

A Preference-Aware Interactive System for Online Shopping

Eisa Alanazi^{1,2}, Malek Mouhoub¹ & Bandar Mohammed^{1,2}

¹ Department of Computer Science, University of Regina, Regina, Canada

² Sponsored by Ministry of Higher Education, Saudi Arabia

Correspondence: Malek Mouhoub, Department of Computer Science, University of Regina, Regina, SK., S4S 0A2, Canada. Tel: 1-306-585-4700. E-mail: alanazie@gmail.com; mouhoubm@uregina.ca; mohammeb@uregina.ca

Received: July 24, 2012 Accepted: September 13, 2012 Online Published: October 18, 2012

doi:10.5539/cis.v5n6p33

URL: <http://dx.doi.org/10.5539/cis.v5n6p33>

Abstract

Preference Elicitation is very important for online shopping interactive applications. The potential buyers usually have interest in some of the attributes of the product they want to purchase. While the current online shopping systems allow the users to provide some keywords and other information in order to filter and get only what they need, these latter feel that what they get does not necessarily meet their satisfaction. In this paper, we propose a new shopping system that enables the customers to express their needs when buying a product online. More precisely, the users are given the ability to provide their requirements and desires in a friendly and interactive way. The system will then provide a list of suggestions meeting the users' requirements and maximizing their desires. Requirements and desires are managed, in a unique model, respectively as a set of hard constraints and preferences where these latter can be quantitative (numerical), qualitative (ordinal) or both. These constraints and preferences represent a constraint optimization problem where optimal solutions (best outcomes) are those satisfying the hard constraints and maximizing the user's preferences. The branch and bound method is applied in order to provide the user with a list of best outcomes.

Keywords: e-commerce, constraint optimization, preferences, artificial intelligence

1. Introduction

Many online systems involve some interaction with the users in order to understand and answer their needs. In order to succeed, these systems should maximize the user experience with the system. Maximizing user satisfaction can be done by taking his/her desires/preferences into consideration when he/she is looking for a given item. This will result in making the system more popular and earning more reputation among customers. In order for the systems to handle user desires, they should have a proper way to represent, elicit and evaluate user preferences. Thus, reasoning about preferences is a core concept for systems involving a relatively large space of interaction with the user. Moreover, many of these systems exist in a constrained environment. For example, in an online PC configuration system, the user cannot choose two different parts that are incompatible with each other. Therefore, we should consider the situation where constraints and preferences co-exist together. In the current shopping systems, shoppers might face a difficult time to get an appropriate product among a huge set of possibilities of outcomes. Moreover they have the impression that their preferences and needs are not taken into consideration by the system. This has motivated us to develop an online shopping system taking into account the problem requirements and maximizing the users desires. Requirements and desires are managed, in a unique model, respectively through a set of hard constraints and preferences where these latter can be quantitative (numerical), qualitative (ordinal) or both. Hard constraints correspond here to both user and problem requirements and are managed with the Constraint Satisfaction Problem framework (CSP) (Dechter, 2003). We use the C-semiring (Stefano, 1997) and the CP-net (Craig, 2004) 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 outcomes satisfying the hard constraints and optimizing the preferences.

To the best of our knowledge there is no existing interactive system handling problem requirements as well as user preferences expressed in both a qualitative and a quantitative way. Moreover, our system is the first real attempt to cope and use both the C-semiring and the CP-net into the same interactive system.

The rest of the paper 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 (Carmel, 2006) 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.

2. 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 the customers 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 the customers preferences and state, and it predict the customers needs and desires (Peintner, 2008). In this regard, in Jiang (2012), the authors propose an interest-based offer evaluation system for semantic matchmaking. This system recommends the best offer by ranking the offers (that match the user's request) according to the users interests and preferences.

Handling preferences plays an important role in many interactive applications. In Peintner (2008), a discussion of interactive systems challenges has been surveyed. They broadly examined different available interactive systems along with their limitations. SmartClient (Pearl, 2000) is a planning tool to manage travelers activities. SmartClient provide list of suggestions by using example critiques to adapt to user satisfaction. It uses the CSP (Dechter, 2003) as a representation for the preference part of the problem. Teaching Salesman (Markus, 2004) is another interactive system which aids the client to compare diverse goods depending on how he matches their desire and needs. Preferences are represented through utility function. In (Ghavamifar, 2010) a preference elicitation model has been proposed for the winner determination problem in Multi-Attribute Reverse Auction (MARA). This model integrates both the the Multi-Attribute Utility Theory (MAUT) and constrained CP-nets in order to handle quantitative and quantitative constraints respectively.

