Handling Qualitative and Quantitative Preferences with Constraints in Interactive Applications

A Thesis
Submitted to the Faculty of Graduate Studies and Research
In Partial Fulfilment of the Requirements
For the Degree of
Doctor of Philosophy (Ph.D.) of Science
In
Computer Science
University of Regina

By
Bandar Ghalib Mohammed
Regina, Saskatchewan
March, 2017

Copyright 2017: Bandar Ghalib Mohammed
UNIVERSITY OF REGINA

FACULTY OF GRADUATE STUDIES AND RESEARCH

SUPERVISORY AND EXAMINING COMMITTEE

Bandar Ghalib Mohammed, candidate for the degree of Doctor of Philosophy in Computer Science, has presented a thesis titled, Handling Qualitative and Quantitative Preferences with Constraints in Interactive Applications, in an oral examination held on February 8, 2017. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner: *Dr. Behrouz Far, University of Calgary

Co-Supervisors: Dr. Malek Mouhoub, Department of Computer Science

Dr. Samira Sadaoui, Department of Computer Science

Committee Member: Dr. Boting Yang, Department of Computer Science

Committee Member: Dr. Andrei Volodin, Department of Mathematics and Statistics

Committee Member: Dr. Lisa Fan, Department of Computer Science

Chair of Defense: Dr. Aram Teymurazyan, Department of Physics

*Via Skype
ABSTRACT

Preference elicitation is very important for interactive systems. A potential buyer typically has specific expectations in the attributes of the product he or she is interested in. While the current interactive systems allow users to provide some keywords and other information in order to filter and obtain only what they need, users feel the product they get does not necessarily meet their satisfaction. This thesis proposes a new interactive system that enables buyers to express their needs and desires. Users are given the ability to elicit their requirements and preferences in a friendly and interactive way. The system will then provide a list of suggestions satisfying user requirements and maximizing desires. Requirements and preferences are managed respectively as a set of hard constraints and soft constraints where the latter can be quantitative (numerical), qualitative (ordinal), or both. This is an optimization problem where the optimal solutions (best outcomes) are those satisfying hard constraints and maximizing user preferences. We use the C-semiring and the CP-net to represent the set of quantitative and qualitative preferences respectively. The branch and bound method is then applied in order to provide the users with a list of Pareto optimal outcomes satisfying the hard constraints and optimizing the preferences. We use approximation techniques in order to convert conditional and qualitative preferences to soft constraints. The proposed interactive system is enhanced with a component that learns from other buyer preferences and makes a set of recommendations using data mining techniques including classification, association rules, and cluster analysis.
High-fidelity prototyping, an evaluation framework, and a Volere requirements specification template are used in order to obtain user feedback on the interactive design for the proposed system. For better management of constraints and preferences, the well-known constrained CP-Net model is extended to quantitative constraints. This extended model, named weighted constrained CP-net, takes a set of constraints and preferences expressing user requirements and desires. It then returns a set of outcomes provided in the form of a list of suggestions. The results of experimental tests, conducted to evaluate the performance of the interactive system, are very promising.
ACKNOWLEDGEMENT

I would like to say that it was my pleasure to study at the University of Regina. I have met excellent people at the University who are quite knowledgeable in this area of study and I have learned many things from them. I would like to thank them for aiding me in various ways during my study. It was a pleasure to work under my supervisor Dr. Malek Mouhoub and I would like to thank him for his guidance, assistance, suggestions, advice, encouragements, and patience while I completed this research and project. It was also a pleasure to work under my co-supervisor Dr. Samira Sadaoui and I would like to thank her for her guidance, advice and encouragements while I completed my Ph.D. research. I would like to thank the supervisory committee members, the internal examiner from the University of Regina Dr. Lisa Fan, Dr. Boting Yang, Dr. Andrei Volodin, and the external examiner from the University of Calgary, Dr. Behrouz Far, for their reviews and suggestions. Finally, I would like to thank and express appreciation to the Ministry of Higher Education in the Kingdom of Saudi Arabia and the Saudi Cultural Bureau in Ottawa, Canada for everything they have done in terms of their generous financial, material and moral support.
DEDICATION

I would like to dedicate my Ph.D. Thesis to my wonderful and loving parents. To my Mom - Takwa and Dad - Ghalib, I thank them for their guidance, assistance, encouragement and beautiful advice and lessons they have given me. Even though my Dad passed away in 2010, I still remember his lessons and encouragement. My parents have set a good example that I always strive to follow. This Ph.D. Thesis is a small present that I would like to dedicate to my loving parents. I would also like to dedicate my Ph.D. Thesis to my great sisters (Kholoud and Areej) and brothers (Ibrahim, Mohammed and Majed) for their support, encouragement and generous financial, material and moral support.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>i</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENT</td>
<td>iii</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>iv</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>ix</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xii</td>
</tr>
<tr>
<td>LIST OF ALGORITHMS</td>
<td>xiii</td>
</tr>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Problem Statement and Motivations</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Proposed Solution and Contributions</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Thesis Organization</td>
<td>3</td>
</tr>
<tr>
<td>2 BACKGROUND</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Preference Reasoning</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1 Preferences</td>
<td>5</td>
</tr>
<tr>
<td>2.1.2 Qualitative Preference: Conditional Preference Networks (CP-nets)</td>
<td>6</td>
</tr>
</tbody>
</table>
LIST OF TABLES

2.1 Comparison in terms of different types of information . . . . . . . . . . 13

4.1 10 Transactions with 6 itemsets . . . . . . . . . . . . . . . . . . . . . . 33

4.2 Information and budgets for 7 users . . . . . . . . . . . . . . . . . . . . . 39

4.3 Comparison in terms of types of information . . . . . . . . . . . . . . . 43

4.4 4 Transactions with 5 itemsets . . . . . . . . . . . . . . . . . . . . . . . 46
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Software architecture of the proposed interactive system</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>An example of conditional preferences network</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Representation of qualitative and quantitative preferences</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>Soft constraint problem</td>
<td>11</td>
</tr>
<tr>
<td>3.1</td>
<td>Qualitative part of the problem</td>
<td>22</td>
</tr>
<tr>
<td>3.2</td>
<td>Quantitative part of the problem</td>
<td>22</td>
</tr>
<tr>
<td>3.3</td>
<td>The complete network for the example</td>
<td>23</td>
</tr>
<tr>
<td>3.4</td>
<td>Execution of the branch and bound algorithm on a part of the search space</td>
<td>25</td>
</tr>
<tr>
<td>3.5</td>
<td>Architecture of our online system</td>
<td>26</td>
</tr>
<tr>
<td>3.6</td>
<td>Home page of our recommender system</td>
<td>26</td>
</tr>
<tr>
<td>3.7</td>
<td>Basket page of our recommender system</td>
<td>27</td>
</tr>
<tr>
<td>3.8</td>
<td>Sign in page of our recommender system</td>
<td>28</td>
</tr>
<tr>
<td>3.9</td>
<td>User preferences</td>
<td>29</td>
</tr>
<tr>
<td>3.10</td>
<td>User optimal solution</td>
<td>30</td>
</tr>
<tr>
<td>4.1</td>
<td>Example of the Apriori algorithm</td>
<td>36</td>
</tr>
<tr>
<td>4.2</td>
<td>Example of the AprioriTid algorithm</td>
<td>38</td>
</tr>
<tr>
<td>4.3</td>
<td>Software architecture of the interactive system</td>
<td>45</td>
</tr>
<tr>
<td>4.4</td>
<td>Preference learning page</td>
<td>47</td>
</tr>
</tbody>
</table>
7.4 Running time should be inside the circle in the GUI

7.5 Running time for UCI KDD datasets with threshold: 5, 7, 10, and 15

7.6 Average running time needed to return a set of suggestions
## List of Algorithms

1. Approximation procedure ........................................... 14
2. Computing weights to variables ...................................... 15
3. Branch and bound ...................................................... 24
4. Apriori algorithm - frequent itemset generation .................. 35
5. K-means algorithm ..................................................... 41
6. PrefC method - finding Pareto set .................................... 55
7. PrefC method - assigning weights to all the tuples of the CP-Net . 55
8. PrefC method - combining qualitative and quantitative together .... 55
9. WPrefC method - finding Pareto set .................................. 62
10. WPrefC method - assigning weights to all the tuples of the CP-Net . 62
11. WPrefC method - combining qualitative and quantitative together .......... 63
Chapter 1

INTRODUCTION

1.1 Problem Statement and Motivations

Many interactive systems involve some interaction with users in order to understand and answer their needs. In order to succeed, these systems should maximize the user experience and satisfaction. Taking user preferences into consideration while he/she is looking for a given item can achieve this. This will result in making the system more popular and will earn a greater reputation among users. In order for the systems to handle user desires, they need a proper way to elicit, represent, and evaluate user preferences. Reasoning in the selection of preferences is a core concept for systems involving a relatively large space of interaction with users. Many of these systems exist in a constrained environment. For example, in an online computer store, the user cannot choose two parts that are incompatible with each other. One could then consider the situation in which constraints and preferences co-exist. In current shopping systems, shoppers may face a difficult time getting an appropriate product among a large set of possible outcomes. They may have the impression that their preferences and needs are not fully and properly taken into consideration by the system. The last decade has shown a growing number of interactive systems and e-commerce
applications. This is evident from the fact that the number of online shoppers has increased drastically. This brings many challenges to interactive systems. One of these challenges is to be able to learn about and reason buyer interests and preferences. Most existing systems gather preference information without engaging the users. In principle, most of such systems keep track of the user history and reason about the user preferences passively without engaging him/her with the process. Needless to say, such techniques are prone to inaccurate and missing preference information.

1.2 Proposed Solution and Contributions

A proposal is outlined for an interactive application that takes into account problem requirements (user constraints and system constraints) and customer preferences (quantitative and qualitative). Requirements and desires are managed in a unique model, respectively through a set of hard constraints and preferences where the latter can be quantitative. For example, one’s preference for playing Xbox could be 90 percent and playing Playstation 70 percent, qualitative (ordinal); one’s preference in laptops could be Toshiba rather than Hp. Hard constraints correspond here to both users and problem requirements and are managed with the Constraint Satisfaction Problem framework (CSP) [23]. We use the C-semiring [9] and the CP-net [13] to represent the set of quantitative and qualitative preferences respectively. The branch and bound method is then applied in order to provide the users with a list of Pareto optimal outcomes satisfying the hard constraints and optimizing the preferences [4][6][46][47]. Approximation techniques are used in order to convert conditional and qualitative preferences to soft constraints [6][28][46]. The proposed system is extended with a learning component that is able to suggest products to clients depending on data mining techniques [42]. High-fidelity prototyping, an evaluation framework, and a Volere requirements specification template are used in order to obtain user’s feedback
on the interactive design for the proposed system [40]. For a better management of constraints and preferences, the well-known constrained CP-Net model is extended to quantitative constraints and integrated into the system. This extended model, named weighted constrained CP-net, takes a set of constraints and preferences expressing user’s requirements and desires, and returns a set of outcomes provided in the form of a list of suggestions. This list is sorted according to user preferences. An experimental evaluation has been conducted in order to assess the time efficiency of the proposed system to return the list of suggestions to the user. The results show that the response time is acceptable when the number of attributes is of a manageable size. It is believed that there is no existing interactive system handling problem requirements as well as user preferences expressed in both a qualitative and a quantitative way. This system is the first attempt to cope and use both quantitative and qualitative preferences in the same interactive system. Figure [1.1] illustrates the software architecture of the interactive system. This architecture has three layers: graphic user interface (presentation layer), business logic layer, and database (data layer). The presentation layer allows the user to input the set of constraints and preferences. The business logic layer includes the following solving techniques: data mining techniques, PrefC solver, and WPrefC solver. The data layer takes care of the information storage and retrieval.

1.3 Thesis Organization

This thesis is organized into eight chapters. The following chapter provides a background on qualitative preferences, constrained CP-nets, quantitative preferences, weighted CP-nets, preference-aware interactive systems, approximation method, preference learning, and recommender systems. The preference-aware interactive system for online shopping is presented in chapter 3. This system extends into chapter 4
in order to include a recommendation component based on learning using data mining techniques. These techniques include association rules, classification, and cluster analysis. In order for the system to be more expressive, a new framework managing constraints as well as qualitative and quantitative preferences in interactive applications is proposed in chapter 5. The evaluation of the graphical user interface of the interactive system is presented in chapter 6. Also in chapter 6, the Volere requirements specification template (project drivers, project constraints, functional requirements, non-functional requirements, and project issues) for the interactive system under preferences and constraints with the design and high-fidelity is also provided. The experimentations evaluating the systems presented in chapters 3, 4, and 5 are reported in chapter 7. Finally, concluding remarks and future works are listed in chapter 8.
Chapter 2

BACKGROUND

2.1 Preference Reasoning

2.1.1 Preferences

Managing preferences is an important subject in the area of Computer Science, particularly in the area of Artificial Intelligence (AI) and in many other applications such as finance [16, 18, 67]. The generation and management of preferences are very popular as they allow customers to specify or state their choices and search for the best outcome or solution to meet their needs. For example, if a customer would like to specify their choice for a brand name of laptop or desktop computer, the customers preferences could include the size of the memory or the speed of processor or could be both. Producing or establishing a model or application to manage the preferences is a significant way to determine a decision making method [20, 56, 67]. There are several challenges when reasoning is used, particularly for preferences [18, 56]. First, customers must have flexibility or the ability to choose and signify their preferences. Second, the application must have the ability to represent different kinds of preferences. Lastly, the application has to offer the customer a simple technique or way in which to choose their preferences. When the model gives the customers a simple and
flexible technique to choose their preferences, it is called a representation of preference and the procedure for acquiring and registering the customers preferences is called elicitation of preference. The model that understands the preferences of the user and recommends products based on those preferences is called learning of preferences. The representation of preference should take into consideration the customers ability to represent their own preferences \([16,56]\). Preferences can exist in various forms, and when the model or the application gives the users the ability to choose their preferences, it is very significant in the decision making method \([20,67]\). Preferences can come in various forms \([18]\) and can be classified into two major categories; qualitative and quantitative. Quantitative preferences are also called cardinal or soft constraints. Qualitative preferences are also called ordinal or conditional preferences.

