 \times 5 + 0 \times 0 + 0 \times 0 + (-3) \times (-2) + (-2) \times (-4)}{\sqrt{5^2 + 0^2 + 0^2 + (-3)^2 + (-2)^2}\sqrt{5^2 + 0^2 + 0^2 + (-2)^2 + (-4)^2}} = 0.94$$

$$\text{Distance Matrix} = \begin{array}{c c} & \begin{array}{ccc} User1 & User2 & User3 \end{array} \\ \begin{array}{c} User1 \\ User2 \\ User3 \end{array} & \begin{pmatrix} 1 & 0.94 & -0.52 \\ 0.94 & 1 & -0.65 \\ -0.52 & -0.65 & 1 \end{pmatrix} \end{array}$$

3.4 Clustering CP-nets

The similarity matrix created in the previous step can be used as an input for clustering CP-nets. If we consider the matrix as a graph, the CP-nets can be considered as nodes in this graph and the degree of similarity between each pair of CP-nets as weighted edges between the nodes. If we give this graph as an input to Gephi [1], we would be able to cluster CP-nets.

One of the clustering methods used by Gephi is based on modularity, one of the measurements for the structure of networks and graphs. The most important goal of modularity is to measure the strength of dividing a network into different modules by looking for nodes that are more densely connected to each other in comparison with the rest of the nodes in the graph[3]. By defining the modularity ranking class and running the tool in Gephi, we can find the clusters with which each node has been associated.

3.5 Normalizing the Weights of Attributes

After clustering CP-nets, we follow a procedure to go through the attributes and find which clusters contain users who place a high importance on each attribute. However, we should first normalize the weights of attributes. The normalization method normalizes the distribution of weights for each attribute between the highest and the lowest weight for that attribute. As a

[1] Gephi is an open source software tool for exploring, analyzing, filtering, clustering and manipulating all types of networks[15].

result, it helps us to compare the importance of attributes in a fair way.

Let $w_{i,j}$ be the weight assigned in CP-net i to attribute j, let $max_{i,*}$ be the maximum weight for any attribute in CP-net i, and let $Max_{*,*}$ be the maximum weight for any attribute in any CP-net in the set. The normalization formula for attributes can be defined as follows:

$$NormalizedWeight_{i,j} = w_{i,j} \times Max_{*,*}/max_{i,*}$$

For example, suppose we have 12 users and, based on their preferences, we have created weighted CP-nets (with the most important attributes for each user) in three clusters, depicted in Tables 3.7, 3.8 and 3.9.

Table 3.7: Users in First Cluster and Their Weights for Attributes

	Attributes							
Cluster One	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
User 1	0	6	5	2	0	2	0	0
User 2	0	3	6	0	2	0	0	0
User 3	0	3	3	7	0	0	0	0

Table 3.8: Users in Second Cluster and Their Weights for Attributes

	Attributes							
Cluster Two	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
User 4	0	2	2	8	3	4	0	0
User 5	0	6	0	8	7	0	0	0
User 6	0	2	0	0	7	4	0	0
User 7	0	4	0	5	0	8	6	0

Table 3.9: Users in Third Cluster and Their Weights for Attributes

	Attributes							
Cluster Three	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
User 8	7	0	6	0	0	0	3	2
User 9	8	0	2	0	0	3	4	0
User 10	0	0	4	0	0	0	8	2
User 11	4	0	7	0	0	0	0	8
User 12	0	0	5	0	0	2	2	0

In this specific example, the maximum weight given by any user to any attribute is 8. The maximum weight given to any attribute by *User 1* is 6. Therefore, for example, the normalized weight of attribute X_3 for *User 1* would be:

$$5 \times 8/6 = 6.67$$

Normalized weights for each attribute have been shown for this example in Tables 3.10, 3.11 and 3.12.

Table 3.10: Users in First Cluster and Their Normalized Weights for Attributes

	Attributes							
Cluster One	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
User 1	0	8	6.67	2.6	0	2.6	0	0
User 2	0	4	8	0	2.6	0	0	0
User 3	0	3.42	3.42	8	0	0	0	0

Table 3.11: Users in Second Cluster and Their Normalized Weights for Attributes

	Attributes							
Cluster Two	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
User 4	0	2	2	8	3	4	0	0
User 5	0	6	0	8	7	0	0	0
User 6	0	2.28	0	0	8	4.57	0	0
User 7	0	4	0	5	0	8	6	0

Table 3.12: Users in Third Cluster and Their Normalized Weights for Attributes

	Attributes							
Cluster Three	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
User 8	8	0	6.85	0	0	0	3.42	2.28
User 9	8	0	2	0	0	3	4	0
User 10	0	0	4	0	0	0	8	2
User 11	4	0	7	0	0	0	0	8
User 12	0	0	8	0	0	3.2	3.2	0

3.6 Choosing Clusters for Which Each Attribute is Important

After normalizing the weights of attributes, we need a procedure to go through the attributes and find clusters for which this attribute is important. This procedure can be done by following the steps below.

In the first step, we find the average weight given to each attribute by the users in each cluster. Table 3.13 shows the average weight of attributes for each cluster, for the example in previous section.

Table 3.13: Average Normalized Weight for Attributes in each Cluster

	Attributes							
Cluster Name	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Cluster One	0	5.14	6.02	3.53	0.86	0.86	0	0
Cluster Two	0	3.57	0.5	5.25	4.5	4.14	1.5	0
Cluster Three	4	0	5.57	0	0	1.24	3.72	4.45

After finding the average weight of attributes for each cluster, we define a threshold to choose the clusters for which each attribute is important. This threshold can be defined based on finding the maximum and minimum of the average weights given for each attribute in each cluster in the previous step. The formula below is used to get this threshold.

$$Threshold = (min + (3 \times max))/4$$

where max and min are the maximum and minimum of the weights given for each attribute in each cluster in the previous step. This calculation gives the value that is 3/4 of the way between the minimum and maximum values. For example, the threshold for attribute X_2 is:

$$Threshold = (0 + (3 \times 5.14))/4 = 3.855$$

A cluster is then selected for an attribute if the average weight for the attribute in that cluster exceeds the threshold. For example, $Cluster1$ is chosen as a suitable one for deciding over the value of attribute X_2, because in this cluster the normalized weight for this attribute exceeds the value above. Table 3.14 shows the clusters for which each attribute is considered sufficiently important. Note that most attributes are found to be important for only one cluster, but attribute X_3 was found to be important for two clusters.

