IJCT International Journal of Information & Communication Technology

ITRC

Volume 2- Number 4- September 2010 (9-18)

Towards Faster Performance of PROMETHEE II in a Lower Class of Complexity

Arash Niknafs
Department of Information Technology Engineering
Tarbiat Modares University
Tehran, Iran
arash.niknafs@gmail.com

Nasrollah Moghaddam Charkari
Faculty of Electrical and Computer Engineering
Tarbiat Modares University
Tehran, Iran
charkari@modares.ac.ir

Ali Akbar Niknafs
Department of Computer Engineering
Shahid Bahonar University of Kerman
Kerman, Iran
niknafs@uk.ac.ir

Received: August 1, 2010 - Accepted: September 21, 2010

***Abstract*—PROMETHEE II is one of the most popular members of the well-known family of multi-criteria decision-making methods. One of the main concerns in developing PROMETHEE-based systems is the rapid growth of the response time as the number of alternatives (*n*) and criteria (*k*) grow. PROMETHEE II belongs to the computational complexity class of $O(n^2)$. In this paper, a simplified version of PROMETHEE II is proposed and a novel estimation of the simplified PROMETHEE II is introduced. This simplified version reproduces the results of the original method and requires fewer operations. The estimation belongs to the complexity class of *O(n log n)* and consequently has a shorter response time than that of the simplified version.**

The proposed simplification and estimation are tested and evaluated with real-world data. When compared to the original PROMETHEE II and even other similar MCDM methods, such as AHP, ELECTRA, and TOPSIS, the experiments reveal the satisfactory results with a considerably reduced computational complexity and response time.

Keywords-multi criteria decision making; PROMETHEE II; decision support systems; recommender systems

I. Introduction

Multi-criteria decision making (MCDM) finds an optimal solution from all available alternatives evaluated on multiple (both qualitative and quantitative) and usually conflicting criteria [1]. A well-known family of MCDM methods is called PROMETHEE (Preference Ranking Organization METHod for Enrichment Evaluations) [2]. PROMETHEE II, which has been applied to different fields, is the most popular and widely used method among the members of this family.

One of the main concerns of developing information systems (e.g. Decision Support Systems (DSSs), Recommender Systems (RSs)) based on PROMETHEE II is its high computational complexity and consequently its long response time. The response time becomes a more critical issue in systems for which the number of alternatives in the MCDM problem is high. In other words, we face with a computational challenge when the problems are not of trivial dimension. In such circumstances, the high response time of PROMETHEE II strongly reduces the advantage of this method for information systems. Dias et al proposed a parallel processing architecture to deal with this issue [3]. Recommender systems, known as a type of decision-support system [4, 5], deal with a large number of alternatives. Consequently, we chose recommender systems to examine the performance of the two proposed modifications of PROMETHEE II. The application of MCDM methods within recommender systems has attracted researchers' attention. For example, PROMETHEE-based and AHP-based recommender systems were developed in [5] and [6], respectively.

Today, customers encounter an increasing amount of information in selecting products. In e-commerce, this is also referred to as the information overload problem [7]. One solution is the recommender system. Recommender systems support users by identifying interesting products and services in situations where the number and the complexity of offered items and services outstrips the user's capability to survey them and make a decision [8]. There are also a lot of applications of MCDM methods in the field of information technology and communication (ICT) about which there are some examples that are presented in section 5.

There are different approaches to categorizing recommender systems. The most common approach suggests that recommender systems can be divided into three categories [7, 9, 10]: content-based, collaborative filtering (CF), and a hybrid.

Content-based recommender systems store the content of information about each item to be recommended. This information will be used to recommend items similar to those the user has preferred in the past [11]. CF systems make recommendations based on their past experiences. They focus on the similarity among users or the similarity among items using users' ratings [12]. Hybrid recommender systems combine two or more techniques, and in this way they overcome the weaknesses of separately applying each technique [13, 14]. The recommender system developed in our research is a content-based system.

The following three MCDM methods were used for evaluation and comparison purposes in this paper: 1- AHP [15] 2- ELECTRE III [16] (In order to get a complete ranking in ELECTREE III, we used the method proposed in [17] that combines the concordance and discordance indexes.) 3- TOPSIS [18].