Furthermore, in (Myrers, 2007) the authors propose the PExA system. The latter 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 provides some suggestions. In PExA, the preferences are expressed as an instance of soft constraint satisfaction problem. However, the problem of handling both qualitative and quantitative had not been addressed in the previous work. All the aforementioned work are 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 us to propose a new system for handling both quantitative and qualitative preferences in a friendly manner so the user can express his desires in a qualitative, quantitative or both forms. Table 1 lists some interactive systems along with their expressiveness in handling constraints and different forms of preferences.

Table 1. Comparison in terms of different types of information

Application	Quantitative	Qualitative	Mixed	Constraints
Amazon	Yes	No	No	No
Netflix	Yes	No	No	No
SmartClient	Yes	No	No	No
Teaching Salesman	Yes	No	No	No
Project Execution Assistant	Yes	No	No	No
Online Shopping System	Yes	Yes	Yes	Yes

3. Constraints and Preferences Representations

3.1 Conditional Preference Networks (CP-nets)

A Conditional Preferences network (CP-net) (Craig, 2004; Dimopoulos, 2003) 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 $Pa(X)$. Consider the CPT represented by the Figure 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 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 (Steve Prestwich, 2004). 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 1.

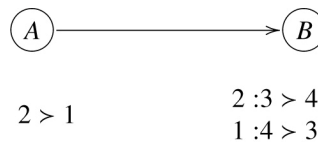


Figure 1. An example of conditional preferences network

3.2 Constraint Satisfaction and Soft Constraints

A Constraint Satisfaction Problem (CSP) (Dechter, 2003) 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 these constraints are satisfied. Unlike hard constraints, soft constraints are associated with degrees of satisfaction (Rossi, 2008) and the goal is to find the optimal outcome or solution. In Soft CSPs (SCSPs), an optimization problem is investigated in (Francesca, 2008) where the optimal solution is one solution with the best objective function according to the soft constraints. An SCSP (Stefano, 1995) 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 (Stefano, 1997; Rossi, 2008) (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 $(A, +, \times, 0, 1)$ where (Rossi, 2008): 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. \times 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 (Bistarelli, 1999). 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 minimum and the ultimate goal is to minimize the global cost of the problem: $WCSP = \{R^+, \min, +, +\infty, 0\}$. 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 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) .