Table 3.14: Important Attributes in each Cluster

	Name of Attributes							
Cluster Name	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8
Cluster One	χ	✓	✓	χ	χ	χ	χ	χ
Cluster Two	χ	χ	χ	✓	✓	✓	χ	χ
Cluster Three	✓	χ	✓	χ	χ	χ	✓	✓

After finding the most suitable clusters for each attribute, we would be able to choose desirable outcomes for the clusters not by considering all attributes, but by considering the ones that have been identified for this cluster. This decreases the number of (partial) attribute assignments considered for each cluster, based on the number of attributes chosen for it. In this example, for finding users' preferences in the first cluster, instead of comparing users' preferences over 2^8 outcomes (with 8 attributes), we would be able to get the final outcome by comparing 2^2 partial assignments: x_2x_3, $x_2\overline{x_3}$, $\overline{x_2}x_3$ and $\overline{x_2x_3}$.

3.7 Finding Users' Preferences Based on Chosen Attributes for Each Cluster

In this section, we are going to explain an approach for deciding the best combination of values for the attributes that are important for a cluster, even when the attributes that are important to each individual user are not exactly the same.

This procedure will be illustrated with an example. Suppose that we have three users with different important attributes, in a cluster in which two attributes X_1 and X_3 have been identified as important for the cluster as a whole. Figure 3.4 displays users' preferences over their important attributes.

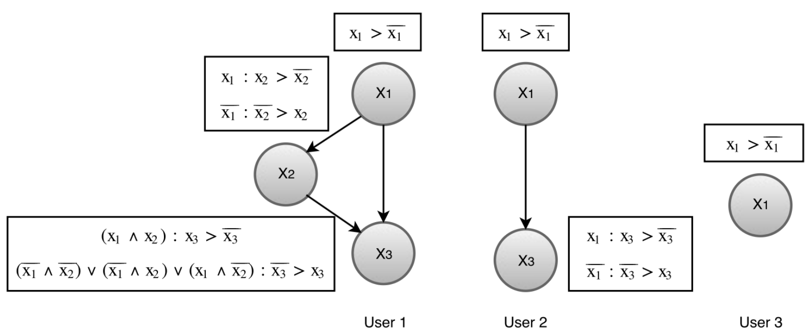

Figure 3.4: Users' Preferences Over Attributes

Now we are going to find the best partial attribute assignment for this cluster based on users' preferences. As in this cluster two attributes are important, we will consider four different assignments: x_1x_3, $x_1\overline{x_3}$, $\overline{x_1}x_3$ and $\overline{x_1x_3}$. Now we should be able to map the weights of users' preferences over attribute assignments to find the best assignment for each cluster. Tables 3.15, 3.16

and 3.17 show the outcomes and their penalty scores based on each user's preferences over their important attributes.

Table 3.15: Penalty Scores for Each Assignment for User 1

User 1	
Assignments	Total Weight
$x_1x_2x_3$	0
$x_1x_2\overline{x}_3$	1
$x_1\overline{x}_2x_3$	3
$x_1\overline{x}_2\overline{x}_3$	2
$\overline{x}_1x_2x_3$	7
$\overline{x}_1x_2\overline{x}_3$	6
$\overline{x}_1\overline{x}_2x_3$	5
$\overline{x}_1\overline{x}_2\overline{x}_3$	4

Table 3.16: Penalty Scores for Each Assignment for User 2

User 2	
Assignments	Total Weight
x_1x_3	0
$x_1\overline{x}_3$	1
$\overline{x}_1x_3$	5
$\overline{x}_1\overline{x}_3$	4

Table 3.17: Penalty Scores for Each Assignment for User 3

User 3	
Assignments	Total Weight
x_1	0
$\overline{x}_1$	4

Now it is time to map users' assignments and their penalty scores into cluster assignments. For instance, Table 3.18 demonstrates the mapping of the 8 assignments for user 1 into the 4 assignments for the cluster. For the first assignment of this cluster which is x_1x_3, we can map two assignments of user 1, which are $x_1x_2x_3$ and $x_1\overline{x_2}x_3$; therefore, we should consider the penalty scores of these two outcomes for user 1 and then take the average.

Table 3.18: Mapping Penalty Scoring Weights for Cluster Assignments Based on Penalty Scoring of User 1

Assignments	Mapped Weights	Average Weight
x_1x_3	$0, 3$	1.5
$x_1\overline{x}_3$	$1, 2$	1.5
$\overline{x}_1x_3$	$7, 5$	6
$\overline{x}_1\overline{x}_3$	$6, 4$	5

Table 3.19 depicts the average weights of assignments for this cluster based on users' preferences. Therefore, after going through all steps, the assign-

ment with the lowest penalty scoring weight would be the most favourable assignment for this cluster; in this specific example assignment x_1x_3 is the most favourable one for the members of the cluster.

Table 3.19: Penalty Scoring Weights for Cluster Assignments Based on Users' Penalty Scoring

	Name of Users			
Assignments	User 1	User 2	User 3	Final Weight
x_1x_3	1.5	0	0	1.5
$x_1\overline{x}_3$	1.5	1	0	2.5
$\overline{x}_1x_3$	6	5	4	15
$\overline{x}_1\overline{x}_3$	5	4	4	13

3.8 Deciding About Common Attributes Between Clusters

After finding the best attribute assignment for each cluster, it is time to decide what to do with attributes that are important in more than one cluster. If attribute X_i is important in two clusters, but cluster A prefers value x_i while cluster B prefers value $\overline{x_i}$, how should this be handled?