Any increase in the number of alternatives causes a big increase in the response time, which discourages researchers from using this method. Yet, some interesting advantages of PROMETHEE II in a recommender system motivated us to investigate reducing the response time and the computational complexity class of this method.

The purpose of this paper is to reduce the complexity class of PROMETHEE II ($O(n^2)$). First, a simplified version of PROMETHEE with a shorter response time is introduced. Then an estimation ($O(n \log n)$) of the simplified version is proposed. A prototype system is developed to evaluate and analyze the performance of the proposed modifications of PROMETHEE II. The results show a considerable reduction in the computational complexity and response time while outputs maintain a high level of correlation with that of (simplified) PROMETHEE II. Estimated PROMETHEE II can be used in complex decision-making problems where a great number of alternatives and criteria exist.

The structure of the remainder of this paper is as follows. Section 2 describes PROMETHEE II and the proposed simplified PROMETHEE II. The estimation of the simplified version is discussed in this section as well. Section 3 explains the architecture of the system and the implementation issues. Section 4 shows the experimental results. Finally, Section 5 draws some conclusions and discusses future work.

II. THE ORIGINAL PROMETHEE II

In this section a brief description of PROMETHEE II [19] is presented.

PROMETHEE uses preference functions to form the PCM matrix. Moreover, in contrast to AHP, the preference scores are not limited to the scale of 1 to 9 (integers) in this method. In PROMETHEE, $P_j(a,b)$ denotes the preference of a over b with regards to criterion j. It is computed by $H_j(d_j)$, where H_j is one of the functions shown in Appendix A. $d_j = f_j(a) - f_j(b)$ is the difference between the values of two alternatives at criterion j.

Suppose a and b are two items that belong to A (a set of n items) and k is the number of criteria. Let $f_j(a)$ denotes the value of item a at criterion j, $(0 < j < k)$. The following steps derive the optimal solution using PROMETHEE II:

(1) "*Identify the preference function P_j and H_j for each criterion.*"

Depending on whether a criterion in our MCDM problem gets maximized or minimized, the preference function of a over b at criterion j is defined as follows (where $d = f_j(a) - f_j(b)$ and $H_j(d)$ is one of the six functions described in Appendix A). If criterion j is being maximized, then:

$$P_j(b,a) = 0 \text{ and } P_j(a,b) = H_j(d) \quad \text{if } (d > 0)$$
$$P_j(b,a) = H_j(d) \text{ and } P_j(a,b) = 0 \quad \text{if } (d < 0)$$

Similarly, if it is being minimized, then:

$$P_j(a,b) = 0 \text{ and } P_j(b,a) = H_j(d) \quad \text{if } (d > 0)$$
$$P_j(a,b) = H_j(d) \text{ and } P_j(b,a) = 0 \quad \text{if } (d < 0)$$

(2) "*Weight of each criterion (w_j)*".

Here, the relative importance of each criterion is used to weigh the criteria.

(3) "*Calculate the outranking degree of a relative to b*".

The higher the value of the outranking degree ($\pi(a,b)$), the more a is preferred.

$$\pi(a,b) = \frac{1}{W}\sum_{j=1}^{k} w_j P_j(a,b) \quad , \quad \mathrm{W} = \sum_{j=1}^{k} w_j$$

(4) *"Calculate the complete preorder (net flow), $\varphi(a) = \varphi^{+}(a) - \varphi^{-}(a)$, for each item".*

$$\varphi^{+}(a) = \frac{1}{n-1} \sum_{\substack{b \in A, \\ b \neq a}} \pi(a,b)$$

$$\varphi^{-}(a) = \frac{1}{n-1} \sum_{\substack{b \in A, \\ b \neq a}} \pi(b,a)$$

The item with the highest net flow value will be the best item for recommendation. The items are sorted according to their net flow values in the final ranked list.

A. *Simplified PROMETHEE II*

Any increase in the number of alternatives or criteria will increase the number of operations and consequently response time in PROMETHEE II (which is $3kn^2$). Here a simplified version of PROMETHEE II, which requires a smaller number of operations to compute the net flow of an item, is introduced. First, we present our new definition of function H followed by Theorem 1which describes how the simplified version is formulated.