2.1.2 Qualitative Preference: Conditional Preference Networks (CP-nets)

A Conditional Preferences network (CP-net) \([13,17]\) is a graphical model to represent qualitative preferences statements including conditional preferences such as: I prefer A to B when X holds. A CP-net works by exploiting the notion of preferential independency based on the ceteris paribus (with all other things being without change) assumption. Ceteris Paribus (CP) assumption gives us a clear way to interpret the user preferences. For instance, I prefer A more than B means I prefer A more than B if there was no change in the main properties of the objects. A CP net can be represented by a directed graph where nodes represent features (or variables) along with their possible values (variables domains) and arcs represent preference independencies among features. Each variable X is associated with a ceteris paribus table (denoted as CPT(X)) expressing the order ranking over different values of X given the set of parents \(\text{Pa}(X)\). Consider the CPT represented by the figure 2.1. The user prefers 2 to 1. Also, the user prefers 3 to 4 for B when A=2. An outcome for a CP-net is an
A CP-net is a graphical model to compactly represent user conditional preferences. A CP-net with respect to a set of variables $V$ is a directed a cyclic graph (DAG) $G$ with vertex set $V$. For every variable $V_i \in V$, the user chooses a (possibly empty) set $Pa(V_i) \subseteq V \setminus \{V_i\}$ representing the direct predecessors of $V_i$. Every vertex $V_i$ is associated with a conditional preference table $cpt(V_i)$ showing, for every assignment of $Pa(V_i)$, a total order over the values of $V_i$.

![Figure 2.1: An example of conditional preferences network](image)

Figure 2.1: An example of conditional preferences network

assignment for each variable from its domain. Given a CP net, the users usually have some queries about the set of preferences represented. One of the main queries is the best outcome given the set of preferences. We say outcome $o_i$ is better than outcome $o_j$ if there is a sequence of worsening flips going from $o_i$ to $o_j$ [51]. A Worsening flip is a change in the variable value to a less preferred value according to the variables CPT. For instance; going from 2 to 1 is a worsening flip according to the $CPT(A)$ of the figure 2.1.

A CP-net is a graphical model to compactly represent user conditional preferences [13]. A CP-net with respect to a set of variables $V$ is a directed a cyclic graph (DAG) $G$ with vertex set $V$. For every variable $V_i \in V$, the user chooses a (possibly empty) set $Pa(V_i) \subseteq V \setminus \{V_i\}$ representing the direct predecessors of $V_i$. Every vertex $V_i$ is associated with a conditional preference table $cpt(V_i)$ showing, for every assignment of $Pa(V_i)$, a total order over the values of $V_i$.

![Figure 2.2: Representation of qualitative and quantitative preferences](image)

Figure 2.2: Representation of qualitative and quantitative preferences
Example 1 (CP-net)  The CP-net in Figure 2.2 defines a set of three variables \{Resolution, USB, Wifi\} each associated with its preference table. For variable Resolution, the user unconditionally prefers 1080HD to 720HD. However, for Wifi, the user preferences are conditional upon the values assigned to both Resolution and USB.

The CP-net defines a dominance relation over the set of outcomes \(O\) such that for any two outcomes \(o_i, o_j \in O\), \(o_i\) dominates \(o_j\) if and only if there is a sequence of worsening flips from \(o_i\) to \(o_j\). A worsening flip is a change in the variable value to a less preferred one where all other things being equal (ceteris paribus).

Example 2 (Dominance Relation)  Recall the running example in Figure 2.2. Let \(o_1 = (1080HD, USB2, Wifi)\) and \(o_2 = (1080HD, USB2, SuperWifi)\). Going from \(o_1\) to \(o_2\) is a worsening flip to the variable Wifi. \(o_1\) dominates \(720HD, USB3, Wifi\) because there is a sequence of improving flips from the former to the latter. In particular, the sequence is \((1080HD, USB2, Wifi), (720HD, USB2, Wifi), (720, USB3, Wifi)\).

For example, one may prefer to play football rather than hockey or one may prefer to drink Pepsi to juice when eating meat. One may prefer a brown Honda civic to a red Honda civic. This means a preference of a brown Honda civic over a red Honda civic if there is no change in the features of the object [13, 17]. Even though CP.net is a good tool to reason and signify with conditional preference, it is affected with two issues [68]. The first issue is that users cannot signify their preferences. For example, a user prefers to buy a Dell Laptop rather than an HP Laptop. In this situation, the user can only state that a Dell Laptop is preferred over the HP Laptop. The user is not able to specify to what level one preference is preferred over the other [68]. The consumer may wish to show that the Dell Laptop is highly preferred to the HP Laptop [68]. The second issue is regarding the relation significance between attributes and the dependence association in CP-nets. This only specifies that children nodes are less significant than their parent nodes [68]. This statement concludes that several patterns are not comparable [68].
2.1.3 Constrained CP-nets

Preferences usually take place in a constrained environment. Thus, solving CP-nets with the presence of constraints is an important step towards applying such models into real world applications. A Constrained CP-net \([14]\) is a tuple \((\mathcal{N}, \mathcal{C})\) where \(\mathcal{N}\) is a CP-net and \(\mathcal{C}\) is a set of constraints restricting the values that some of the CP-net variables can take. Solving a Constrained CP-net consists of finding one or more feasible outcomes (called Pareto optimal solutions) that are not dominated by any other feasible outcome.

Example 3 (Pareto Set)

Consider the CP-net in Figure 2.2 and assume \((1080\text{HD}, \text{USB2})\) is infeasible. The best outcome of this CP-net \((1080\text{HD}, \text{USB2}, \text{Wifi})\) is no longer valid according to the above constraint. Instead we have \((1080\text{HD}, \text{USB3}, \text{SuperWifi})\) and \((720\text{HD}, \text{USB2}, \text{Wifi})\) as the Pareto set.

In \([14]\) a method, called Search-CP, has been proposed to solve the constrained CP-net problem. Search-CP is a backtracking search algorithm that assigns the variables to values from their domains, in a topological order. Following the semantics of CP-nets, the first solution generated by Search-CP is guaranteed to be a Pareto optimal solution. However, when looking for more than one solution, a dominance test needs to be performed with earlier solutions every time a feasible solution is found. In order to improve its time performance in practice, Search-CP has been updated with constraint propagation and the most constrained variable ordering heuristic \([5]\).
2.1.4 Quantitative Preference: Constraint Satisfaction and Soft Constraints

A Constraint Satisfaction Problem (CSP) is a well-known framework for constraint problems. More formally, a CSP consists of a set of variables each defined on a set of possible values (variable domain) and a set of relations (or hard constraints) restricting the values that each variable can take. A solution to a CSP is a complete assignment of values to variables such that all constraints are satisfied. Unlike hard constraints, soft constraints are associated with degrees of satisfaction and the goal is to find the optimal outcome or solution. In Soft CSPs (SCSPs), an optimization problem is investigated where the optimal solution is one with the best objective function according to the soft constraints. An SCSP is a generalization of the classical CSP where constraints have several levels of satisfiability that are totally or partially ordered according to the C-semiring structure. The C-semiring (constraint semiring) is a mathematical model based on the semiring formalism with which to handle different soft constraint problems and extensions. Using the C-semiring, different constraint problems in a unified framework can be represented. More formally, a C-semiring is a tuple \((A,+,\cdot,0,1)\) where \(A\) is a set and 0, 1 are elements of \(A\). + is the additional operation defined over a set of elements of \(A\) and 0 is the unit element. \(\cdot\) is the multiplication operation and 1 is the unit element and 0 is the absorbing element. + is an idempotent operation. That is \(a+a=a\) and this can be used to find partial order over \(A\). The partial order over \(A\) can be found as \(a < b\) iff \(b\) is better than \(a\) (that is, \(a+b=b\)). Using the C-semiring definition, different CSP extensions can be represented in a unified framework by giving different semantics to the addition and the multiplication operations. A Weighted CSP (WCSP) is an instance of the C-semiring where the associated value for each tuple represents the cost of that tuple in the final solution. Therefore, the optimal solution in a WCSP is the solution where the associated value is the mini-
minimum and the ultimate goal is to minimize the global cost of the problem: \( WCSP = (\mathbb{R}^+, \min, +, +\infty, 0) \). In adding two constraints, a WCSP takes the constraint with the minimum tuple value and the total cost could become more than 1. Figure 2.3 shows a SCSP (where numeric values can be interpreted as costs) involving three variables \( A, B \) and \( C \) with two soft constraints \((A, B)\) and \((B, C)\).

## 2.2 Weighted CP-nets

WCP-nets is a weighted extension to CP-nets by targeting to resolve the issues of CP-nets and permitting users to state their preferences in a hard-grained way [68]. More precisely, user preferences can be defined at multiple stages scales: level 1 - two values are equally preferred, level 2 - first value is slightly preferred to the second value, level 3 - first value is highly preferred to the second value, level 4 - first value is very highly preferred to the second value, level 5 - first value is extremely significant preferred to the second value [68, 71]. As an example, an attribute value, Dell Laptop, can be preferred to another attribute value, HP Laptop, equally at level 1, or at level 3, or at the maximum level 5. The relation significance (weight) among attributes can be further defined by showing the range to which an attribute is more significant than another attribute [68, 71].
2.3 Preference-aware Interactive Systems

The diversity of preference-aware interactive applications includes recommender systems, such as Amazon.com and Netflix which recommend items based on customer similarity to other customers or on past viewed items and conversational systems that interact with the customer in an easy dialogue to execute a job. Here, the user interfaces are accustomed to customer preferences and state, and it predicts customer needs and desires [50]. In this regard, in [33], the authors propose an interest-based offer evaluation system for semantic matchmaking. This system recommends the best offer by ranking the offers (that match user requests) according to user interests and preferences.

Handling preferences plays an important role in many interactive applications. In [50], a discussion of interactive system challenges has been surveyed. They broadly examined different available interactive systems along with their limitations. SmartClient [53] is a planning tool to manage traveler activities. SmartClient provides a list of suggestions by using example critiques to adapt to user satisfaction. It uses the CSP [23] as a representation for the preference part of the problem. Teaching Salesman [63] is another interactive system which aids the client to compare diverse goods depending on how he/she matches their desires and needs. Preferences are represented through utility function. In [30] a preference elicitation model has been proposed for the winner determination problem in Multi-Attribute Reverse Auction (MARA). This model integrates both the Multi-Attribute Utility Theory (MAUT) and constrained CP-nets in order to handle quantitative and quantitative constraints respectively.

Furthermore, in [49] the authors propose the PExA system. This is an example of a personal agent that describes activities and knowledge of task forms. The agent can deduce the situation of a collective customer system of jobs and proactively provide some suggestions. In PExA, the preferences are expressed as an instance
of a soft constraint satisfaction problem. However, the problem of handling both qualitative and quantitative preferences had not been addressed in previous work. All the aforementioned work is based on the quantitative format of the preferences (i.e. preferences expressed as means of numbers). They differ in the way these quantitative preferences are represented. For instance, PExA uses soft constraints while SmartClient uses utility functions. This motivates the proposal of a new system for handling both quantitative and qualitative preferences in a friendly manner so the user can express his desires in a qualitative or quantitative form, or both. Table 2.1 lists some interactive systems along with their expressiveness in handling constraints and different forms of preferences.

Table 2.1: Comparison in terms of different types of information

<table>
<thead>
<tr>
<th>Application</th>
<th>Quantitative</th>
<th>Qualitative</th>
<th>Mixed</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Netflix</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SmartClient</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Teaching Salesman</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Project Execution Assistant</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Proposed Interactive system</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

2.4 Approximation Method

When given a CP-net N, it can be approximated to a Soft Constraint Satisfaction Problem (SCSP) [27][28]. This approximation is favourable in many domains, specifically those domains where there is heavy interaction with the user as in online configuration problems [27]. Another advantage of approximating a CP-net to an SCSP is to overcome the complexity of reasoning with CP-nets. Approximation aims to convert the preference statements in the CP-net to soft constraints and reflect the
ordering for each statement by giving a higher value to the preferred instantiations for the tuples. Test dominancy (a given outcome better than another) is known to be computationally difficult in CP nets. However, an SCSP provides a linear time in response to dominance testing between two outcomes [11,27]. The main characteristic with which to distinguish different approximations for a particular CP-net is the information preserving property [52]. Information preserving simply means what is already preferred in the CP-net must also be preferred in the approximated SCSP. The process of approximation can be completed in two steps by creating the constraint graph and by calculating value preferences for each variable in the constraint graph. Algorithm 1 [28] shows the steps to create an SCSP graph with a set of hard and soft constraints and Algorithm 2 [28] assigns weights to each variable in the constraint network. One final step to be completed in order to make the approximated SCSP compatible with the C-semiring framework, is to reflect the weights of the variables to the constraint tuples. This can simply be done by multiplying the weight for each variable X to the set of constraints where X is mentioned.