The only factor that we are going to consider in choosing the final value for each common attribute is the number of users in each cluster. If attribute value x_i is preferred by clusters that include n users and if $\overline{x_i}$ is preferred by clusters that include m users, and if $n > m$, then we will choose value x_i. In this way, we are deciding on the value of common attributes based on the preferences of users in the cluster with the higher number of users and therefore we are making a larger number of users satisfied with this selection. In this example, attribute X_3 is common between $Cluster1$ and $Cluster3$ and if these two clusters disagree on X_3's value, the only factor for choosing one of the values is the number of users in each cluster.

3.9 Two Approaches for Recommending Outcomes

In this research we are going to define two different kinds of recommended outcomes. The first one is to determine a global outcome to be recommended if one outcome has to be chosen for all of the users. The second one is to determine a recommended outcome that is specific to each cluster. In the second case, we will decide on the values of attributes that are not important in the cluster based on the values defined for the global recommended outcome. For example, if the global outcome for the previous example becomes $\overline{x_1}\overline{x_2}x_3x_4\overline{x_5}\overline{x_6}\overline{x_7}x_8$ we are going to define $x_1\overline{x_2}x_3x_4\overline{x_5}\overline{x_6}\overline{x_7}x_8$ as the final outcome for $Cluster1$, because the two important attributes for this cluster are X_1 and X_3 and the most favorable values for these two attributes are x_1 and x_3. We then define the values for the rest of attributes based on the values defined for the global outcome. After defining these two kinds of outcomes, we are going to evaluate them by the approaches that will be defined in Chapter 4.

Chapter 4

Evaluation of Proposed Approaches

After producing a recommended global outcome and an outcome for each cluster, we wish to evaluate these outcomes by analyzing how favourable they are for the users. Our goal for this work is to compare the results of the different weighting methods developed in this thesis with each other by using several different evaluation approaches, which will be explained in the following subsections. [1]

[1] As acknowledged in Chapter 5, one direction for future work is to further evaluate these proposed techniques by comparing their performance directly to the performance of other approaches in the literature.

4.1 Generating CP-nets

Finding a suitable data set of CP-nets is one of the challenges of this research. CP-net generator (GenCPnet), a program inplemented in C++, released in December 2015, has been used for randomly generating acyclic conditional preference networks [1]. This method helps us to define CP-nets based on the number of nodes, and maximum number of parents that a node can have. For example, the following command generates $g = 10$ CP-nets with $n = 10$ nodes, $c = 4$ as the in-degree bound (each node can have at most four parents) and with a binary domain of $d = 2$ for each node.

```
gencpnet -n 10 -c 4 -d 2 -g 10 temp
```

This method helps us to create an unlimited number of CP-nets with different characteristics in separate XML files. For example, the XML file fragment below shows that in this specific CP-net we have a node named $x1$ with a domain size of two; therefore, this node can have two different values, 1 and 2, which corresponds to $Dom(X_1) = \{x_1, \overline{x}_1\}$.

<PREFERENCE-VARIABLE>

<VARIABLE-NAME> $x1$ < /VARIABLE-NAME>

<DOMAIN-VALUE> 1 < /DOMAIN-VALUE>

<DOMAIN-VALUE> 2 < /DOMAIN-VALUE>

< /PREFERENCE-VARIABLE>

For each one of these nodes, we should show the preference over the domain of each attribute. For example, in the XML file fragment below, we are deciding over the value of attribute $x2$ when the parents of this attribute are $x1$ and $x3$. This part of code indicates that when the parents' values are $x1 = 1$ and $x3 = 1$ in this CP-net, the value of $x2 = 1$ is preferred over $x2 = 2$. Based on the number of parents for each node, there would be $2^{NumberOfParents}$ different combinations shown in the values for XML file. For this specific example, we would have four different combinations of values for the parents $x1$ and $x3$, and the preference for attribute $x2$ would be shown for each of these combinations.

<PREFERENCE-STATEMENT>

<STATEMENT-ID> $p2 - 1$ < /STATEMENT-ID>

<PREFERENCE-VARIABLE> $x2$ < /PREFERENCE-VARIABLE>

<CONDITION> $x1 = 1$ < /CONDITION>

<CONDITION> $x3 = 1$ < /CONDITION>

<PREFERENCE> $1 : 2$ < /PREFERENCE>

< /PREFERENCE-STATEMENT>

$\dots$

In our testing, we have generated some number of random CP-nets by the use of GenCPnet [1]. For example, we have generated 20 random CP-nets with 6 binary attributes. The next step is weighting the attributes in order to find the most important attributes for each CP-net, based on the different weighting approaches defined in Section 3.2. By the use of these weighting methods, we would be able to find the similarity between each pair of CP-nets by the use of cosine similarity introduced in Section 3.3. The final output of the cosine similarity is a matrix that shows the similarity of each pair of CP-nets, which can be used as an input to Gephi to divide users into clusters. Table 4.1 displays the result of clustering users for this set of 20 users, based on different weighting approaches. This shows that the clusters formed by the three weighting methods are quite similar, but that they do differ.

Table 4.1: Clustering Results By the Use of Different Weighting Approaches

Cluster Name	Plain Weighting	Best Outcome Weighting	Attribute-Value-Based Weighting
Cluster 1	User 1	User 1	User 1
			User 2
	User 4	User 4	
	User 6	User 6	User 6
	User 9		
	User 13	User 13	User 13
	User 15		User 15
	User 16		
	User 17	User 17	User 17
		User 18	
Cluster 2	User 5	User 5	
	User 7	User 7	
		User 9	User 9
	User 11	User 11	
	User 12		User 12
			User 16
		User 19	User 19
	User 20		User 20
Cluster 3	User 2	User 2	
	User 3	User 3	User 3
			User 4
			User 5
			User 7
	User 8	User 8	User 8
	User 10	User 10	User 10
			User 11
		User 12	
	User 14	User 14	User 14
		User 15	
		User 16	
	User 18		User 18
	User 19		
		User 20	

After clustering the users, we can find global and cluster-specific recommendations. Table 4.2 shows an example of these recommendations for each cluster by the use of different weighting methods, for one set of 20 users. In each cluster, important attributes for which values are recommended are shown as x_i or $\overline{x_i}$, while unimportant attributes without any defined values are shown as X_i.