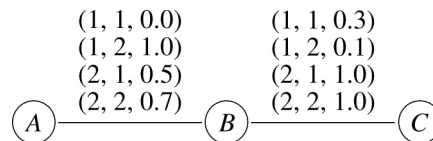


Figure 2. Soft constraint problem

3.3 Approximation Method

When given a CP-net N , it can be approximated to a Soft Constraint Satisfaction Problem (SCSP) (Carmel, 2006; Carmel, 2009). This approximation is favourable in many domains, specifically those domains where there is heavy interaction with the user as in online configuration problems (Carmel, 2006). 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 dominance (is 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 (Rossi, 2008; Carmel, 2006). The main characteristic with which to distinguish different approximations for a particular CP-net is the information preserving property (Prestwich, 2005). 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 (Carmel, 2009) shows the steps to create an SCSP graph with a set of hard and soft constraints and Algorithm 2 (Carmel, 2009) 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 acyclic 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
       $W_X = \sum W_y \cdot |dom(y)| \quad \forall y \in Children(X)$ 
  
```

Note that the approximation process approximates the entire CP-net to SCSP. This can lead to further computational time and the loss of precision (Steve, 2004). 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 acyclic CP-nets. 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 100 variables in the constraint problem and only 2 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) (Craig, 1997; Brafman, 2002; Wilson, 2011). However, there is no such thing forcing the preferences and constraints to be coupled.

The steps for converting acyclic CP-net to 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.

4. The Proposed Model

In this paper 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 online shopping 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.

4.1 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 is 1 GB and 2.3 GHz otherwise. For the hard drive options, he prefers hard drives with larger capacity. These information is transformed into CP-net as shown in Figure 3. For other variables, the user specified his preferences quantitatively as described in Figure 4.

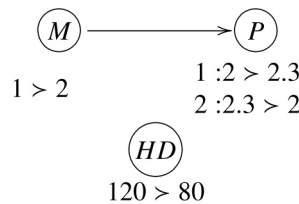


Figure 3. Qualitative part of the problem

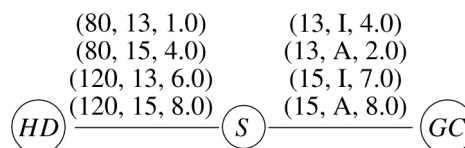


Figure 4. Quantitative part of the problem

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 5. In addition to the information shown in Figure 5, 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.

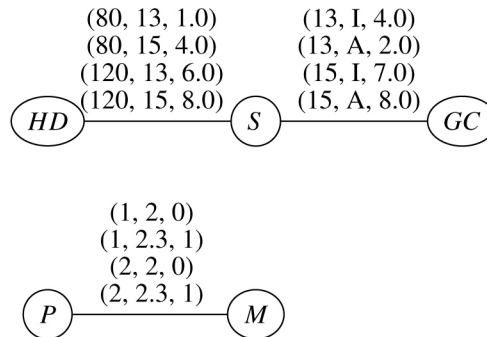


Figure 5. The complete network for the example

4.2 Branch and Bound Algorithm

The branch and bound algorithm is a systematic search method for discovering the best solutions to optimization problems (Dechter, 2003). We have used the implementation of branch and bound as shown in Algorithm 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 6, 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 6. 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** $S.\text{objective} < \text{objective}$
 $\text{objective} = S.\text{objective}$
 $\text{bestSolution} = S$
 - else**
 5. List $\text{next} = V.\text{children}$
 6. **for each** $\text{current} \in \text{next}$
 if current is consistent
 if $\text{current}.\text{objective} < \text{objective}$
 Branch and Bound(current)
-

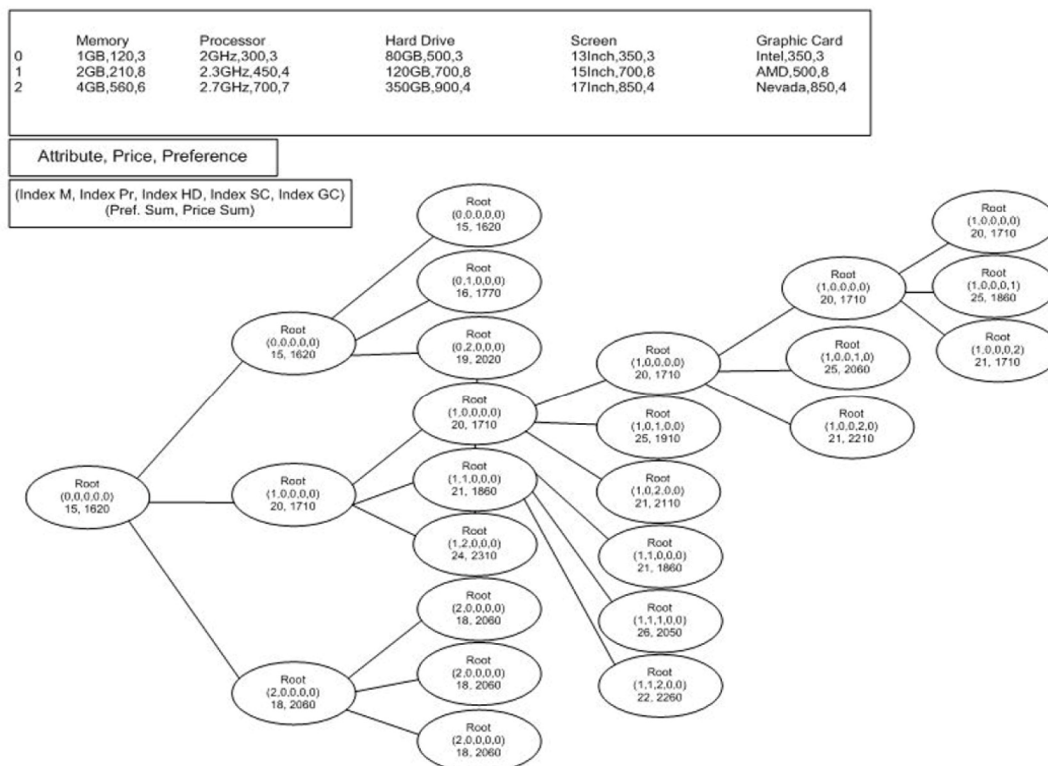