**Algorithm 1** Approximation procedure

Let $N$ be a cyclic CP-net
Let $S$ be SCSP

**BEGIN**

/*create nodes*/

for each $X$ in $N$
    create node $X$ in $S$

for each $X$ where $Pa(X) > 1$
    create node $Pa(X)$ in $S$

/*create arcs*/

for each $X$ in $N$
    if $Pa(X) = 0$
        create soft unary constraint over $X$
    if $Pa(X) = 1$
        create soft constraint($Pa(X),X$)
    if $Pa(X) > 1$
        create hard constraint($A,X$); $\forall A \in Pa(X)$

**END**
Algorithm 2 Computing weights to variables

Let $S$ be SCSP

for each $X \in S$

if $Children(X) = \emptyset$

then $W_X = 1$

else

$W_X = \sum W_y \cdot |dom(y)| \, \forall y \in Children(X)$

Note that the approximation process approximates the entire CP-net to SCSP. This can lead to further computational time and the loss of precision [51]. However, in many online systems where the user directly interacts with the system, it is sufficient to use the approximation technique because users usually do not have the patience to wait for their perfect outcome. Rather, they usually accept the outcome which is closest to their optimal, even though it has lost some precision. In addition to converting the entire CP-net, the approximation technique functions solely in a cyclic CP-net. This can be beneficial in some cases where the problem is a more constrained problem than a preferential problem. This is critical because if the problem has a hundred variables in the constraint problem and only two variables in the CP-net, the complexity of the dominance testing in the CP-net can simply be ignored and converted into an SCSP. By this, even if some precision were lost, the complexity time for the entire problem was reduced. The approximation process is one way of handling qualitative preferences with constraints. This process deals with preferences and constraints via a single framework. That is, constraints and preferences are described via the same model or language (coupled) [15, 26, 69]. However, there is no such thing as forcing the preferences and constraints to be coupled.

The steps for converting a cyclic CP-net into WCSP are summarized as follows:

- Construct the graph of nodes.
- Create arcs or constraints.
- Calculate the penalties.
• Calculate the weights for each variable.

• Assign or replace the weights to tuples rather than the penalties or constraints which mean multiple weights with the penalties and replace the result instead of or rather than the penalties.

2.5 Preference Learning

The preference learning work focused mainly on rankings over individual items and pairwise preferences [29]. It has gained significant attention in the area of artificial intelligence, especially in the last few years. It involves knowledge and learning from explanations that expose information and knowledge about the preferences of a person or group of people [29]. Processing knowledge representing preferences permits anyone to state desires in a declarative method to have qualitative and quantitative forms of reasoning. It deals with exceptions and variations in a flexible manner. This leads to the creation and implementation of models to learn the preferences of users, and might be utilized for prediction of preferences [29]. The idea of preference learning is very easy to understand but hard to manage and implement. Every problem of learning preferences depends on the number of dimensions used to show the set of potential preferences [22, 25]. For instance, when the user would like to purchase a laptop or desktop computer, and the only option considered is budget, the recommendations offered to the user are going to be easy to provide since there is just one dimension. If there are more options, such as the size of the memory or the speed of the processor, the problem becomes harder in terms of managing user preferences in an effective way. A method is needed to consider all the options and generate a list of recommendations to the user. Feedback is required from the user in order to gain information about user preferences to model the preferences by utilizing a particular representation. The information can be processed with machine learning techniques in order to learn user
preferences for use in the recommendation procedure [22,25].

2.6 Recommender Systems

The appearance of several web applications has matched with growing acknowledgment that several of the works people achieve with computers can be executed better when the application becomes accustomed to its user [2]. There are many examples of these web applications, for instance, internet media and electronic commerce. Netflix is one of the popular examples offering customized movie suggestions to users via precedent movie ratings [2]. Netflix, Amazon.com and many preference-aware interactive systems share the common target of helping the customer find a product by choice [2]. These preference-aware interactive applications recommend items based on customer similarity or on customer past viewed items. These conversational systems interact with the customer in an easy dialogue or conversation to execute a job [2]. Their interfaces are becoming accustomed to customer preferences and states [2]. Given these features, recommender systems have changed from originality and creativity used by a small number of E-commerce websites into serious and significant commerce tools that are re-shaping the earth of e-commerce [2]. As well, more and more major e-commerce websites are using recommender systems to assist consumers to find goods or products [2]. Recommender systems are usually used in electronic stores or shops in order to advise on related or similar goods or products [2]. The majority of recommender systems have used the technique of collaborative filtering in order to offer personalized information [2]. This technique is extremely efficient, capable, well-organized, proficient, suitable, and convenient for achieving personalized information [2]. We no longer need to bring in semantic information about the goods or manually tag or link goods and customers [2]. There are three ways which recommender applications improve E-commerce sales [60]:
1. **Browsing websites by buyers:** usually, people who visit websites browse without buying anything. Recommender applications can assist clients find goods that they would like to purchase [60].

2. **Cross-sell:** recommender applications develop cross-sell by recommending additional goods for the client to buy. If the suggestions are excellent, the standard order size should increase. For example, a website may suggest additional goods in the checkout procedure, based on those goods in the shopping cart [60].

3. **Loyalty:** recommender applications develop loyalty by producing a value-added association between the website and the client [60].

There are three groups of recommender systems. These three groups can be based on two things. First, how these recommendations are made, and second the method used to produce the recommendations [2, 7, 38]:

1. **Content-based recommendations:** in this feature, the customer is suggested items or products similar to the ones that the customer preferred previously [2].

2. **Collaborative recommendations:** in this feature, the customer is suggested products which people with related preferences and tastes liked previously [2].

3. **Hybrid approaches:** in this feature, these techniques are combined between collaborative and content-based methods [2].

### 2.7 Conclusion

Managing preferences is an important subject in the area of computer science, particularly in the area of Artificial Intelligence (AI) and in many other applications such as finance [16, 18, 67]. The generation and management of preferences are very popular as they allow customers to specify or state their choices and search for the
best outcome or solution to meet their needs. A Conditional Preferences network (CP-net) \[13, 17\] is a graphical model to represent qualitative preference statements including conditional preferences such as: I prefer A to B when X holds. Preferences usually take place in a constrained environment. Solving CP-nets with the presence of constraints is an important step towards applying such models in real world applications. A Constraint Satisfaction Problem (CSP) \[23\] is a well-known framework for constraint problems. More formally, a CSP consists of a set of variables each defined on a set of possible values (variable domain) and a set of relations (or hard constraints) restricting the values that each variable can take. A solution to a CSP is a complete assignment of values to variables such that all the constraints are satisfied. Unlike hard constraints, soft constraints are associated with degrees of satisfaction \[11\] and the goal is to find the optimal outcome or solution. WCP-nets is a weighted extension to CP-nets that target to resolve the issues of CP-nets and permit users to state their preferences in a hard-grained way \[68\]. The diversity of preference-aware interactive applications includes recommender systems, such as Amazon.com and Netflix. These systems recommend items based on customer similarity or on past viewed items and conversational systems that interact with the customer in an easy dialogue to execute a job. The work on preference learning is mostly focused on rankings over individual items and pairwise preferences \[29\]. It has gained significant attention in the area of artificial intelligence, especially in the last few years. It involves knowledge and learning from explanations that expose information and knowledge about the preferences of a person or group of people \[29\]. The appearance of several web applications has matched with growing acknowledgment that several of the works people achieve with computers can be executed better when the application becomes accustomed to its user \[2\].
Chapter 3

A PREFERENCE-AWARE
INTERACTIVE SYSTEM FOR
ONLINE SHOPPING

3.1 Introduction

Constraints and preferences coexist in a wide variety of real world applications. In a previous work we have proposed a preference-based interactive system that handles both constraints as well as preferences where these latter can be in a qualitative or a quantitative form. Given online shoppers’ requirements and preferences, the proposed system provides a set of suggested products meeting the users’ needs and desires. This is an improvement to the current shopping websites where the clients are restricted to choose among a set of alternatives and not necessarily those meeting their needs and satisfaction. The rest of the chapter is structured as follows. The next section provides an overview about preference-aware interactive systems. In section 3, CP Nets, soft constraints and the approximation process as described in [27] are discussed. Our proposed system is then presented in Sections 4 and 5. Experimental
results are shown in section 6. Finally, in Section 7 we list some concluding remarks and promising future works.

3.2 The Proposed Model

In this chapter we propose a new online interactive system that takes user preferences into account when looking for the optimal outcome. More precisely, the problem consists of deciding which item is the most preferred given the user preferences in an online shopping website. The problem of co-existence between problem requirements and user preferences is studied assuming the preferences can be expressed as a quantitative (numerical) or qualitative (ordinal) or both. The purpose for developing this interactive system is to give the buyer more flexibility to express his preferences. Our system is presented through the problem of choosing an optimal computer according to a set of quantitative and qualitative preferences as well as hard constraints. It is assumed that the customer is capable to state his preferences quantitatively or qualitatively. The hard constraints can be global or local. The global constraint can be the user’s budget (maximum price) while local constraints can express the incompatibility between different components of the computer. Users have the ability to express their preferences in qualitative and quantitative manner. The set of qualitative preferences are approximated to the SCSP to overcome the computational complexity associated with the CP-net. As a result, the set of quantitative preferences along with the approximated SCSP form the SCSP network. The latter is the result of the approximation process in addition to the quantitative preferences. In combining preference information the whole network for the quantitative preferences and the approximated CP-net will be generated. Finally, the problem can be solved like any other WCSP problem where the optimal solution is the solution whose associated value is minimal.
3.2.1 Example

We consider the PC configuration problem as an example for our system. Assume we have five variables constitute a valid PC configuration. These variables are: Memory, Processor, Hard Drive, Screen and Graphic Card. We refer to each variable by its initials for simplicity purposes. Let the domain for each variable be as follow: $D_M = \{1, 2\}$, $D_P = \{2, 2.3\}$, $D_{HD} = \{80, 120\}$, $D_S = \{13, 15\}$ and $D_{GC} = \{I, A\}$. Now the user login to the system and expresses his preferences. The user has the flexibility to express his preferences quantitatively or qualitatively. For instance, the user always prefers to have 1 GB to 2 GB for the memory. Also, he prefers processor with 2 GHz if the memory 1 GB is preferred than 2.3 GHz. For the hard drive options, he prefers hard drives with larger capacity. These information is transformed into CP-net as showin in Figure 3.1. For other variables, the user specified his preferences quantitatively as described in Figure 3.2.

![Figure 3.1: Qualitative part of the problem](image)

![Figure 3.2: Quantitative part of the problem](image)

When the user ends specifying his preferences, the system transforms the qualitative preferences into the WCSP instance following the approximation procedure.
The result is then combined with the existing quantitative information to form the final network. The latter is a WCSP instance as shown in Figure 3.3. In addition to the information shown in Figure 3.3, variable \( M \) has a soft unary constraint with the following two tuples: \((1, 0)\) and \((2, 2)\), and the \( HD \) variable has a soft unary constraint with the following two tuples: \((120, 0)\) and \((80, 1)\). Since the final result is a WCSP instance, we adapt the branch and bound algorithm to systematically search for the optimal outcomes in the network.

\[
\begin{align*}
(80, 13, 1.0) & \quad (13, I, 4.0) \\
(80, 15, 4.0) & \quad (13, A, 2.0) \\
(120, 13, 6.0) & \quad (15, I, 7.0) \\
(120, 15, 8.0) & \quad (15, A, 8.0)
\end{align*}
\]

\( \text{HD} \) \quad \( S \) \quad \( GC \)

\[
\begin{align*}
(1, 2, 0) & \\
(1, 2.3, 1) & \\
(2, 2, 1) & \\
(2, 2.3, 0) & \quad \text{M}
\end{align*}
\]

Figure 3.3: The complete network for the example

### 3.2.2 Proposed Solving Methods

The branch and bound algorithm is a systematic search method for discovering the best solutions to optimization problems \cite{23}. We have used the implementation of branch and bound as shown in Algorithm \texttt{3}. We also show the execution of branch and bound when applied to the example we described earlier. Note that only a part of the search space is shown here. This algorithm pertains to the Quantitative and Qualitative pages. It considers the problem of choosing an optimal laptop according to a set of quantitative preferences in addition to a set of hard constraints. It is assumed that the user is able to specify her/his preferences quantitatively as shown in Figure 3.4 with a budget of $900. Therefore, the system is constrained to choose the laptops that are less than or equal to $900, and lists all the results as shown at
the bottom of Figure 3.4. The budget is considered here as a global hard constraint.

Algorithm 3 Branch and bound
1. Let $S$ be current assignment
2. Let $V$ be current variable
3. if $S$ is solution
4. if $S\text{.objective} < $ objective
   objective=$S\text{.objective}$
   bestSolution=$S$
else
5. List next=$V\text{.children}$
6. for each current $\in$ next
   if current is consistent
   if current.objective $<$ objective
      Branch and Bound(current)

3.3 System Architecture

We present here the implementation of our system described in the previous section. To make the process easier from client perspective, we have designed a friendly graphic user interface with which the user expresses his preferences quantitatively or qualitatively. The system will then convert these information into their corresponding representations (i.e. C-semiring and CP-net). The branch and bound algorithm we describe in the previous section is then applied to return the best outcomes to the user. Figure 3.5 shows the different components of our system. After the list of results suggested to the user are displayed, this latter can save them in his/her basket for further use. Figures 3.6 3.7 3.8 represent the homepage, basket page and sign
Figure 3.4: Execution of the branch and bound algorithm on a part of the search space

in page for our recommender system.

Figure 3.9 shows a snapshot of the user interaction with the system. The system gives the user the ability to specify his preferences in different forms. Also, the system allows the user to specify hard constraints over potential outcomes. In our system, the user has a complete control over preferences and constraints.