Table 4.2: Cluster-specific and Global Recommendations By the Use of Different Weighting Approaches

Cluster Name	Plain Weighting	Best Outcome Weighting	Attribute-Value-Based Weighting
Cluster 1	$\overline{x_1}x_2X_3x_4X_5X_6$	$X_1x_2X_3x_4X_5X_6$	$\overline{x_1}X_2X_3x_4X_5X_6$
Cluster 2	$X_1x_2x_3X_4x_5X_6$	$X_1X_2x_3X_4X_5\overline{x_6}$	$X_1\overline{x_2}\overline{x_3}\overline{x_4}X_5\overline{x_6}$
Cluster 3	$\overline{x_1}X_2X_3X_4x_5x_6$	$\overline{x_1}\overline{x_2}X_3X_4x_5x_6$	$X_1x_2x_3X_4x_5x_6$
Global Outcome	$\overline{x_1}x_2x_3x_4x_5x_6$	$\overline{x_1}\overline{x_2}x_3x_4x_5x_6$	$\overline{x_1}x_2x_3x_4x_5x_6$

4.2 Comparing the Penalty Scores of Outcomes

4.2.1 Evaluation Method

In this procedure, by referring to the users' preferences, we are going to find penalty scoring weights[23] (as described in Section 3.3) for the recommended global outcome, for the cluster-specific outcome, and for the worst outcome. If the penalty scoring weights for the global outcome and the cluster-specific outcome are less than half of the penalty scoring weight for the worst outcome, we are going to count that outcome as an acceptable one for this user. By counting the number of users for whom these outcomes are acceptable, we would be able to evaluate the recommended global outcome and cluster-specific outcome.

For example, if the final recommended outcome is $x_1\overline{x_2}x_3x_4\overline{x_5x_6x_7}x_8$ and for $User1$ two attributes X_2 and X_3 are important (with weights of 8 and 6.67 respectively), we are going to give a penalty scoring weight for the final outcome based on the preferences of $User1$ over these two attributes; therefore, the penalty scoring weight for this outcome is $1 \times 8 + 0 \times 6.67 = 8$. The penalty scoring weight for the worst outcome of this user is when $\overline{x_2x_3}$; therefore, the penalty scoring weight for the worst outcome is $1 \times 8 + 1 \times 6.67 = 14.67$. Finally, the weight given for the final recommended outcome based on the user's preferences over the most important attributes is not less that half of the weight for the worst outcome for this user. Therefore, the status of this user will be unsatisfied. We will follow this procedure for all users and then by counting the number of users who would be satisfied we can evaluate the global outcome. We can follow the same procedure for the cluster-specific recommendations.

4.2.2 Evaluation Results

Table 4.3 displays the results of the evaluation method for global and cluster-specific outcomes by determining if the penalty scoring weights for the recommended outcomes are less than half of the maximum penalty score for any outcome. These results are averages over three trials involving 20 users with randomly generated CP-nets (with 6 binary attributes) and three trials involving 40 users (with 8 binary attributes).

Table 4.3: Percentage of Satisfied Users for Global and Cluster-specific Outcomes Based on Comparing Penalty Scoring Weights of Outcomes

		Plain Weighting	Best Outcome Weighting	Attribute-Value-Based Weighting
Global Outcome	20 Users	65%	71.7%	70%
	40 Users	73.3%	73.3%	65.8%
Cluster-specific Outcomes	20 Users	90%	93.3%	98.3%
	40 Users	93.3%	98.3%	100%

As Table 4.3 depicts, the results for cluster-specific outcomes are substantially better than those for the global recommended outcomes, with the best outcome and attribute-value-based weighting methods outperforming the plain weighting method. It is acknowledged that having a penalty score that is less than half of the maximum is a fairly low standard, so we will look at other ways of evaluating these recommendations in the remaining sections of this chapter.

4.3 Comparing the Dominance Between Recommended Outcomes and Random Outcomes

4.3.1 Evaluation Method

In this procedure, after generating a set of R random outcomes, we perform dominance checking between the recommended outcomes and these random outcomes for each of a set of N users, based on the work by Santhanam et al. [29] explained in Section 2.3.3. We find the number of times that the recommended outcome dominates a random outcome (Dom), the number of times that the recommended outcome is dominated by a random outcome ($revDom$) and the number of times that there is no dominance between them ($noDom$). These numbers will help us to find an overall score for the recommended outcome. The formula for this score is defined as follows:

$$Score = \frac{Dom + 0.5 \times noDom}{N \times R} \times 100$$

As another measure, we also record the number of users for which this recommended outcome is at least as preferred as 60%, 70%, 80% and 90% of the random outcomes.

4.3.2 Evaluation Results for Global Recommended Outcomes

We created three different sets of CP-nets for 20 users (with 6 binary attributes) and 40 users (with 8 binary attributes). Then for each set, we found the global outcome driven by the different weighting approaches, and then we used dominance checking to compare these global outcomes to 64 random outcomes (for the 20-user cases) or 50 random outcomes (for the 40-user cases).

Table 4.4 summarizes the evaluation results of three trials for the global outcomes driven by different weighting approaches.

Table 4.4: Summarizing Evaluation Results of Three Trials for Global Outcomes Driven by Different Weighting Approaches

Weighting Method	Set of CP-nets	o^* dominates random outcome	no dominance	o^* is dominated by random outcome	Is o^* better than or "tied" with ___ of random outcomes? $\geq$ 60%	$\geq$ 70%	$\geq$ 80%	$\geq$ 90%
Plain Weighting	20 Users	46.2%	23.1%	30.7%	53.3%	43.3%	33.3%	21.7%
	40 Users	44.3%	26.1%	29.6%	52.5%	39.2%	29.2%	15%
Best Outcome Weighting	20 Users	47.5%	25.7%	26.8%	61.7%	43.3%	31.7%	15%
	40 Users	43.6%	28.2%	28.9%	59.2%	45%	27.5%	15%
Attribute-Value-Based Weighting	20 Users	47.8%	24.8%	27.4%	63.3%	46.7%	26.7%	16.7%
	40 Users	39.4%	29.2%	32.1%	50.9%	36.7%	20.8%	12.5%

Table 4.5 summarizes the results of Table 4.4 based on the scoring method described in Section 4.3.1.