Figure 6. Execution of the branch and bound algorithm on a part of the search space

5. 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 7 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.

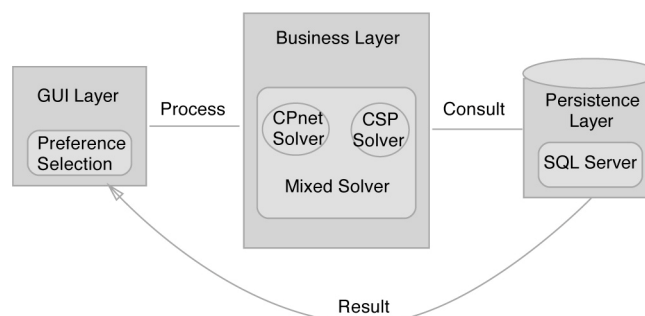


Figure 7. Architecture of our online system

Figure 8 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.

Figure 8. User preferences

Figure 9. User optimal solution

Considering the previous example in Section 4.1, Figure 9 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.

6. Experimentation

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 $y : x_1 > x_2$ in the $CPT(X)$ is a statement asserting that x_1 is preferred to x_2 when y holds. Figure 10 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 acyclic CP-nets (Francesca, 2008). For the mixed approach, it is clear that it requires more time due to the approximation process involved.

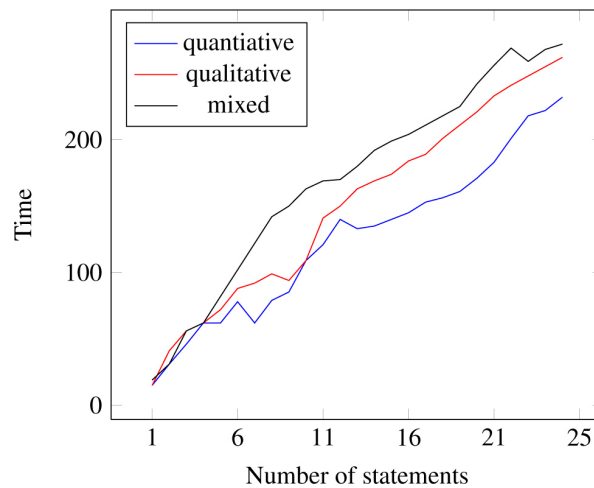


Figure 10. Experiments for the time needed to find optimal solution

7. Conclusion and Future Work

In this paper, we proposed an online shopping system based on user preferences and constraints. We considered the case where user preferences can take qualitative, quantitative or both forms. We used the C-semiring and the CP-net 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 online shopping 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, we can use a dynamic variant of the constraint solving techniques (Mouhoub, 2004; Mouhoub, 2012). In order to improve the time efficiency of our branch and bound technique, we will investigate several variable and value ordering heuristics (Mouhoub, 2011) that allow the branch and bound to return the optimal solution in a better running time. We will also explore other techniques based on parallel genetic algorithms (Abbasian, 2011) and ant colony optimization (Mouhoub, 2006). 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.

References

- Abbasian, R., & Mouhoub, M. (2011). An efficient hierarchical parallel genetic algorithm for graph coloring problem. In *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO 2011)*, ACM, pp. 521-528, Dublin, Ireland, July 12-16.
- Bart, P., Paolo, V., & Neil, Y. (2008). Preferences in Interactive Systems Technical Challenges and Case Studies. *AI Magazine*, 29(4), 13-25.
- Ben, J., & Schafer, J. A. (1999). Recommender systems in e-commerce. In *Proceedings of 1st ACM Conference on Electronic Commerce*, pp. 158-166. <http://dx.doi.org/10.1145/336992.337035>
- Bistarelli, S. U. M. (1999). Semiring-based cps and valued cps: Frameworks, properties, and comparison. *Constraints*, 4(3), 199-240. <http://dx.doi.org/10.1023/A:1026441215081>
- Brafman, C. D. (2002). Cp-nets - reasoning and consistency testing. In *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning, Morgan Kaufmann*, pp. 121-132.
- Carmel Domshlak, F. R. (2009). Reasoning about soft constraints and conditional preferences: complexity results and approximation techniques. *CoRR*, vol. abs/0905.3766.
- Carmel Domshlak, S. D. (2006). Hard and soft constraints for reasoning about qualitative conditional preferences. *J. Heuristics*, 12(4-5), 263-285. <http://dx.doi.org/10.1007/s10732-006-7071-x>
- Craig Boutilier, R. B. (1997). A constraint-based approach to preference elicitation and decision making. *AAAI spring symposium on qualitative decision theory*.

- Craig Boutilier, R. I. (2004). Cp-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res. (JAIR)*, 21, 135-191.