Definition 1. Suppose $H'_j(d_j)$ denotes the original definition of function $H(d)$ for criterion j. Let $d_j = f_j(a) - f_j(b)$, then $H(d)$ in the simplified version of PROMETHEE II is defined as follows (Ref. to Appendix A):

$$H_j(d_j) = \begin{cases} H'_j(d_j) & d_j \geq 0 \\ -H'_j(d_j) & d_j < 0 \end{cases}$$

Theorem 1. Let $d_h = f_h(a) - f_h(b)$ and:

$i = \{t \in N \mid 0 \leq t \leq k, k = \#$ of criteria, criterion t to be minimized$\}$

$j = \{t \in N \mid 0 \leq t \leq k, k = \#$ of criteria, criterion t to be maximized$\}$

Using the new definitions of $H(d)$, all the steps in PROMETHEE II, for net flow calculations, can be abstracted to:

$$\varphi(a) = \frac{1}{W(n-1)} [\sum_{x \neq a} \sum_{j=1} w_j H_j(d_j) - \sum_{x \neq a} \sum_{i=1} w_i H_i(d_i)]$$

Proof. From the description of the original PROMETHEE II in section 2, it can be inferred that:

$$\varphi(a) = \frac{1}{n-1} \sum_{\substack{b \in A, \\ b \neq a}} \pi(a,b) - \frac{1}{n-1} \sum_{\substack{b \in A, \\ b \neq a}} \pi(b,a) =$$

$$\mathrm{k} = \frac{1}{n-1} \sum_{\substack{b \in A, \\ b \neq a}} (\frac{1}{W} \sum_{h=1}^{k} w_h P_h(a,b)) - \frac{1}{n-1} \sum_{\substack{b \in A, \\ b \neq a}} (\frac{1}{W} \sum_{h=1}^{k} w_h P_h(b,a)) =$$

$$= \frac{1}{n-1} * \frac{1}{W} (\sum_{\substack{b \in A, \\ b \neq a}} \sum_{h=1}^{k} (w_h P_h(a,b) - w_h P_h(b,a))$$

Whether h has to be maximized or minimized, one of the two functions $P_h(a,b)$ and $P_h(b,a)$ always becomes zero while the other one will be equal to $H'_h(d_h)$, where $d_h = f_h(a) - f_h(b)$. This means:

$\forall b \in A,\ b \neq a$:

$P_h(a,b) - P_h(b,a) =$

$$= \begin{cases} +H'_h(d_h) & (\mathrm{d_h} > 0 \& h\,\text{maximized})\text{or}(\mathrm{d_h} < 0 \& h\,\text{minimized} \\ -H'_h(d_h) & (\mathrm{d_h} > 0 \& h\,\text{minimized})\text{or}(\mathrm{d_h} < 0 \& h\,\text{maximized} \end{cases}$$

Changing the definition of $H'_j(d)$, as indicated in Definition 1, the following is valid:

$$\forall b \in A\ ,\ b \neq a:$$

$$\varphi(a) = \frac{1}{W(n-1)} [\sum_{x \neq a} \sum_{j=1} w_j H_j(d_j) - \sum_{x \neq a} \sum_{i=1} w_i H_i(d_i)]$$

As Theorem 1 suggests the net flow can be calculated by simply using the above formula, which requires a smaller number of operations to provide the same output as PROMETHEE II.

B. *An Estimation of Simplified PROMETHEE II*

Although the simplified PROMETHEE II requires fewer operations than PROMETHEE II, its computational complexity order is the same as PROMETHEE II. In this regard, we introduce the estimated PROMETHEE II, which not only has a shorter response time but also has lower class of complexity. In this section the estimation of the simplified PROMETHEE II is presented. The estimation works much faster than the simplified version and produces outputs that are highly correlated with (simplified) PROMETHEE II outputs.

Theorem 2.