Table 4.5: Evaluation Results of Three Trials Based on Dominance, Comparing Global Recommendations Driven by different Weighting Approaches to Random Outcomes

Weighting Method		Score
Plain Weighting	20 Users	57.7
	40 Users	57.4
Best Outcome Weighting	20 Users	60.3
	40 Users	57.7
Attribute-Value-Based Weighting	20 Users	59.5
	40 Users	54.0

Based on the results depicted in Table 4.5, we can see that the scores of the plain weighting, best outcome weighting and attribute-value-based weighting methods in finding the global outcome are quite close to each other. From a series of paired t-tests performed on these results, it was found that these differences in scores were not statistically significant.

Moreover, Table 4.5 shows that when we are increasing the number of users, making a higher percentage of users satisfied by offering a global outcome is hard, because as we increase the number of users, we see a more diverse set of preferences.

Finally, it is noted from Table 4.4 that more than half of the recommendations are at least as good as 60% of random outcomes, while roughly 1/4 - 1/3 of recommendations are at least as good as 80% of random outcomes.

4.3.3 Evaluation Results for Cluster-specific Outcomes

In the next step of evaluating the outcomes, we have done the same comparisons, but for the outcomes that were chosen specifically for each cluster. Table 4.6 summarizes the evaluation results of three trials for the cluster-specific recommendations driven by different weighting approaches, comparing them to randomly-chosen outcomes.

Table 4.6: Evaluation Results of Three Trials for Cluster-specific Outcomes Driven by Different Weighting Approaches

Weighting Method	Set of CP-nets	o^* dominates random outcome	no dominance	o^* is dominated by random outcome	Is o^* better than or "tied" with ___ of random outcomes? $\geq 60\%$	$\geq 70\%$	$\geq 80\%$	$\geq 90\%$
Plain Weighting	20 Users	46.4%	22.7%	30.9%	55%	45%	33.3%	23.3%
	40 Users	47.2%	25.1%	28.4%	57.5%	45%	31.7%	15.8%
Best Outcome Weighting	20 Users	60.2%	23.1%	16.7%	73.3%	65%	48.3%	26.7%
	40 Users	56.4%	24.6%	19.6%	74.2%	63.3%	45.8%	25%
Attribute-Value-Based Weighting	20 Users	57.7%	23%	19.2%	78.3%	55%	40%	21.7%
	40 Users	59.2%	23.5%	18%	79.2%	65.8%	50%	30%

Table 4.7 summarizes the results of Table 4.6 based on the scoring method defined in Section 4.3.1.

Table 4.7: Evaluation Results of Three Trials Based on Dominance, Comparing Cluster-specific Outcomes Driven by different Weighting Approaches to Random Outcomes

Weighting Method		Scoring
Plain Weighting	20 Users	57.7
	40 Users	59.7
Best Outcome Weighting	20 Users	71.8
	40 Users	68.7
Attribute-Value-Based Weighting	20 Users	69.2
	40 Users	71.0

Table 4.7 shows that the best outcome and attribute-value-based weighting method have outperformed the plain weighting method by a substantial margin. In fact, from a series of paired t-tests that were performed, the increases

in performance for both the best outcome method and the attribute-value-based method, over the plain weighting method, were found to be statistically significant ($p < 0.05$) for the $20-$user and $40-$user cases.

However, perhaps the most meaningful insights come from comparing Table 4.7 (for cluster specific recommendations) to Table 4.5 (for a global recommendation). In Table 4.7, the scores for the best outcome and attribute-value-based weighting method are in the 68-72 range, while they were between 54 and 61 for the global recommendation. This shows the degree to which we can make users more satisfied if our recommendations are tailored to clusters of users, rather than trying to make one recommendation for all users.

Finally, it is noted from Table 4.6 that roughly 3/4 of recommendations from these two methods are at least as good as 60% of random outcomes, while almost half of the recommendations are at least as good as 80% of random outcomes. This is substantially stronger than what we saw for the global recommendations in Table 4.4

4.4 Comparing the Dominance Between Recommended Outcomes and Baseline Outcomes

4.4.1 Evaluation Method

In this method, we are going to define a simple baseline outcome and then by the use of the procedure for dominance testing between two outcomes [29], we find the dominance between the recommended outcomes and the baseline. In this procedure, the baseline outcome has been defined in an iterative way, using the preferences of each user over their most important attributes. We choose the most important attribute for each user and then find the preferred value for this important attribute for this user. If an attribute is the most important for more than one user, we will choose that attribute's value by considering the importance weights of the attribute for each of these users. In the next iteration, we will give a value for attributes that have not had a value assigned yet, by looking at the second-most important attribute for each user. This continues until we have defined a value for all attributes.
After finding the baseline outcome, we evaluate the recommended outcomes by comparing the dominance between the outcomes and this baseline for all users and counting the number of users for which the recommended outcome dominates the baseline.
As a simplified example, suppose we have two attributes X_2 and X_3 and

we want to define the baseline outcome for the users in Cluster 1. In the first iteration we will go through each user's most important attribute and will choose a value for them. The most important attribute for $User1$ is X_2 (with a weight of 8) and the preference of this user over this attribute is x_2. If we use 1 to represent the value x_2 and use -1 to represent the value $\overline{x_2}$ then since this user prefers x_2, we can add the weight of 8×1 into the final value of this attribute. We can follow the same procedure for all the users of $Cluster1$. After finishing the first iteration over all users' preferences in $Cluster1$ we will check all the attributes and assign x_n or $\overline{x_n}$, depending on whether the final value for X_n is positive or negative. If some attributes do not have a value yet, we will continue the next iteration over the next most important attribute for each user until finding a value for all the attributes. Table 4.8 shows users and the final value of attributes in $Cluster1$ based on the preferences of users in this cluster. In this example, the baseline outcome would be x_2x_3. The idea behind this approach is to give us a relatively simple baseline outcome to which we can compare our recommended outcomes.