Considering the previous example in Section 3.2.1, Figure 3.10 shows more than one optimal solution to the customer, the first optimal solution for this network is Memory =1, Processor=2.0, Hard Drive=80, Screen=15 Inch and Graphic Card=Intel with 4 as the total cost or preference, and the price is $700.
Figure 3.5: Architecture of our online system

Figure 3.6: Home page of our recommender system

3.4 Conclusion

In this chapter, an interactive system based on user preferences and constraints has been proposed. The interactive system considered the case where user preferences can take qualitative, quantitative or both forms. C-semiring and the CP-net have
been used to represent the set of quantitative and qualitative preferences respectively. The user specifies his/her preferences then the system looks for the optimal outcome satisfying the set of requirements expressed as hard constraints and optimizing the user preferences. Here we focused upon the interactive system, however, our work can be generalized to any interactive system where the user is involved in the process of choosing an item among a set of others. Another future work is to manage preferences in the presence of dynamic hard constraints. This can be the case where the user can interact with the system by adding or removing some constraints and see the effect of these changes on the solutions returned. In this case, it can be used a dynamic variant of the constraint solving techniques \cite{44, 47}. In order to improve the time efficiency of our branch and bound technique, it will investigate several variable and value ordering heuristics \cite{30} that allow the branch and bound to return the optimal solution in a better running time. It will also explore other techniques based on parallel genetic algorithms \cite{1} and ant colony optimization \cite{48}. While these
techniques do not guarantee the best outcome, they are in general very efficient in terms of response time needed to reach the optimal (or near to optimal) solution.
Figure 3.9: User preferences
Figure 3.10: User optimal solution
Chapter 4

DATA MINING TECHNIQUES
AND PREFERENCE LEARNING
IN RECOMMENDER SYSTEMS

4.1 Data Mining Techniques

Data mining is the field of extracting valuable information and knowledge from large amounts of data stored in databases. It is the process of finding out formerly unknown, useful and valuable patterns from a large amount of data stored in a database \cite{31,36,64}. Data mining deals with the data stored in a database administration scheme/system. The tools and techniques for data mining identify business trends which may occur in the future. It also answers many questions of businesses with regard to time consumption for decision making \cite{36}. There are two significant reasons why data mining has attracted and gained a lot of attention in the last few years \cite{36,64}. It has the capability to store and collect a large amount of data while this storage quickly increases every day. As a result of improvements in processing power, there is the potential to store a large amount of relevant data which can be
processed anytime. The most significant reason is the need to transform data into useful and valuable knowledge and information \[31, 36, 64\]. Data mining examines databases in order to discover hidden patterns and valuable information that sometimes experts may not observe as it occurs outside their expectations \[36, 64\]. The discovered patterns are accessible to the user and could be stored as new information in the information database \[31, 34, 36\].

4.1.1 Data Mining Association Rules

Association rule mining \[31, 36, 64\] is a data mining task for finding and discovering hidden associations between items in a transaction. It is a well-known technique to find and discover interesting and attractive relationships between variables and items in large databases \[31, 36\]. This method relies on the extraction of an association rule with algorithmic techniques such as the FP-tree, Apriori and AprioriTid algorithms to obtain and generate the appropriate association rules between items in a transaction \[31, 34, 36\]. More precisely, it is based on association rule evolution by utilizing different measures such as support and confidence factors. Support \((s)\) defines how frequently a rule is appropriate, and applicable to a particular data set, whereas confidence \((c)\) defines how often items in set B appear in transactions containing set A \[64\]. The next two equations are the formal definitions for support \((s)\) and confidence \((c)\) \[64\]:

\[
\begin{align*}
    s(A \rightarrow B) &= \sigma(A \cup B) / T \\
    c(A \rightarrow B) &= \sigma(A \cup B) / \sigma(A)
\end{align*}
\]
where \( s \) is a support and \( c \) is a confidence. \( A \) and \( B \) are sets, and \( T \) is a transaction. 

\( \sigma \) is the support and confidence count and \( \sigma(A) \) is the union count of \( A \):

\[
\sigma(A) = |\{t_i : A \text{ is a subset of } t_i \text{ and } t_i \in T\}|
\]

(4.3)

Association rules are utilized in several areas, such as medical diagnosis and research, website navigation analysis, churn analysis and prevention, market basket analysis, and retail data analysis [31, 34, 36]. A classic example is the market basket analysis where retailers identify and analyze what customers would like or prefer to purchase to find an association between items that customers have purchased. Retailers can identify frequent items between customers to aid and assist them in order to plan diverse item placement, advertising and inventory administration [31, 36, 64]. There are many algorithmic techniques used for association rule mining. The most popular are the Apriori, AprioriTid, Partition, FP growth, and Eclat algorithms [32].

Table 4.1 shows an example of 10 transactions with 6 itemsets (Dell, Apple, Samsung, Sony, LG, Toshiba). In this example, the method for computing and calculating the support \((s)\), and confidence \((c)\) from 10 transactions with 6 itemsets is shown. As mentioned in the previous section, support \((s)\) defines how frequently a
rule is appropriate and applicable to a particular data set, whereas confidence \((c)\) defines how often items in \(Y\) appear in transactions which contain \(X\) \cite{64}. For example, \(\{\text{Apple, Samsung}\} \rightarrow \{\text{LG}\}\), where \(X = \{\text{Apple, Samsung}\}\) AND \(Y = \{\text{LG}\}\). The number of times the subsets \(\{\text{LG, Apple}\}\) and the subset \(\{\text{Samsung}\}\) appear in 10 transactions is determined to calculate the support \((s)\) for the following rule: \(\{\text{Apple, Samsung}\} \rightarrow \{\text{LG}\}\). The number of times \(X \cup Y\) \{Apple, Samsung, LG\} appear together in the transactions table is divided by \(T\) which is the total number of transactions \((10)\) in the example.

\[
S = \frac{(X \cup Y)}{|T|} = \frac{((\text{Apple, Samsung, LG}))}{|T|} = \frac{3}{10} = 0.3 \tag{4.4}
\]

Once the support is calculated, the confidence \((c)\) is calculated for the following rule: \(\{\text{Apple, Samsung}\} \rightarrow \{\text{LG}\}\). The number of times \(X \cup Y\{\text{Apple, Samsung, LG}\}\) appear together in the transactions table divided by the number of time the itemsets of \(X\) \{Apple, Samsung\} appear together in the transactions table.

\[
C = \frac{(X \cup Y)}{X} = \frac{((\text{Apple, Samsung, LG}))}{3} = 1 \tag{4.5}
\]

The Apriori algorithm \cite{34,36} is one of the most significant and classic algorithmic techniques for learning data mining association rules. The main purpose of the Apriori algorithm is to discover associations between different data sets. The Apriori algorithm is implemented for databases that have transactions such as a list of products or items that customers or buyers have purchased from any store or supermarket \cite{70}. The main problem with the Apriori algorithm is that it exchanges information with the database in order to compute the number of transactions for each item to occur, lowering its efficiency \cite{39,70}. The best solution to this issue is to reduce the number of itemsets in the transactions, and establish a better way to exchange information with the database in order to compute the number of transactions for
each item to occur \[39,70\]. Algorithm \[4\] which implements the Apriori algorithm, to
generate frequent itemsets \[64\] is presented below. \(C_K\) denotes the set of candidate
K-itemsets and \(F_K\) indicates the set of frequent K-itemsets. The algorithm makes a
single pass over the dataset to define the support for each item. Once the first step
is completed, the set of all frequent 1-itemsets and \(F_1\) will be recognized. Then the
algorithm will iterate in order to produce, and create, a new candidate K-itemsets
using the frequent (K-1)-itemsets found in the previous step. In the following ex-

**Algorithm 4 Apriori algorithm - frequent itemset generation**

BEGIN

\(k = 1\)

\(F_K = \{i|i \in I \land \sigma(\{i\}) \geq N \times minsup\}. \) \{Find all frequent 1-itemsets\}

repeat

\(k = k + 1\)

\(C_k = \text{apriori-gen}(F_{k-1}). \) \{Generate candidate itemsets\}

for each transaction \(t \in T\) do

\(C_t = \text{subset}(C_k, t). \) \{Identify all candidates that belong to \(t\)\}

for each candidate itemset \(c \in C_t\) do

\(\sigma(c) = \sigma(c) + 1.\) \{Increment support count\}

end for

end for

\(F_K = \{c|c \in C_k \land \sigma(c) \geq N \times minsup\}. \) \{Extract the frequent k-itemsets\}

until \(F_k = \emptyset\)

Result = \(\cup F_k = \emptyset\)

END

ample, the frequent itemsets will be found using the aforementioned algorithm. The
following steps, including the list of transactions, are visualized in Figure 4.1. An
item or itemset is frequent if it occurs at least twice. This is the general rule in this
example.

- Step 1: Calculate the number of transactions for each item to occur.
- Step 2: Remove all the items that occur less than twice (the items are identified
  in red in Step 1 of Figure 2).
- Step 3: Generate all the possible pairs of itemsets in Step 2.
- Step 4: Calculate how many times each pair occurs in the list of transactions. For instance, A, L occur together in A, L, SO and A, SA, L, D, so they occur twice in this example.

- Step 5: Remove all the item pairs appearing less than twice (identified in red in Step 4).

- Step 6: Produce a set of three distinct items from the table in step 5.

- Step 7: From step 6, calculate the number of transactions for SA, L, D. Therefore, the set of three items that are frequently together are SA, L, D.

**Figure 4.1: Example of the Apriori algorithm**

The AprioriTid algorithm improves upon the classic Apriori algorithm for learning
data mining association rules. The main objective of the AprioriTid algorithm is to
discover associations among different sets of data \[39, 70\]. A component is added
to improve the efficiency of the AprioriTid algorithm. A database is not used to
compute the support or the number of transactions for each item to occur and it is
only used once in the first pass \[3, 36, 70\]. Instead of using the database every time,
it utilizes storage \(C_k\). This storage is used instead of the database to compute the
support each time \[3, 39\]. \(C_k\) is \{TID, set of Itemsets\}, where TID is the transaction
number associated with the item set or candidate. \(C_k\) is the set of itemsets, and \(L_k\)
is the set of large K-itemsets, and it has two columns (itemsets and support count).
Overall, \(C_k\) provides an advantage for the in the AprioriTid algorithm as it is a more
efficient technique to avoid scanning the database \[32, 39\]. However, the disadvantage
of the AprioriTid algorithm is that a similar item will appear in several candidate
itemsets for the transaction in storage \(C_k\) and it will be repeatedly stored, increasing
the range of query data \[32, 39\]. The AprioTid algorithm is implemented via the
following steps (see also Figure 4.2): An item or itemset is frequent if it occurs at
least twice. This is the general rule in this example. this example.

- Step 1: Create and build a set of \{TID, set of Itemsets\} for the first item set
  \(C_1\).

- Step 2: Define the large first itemsets in \(L_1\) with a minimum support = 2 (less
  than two occurrences) with two columns (itemsets and support count).

- Step 3: Initiate the creation of pairs from the first items (Apple). For instance,
  \{Apple, Samsung\}, \{Apple, LG\}, Apple, Dell and repeat the operation for the
  second item (Samsung). Produce the pairs and get the outcomes in \(C_2\).

- Step 4: Check the itemsets in \(C_2\) with storage \(C_1\) in order to define \(C_2\) \{TID,
  set of Itemsets\} for the second round or scan.
• Step 5: Start the second round or scan in \( \overline{C_2} \) to calculate how often each pair occurs together to define the large second itemsets in \( L_2 \) with minimum support = 2 (less than two occurrences) with two columns (itemsets and support count).

• Step 6: Define \( C_3 \) and produce a set of three items from \( L_2 \) in order to define \( \overline{C_3} \) {TID, set of Itemsets} for the third round or scan. Add one more rule, which is two pairs with a similar first Alphabet. Therefore, \{Samsung, LG\} and \{Samsung, Dell\} have the same first Alphabet which produces \{Samsung, LG, Dell\}.

• Step 7: The third round or scan is started in \( \overline{C_3} \) to calculate the number of transactions for \{Samsung, LG, Dell\} in order to define the large second itemsets in \( L_3 \) with minimum support = 2 (less than two occurrences) with two columns (itemsets and support count). Therefore, the set of three items that are frequently together are \{Samsung, LG, Dell\}.

![Diagram](image)

**Figure 4.2:** Example of the AprioriTid algorithm
4.1.2 Data Mining Classification Technique

The general definition of a classification technique is the operation of assigning items to single of many predefined types [64]. Also, it is a general problem which includes many applications [64]. There are many methods and techniques for solving a classification problem [64]. A classification method, also called a classifier, is a methodical technique for constructing classification models from an input data set [64]. Examples of such techniques include Rough set, support vector machines, rule-based classifiers, naive Bayes classifiers, decision tree classifiers, and neural networks [64]. Every technique for solving a classification problem needs to employ a learning algorithm in order to identify a model which can best fit the association between the class label for inserting data and the attribute set [64]. The main objective of the learning algorithm is to construct models with sufficient capability to generalize [64].

A data mining classification technique is applied to the recommender system in order to classify the users as groups based on their budgets. Seven users have stated their budgets in the column Budget of Table 4.2. The users are grouped according to their budgets, and once the recommender system groups the users, it recommends products and items based on their budgets.

Table 4.2: Information and budgets for 7 users

<table>
<thead>
<tr>
<th>TID</th>
<th>First Name</th>
<th>Last Name</th>
<th>Email</th>
<th>Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Allan</td>
<td>Cooper</td>
<td><a href="mailto:a@a.com">a@a.com</a></td>
<td>&gt; 500</td>
</tr>
<tr>
<td>2</td>
<td>John</td>
<td>Smith</td>
<td><a href="mailto:j@j.com">j@j.com</a></td>
<td>&gt; 1500</td>
</tr>
<tr>
<td>3</td>
<td>Zhang</td>
<td>Ming</td>
<td><a href="mailto:z@z.com">z@z.com</a></td>
<td>&gt; 1000</td>
</tr>
<tr>
<td>4</td>
<td>Eisa</td>
<td>Ayed</td>
<td><a href="mailto:e@e.com">e@e.com</a></td>
<td>&gt; 1500</td>
</tr>
<tr>
<td>5</td>
<td>Yao</td>
<td>Hua</td>
<td><a href="mailto:y@y.com">y@y.com</a></td>
<td>&gt; 500</td>
</tr>
<tr>
<td>6</td>
<td>Jonathan</td>
<td>William</td>
<td><a href="mailto:w@w.com">w@w.com</a></td>
<td>&gt; 1000</td>
</tr>
<tr>
<td>7</td>
<td>Bandar</td>
<td>Ghalib</td>
<td><a href="mailto:b@b.com">b@b.com</a></td>
<td>&gt; 500</td>
</tr>
</tbody>
</table>
4.1.3 Data Mining Cluster Analysis Technique

The cluster analysis technique \cite{64} splits data into groups or clusters which are useful or expressive. The cluster analysis technique groups data items according to information which has been found in the data that define the items and their relationships \cite{64}. The main goal of the cluster analysis technique is to separate the clustering \cite{64}. Nevertheless, there are some cases where the cluster analysis technique is only useful for obtaining a result for other purposes; for instance, data summarization \cite{64}.