For each increasing function like H the following is true:

$$\forall a, x \in A\ , \mathrm{x} \neq a:$$

$$\sum_x H_t(f_t(a) - f_t(x)) > \sum_x H_t(f_t(b) - f_t(x)) \Rightarrow$$

$$f_t(a) > f_t(b)$$

Proof.

$$\forall d_t, d_{t+1}, t \in N \text{ and } t < n:$$

$$if\,(H(d_t) > H(d_{t+1})\ then: d_t > d_{t+1}$$

$$\sum_x H_t(f_t(a) - f_t(x)) > \sum_x H_t(f_t(b) - f_t(x))$$

$$\Rightarrow \sum_x f_t(a) - f_t(x) > \sum_x f_t(b) - f_t(x) \Rightarrow$$

$$\sum_x f_t(a) - f_t(x) - f_t(b) + f_t(x) > 0 \Rightarrow$$

$$\sum f_t(a) - f_t(b) > 0 \Rightarrow\ f_t(a) > f_t(b)$$

If we could generalize Theorem 2 to:

$$\sum_t \sum_x w_t H_t(f_t(a) - f_t(x)) > \sum_t \sum_x w_t H_t(f_t(b) - f_t(x))$$

$$\Rightarrow \sum_t f_t(a) > \sum_t f_t(b)$$

Then we could infer that:

$$\varphi(a) > \varphi(b) \Rightarrow$$

$$\sum_j w_j f_j(a) - \sum_i w_i f_i(a) > \sum_j w_j f_j(b) - \sum_i w_i f_i(b)$$

However the above mentioned generalization is not always true. That is because $f_k(a)$ at each criterion has a different scale, meaning and concept. As a primary step towards introducing an estimation of net flow ($\varphi_{est.}(a)$), the following introduces the relationship (function) T, which maps $f_k(a)$ values into a unit scale.

Suppose *f(i,k)* represents the value of item *i*, ($i = \{0,1,...,n-1\}$) at criterion *k*, ($k = \{1,...,K\}$) then through the following steps, the values of each item at different criteria are mapped into a new scale with the range of 1 to n (underlined values show the # of operations in each step). Fig. 1 depicts the pseudo code for these steps.

Step 1. Sort all values at each criterion and assign the indices of 1 to n to the corresponding items (in each criterion separately).

If a criterion is meant to be maximized (minimized), the items are sorted ascending by their values in that criterion. (descending). ($\underline{kn\log n}$)

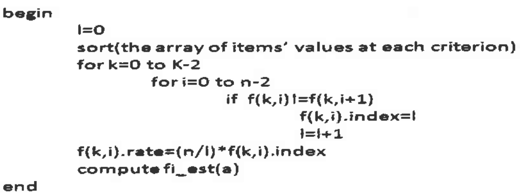
Step 2. Now we count the number of unique values at each criterion; this number is denoted by l. Then the sorted array of values of items at a certain criterion is divided into l parts ($0 < l \le n$). Starting from the first (smallest) index in the sorted array, all parts are indexed with a unique number from 1 to l. ($\underline{kn + kn}$)

Step 3. To each item ($\forall a \in A$) in each of the above-mentioned l parts, a rating value of $R(a) = \frac{n}{l} * t$ is assigned; where $t = \{1,2,...,l\}$ denotes the index of the part (Ref. to step 2) to which an item belongs. ($\underline{kl}$)

Step 4. Now that we have mapped $f_k(a)$ values to $R_k(a)$ through the strictly increasing function $T_k(f_k(a))$, ($T_i : f_i(a) \rightarrow R_i(a)$), an estimation of $\varphi(a)$ can be denoted as follows: ($\underline{k}$)

$$\sum_{j=1} w_j T_j(f_j(a)) - \sum_{i=1} w_i T_i(f_i(a))$$

The following Theorem, which is based on the above-mentioned mapping relationship, introduces the proposed estimation of the simplified version of PEOMETHEE II. While the simplified PROMETHEE II has the same accuracy (output) as the original one, the proposed estimation of PROMETHEE II does not provide exactly the same outputs (a compromise has been made between response time and accuracy in following sections).



```
begin
        l=0
        sort(the array of items' values at each criterion)
        for k=0 to K-2
                for i=0 to n-2
                        if  f(k,i)!=f(k,i+1)
                                f(k,i).index=l
                                l=l+1
        f(k,i).rate=(n/l)*f(k,i).index
        compute fi_est(a)
end
```

Figure 1. Pseudo code for the mapping process & computing $\varphi_{est.}(a)$.