Table 4.8: Baseline Values for Most Important Attributes of Cluster 1

	Name of Attributes	
	X_2	X_3
User 1	8×1	
User 2		8×1
User 3	$3.42 \times (-1)$	3.42×1
Weight	4.58	11.42
Value	x_2	x_3

4.4.2 Evaluation Results for Global Recommended Outcomes

Table 4.9 summarizes the evaluation results of three trials for each number of users, comparing the global outcomes driven by different weighting approaches to the baseline outcome.

Table 4.9: Summarizing Evaluation Results of Three Trials Based on Dominance, Comparing Global Outcomes Driven by different Weighting Approaches to the Baseline Outcome

Weighting Method	Set of CP-nets	o^* dominates baseline outcome	no dominance	o^* is dominated by baseline outcome
Plain Weighting	20 Users	40%	21.7%	38.3%
	40 Users	39.2%	31.7%	29.2%
Best Outcome Weighting	20 Users	33.3%	43.3%	23.3%
	40 Users	44.2%	25%	30.8%
Attribute-Value-Based Weighting	20 Users	50%	3.3%	45%
	40 Users	37.5%	25.8%	36.7%

Table 4.10 shows a summary score for each of these conditions based on the formula defined in Section 4.3.1, but using the baseline outcome instead of a set of randomly-chosen outcomes.

Table 4.10: Ranking of Evaluation Results of Three Trials Based on Dominance, Comparing Global Outcomes Driven by different Weighting Approaches to the Baseline Outcome

Weighting Method		Score
Plain Weighting	20 Users	50.8
	40 Users	55.0
Best Outcome Weighting	20 Users	55.0
	40 Users	56.7
Attribute-Value-Based Weighting	20 Users	51.7
	40 Users	50.4

The comparisons to the baseline outcome depicted in Table 4.10 are similar to the results in Table 4.5 and show that the global outcome offered by the best outcome weighting approach is the strongest.

The attribute-value-based weighting method, on the other hand, only slightly outperforms the plain weighting method for 20 users and is actually weaker for 40 users.

Not surprisingly, the plain weighting method does not have strong results because it clusters users only based on their important attributes and not on users' preferences over those attributes. However, the results of the other weighting methods were somewhat disappointing. Using McNemar's Test [34], it was found that these methods did not outperform the baseline method by enough of a margin to be statistically significant.

4.4.3 Evaluation Results for Cluster-specific Outcomes

In this step, we have evaluated the favourability of the cluster-specific outcomes by finding the dominance between them and the baseline outcome. Table 4.11 summarizes the evaluation results of three trials for each number of users, comparing the cluster-specific outcomes driven by different weighting approaches to the baseline outcome.

Table 4.11: Summarizing Evaluation Results of Three Trials Based on Dominance, Comparing Cluster-specific Outcomes Driven by different Weighting Approaches to the Baseline Outcome

Weighting Method	Set of CP-nets	o^* dominates baseline outcome	no dominance	o^* is dominated by baseline outcome
Plain Weighting	20 Users	38.3%	18.3%	43.3%
	40 Users	43.3%	25.8%	30.8%
Best Outcome Weighting	20 Users	58.3%	21.7%	20%
	40 Users	58.3%	22.5%	20%
Attribute-Value-Based Weighting	20 Users	51.7%	21.7%	26.7%
	40 Users	62.5%	20%	16.7%

Table 4.12 shows a summary score for each of these conditions based on the formula defined in Section 4.3.2, but using the baseline outcome instead of a set of randomly-chosen outcomes.

Table 4.12: Ranking of Evaluation Results of Three Trials Based on Dominance, Comparing Cluster-specific Outcomes Driven by different Weighting Approaches to the Baseline Outcome

Weighting Method		Score
Plain Weighting	20 Users	47.5
	40 Users	56.2
Best Outcome Weighting	20 Users	69.1
	40 Users	69.6
Attribute-Value-Based Weighting	20 Users	62.5
	40 Users	72.5

The comparisons to the baseline outcome depicted in Table 4.12 show that the cluster-specific outcomes offered by the best outcome and attribute-value-based weighting approaches are much stronger than those offered by the plain method. Using McNemar's Test, these two methods did outperform the baseline method by enough of a margin to be statistically significant ($p < 0.05$) for the $20-$user and $40-$user cases.

Not surprisingly, the plain weighting method does not have strong results because it clusters users only based on their important attributes and not on users' preferences over those attributes.

4.5 Evaluating Recommendations on the Basis of a Collective Penalty Scoring Function

4.5.1 Evaluation Method

Li et al. [23] discuss using a collective penalty scoring function to evaluate the quality of a choice for an entire group of users. They suggest that one reasonable choice for such a measure is to take the *sum* of the individual penalty values for each user.

Our final evaluation method involves calculating the penalty score for all of the outcomes of each CP-net based on users' preferences over all attributes, and then using a collective penalty scoring function to evaluate the quality of outcomes.

We go through all users' preferences over all attributes and then we give penalty scores for all the outcomes based on each user's preferences. In the next step, by taking the sum of these penalty scores for all users, we would be able to order the outcomes based on their favourability for users. This ordering will help us to find the rank of the global outcome suggested by our weighting approaches among all possible outcomes.

4.5.2 Evaluation Results

In Table 4.13, we show where our recommended global outcomes rank in the set of all 64 outcomes for the three trials using 6 attributes for 20 users, using the *sum* of individual penalty values.

Table 4.13: Global Outcomes' Ranking Based on a Collective Penalty Scoring Function

Trial	Weighting Method	Rank of Outcome
Trial One	Best Outcome Weighting	2^{nd}
	Attribute-Value-Based Weighting	1^{st}
	Plain Weighting	4^{th}
Trial Two	Best Outcome Weighting	6^{th}
	Attribute-Value-Based Weighting	5^{th}
	Plain Weighting	5^{th}
Trial Three	Best Outcome Weighting	5^{th}
	Attribute-Value-Based Weighting	7^{th}
	Plain Weighting	6^{th}

As Table 4.13 depicts, the global outcomes that we are offering are among the best and we can claim that, on average, the global outcome that we are offering for users has the rank of 4^{th} in the set of 64 outcomes.