Cluster analysis techniques have played a significant part in a wide variety of fields including machine learning, information retrieval, pattern recognition, statistics, psychology, biology, data mining, and social science \cite{64}. There are several techniques for cluster analysis data mining including K-means and Agglomerative Hierarchical Clustering \cite{64}.

A cluster analysis technique can be applied to a recommender system to recommend products based on similarities between users. The users are clustered when they have similar products in their wish list or if the users purchased many similar, or the same, products. Then, the recommender system will recommend products to the users who have purchased similar products. For example, assume users A and B have purchased three similar products from Dell, and user A has purchased two more products from Apple. User B has similarities user A because they purchased three similar products from Dell, and in this case, the recommender system will recommend two products from Apple to user B. The recommender system does the same if users have similar products in their basket. The recommender system uses a basic K-means algorithm to execute the cluster analysis technique. The following algorithm describes the basic K-means algorithm \cite{64}. 

\begin{algorithm}
\end{algorithm}
Algorithm 5 K-means algorithm

BEGIN
Select K points as initial centroids
repeat
Form K cluster by assigning each point to its closest centroid.
Recompute the centroid of each cluster.
until Centroids do not change.
END

4.2 Case Study: Online Shopping System

The importance of implementing recommender systems has significantly increased during the last decade. The majority of available recommender systems do not offer clients the ability to make selections based on their choices or desires. This has motivated the development of a web based recommender system in order to recommend products to users and customers. The new system is an extension of an online application previously developed for online shopping under constraints and preferences. In this chapter, the system is enhanced by introducing a learning component to learn user preferences and suggests products based on them. More precisely, the new component learns from other customers’ preferences and makes a set of recommendations using data mining techniques including classification, association rules and cluster analysis techniques. The results of experimental tests, conducted to evaluate the performance of this component when compiling a list of recommendations, are very promising. Designing an appropriate recommender system, to meet the business needs of clients is the first and foremost consideration of this research. A recommender system for online shopping, based on preference learning, is a potential tool for business development and marketing. In this thesis, an interactive system is extended and based on preference elicitation [6,46], to recommend products based on customer suggestions. Recommender systems have significantly increased in the past decade. Preference learning in a recommender system is considered one of the most popular and significant techniques from Information Filtering [22,25]. Information filtering assists in
the removal of insignificant information and content that does not need to be stored in a customer profile. When a recommender system is applied, for instance, to learn the interests of users, it will study and learn some of the users’ behavioural aspects in order to generate and recommend a list of products. Learning the users’ preferences is one technique to discover the best outcomes to recommend items. Currently, it is important for clients to be assisted with their choices due to the exponential increase in existing data. Adaptive tools, algorithms, and user profiles are the three most significant components for designing and managing personalized recommendations. The popular recommender systems approaches are Content-Based, Collaborative, Demographic, Knowledge-Based and Hybrid. There are many techniques for learning user profiles including probabilistic approaches, neural networks, decision trees and association rules. The idea of preference learning is easy to understand but challenging to implement. A line of investigation is presented as follows: Can we learn, and know the preferences of users especially when there are missing data? Also, Are there any application platforms or recommender system for online shopping based on learning preferences? According to the current position in the field, the answer is no. In addition, there do not appear to be any recommender systems representing both numeric (quantitative) and ordinal (qualitative) preferences. Preference interactive systems include recommender systems; For instance, Amazon.com and Netflix which recommend products based on a clients similarities to other clients or viewed products and conversational applications which interact with the client with a simple dialogue to execute a task. However, Amazon.com and Netflix cannot handle the constraints and different forms of preferences such as numeric (quantitative) and ordinal (qualitative) preferences. As well, their learning preference components are limited in comparison to the learning preferences components, constraints and forms of preferences in the present research. Also, they ask clients questions that may not help them state their preferences. Table }
some interactive systems along with their parameters for handling constraints and forms of preferences in comparison to our interactive system.

Table 4.3: Comparison in terms of types of information

<table>
<thead>
<tr>
<th>Application</th>
<th>Quantitative</th>
<th>Qualitative</th>
<th>Mixed</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Netflix</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>SmartClient</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Teaching Salesman</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Project Execution Assistant</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Interactive system</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The new learning component added to the present interactive system will learn some of the customers preferences by applying data mining association rules and using the AprioriTid algorithm in order to provide and recommend products to users. The main objective is to recommend products to customers using two different methods. The first method learns some of the users preferences and recommends products based on the preferences. The second method recommends products based on the users preferences by applying the AprioriTid algorithm and data mining association rules. The system generates general recommendations for all customers with the AprioriTid algorithm which reviews all transactions to correlate frequent itemsets between customers.

4.3 System Description

Our proposed recommender system has been implemented using the following tools and environment.

- Programming languages: JavaScript, ASP.NET, C Sharp, and HTML.

- Database system: SQL Server Database 2012.

- Windows operating system supporting .NET Runtime Version ASP.NET 4, and PHP Version (PHP 5.2, PHP 5.3).

As illustrated in Figure 4.3, the interactive system is designed with three layers. The first layer is a graphic user interface where the user can register or create a new account in order to sign in and use the systems features. This layer also shows the results of the user queries and additional information. The user can choose the representation or elicitation of preferences or learning of preferences as the system learns the users preferences (Figure 4.4 and Figure 4.6 provide a screenshot of the graphic user interface). The second layer is a business layer where the user can interact or deal with two types or pages. The first page is the Quantitative and Qualitative Mixed page, and the second page is the Datea Mining Task for learning of preferences. The third layer is a database layer where the user can save data and import and export them into a database. A web server associated with the domain on Godaddy Web Hosting was reserved for the online system. The features for web hosting 150 GB of space, unlimited websites and bandwidth, 500 email Accounts, and SQL Server 2005 and MySQL databases.

4.4 System Methodology

An interactive system has been proposed to take into account user preferences when looking for an optimal solution [6][46]. The objective is to decide which configuration (i.e. combination of items) is the most preferred according to a set of preferences while respecting some requirements provided by the user. Requirements and preferences are represented through the Semiring-based Constraint Satisfaction Problem (SCSP) and CP-nets, respectively. The previous work has been extended to investigate the applicability of learning some user preferences. A new component is proposed to learn
preferences and suggest products for users according to their interests. For the time being, only one attribute (brand) is considered for new users. It is assumed there is a set of existing user profiles. Then the AprioriTid algorithm is applied to generate the frequent itemsets or products. Once the user is registered and an account is created, the user can choose his or her preferred brands. For example, the user may
show interest in one of the following brands: Apple, Samsung, LG, Dell, Toshiba and Sony. The system learns the users interests in order to recommend new products. It is assumed the user is able to specify the preferences or brands as shown in the graphic user interface illustrated in Figure 4.4 and the minimum support is equal to 20 percentage. It is also assumed there are four transactions or registered users, and that the users have chosen their brands as shown in Table 4.4. Figure 4.5 shows the itemsets represented with a lattice, and the minimum support is 20 percentage. Each node has its support number in order to compare it with the minimum support of 20 percentage to reduce the size of the tree to find the frequent itemsets. The interests for the first transaction are \{Apple, LG, Sony\}, and the interests for the second transaction are \{Samsung, LG, Dell\}. Once the system learns the user preferences, it will apply the AprioriTid algorithm and generate the frequent itemsets in order to suggest products to all users who share common interests as shown in Figure 4.6. The system will generate a specific page called customers recommendation to each user in order to suggest products based on their preferences as shown in Figure 4.6.

Table 4.4: 4 Transactions with 5 itemsets

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Apple, LG, Sony</td>
</tr>
<tr>
<td>2</td>
<td>Samsung, LG, Dell</td>
</tr>
<tr>
<td>3</td>
<td>Apple, Samsung, LG, Dell</td>
</tr>
<tr>
<td>4</td>
<td>Samsung, Dell</td>
</tr>
</tbody>
</table>

4.5 Conclusion

In this chapter, an extension of the constrained CP-net model in order to handle quantitative preferences has been proposed. This extended model has been integrated into our interactive system that we previously developed for managing constraints and preferences when buying online products.
As well, a recommender system based on user preferences and constraints has been proposed. The AprioriTid algorithm has been used to find frequent itemsets in order to recommend products to users. Experimental tests conducted on random datasets show the running time of the AprioriTid algorithm increases when the minimum support has a low value and decreases when the value is high. Additional solutions and algorithmic techniques can be used in this system in order to recommend products, and make it easier for users. New techniques for preferences and constraints can be implemented and tested on the system to see if the latter can handle more complex preferences and constraints. This recommender system can be generalized, and added to any interactive recommender application where the user and customer are involved in the procedure of choosing their products. In the near future we will explore other variable ordering heuristics [45] that can be used to better improve the time performance of our search method, and also plan to study the case of solving the extended
Figure 4.5: Itemsets represented with a lattice - reduce the size of the tree

constrained CP-net in a dynamic environment. This can be in the situation where the problem is over constrained and no suggestions can therefore be returned to the user. In this particular situation the user will be assisted in the selection of the constraints that need to be retracted in order to restore the consistency of the problem. Another scenario is when there is a large number of Pareto optimal solutions to return to the user. In this case, the user needs to constrain the problem even more and eventually adds more preferences in order to obtain a manageable number of options. To handle both the addition and retraction of constraints, we will integrate the dynamic arc consistency algorithm proposed in [43] into the constraint propagation of our search method.
Figure 4.6: The user’s recommendation page
Chapter 5

MANAGING HARD AND SOFT PREFERENCES WITH CONSTRAINTS IN INTERACTIVE APPLICATIONS

5.1 Introduction

Developing intelligent and interactive systems with visual user interfaces is very significant for websites and devices. In past years, interactive systems, recommender systems and online shopping websites have been enhanced and commonly used all over the world. The preference of the client plays a key part in customized AI systems/applications, e.g., the selection of web service [19,59]. Several online websites include some communication with the customer in order to understand and respond to their needs. Many online buying sites and recommender systems offer outcomes depending on their solutions and their costs. These do not always meet client cost ranges. When a client is enthusiastic about buying a new item, he or she will ac-
cess the web to find it. Some internet shopping sites offering clients new goods are problematic. Clients may be unhappy if these sites are difficult to use. Customers do not want to deal with complications when buying a product nor will they spend their time looking through sites. Corporations may not understand that it is a serious problem when they develop sites or internet purchasing websites that are difficult to use. The choices that websites offer may not be meeting the choices or specifications of the client and sometimes the choices offered are not enough. To be successful, these applications must increase the consumer encounter with the websites. Increasing the satisfaction of users can be achieved by knowing their preferences and taking their desires into account when they are looking for a given product. Several of these websites are available in a restricted environment. For instance, on the internet PC settings system, the customer cannot select two different parts that are not compatible. Therefore, the scenario where both preferences and constraints co-exist with each other is considered. The interest lies in an interactive system with combinatorial nature. Such systems have been intensively studied in the literature [49, 53, 58, 61, 63]. However, many of these existing systems assume feature aggregation to be additive and do not allow dependencies among the features. In previous work [6, 42], a system that allows these dependencies and tackles two well-known classes of preferences: qualitative and quantitative was proposed. In qualitative preferences, the buyer expresses preferences comparatively over the feature values. For instance, a user may be interested in 8GB of RAM more than 4GB for a laptop. This induces that the RAM with 8GB is more preferred to the 4GB one. In contrary, quantitative preferences assign numeric values to the feature values. Qualitative preferences are known to be more intuitive and easier to elicit from the user. At the same time, unlike quantitative ones, qualitative preferences are computationally problematic when it comes to deciding which alternative is preferred to another. As a result, we have also introduced an approximation method to approximate qualitative information into quantitative ones. This allowed
the system to process both types of preferences together and to be able to response faster than solving the qualitative information alone. In this chapter, an interactive system with two new methods (PrefC and WPrefC) that allows clients to show their preferences when purchasing a product online is proposed. More specifically, the clients are given the capability to provide their preferences and requirements in a helpful, interactive and entertaining technique. Desires and requirements are managed in an exclusive model, correspondingly through hard constraints and numerical (quantitative) and ordinal (qualitative) data. The interactive system will then show a list of optimal solutions that are intended to satisfy the hard constraints while satisfying client tastes and requirements. For a better management of constraints and preferences, the well-known constrained CP-net graphical model [14] is extended to (PrefC and WPrefC) and integrate it into the system. This extended constrained CP-Net first takes a set of constraints and qualitative preferences and returns a list of Pareto optimal outcomes (outcomes that are not dominated by any other outcome). It is believed this is the first time constrained CP-nets have been used together with quantitative preferences in interactive system.