Theorem 3.

$$\forall a,b \in A \text{ , } b \neq a:$$

$$\varphi(a) > \varphi(b) \Rightarrow$$

$$\sum_j w_j T_j(f_j(a)) - \sum_i w_i T_i(f_i(a)) > \sum_j w_j T_j(f_j(b)) - \sum_i w_i T_i(f_i(b))$$

Proof.

$$\varphi(a) = \frac{1}{W(n-1)}[\sum_{x \neq a} \sum_{j=1} w_j H_j(d_j) - \sum_{x \neq a} \sum_{i=1} w_i H_i(d_i)]$$

From Theorem 1 it can be concluded that:

If: $\varphi(a) > \varphi(b)$, then:

$$\sum_{x \neq a} \sum_{j=1} w_j H_j(f_j(a) - f_j(x)) - \sum_{x \neq a} \sum_{i=1} w_i H_i(f_i(a) - f_i(x))$$

$$>$$

$$\sum_{x \neq a} \sum_{j=1} w_j H_j(f_j(b) - f_j(x)) - \sum_{x \neq a} \sum_{i=1} w_i H_i(f_i(b) - f_i(x))$$

Using the above mentioned mapping technique and Theorem 2:

$$\sum_{x \neq a} \sum_{j=1} w_j (T_j(f_j(a)) - T_j(f_j(x))) - \sum_{x \neq a} \sum_{i=1} w_i (T_i(f_i(a)) - T_i(f_i(x)))$$

$$>$$

$$\sum_{x \neq a} \sum_{j=1} w_j (T_j(f_j(b)) - T_j(f_j(x))) - \sum_{x \neq a} \sum_{i=1} w_i (T_i(f_i(b)) - T_i(f_i(x)))$$

$$\Rightarrow$$

$$\sum_j w_j T_j(f_j(a)) - \sum_i w_i T_i(f_i(a)) > \sum_j w_j T_j(f_j(b)) - \sum_i w_i T_i(f_i(b))$$

Theorem 3 suggests $\varphi_{est.} = \sum_j w_j T_j(f_j(a)) - \sum_i w_i T_i(f_i(a))$ as an estimation of $\varphi(a)$ in PROMETHEE II. Based on this estimation, net flows for a set of alternatives can be calculated a smaller number of operations.

III. THE PROTOTYPE'S DESIGN ATTRIBUTES

In PROMETHEE, the first step is to identify decision criteria and corresponding $H(d)$ function types. The set of selected criteria should be complete and not redundant. This means that all major aspects should be taken into consideration, while keeping the number of criteria as small as possible; a double counting of impacts should also be avoided [20]. The criteria were selected from a set of studies in the literature of business and marketing [21-27] (considering the fact that the system is going to recommend from among different, less frequently purchased items) and then discussed with experts. The following four criteria were selected: overall reviewer (previous users/customers) ranking, price, brand reputation, and customer's interests.

A prototype of the system is developed in a single window with different tabs. In the first tab, users specify the weights of each factor (criterion) on a scale of 1 to 7 (integers). In the second tab, users specify their favorite categories by choosing from a list. Customer purchased items are then added to a cart in the third tab by their corresponding ID number in the database. The fourth tab, as shown in Fig. 2, is used to recommend items to the customer with details on the product features so that the customer can provide feedback based on the detailed information on each item.

The information about all items is stored in a table in SQL Server 2000 on the local host. The item's ID number is set as primary key of the table. Part of the data table which stores items' information in the database is shown in Fig. 3.

IV. Experiments and Results

This section describes some experiments conducted on real-world data to measure the response time and the deviation of outputs of the two proposed modifications from that of the original PROMETHEE II. A prototype system was developed by Visual C# .NET 2005 and SQL Server 2000 Enterprise Edition.

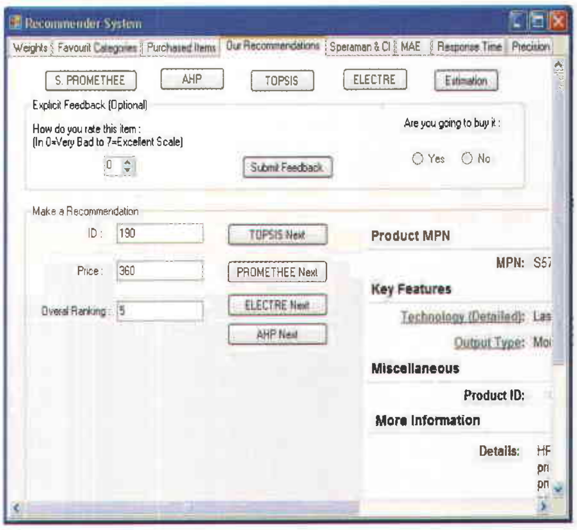
A. *Evaluation measures*

Three metrics, the Spearman rank correlation, response time (in seconds) and mean absolute error (MAE), are commonly used in related works to measure correlation, speed and deviation respectively.