- Dechter, R. (2003). Constraint processing. Elsevier Morgan Kaufmann.
- Dimopoulos, R. I. (2003). A new look at the semantics and optimization methods of cp-networks. in IJCAI, pp. 1033-1038.
- Francesca Rossi, K. B. (2008). Preferences in constraint satisfaction and optimization. *AI Magazine*, 29(4), 58-68.
- Ghavamifar, F., Sadaoui, S., & Mouhoub, M. (2010). Winner Determination Based on Preference Elicitation Methods. Canadian Conference on Artificial Intelligence (AI'2010), pp. 332-335.
- Jiang, W., & Sadaoui, S. (2012). Evaluating and Ranking Semantic Offers According to Users' Interests. *Journal of Electronic Commerce Research. JECR*, 13(1), 1-22.
- Markus, S., & Michael, B. (2004). Recommending as personalized teaching: towards credible needs-based ecommerce recommender systems. Norwell, MA, USA: Kluwer Academic Publishers.
- Mouhoub, M. (2004). Systematic versus non systematic techniques for solving temporal constraints in a dynamic environment. *AI Communications*, 17(4), 201-211.
- Mouhoub, M., & Jafari, J. B. (2011). Heuristic techniques for variable and value ordering in CSPs. In *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO 2011)*, ACM, pp. 457-464, Dublin, Ireland, July 12-16.
- Mouhoub, M., & Sukpan, A. (2012). Managing Dynamic CSPs with Preferences. *Applied Intelligence*, 37(3), 446-462. <http://dx.doi.org/10.1007/s10489-012-0338-z>
- Mouhoub, M., & Wang, Z. (2006). Ant Colony with Stochastic Local Search for the Quadratic Assignment Problem. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*. pp. 143-149, Washington, November 13-15. <http://dx.doi.org/10.1109/ICTAI.2006.34>
- Myers, K. L., Berry, P., Bylthe, J., Conley, K., Gervasio, M. T., McGuinness D. L., ... Tambe, M. (2007). An Intelligent Personal Assistant for Task and Time Management. *AI Magazine*, 28(2), 47-61.
- Pearl, P., & Boi, F. (2000). Enriching buyers' experiences: the SmartClient approach. In the proceedings of the SIGCHI conference on Human factors in computing systems. pp. 289-296. The Hague, The Netherlands.
- Rossi, S. B. (2008). Semiring based soft constraints. in *Concurrency, Graphs and Models* , pp. 155-173.
- Prestwich, S. F. R. (2005). Constraint-based preferential optimization. in *Proceedings of the 20th national conference on Artificial intelligence*, Vol. 1, AAAI Press , pp. 461-466.
- Shoham, M. B. (1997). Fab: Content based. collaborative recommendation. *Communications of the ACM*, 40, 66-72. <http://dx.doi.org/10.1145/245108.245124>
- Stefano, B. U. M. (1995). Constraint solving over semirings. in *Proceedings of the 14th international joint conference on Artificial intelligence*. vol. 1, San Francisco, CA, USA , pp. 624-630.
- Stefano, B. U. M. (1997). Semiring-based constraint satisfaction and optimization. *Journal of the ACM*, 44(2), 201-236. <http://dx.doi.org/10.1145/256303.256306>
- Steve, P. F. R. (2004). Constrained cpnets. in *Proceedings of CSCLP04*.
- Stormer, U. L. (2007). Collaborative recommender systems for online shops. In *Proceedings of the Thirteenth Americas Conference on Information Systems*, Keystone (CO), USA, 08/2007.
- Tuzhilin, G. A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6), 734-749. <http://dx.doi.org/10.1109/TKDE.2005.99>
- Wilson, N. (2011). Computational techniques for a simple theory of conditional preferences. *Artif. Intell.*, 175(7-8), 1053-1091. <http://dx.doi.org/10.1016/j.artint.2010.11.018>