5.2 PrefC Algorithm

5.2.1 Problem Statement

We assume a multivalued domain over a set of \( n \) variables \( V = \{V_1, V_2, \ldots, V_n\} \). Each variable \( V_i \) is associated with a set of possible values \( \text{dom}(V_i) = \{v_{i1}, v_{i2}, \ldots, v_{im_i}\} \). The set of variables \( V \) is the result of combining two disjoint sets \( V = V_{\text{Qual}} \cup V_{\text{Quan}} \). \( V_{\text{Qual}} \) is the set of variables that represent the CP-net while \( V_{\text{Quan}} \) is the set of variables where each variable \( V_i \in V_{\text{Quan}} \) has a utility function \( \mu(V_i) \) associated to it. A utility function for variable \( V_i \) is a mapping from \( \text{dom}(V_i) \) to \([1, 10]\). We also consider the existence of a set of constraints \( C \) over the variables in \( V \). Therefore, a solution (i.e.
mapping from $V$ to its domain) is feasible if and only if it satisfies all the constraints in $C$. Thus, the problem can be viewed as a triple $\varphi = \langle \mathcal{N}, \mathcal{U}, \mathcal{C} \rangle$ where:

- $\mathcal{N}$ is a CP-net over a subset of variables $V_{\text{Qual}} \subset V$,
- $\mathcal{U}$ is a set of utility functions over $V_{\text{Quan}} \subset V$,
- and $\mathcal{C}$ is a set of constraints over $V$.

Given $\varphi$, we are interested in finding the set of solutions that are feasible according to $\mathcal{C}$ and best according to both $\mathcal{N}$ and $\mathcal{U}$. In other words, our goal is find the Pareto set of the problem. A feasible solution is Pareto if its not dominated by any other feasible solution. The Pareto set is the set of all Pareto solutions.

### 5.2.2 Solving Method

The problem is solved by finding the Pareto set of the qualitative (i.e. CP-net) part first. Then, the set is approximated lexicographically. The method is based on the algorithm described in [5]. It should be noted that the latter algorithm finds one Pareto but it is straightforward to make it look for all the Pareto set. When looking for the Pareto set, one needs to perform dominance testing every time a solution is encountered. Algorithms 6, 7 and 8 describe the general steps in order to find the Pareto set. At each step, the current assignment ($n$) is determined if it is a complete assignment or not (line 6). If it is a solution, a dominance test is performed with the current solutions found to that point (line 7). Otherwise, one moves to the next variable. When the search ends, $\mathcal{S}$ contains all the Pareto solutions of the problem.

Once the Pareto is set, Algorithm 7 is used in order to obtain a total order over $\mathcal{S}$ following the lexicographic ordering. This is done by posing a total order over the set of variables $V$ (consistent with the CP-net $\mathcal{N}$). Afterwards, for every two Pareto solutions, the ties are broken by the first different value they have. The result is a total order over the Pareto set $s_1 > s_2 > \cdots > s_{|\mathcal{S}|}$. Thus, every Pareto solution is...
mapped to a number $i \in [1, \ldots, |S|]$ representing the position of that solution in the lexicographic order $>$. 

For the quantitative part, every outcome is defined over $V_{Quan}$. The utility of an outcome $o \in O_{Quan}$ is defined as follows:

$$\mu(o) = \sum_{i=1}^{|V_{Quan}|} \mu_i(o[i])$$  \hfill (5.1)

where $\mu_i(x)$ is the utility value when $V_i = x$. Therefore, the set of outcomes over $V$ (denoted as $O$) is the product of $O_{Qual} \times O_{Quan}$. Let $f^1 : O_{Qual} \rightarrow N$ and $f^2 : O_{Quan} \rightarrow N$ be the two mapping functions for qualitative and quantitative respectively. Then the utility of an outcome $o \in O$ is defined as follows:

$$u(o) = f^1(o_1) + f^2(o_2)$$  \hfill (5.2)

where $o_1$ and $o_2$ correspond to the outcome $o$ projected to $V_{Qual}$ and $V_{Quan}$ respectively. Once the Pareto set is determined and there is a total order to break Pareto Set Lexicographically, then qualitative and quantitative preferences are combined together as shown in Algorithm PrefC Method for interactive applications. More specifically, all tuples are combined with their weight/cost for qualitative and quantitative preferences and the PrefC method is applied, which consists of computing the sum of the quantitative and qualitative weights while satisfying the budget global constraints.

The extended constrained CP-net is integrated into the interactive system, presented in this section, via the problem of selecting products according to a set of constraints and preferences. Here the user has the ability to express budget constraints as well as preferences either in a quantitative (assigning numeric values to the attributes features) or in a qualitative manner (conditional of unconditional ranking of attributes features). Figure 5.1 shows a screenshot of the system Graphic User
**Algorithm 6** PrefC method - finding Pareto set

Input: CP-net $\mathcal{N}$ and a set of constraints $\mathcal{C}$
Output: The Pareto Set $\mathcal{S}$ (initially empty)

1. Let $\succ$ be topological ordering over $\mathcal{N}$
2. Let $T$ be a stack.
3. push $\{\}$ into $T$.
4. while (T is not empty)
   5. $n \leftarrow$ pop first element
   6. if ($n$ is complete assignment)
      7. if ($n$ is not dominated by any element in $\mathcal{S}$)
         8. add $n$ to $\mathcal{S}$
      9. else:
         10. for every value $v_i$ of the next variable $V_i$ according to $\succ$
            11. if ($n \cup (V_i, v_i)$ is consistent with $\mathcal{C}$)
               12. push $n \cup (V_i, v_i)$ into $T$

Return $\mathcal{S}$

**Algorithm 7** PrefC method - assigning weights to all the tuples of the CP-Net

Input: Pareto Solutions $\mathcal{S}$
Output: Lexicographic ordering and assigning weights to CP-Net tuples $\succ_{QualP}$ over $\mathcal{S}$

13. Let $\succ_{QualP} \leftarrow \{\}$
14. For every two elements $s_i, s_j \in \mathcal{S}$
15.   For $k = 1$ to $|V|$
16.      if $V_k(s_i) \neq V_k(s_j)$
17.      if $V_k(s_i)$ is better than $V_k(s_j)$
18.         add $(s_i, s_j)$ to $\succ_{QualP}$
19.      else
20.         add $(s_j, s_i)$ to $\succ_{QualP}$

Return $\succ_{QualP}$

**Algorithm 8** PrefC method - combining qualitative and quantitative together

Input: $f^1$ and $f^2$ mapping functions, constraints $\mathcal{C}$
Output: A set of ranked feasible solutions $\mathcal{S}$

21. $\mathcal{S} \leftarrow \emptyset$
22. while ($\mathcal{O}_{Qual} \neq \emptyset$)
23.   $o_1 \leftarrow \min(f^1(\mathcal{O}_{Qual}))$
24.   copy $\leftarrow Q_{Quan}$
25.   while (copy $\neq \emptyset$)
26.      $o_2 \leftarrow \min(f^2(copy))$
27.      if ($o_1 \times o_2$ consistent with $\mathcal{C}$)
28.         add $o_1 \times o_2$ to $\mathcal{S}$
29.      remove $o_2$ from copy
30.     remove $o_1$ from $\mathcal{O}_{Qual}$

Return $\mathcal{S}$
Interface (GUI). It is assumed that the client is capable of expressing preferences in either a quantitative or qualitative manner. As shown in Figure 5.2 WCP-nets are defined and a set of three variables (USB, Connection, HDMI) are linked, each with its own preference table. For example, the client prefers USB > Connection > HDMI. As shown in Figure 5.2, the relation and level among attributes can be clearly defined as well. For example, the variable USB (parent) is preferred more than the variables Connection and HDMI (children). Also, Figure 5.2 shows where quantitative preferences are defined as a set of two variables (Resolution, Display), each linked with its preference table and values level between 1 - less weight to 10 - high weight). For the variable Resolution, the client has a preference for a 1080pHD at a value level of 8 and a 720pHD at a value level of 4. For the Display variable, the client shows a value level of 5 for a LED and for a 3D LED, a value level of 2. The best outcome/tuple has less weight for the qualitative part plus the quantitative preferences part (USB3, Wifi, HDMI3 + 720pHD + 3D LED) = (0+4+2) = 6 and the worst outcome/tuple has a high weight (USB2, Ethernet, HDMI4 + 1080pHD + LED) = (15 + 8 + 5) = 28. It is assumed that there are two hard constraints, a global constraint being client budget and a local constraint being available attributes such as resolution, display size, and connection. Maximum price and local constraints can be expressed as incompatible among different components of products such as a laptop, e.g. components such as (USB3, HDMI4) and (Ethernet, 1080pHD) are incompatible, therefore, any tuple in the optimal solution with those components will not list as a set of optimal solutions because they do not satisfy the local constraints. Once the set of qualitative data along with quantitative preferences is determined, the list of optimal solutions must satisfy the hard constraints. For example, tuple (USB3, Wifi, HDMI3 + 720pHD + 3D LED) satisfied the local constraints.
Figure 5.1: Graphic user interface of the interactive system
5.3 Weighted PrefC Algorithm

5.3.1 Problem Statement

A variant of the problem stated in section 5.2.1 is considered here, where the qualitative preferences come with weights corresponding to their degree of preference. In this particular case, a Weighted CP-net instead of the traditional CP-net for representing the qualitative preferences was used. As shown in Algorithms 9, 10 and 11, the proposed method is implemented using Algorithms 6, 7, 8 and changing lines 14 to 26 as shown the steps of Weighted CP-net corresponding to their degree of preference. More precisely, the problem is defined as follows: a multivalued domain over a set of \( n \) variables \( V = \{ V_1, V_2, \ldots, V_n \} \) is assumed. Each variable \( V_i \) is associated with a set of possible values \( \text{dom}(V_i) = \{ v_{i1}, v_{i2}, \ldots, v_{im_i} \} \). The set of variables \( V \) is the result of combining two disjoint sets \( V = V_{\text{Qual}} \cup V_{\text{Quan}} \). \( V_{\text{Qual}} \) is the set of variables that represent the Weighted CP-net while \( V_{\text{Quan}} \) is the set of variables where each variable \( V_i \in V_{\text{Quan}} \) has a utility function \( \mu(V_i) \) associated to it. A utility function for variable
\( V_i \) is a mapping from \( \text{dom}(V_i) \) to \([1, 10]\). We also consider the existence of a set of constraints \( \mathcal{C} \) over the variables in \( V \). Therefore, a solution (i.e. mapping from \( V \) to its domain) is feasible if and only if it satisfies all the constraints in \( \mathcal{C} \). Thus, the problem can be viewed as a triple \( \varphi = \langle \mathcal{N}, \mathcal{U}, \mathcal{C} \rangle \) where:

- \( \mathcal{N} \) is a Weighted CP-net over a subset of variables \( V_{\text{Qual}} \subset V \),
- \( \mathcal{U} \) is a set of utility functions over \( V_{\text{Quan}} \subset V \),
- and \( \mathcal{C} \) is a set of constraints over \( V \).

Given \( \varphi \), in the goal is in finding the set of solutions that are feasible according to \( \mathcal{C} \) and best according to both \( \mathcal{N} \) and \( \mathcal{U} \). In other words, the goal is find the Pareto set of the problem. A feasible solution is Pareto if its not dominated by any other feasible solution. The Pareto set is the set of all Pareto solutions.

### 5.3.2 Solving Method

These preferences are either quantitative (values scale between 1 - less weight, to 10 - high weight) and qualitative with a weighted extension to CP-nets (Level 1 - two values are equally preferred, level 2 - first value is slightly preferred to the second value, level 3 - first value is highly preferred to the second value, level 4 - first value is very highly preferred to the second value, level 5 - first value is extremely significant preferred to the second value) \cite{68,71}. The interactive system will then display a list of optimal solutions for meeting the clients needs and satisfying hard constraints. It is assumed that the client is capable of expressing preferences in either a quantitative or qualitative manner with a weighted extension to CP-nets. As shown in Figure 5.3, WCP-nets are defined and a set of two variables (Memory, Screen) are linked, each with its own preference table. For example, for the memory variable, the client prefers 12 GB of memory three times more than 8GB. For the screen variable, when
the memory value is 12 GB, the client prefers a 13-inch screen two times more than a 15-inch and for a memory variable of 8 GB, the client prefers a 15-inch screen four times more than a 13-inch. As shown in Figure 5.3, the relation and level among attributes can be clearly defined as well. For example, the variable memory (parent) is preferred five times more than the variable screen (child). Figure 5.3 shows where quantitative preferences are defined as a set of two variables (Processor, Hard Drive), each linked with its preference table and values level between 1 - less weight to 10 - high weight). For the variable Processor, the client has a preference for a 2.7GHz processing speed at a value level of 8 and a 2.9GHz speed at a value level of 4. For the hard drive variable, the client shows a value level of 5 for a 1TB and for a 2TB, a value level of 2. The next seven equations are the formal expression of how to compute the weighted values of the qualitative data with a weighted extension to CP-nets for two variables (Memory, Screen) in order to sum those weights with the weights/values level of the quantitative data.

\[WM + WS = 1\] \hspace{1cm} (5.3)

\[WM + WS = 5WS + WS = 6WS = WS = 0.16\] \hspace{1cm} (5.4)

\[WM = 5WS = 5 \times 0.16 = 0.84\] \hspace{1cm} (5.5)

\[(12GB, 13\text{ - inch}) = (0 \times 0.84) + (0 \times 0.16) = 0\] \hspace{1cm} (5.6)

\[(12GB, 15\text{ - inch}) = (0 \times 0.84) + (2 \times 0.16) = 0.32\] \hspace{1cm} (5.7)
The best outcome/tuple has less weight for the qualitative part with WCP-nets plus quantitative preferences part (12 GB, 13-inch, 2.9GHz, 2TB) = (0+4+2) =6 and the worst outcome/tuple has a high weight (8 GB, 13-inch, 2.7GHz, 1TB) = (3.16+8+5) =16.16. It is assumed there are two hard constraints, a global constraint being the clients budget and a local constraint being available attributes such as speed, screen size and hard drive capacity. Maximum price and local constraints can be expressed as incompatible among different components of product such as laptop, e.g. components such as (12GB, 15-inch) and (2.7GHz, 2TB) are incompatible, therefore, any tuple in the optimal solution with those components will not list as a set of optimal solutions because they do not satisfy the local constraints. Once the set of qualitative data with a weighted extension to CP-nets along with quantitative preferences is obtained, the list of optimal solutions has to satisfy the hard constraints. For example, tuple (12 GB, 13-inch, 2.9GHz, 2TB) satisfied the local constraints. Algorithms 9, 10 and 11 describe the general steps in order to find a list of optimal solutions to attempt to meet the clients needs and satisfy the constraints. The proposed method is implemented using Algorithms 6, 7, 8 and changing lines 13 to 26 in Algorithm 10 as shown the steps of Weighted CP-net corresponding to their degree of preference.