In this method, if we wanted to find the global outcome with the lowest penalty score, we should go through each user's preferences over all attributes; therefore, we should assign penalty scores for each user's preferences over all outcomes. So, if we have A attributes, this method should find penalty scores for 2^A outcomes for each user, which leads to a large number of calculations when we have a large number of users and attributes. In our approach, we are finding penalty scores for each user only over the important attributes for the cluster the user belongs to, which is a relatively small number.

Moreover, as the results of Table 4.5 display, when we have a large number of users, making them all happy by offering a global outcome is hard. For situations in which it is feasible, our approach has also included the ability to generate cluster-specific outcomes, which have the potential to make users significantly more satisfied. All of this is done without the need to compare all outcomes for all users.

We have also found the rank of each cluster-specific outcome, but the results were not as promising as we were expecting. This problem happens because in the procedure of finding cluster-specific outcomes, we find values for the most important attributes and then we define the values for the non-important ones based on the values in the global outcome, without considering users' preferences over those attributes, which leads to a lower rank.

We did observe that the very best outcomes, according to the collective penalty scoring function, often matched our cluster-specific recommendations on the attributes that were the most important for the cluster. Therefore, there is potential for our recommendations to achieve a very high rank if we can improve our approach to assigning values to the less important attributes. Moreover, as the results of Tables 4.6 and 4.7 display, we can see that cluster-specific outcomes have the potential to make all users more satisfied. Therefore, by investigating different ways of setting the values for the less important attributes in cluster-specific outcomes, we could make the results even stronger.

Chapter 5

Conclusions and Future Work

5.1 Summary

The most important goal of this research is to find the answer to the question of whether we can still get good results in collective decision making in combinatorial domains when we are not considering all users' preferences over all attributes. In real world situations, we have a large range of user preferences over a large range of attributes; therefore, satisfying all users is difficult. In this research, we have applied Conditional Preference networks(CP-nets)[6] to represent users' preferences with different preferential dependencies over attributes and then we proposed a novel procedure for collective decision making when the number of attributes and users is high.

The first step of this approach is finding the most important attributes for each user. In the second step, by the use of cosine similarity and based on different weighting procedures, we would be able to find the distance between each pair of CP-nets, which leads to clustering CP-nets into different groups. By comparing users' preferences in each cluster over important attributes for that cluster, we can determine a partial attribute value assignment for each cluster. We then use these partial assignments to produce a global outcome that is desirable for a majority of users. These processes lead to a reduction in the size of the search space. We have evaluated the desirability of recommended outcomes by applying diverse evaluation methods to sets of random CP-nets. Of the approaches proposed in this thesis, the best outcome and attribute-value-based weighting methods appear to be the most promising, especially in situations in which different recommendations can be made to users in different clusters.

5.2 Research Contributions

In this research, collective decision making in real world situations has been studied. One of the steps to make this study close to the real world is the use of CP-nets, through which users can easily express their preferences in a qualitative way and close to natural language, rather than quantitative ones. Furthermore, the structure of CP-nets allows for each user to have any kind of order over attributes and dependencies among attributes.

Another step is making the process of collective decision making feasible in real world situations, as we should consider a large group of users who might have very different preferences over a large number of attributes. Finding a global recommended outcome based on previous research on collective decision making methods could not be feasible because, in all of these research projects, they are considering all users' preferences over all attributes. In our research, to overcome this problem, we used the idea of clustering users with similar preferences into groups and then considering their preferences only over the important attributes for that group. This leads to smaller number of comparisons over users' preferences over a smaller number of attributes, which would be more practical in real world situations.

When we have a large number of users with a large number of attributes, offering one global recommended outcome for these users as a whole and making them all satisfied with this one outcome is hard. As our evaluation results for global recommended outcomes over 20 and 40 users display(Table 4.5, Table 4.7 and Table 4.13), we can confirm that offering one global outcome for a big group of users can be quite challenging. However, the ranking results shown in Table 4.13 are very encouraging, as they demonstrate that our methods are choosing outcomes with high ranks in terms of a collective penalty scoring function.

For situations in which it might be possible to select a small number of outcomes, each of which can be recommended to a small subset of users, we also investigate the success of our cluster-specific recommendations. As the results of this research depict (Table 4.7), the highest score for cluster-specific outcomes is 71.8, which is a reasonable score in making users satisfied.

One of the contributions of our research is that in the real world with a diverse number of users mentioning their preferences over a diverse number of attributes (in some situations their preferences are totally opposite to each other), offering one global outcome by comparing all users' preferences over all attributes could not make all users as happy as possible. Therefore, we can consider the idea of offering a global recommended outcome as well as cluster-specific ones. Hence, when we need to offer one global outcome to all users, we can offer that, but in some situations in which we have the possibility of offering more than one outcome to make users as satisfied as possible, we can offer cluster-specific outcomes. Moreover, in all these processes of finding outcomes, there is not any need to compare *all* users' preferences on *all* attributes, which has not been considered in previous research.

5.3 Future Work

For further study on this research, we can explore other weighting methods, as well as extending our proposed methods to CP-nets with non-binary attributes.

As research on clustering CP-nets is so limited, investigating other methods of clustering could help to offer more satisfactory outcomes.

Also, extending our proposed methods to the more advanced TCP-nets (tradeoffs-enhanced CP-nets [9]) could be more practical in real world situations.

The results of this thesis show that the proposed methods can provide good recommendations, while reducing the size of the search space. The findings of this research would be further strengthened by also implementing some of the previous approaches in the literature (e.g., [20, 23, 25]) and doing direct performance comparisons between their methods and ours.

Doing further testing of our methods with different combinations of numbers of users and attributes, as well as different number of clusters, would also be valuable.

As mentioned in Section 4.5, investigating different ways of finding the values for the non-important attributes in cluster-specific recommendations could make the results for cluster-specific outcomes even stronger.