5.4 System Description

Our interactive system has been developed using the following.

Algorithm 9 WPrefC method - finding Pareto set

Input: CP-net $\mathcal{N}$ and a set of constraints $\mathcal{C}$
Output: The Pareto Set $\mathcal{S}$ (initially empty)

1. Let $\succ$ be topological ordering over $\mathcal{N}$
2. Let $T$ be a stack.
3. push $\{\}$ into $T$.
4. while (T is not empty)
5. $n \leftarrow$ pop first element
6. if ($n$ is complete assignment)
7. if ($n$ is not dominated by any element in $\mathcal{S}$)
8. add $n$ to $\mathcal{S}$
9. else:
10. for every value $v_i$ of the next variable $V_i$ according to $\succ$
11. if ($n \cup (V_i, v_i)$ is consistent with $\mathcal{C}$)
12. push $n \cup (V_i, v_i)$ into $T$

Return $\mathcal{S}$

Algorithm 10 WPrefC method - assigning weights to all the tuples of the CP-Net

Input: Pareto Solutions $\mathcal{S}$
Output: Ordering and assigning weights to CP-Net tuples $\succ_{WPrefC}$ over $\mathcal{S}$

13. Let $\succ_{WPrefC} \leftarrow \{\}$
14. For every two elements $s_i, s_j \in \mathcal{S}$
15. For $k = 1$ to $|V|$
16. if $V_k(s_i)$ is equally preferred to $V_k(s_j)$
17. $(s_i, s_j) = 1$ and add $(s_i, s_j)$ to $\succ_{WPrefC}$
18. else if $V_k(s_i)$ is mildly preferred to $V_k(s_j)$
19. $(s_i, s_j) = 2$ and add $(s_i, s_j)$ to $\succ_{WPrefC}$
20. else if $V_k(s_i)$ is strongly preferred to $V_k(s_j)$
21. $(s_i, s_j) = 3$ and add $(s_i, s_j)$ to $\succ_{WPrefC}$
22. else if $V_k(s_i)$ is extremely preferred to $V_k(s_j)$
23. $(s_i, s_j) = 4$ and add $(s_i, s_j)$ to $\succ_{WPrefC}$
24. else
25. if $V_k(s_i)$ is better than $V_k(s_j)$
26. add $(s_i, s_j)$ to $\succ_{WPrefC}$
27. else
28. add $(s_j, s_i)$ to $\succ_{WPrefC}$

Return $\succ_{WPrefC}$
Algorithm 11 WPrefC method - combining qualitative and quantitative together

Input: $f_1$ and $f_2$ mapping functions, constraints $C$
Output: A set of ranked feasible solutions $S$

27. $S \leftarrow \emptyset$
28. while ($O_{Qual} \neq \emptyset$)
29. $o_1 \leftarrow \min(f_1(O_{Qual}))$
30. $copy \leftarrow Q_{Quan}$
31. while ($copy \neq \emptyset$)
32. $o_2 \leftarrow \min(f_2(copy))$
33. if ($o_1 \times o_2$ consistent with $C$)
34. add $o_1 \times o_2$ to $S$
35. remove $o_2$ from $copy$
36. remove $o_1$ from $O_{Qual}$

Return $S$

Figure 5.3: An example of WPrefC method for interactive system


3. Operating system: Web Hosting under Windows 7 and supporting .NET Run-
time Version (ASP.NET 4.5) and PHP Runtime Version (PHP 5.4).


The software architecture of the interactive system has three layers as shown in Figure 5.4. The three layers are graphic user interface, business, and database. The graphic user interface is where the client can register and sign in to use the features of the interactive system. It takes care of the user constraint and preference elicitation phase. The business layer links to the graphic user interface and represents the solving techniques PrefC and WPrefC solvers. The database layer takes care of the information storage and retrieval. A sequence diagram of Sequence Diagram of PrefC Method for interactive system (Product - TV) is shown in Figure 5.5. Clients can either log in or sign up and then choose and sort quantitative preferences and choose quantitative preferences in order to express preferences for the product (TV). A sequence diagram of WPrefC (Product - Laptop) is shown in Figure 5.6. Clients can either log in or sign up and then choose WPrefC page in order to express quantitative preferences and qualitative preferences for the product (laptop). Component diagram (3-TIER ARCHITECTURE) is shown in Figure 5.7 and it contains two solvers (PrefC solver and WPrefC solver), Apache Server (Web Hosting) and internet browsers such Internet Explorer and Google Chrome.

5.5 Conclusion

In this chapter, a new interactive system based on PrefC and WPrefC methods that allows the client to express their preferences when buying a product online has been proposed. The two new methods were applied in order to resolve the issues of CP-nets and permit users to hard-code their preferences [68]. Clients can express their preferences either in a numerical (quantitative) way or in an ordinal (qualitative) way. The interactive system will then display a list of optimal solutions to attempt to
meet the clients needs and satisfy the constraints. Future work includes: managing preferences along with dynamic hard constraints where the client is able to either add or delete some hard constraints and notice the influence of those changes on the outcomes returned, applying new techniques to the interactive system in order to represent quantitative and qualitative preferences for clients in a hard-grained way.
Figure 5.5: Sequence diagram of PrefC method for interactive system
Figure 5.6: Sequence diagram of WPrefC method for (product - TV)

Figure 5.7: Component diagram (three-tier architecture)
Chapter 6

EVALUATION OF THE PROPOSED INTERACTIVE SYSTEM

6.1 Introduction

Designing interactive systems with graphic user interfaces is an important step in the development of online devices and websites. Interactive systems and recommender applications have improved in the last decade and they are now widely used all over the world. However, it is important to understand online shoppers needs and preferences and to take them into account. In this regard, several interactive systems rely on customer preference elicitation while others suggest products based on other customers recommendations. The focus of this chapter is the interaction design of a system for Managing Preferences and Constraints (MPC) and Preferences Learning (PL). An evaluation method is utilized to obtain user feedback on how effective the system is and how easy it is to use, compared to other systems. The Volere requirements specification template was used with the six step framework to guide the
evaluation. As well, the focus of this chapter is to evaluate the GUI - graphical user interface for the interactive system.

Human computer interaction and interactive design are considered amongst the most important components of any interactive system, as it shows if the system has an adequate or poor design \[35,37\]. Users should be satisfied when interacting with these systems and not face any obstacles or difficulties \[21\]. It is important in interactive design that designers get an idea of how effective, efficient, and user friendly the system has become \[62\]. They would like to know if the system is confusing, inefficient or difficult to use. There are many fields involved in the interaction design, including Computer Science, Software Engineering, Human Computer Interaction, and so on \[55\]. User interface designers cooperate with developers and experts to create a good design \[12\]. There are two main aspects that must be considered in human computer interaction usability and functionality \[35,65\]. Functionality of an application/system is determined by a set of services and activities which it offers to its customers \[35\]. Usability of an application/system with some functionality is the variety and level by which the application/system can be utilized effectively to achieve certain objectives for a segment of customers \[35\]. Several recommender systems and online shopping websites provide outcomes based on their alternatives, and their prices do not always match a clients budget. When a client is interested in purchasing a new product, he or she may access the internet to find it. Nevertheless, some online shopping websites offering clients and consumers new products are problematic. Clients and consumers may become upset or dissatisfied if the websites are not easy to use. Indeed, they do not want to face difficulties when purchasing a product or waste time searching through websites. Companies may not realize this is a significant problem when they develop websites or interactive systems/website that are not easy to use. Even the options some websites offer do not necessarily match the preferences or requirements of the client and consumer. Sometimes, the options provided to the user
are not sufficient to express his/her desires. In previous work, this research focused on preference elicitation, and applied several models \([4, 33, 45, 47]\) to handle preferences and constraints in order to assist the users when purchasing a product online, giving consideration to their requirements and preferences \([6, 46]\). The work was extended to develop a learning feature into the interactive system in order to give the system the capability to recommend products to customers based on data mining techniques using the AprioriTid algorithm \([42]\). This chapter focuses on evaluating chapter 3 a preference-aware interactive system for online shopping and chapter 4 data mining techniques and preference learning in recommender systems of the interaction design of the interactive system under preferences and constraints. The system interface has been redesigned and improved. Evaluation types have been applied in order to obtain user feedback on how effective the system is and how easy it is to learn compared to other systems. The six steps have been applied to the decision framework to evaluate the system. The Volere requirements specification template has been applied to the interactive system under preferences and constraints. The remainder of the chapter is organized as follows. The Volere requirements specification template (Project Drivers, Project Constraints, Functional Requirements, Non-Functional Requirements and Project Issues) for the interactive system under preferences and constraints is presented in the following section. Design and high-fidelity prototyping are discussed with examples in section 6.3. The evaluation framework is then presented in section 6.4. Finally, concluding remarks and future works are listed in section 6.5.

### 6.2 Requirements Specification

The Volere requirements specification template has been considered and followed \([54, 55]\) in order to direct and help create suitable requirements and specifications for the interactive system. The Volere requirements specification template \([54, 55]\) provides a
good valuable source for institutions worldwide as it saves them time and money when developing products \cite{54,55}. It guides them to suitable requirements specifications \cite{45,55}. The Volere requirements specification template is used to develop the interactive system under preferences and constraints.

6.2.1 Project Drivers

- **The purpose of the project**: the main objective is to develop this recommender system under preferences and constraints so the users can register and create an account and then purchase a product based on their preferences. This online recommender system is capable of recommending and providing the users with a list of options and recommendations corresponding to the best solutions based on a set of the users preferences and interests. Several models are applied to handle the preferences and constraints. This will help customers buy a product online and take into consideration their requirements and preferences. It will recommend products to the customers in different ways.

- **The intended users and audience**: the clients who are going to use and interact with the interactive system are mostly young, adult, teen, and senior people. These are the people that are going to use and interact with the interactive system.

- **Stakeholders**: the Stakeholders of the interactive system are: The designer, developer, administrators, supervisor, user, customer and client.

6.2.2 Project Constraints

- **Mandated constraints**: some online shopping websites and recommender systems provide outcomes based on their alternatives. However, their prices do not always match a clients budget. When a client is interested in purchasing a new
product, he or she may access the internet to find it. Nevertheless, some online shopping websites and recommender systems may not offer clients and consumers the products that meet their preferences. Clients and consumers may become upset or dissatisfied, especially if the system is not easy to use. Clients and consumers do not want to face difficulties when purchasing a product or waste time searching through websites. Even the options some websites offer do not necessarily match client or consumer preferences or requirements. The interactive system is accessible through the following address: www.drbandar.com. Deluxe web hosting on has been reserved on Godaddy for the interactive system. Features of the deluxe web hosting are: Unlimited disk space and bandwidth. It has 500 email Accounts/Addresses, SQL Server 2012 and MySQL databases. The following environment and tools are the implementation details of the interactive system on the local machine and deluxe web hosting. Windows operating system supporting .NET Runtime Version ASP.NET 4. Environments and Tools: Microsoft Visual Studio 2010 and 2012 ASP.NET - C-Sharp. Programming languages: ASP.NET, C-Sharp, HTML, and JavaScript. Database system: SQL Server Database 2012.

- **Naming conventions and terminology**: there are keywords and icons that have been used in the interactive system. Some of the buttons or icons; For example, the admin icon redirects the user to the privileges to assign tasks to the supervisor. The supervisor icon redirects the user to the supervisors page to make adjustments in order to show and display the data and products to the users. The sign in the icon will let the user login and interact with the interactive system. The sign up icon will redirect the user to the sign up page to create a new account to interact and deal with the interactive system. The quantitative preferences icon redirects the user to the page of numeric values to assign the numeric values to his/her preferences. The qualitative preferences
icon redirects the user to the page of non-numeric values, and there is no numeric value to assign to the users preferences in the method. The mixed quantitative and qualitative preferences icon redirects the user to the page of mixed values (a combination between numeric values and non-numeric values).

- **Relevant facts and assumptions:** there are relevant facts in the interactive system. For instance, if the user would like to press the button basket, it will present all the products or items that the user added to his/her basket. If there is a new user and he/she would like to create a new account, he/she would press the sign up button in order to redirect the user to the sign up page to create a new account. When the user would like to press any button or check something on the website, he/she would have a clear idea what it will happen if she/he presses this button. For instance, before the user will hit or press the button or icon to log out, she/he will assume what is going to happen after pressing this button. Therefore, the user will possess assumptions before they interact and press any button or icon in the interactive system.

### 6.2.3 Functional Requirements

- **The scope of work:** currently, the interactive system does not have any evaluation types and none of them have been added to the interactive system to obtain feedback and improve the functionality and interfaces of the interactive system. The interfaces of the interactive system are to be simplified and made flexible. Some of the interactive system interfaces are to be redesigned to make them more attractive.

- **Business data model and data dictionary:** Figure 6.1 represents the Quantitative and Qualitative (Product - Laptop) Sequence Diagram of the interactive system. The user can sign in by inserting a username and password. The user
can choose the Mixed (Quantitative and Qualitative) page and then the user can choose, for example, the laptop product and rank them by their preferences and inserts a price and the number of solutions. The user obtains results based on the search and adds them to their basket. Figure 6.2 represents a component diagram (3-TIER ARCHITECTURE) for the interactive system. It shows SQL Server 2012, Apache Server Godaddy Web Server, and the browsers of the Customer Computer such as Firefox, Google Chrome.

- **The Scope of the Product**: Figure 6.3 shows the use case diagram of the interactive system. Example of Use Case: Search Products - Quantitative

![Quantitative and qualitative (product - laptop) sequence diagram](image1)

**Figure 6.1: Quantitative and qualitative (product - laptop) sequence diagram**

![Component diagram (three-tier architecture)](image2)

**Figure 6.2: Component diagram (three-tier architecture)**
Figure 6.3: Use case diagram

- Initiating actor: buyer who wants to search for a product - Quantitative.
- Precondition: buyer has to sing in/sign up in order to search for a product - Quantitative.

Scenario 1: Searching for a product - Quantitative.

1. If a buyer has an account, then he needs to type in the username and password.
2. Buyer needs to sign in.
3. Buyer selects quantitative page.
4. Buyer selects products.
5. Buyer selects features or aspects with preferences.
6. Buyer enters a budget or price.
7. Buyer enters a number of solutions.
8. Buyer obtains a result based on a budget that the buyer enters and the number of solutions.

9. Buyer can add items from the results to the basket.

**Scenario 2: Searching products - Quantitative for new user.**

1. Buyer needs to sign in.

2. If buyer does not have an account then buyer has to create a new account.

3. Buyer needs to press the button sign up to create a new account.

4. After creating a new account the buyer needs to go to the main page to sign in.

5. Buyer needs to type username and password.

6. After signing in, the buyer selects quantitative page.

7. Buyer selects products.

8. Buyer selects features or aspects with preferences.

9. Buyer enters a budget or price.

10. Buyer enters a number of solutions.

11. Buyer obtains a result based on a budget that the buyer enters and a number of solutions.

12. Buyer can add items from the results to the basket.

- **Postcondition**: showing the results of products - Quantitative along with features and preferences.

- **Benefiting Actor**: buyer, admin and supervisor.
- **Functional and data requirements**: Figure 6.4 shows the software architecture. A third page Evaluation Page has been added to the second layer, Business layer. This new page provides an idea of how the user interacts and deals with the system as well as obtains feedback. The first layer is a graphic user interface where the customer can sign up and sign in order to interact with the system. The second layer is a business layer where the customer can manipulate the pages. The third layer is a database layer where the customer can store data and retrieve and extract them into a database.

![Software Architecture Diagram](image)

Figure 6.4: Software architecture
6.2.4 Non-Functional Requirements

- **Look and feel requirements**: the interactive system has a decent look, and satisfies all the requirements. For example, Figure 6.5 displays the main page where the user can sign in by inserting a username and password or create an account and return to the main page to insert the username and password as shown in Figure 6.6.

- **Usability and humanity requirements**: the interactive system provides adequate usability and meets client requirements as shown in Figures 6.5 and 6.6.

- **Performance requirements**: the online shopping system has adequate performance requirements. For instance, it does not take more than 10 seconds to match the inserted username and password. A buyer obtains a result in 5 or 10 seconds after selecting the features and preferences. Figure 6.7 presents performance testing with benchmarks in the mixed page of the interactive system.

- **Operational and environmental requirements**: the operational and environmental requirements of the online shopping system do not require too much equipment, and can be easily accessed from a laptop, a desktop computer or any device with an internet connection.

- **Maintainability and support requirements**: so far, there are no maintainability or support requirements in the interactive system. However they will be added in the near future.

- **Security requirements**: the interactive system protects the users account. When a buyer types or inserts an incorrect username or password in the login page, the system displays an error message as shown in Figure 6.7.
6.2.5 Project Issues

- **Open issues**: opportunity has been made for the user to provide feedback to improve the functionality and interfaces of the interactive system. It is important to know if the interactive system has good utility, is safe to use, effective, easy to learn and efficient.

- **Addressing problems**: a technical support component will be added to address any problems.
Figure 6.6: Shows sign up page of the interactive system

- **Tasks**: in the near future the system will be improved to ensure the interfaces are more attractive, simpler, and flexible to use. Different types of evaluations will be added to get user feedback in order to improve the functionality, and interfaces of the interactive system.

- **User documentation and training**: so far, there is no user documentation or training for the recommender system. However, future work will add the tools to the interactive system.

- **Costs of hosting**: the cost of reserving a web hosting and domain on the internet for the system will be accounted for in a budget.

- **Risks**: if there is a lack of safety and security in the interactive system, it will cause many problems and put the system data at risk. It will create a hug
• **Ideas for solutions:** new ideas and solutions will be created to solve user problems while they use the interactive system.
6.3 Design and Prototyping

6.3.1 High-Fidelity Prototyping

Some of the high-fidelity prototyping for the interactive system under preferences and constraints is presented in this section. Figure 6.8 shows the main page of the website at Godaddy webhosting (www.drbandar.com), with four sections (PL, MPC, Evaluation Form, Reviews).

![Figure 6.8: Main page of the website at Godaddy web hosting](image)

Figure 6.8: Main page of the website at Godaddy web hosting

Figure 6.9 shows the main page of the MPC with three types of preferences: Numeric (quantitative), ordinal (qualitative) and mixed (quantitative + qualitative).

Figure 6.10 shows the mixed page (quantitative + qualitative) with new features such as a hidden Radio Button; the user can hide any item that they do not want to consider. A textbox has been added to let the user type a number of requested solutions.

Figure 6.11 shows the main page of the PL where the system learns users preferences and interests in order to combine them with other customer preferences and
6.4 Evaluation Framework

Evaluation is considered a very important component to the system. It collects information from users in order to improve the system. The framework makes a set of recommendations using Data Mining Techniques (AprioriTid Algorithm). The system will recommend products to a user based on his/her interests.
Figure 6.10: Mixed page (quantitative and qualitative) with new features
Determine the goals: this interactive system should be capable of recommending and providing the user with a list of options and recommendations based on his/her preferences and interests.

Explore the questions: the following are the important questions regarding the design of the system. Is the interactive system confusing, infuriating, inefficient or difficult to use? Is the interactive system efficient? Is the interactive system easy to learn? Does the interactive system involve the user to evaluate the system and provide a feedback in order to improve the functionality, and interfaces of the interactive system? Does the interactive system have different evaluation types?
• **Choose the evaluation methods:** the questionnaire was used to gather data from the user to improve the system in Figure 6.12.

![Questionnaire](image)

Figure 6.12: Questionnaire in order to gather data from the user

• **Identify the practical issues:** the users who are going to interact with the interactive system are mostly young, adult, teen or senior people.

• **Decide how to deal with the ethical issues:** an informed consent form was developed and explained the main page of the website in Figure 6.1, the goals of the system and the study. Participants did not have to enter their personal information. The informed consent form was available online and the questions were straightforward, as shown in Figure 6.13.

• **Evaluate, analyze, interpret and present the data:** Figure 6.14 presents the reviews page of the interactive system as well as reviews submitted by others.
Figure 6.13: Evaluation form page of the interactive system
Once the users are done evaluating the interactive system, the system will show their feedback without revealing any of their personal information.

Figure 6.14: Reviews page of the interactive system

6.5 Conclusion

In this chapter, the Volere requirements specification template was applied in order to derive suitable requirements specifications for the interactive system. A high-fidelity prototyping was presented to show how the system interfaces were redesigned (PL - MPC), and obtained an adequate interaction design. Finally, the six steps of the decision framework \[55\] were applied in order to evaluate the system and
give users the ability to express their opinions. So far one type of evaluation with anonymous users was used to obtain feedback. In the near future, we will apply new evaluation types such as inspections, predictive models and analytics. Ethical issues were considered and an informed consent form explained the purpose of the evaluation to the evaluators. Students from computer science and other fields were involved in the evaluation.
Chapter 7

EXPERIMENTATION

7.1 Evaluation of Preference-Aware Interactive System for Online Shopping

In order to evaluate the time performance of our system and its related methods, we conducted an experiment on three types of problems: qualitative, quantitative and mixed. More precisely, we randomly generate 100 instances for each problem and take the average time needed to find the optimal outcome. The experiments are done via our solver which contains two mini solvers for CSP and CP-net. Our system is implemented in C# programming language under Microsoft Visual Studio 2010 environment in Windows 7 Professional operating system. The experiments are conducted on an Intel Core i7 PC with 2.7GHz CPU and 4 GB RAM.

For all the problem instances, the domain size of the variables is equal to five possible values and the number of variables is five. The number of preference statements varies from 1 to 25. This is a reasonable number since in online systems users are more likely to provide less than 25 statements. In case of quantitative preferences, each tuple is considered as preference statement. These tuples have the format of $(x, y, p)$ where $x$ and $y$ are variables values for variables $x$ and $y$ respectively and $p$ is
the associated preference. For the qualitative preferences, each entry in the CPT of a variable is considered as a statement. For instance, the statement \( y : x_1 \succ x_2 \) in the \( CPT(X) \) is a statement asserting that \( x_1 \) is preferred to \( x_2 \) when \( y \) holds. Figure 7.1 shows the time needed (in milliseconds) to find the optimal solution for each problem with a given number of statements.

In order to interpret the experimental results we base our discussion on the notion of dominance testing. That is testing whether one outcome is better than the other. Given two outcomes checking which one is better is an easy task in the quantitative preferences. However, the main difficulty in reasoning with CP-nets is the dominance testing. It is known to be expensive even for a cyclic CP-nets 57. For the mixed approach, it is clear that it requires more time due to the approximation process involved.

![Graph showing time needed for different approaches](image)

Figure 7.1: Experiments for the time needed to find optimal solution
7.2 Evaluation of the Proposed Learning Component

The average time required to recommend new items to users is presented in this section. Two experiments have been conducted with two different datasets. The first experiment is based on synthetic data generated below. The second experiment is based on a dataset imported from the UCI KDD Archive (UCI Knowledge Discovery in Databases Archive - http://kdd.ics.uci.edu/). The experiments were performed on the online store website in order to determine the average time needed to obtain the list of recommendations (frequent itemsets). The experiments were completed via an abstract solver containing two mini solvers for CSP, CP-net [6] and AprioriTid. The system is implemented with the C Sharp programming language in a Microsoft Visual Studio 2010 environment operating on a Windows 7 operating system. The experiments were conducted on an Intel Core i7 PC with a 2.7GHz CPU and 4 GB of RAM.

The synthetic data experiments were performed on a dataset with 6 k-itemsets Apple, Samsung, Dell, Sony, Toshiba and LG and a varying number of transactions, from 5 to 40. It is assumed the threshold or minimum support is equal to 30 percentage where the itemset is frequent if it is chosen at least 3 times. Figure 7.2 shows the time needed (in milliseconds) to find the frequent itemsets with a varying number of transactions. Experimental tests conducted on random datasets show the running time of the AprioriTid algorithm increases when the minimum support has a low value and decreases when its value is high.

The second experiment was performed with different transactions from the UCI KDD archive in order to obtain the frequent itemsets. Figure 7.3 represents a sample dataset with 1199 transactions tested with the AprioriTid algorithm in order to obtain the frequent itemsets. The number of features in the data is 257 but Figure 7.4
Figure 7.2: Running time needed to generate frequency displays additional features. Figure 7.5 presents four charts for the dataset with varying minimum supports (5, 7, 10 and 15) along with the time required in seconds, to find the frequent itemsets. The running time increases when the minimum support has a lower value and decreases when the minimum support is higher.
Figure 7.3: Running time needed to generate frequency
Figure 7.4: Running time should be inside the circle in the GUI

Figure 7.5: Running time for UCI KDD datasets with threshold: 5, 7, 10, and 15
7.3 Evaluation of Managing Hard and Soft Preferences with Constraints in Interactive Applications

In order to evaluate the solving methods we presented in chapter 5, we have conducted several experiments on instances of the application we presented in the previous section. The experiments have been conducted on a local machine with 64-bit Operating System, Intel Core i7 with a 2.4GHz CPU and 12.0 GB of RAM. The goal here is to determine the average time needed to return a list of suggestions. Figure 7.6 reports this running time in milliseconds when varying the number of attributes from 3 to 10. As we can see from the chart, while the running time is fast growing it is still under a second even when the number of attributes is equal to 10.

Figure 7.6: Average running time needed to return a set of suggestions
Chapter 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

In summary, an interactive system depending on customer constraints and preferences has been proposed. The clients identify their constraints and preferences then the interactive system looks for the Pareto optimal outcomes fulfilling the set of specifications indicated as hard constraints and enhancing the preferences. Data mining techniques have been used for the new component that learns from other clients preferences and creates a set of suggestions. Also, data mining technique association rules have been used in order to obtain frequent item sets to recommend and suggest products to clients. This general interactive system can be easily customized to serve as an online store where constraints and preferences are provided by the customers. Clients can express their preferences either in a numerical (quantitative) way or in an ordinal (qualitative) way. The Volere requirements specification template was applied in order to derive suitable requirements specifications for this interactive system. A high-fidelity prototyping was presented to show how the GUI was redesigned in order
to obtain an adequate interaction design. Finally, the six steps of the decision framework [55] were applied in order to evaluate this interactive system and give users the ability to express their opinions.

8.2 Future Work

In the near future, plans are to improve this interactive system by adding more sophisticated constraint solving techniques in the application and considering more features. This interactive system can be generalized to any interactive system where the user is involved in the process of choosing an item among a set of others. There are also plans for handling and managing more complex preferences in case of dynamic hard constraints. This can be the case where the user interacts with the system by adding or removing some constraints and can see the effect of these changes on the solutions returned.
REFERENCES