Finally, testing our methods in actual real world applications will help to verify the viability of the approach.

Bibliography

[1] Thomas E Allen, Judy Goldsmith, Hayden Elizabeth Justice, Nicholas Mattei, and Kayla Raines. Generating CP-nets uniformly at random. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 872–878, 2016.

[2] Kenneth J Arrow. *Social choice and individual values*, volume 12. Yale University Press, 2012.

[3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[4] Craig Boutilier, Ronen Brafman, Chris Geib, and David Poole. A constraint-based approach to preference elicitation and decision making. In *AAAI Spring Symposium on Qualitative Decision Theory*, pages 19–28, 1997.

[5] Craig Boutilier, Ronen I Brafman, Holger H Hoos, and David Poole. Reasoning with conditional ceteris paribus preference statements. In

Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, pages 71–80. Morgan Kaufmann Publishers Inc., 1999.

[6] Craig Boutilier, Ronen I Brafman, Carmel Domshlak, Holger H Hoos, and David Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res.(JAIR)*, 21:135–191, 2004.

[7] Craig Boutilier, Ronen I Brafman, Carmel Domshlak, Holger H Hoos, and David Poole. Preference-based constrained optimization with CP-nets. *Computational Intelligence*, 20(2):137–157, 2004.

[8] Ronen I Brafman and Carmel Domshlak. Introducing variable importance tradeoffs into CP-nets. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 69–76. Morgan Kaufmann Publishers Inc., 2002.

[9] Ronen I Brafman, Carmel Domshlak, and Solomon Eyal Shimony. On graphical modeling of preference and importance. *Journal of Artificial Intelligence Research*, 25:389–424, 2006.

[10] Steven J Brams, D Marc Kilgour, and William S Zwicker. The paradox of multiple elections. *Social Choice and Welfare*, 15(2):211–236, 1998.

[11] Michael C. Browne, Edmund M. Clarke, and Orna Grümberg. Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59(1-2):115–131, 1988.

[12] Alessandro Cimatti, Edmund Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. NuSMV 2: An opensource tool for symbolic model checking. In *International Conference on Computer Aided Verification*, pages 359–364. Springer, 2002.

[13] Carmel Domshlak and Ronen I Brafman. CP-nets-reasoning and consistency testing. In *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning*, pages 121–132. Morgan Kaufmann, 2002.

[14] Carmel Domshlak, Steve Prestwich, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Hard and soft constraints for reasoning about qualitative conditional preferences. *Journal of Heuristics*, 12(4-5):263–285, 2006.

[15] Gephi. About Gephi. `https://gephi.org/about/`, 2016. [Online; accessed 16-October-2016].

[16] Judy Goldsmith, Jérôme Lang, Miroslaw Truszczynski, and Nic Wilson. The computational complexity of dominance and consistency in CP-nets. *Journal of Artificial Intelligence Research*, pages 403–432, 2008.

[17] Jérôme Lang. Graphical representation of ordinal preferences: Languages and applications. In *Conceptual Structures: From Information to Intelligence*, pages 3–9. Springer, 2010.

[18] Jérôme Lang and Lirong Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical social sciences*, 57(3):304–324, 2009.

[19] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. An efficient procedure for collective decision-making with CP-nets. In *ECAI 2010: 19th European Conference on Artificial Intelligence*, volume 215, pages 375–380, 2010.

[20] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. An efficient approach for ordering outcomes and making social choices with CP-nets. In *Australasian Joint Conference on Artificial Intelligence*, pages 375–384. Springer, 2010.

[21] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. Efficient heuristic approach to dominance testing in CP-nets. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pages 353–360. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[22] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. Majority-rule-based preference aggregation on multi-attribute domains with CP-nets. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 659–666. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[23] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. Efficient penalty scor-

ing functions for group decision-making with TCP-nets. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pages 1073–1074. International Foundation for Autonomous Agents and Multiagent Systems, 2011.

[24] Minyi Li, Quoc Bao Vo, and Ryszard Kowalczyk. An efficient protocol for negotiation over combinatorial domains with incomplete information. *In Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, pages 436-444. AUAI Press*, 2012.

[25] Minyi Li, Bao Quoc Vo, and Ryszard Kowalczyk. A distributed protocol for collective decision-making in combinatorial domains. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 1117–1118. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[26] Keith Purrington and Edmund H Durfee. Making social choices from individuals' CP-nets. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 179–182. ACM, 2007.

[27] Alan P Reynolds, Graeme Richards, Beatriz de la Iglesia, and Victor J Rayward-Smith. Clustering rules: a comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms*, 5(4):475–504, 2006.

[28] Francesca Rossi, Kristen Brent Venable, and Toby Walsh. mCP nets: representing and reasoning with preferences of multiple agents. In *AAAI*, volume 4, pages 729–734, 2004.

[29] Ganesh Ram Santhanam, Samik Basu, and Vasant Honavar. Dominance testing via model checking. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 357–362, 2010.

[30] Hongbing Wang, Jie Zhang, Yangyu Tang, and Shizhi Shao. Collaborative approaches to complementing qualitative preferences of agents for effective service selection. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, pages 51–58. IEEE, 2011.

[31] Hongbing Wang, Jie Zhang, Hualan Wang, Yangyu Tang, and Guibing Guo. Service selection based on similarity measurement for conditional qualitative preference. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on*, volume 1, pages 612–619. IEEE, 2012.

[32] Wikipedia. Cluster analysis — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Cluster%20analysis&oldid=693316944`, 2016. [Online; accessed 16-January-2016].

[33] Lirong Xia, Jérôme Lang, and Mingsheng Ying. Strongly decomposable

voting rules on multiattribute domains. In *AAAI*, volume 7, pages 776–781, 2007.

[34] Charles Zaiontz. McNemar's Test. `http://www.real-statistics.com/non-parametric-tests/mcnemars-test/`, 2017 Online; accessed 19-February-2017.

Vita

Candidate's full name: Mina Joroughi
University attended: Bachelor of Information Technology Engineering, University of Tabriz , 2005 - 2009