The first evaluation measure is the Spearman's rank correlation coefficient (R) or the Spearman rho (ρ), a non-parametric measure of correlation, which was first introduced in [28] and since then it has been used in many studies, such as [7, 29]. It measures the relationship between two sets of ordinal (ranked) values. The Spearman rho has two advantages. First, a normal distribution is not necessary because it is a nonparametric measure, and, second, it is less affected by outliers [30]. R is computed by the following formula:

$$\mathrm{R} = 1-\frac{6*\sum D_a^2}{A(A^2-1)}$$

$D_a = U_a - V_a$, where U_a and V_a are ranks denoted by two different MCDM methods for an item like a; A $(a=1,2,\ldots,A)$ is the number of alternatives. The greater the value of R is, the more the results of the two methods are correlated.


Recommender System

Explicit Feedback (Optional)

How do you rate this item :
(In 0=Very Bad to 7=Excellent Scale)
0

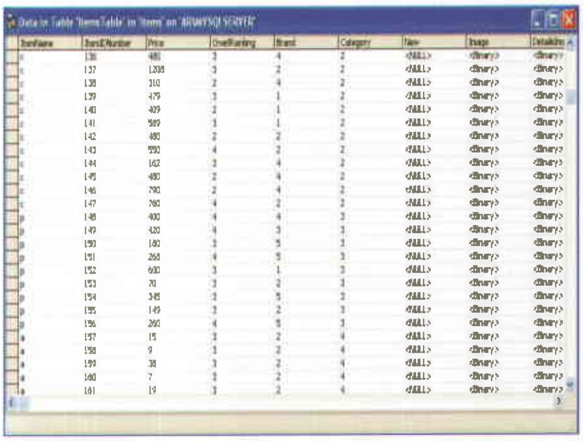
Are you going to buy it :
○ Yes ○ No

Make a Recommendation

ID : 190
Price : 360
Overal Ranking : 5

Product MPN
MPN: S57
Key Features
Technology (Detailed): Las
Output Type: Mo
Miscellaneous
Product ID:
More Information
Details: HF
pri
pri

Figure 2. Interface of the implementation.


Data in Table 'ItemsTable' in 'Items' on 'ARIAMYSQLSERVER'

ItemName	ItemIDNumber	Price	OverallRanking	Brand	Category	New	Image	DetailsIn
c	136	480	3	4	2	<NULL>	<Binary>	<Binary>
c	137	1308	3	2	2	<NULL>	<Binary>	<Binary>
c	138	310	2	4	2	<NULL>	<Binary>	<Binary>
c	139	479	3	1	2	<NULL>	<Binary>	<Binary>
c	140	409	2	1	2	<NULL>	<Binary>	<Binary>
c	141	589	3	1	2	<NULL>	<Binary>	<Binary>
c	142	480	2	2	2	<NULL>	<Binary>	<Binary>
c	143	550	4	2	2	<NULL>	<Binary>	<Binary>
c	144	162	3	4	2	<NULL>	<Binary>	<Binary>
c	145	480	2	4	2	<NULL>	<Binary>	<Binary>
c	146	790	2	4	2	<NULL>	<Binary>	<Binary>
c	147	760	4	2	2	<NULL>	<Binary>	<Binary>
p	148	400	4	4	3	<NULL>	<Binary>	<Binary>
p	149	420	4	3	3	<NULL>	<Binary>	<Binary>
p	150	180	3	5	3	<NULL>	<Binary>	<Binary>
p	151	268	4	5	3	<NULL>	<Binary>	<Binary>
p	152	600	3	1	3	<NULL>	<Binary>	<Binary>
p	153	70	3	2	3	<NULL>	<Binary>	<Binary>
p	154	345	3	5	3	<NULL>	<Binary>	<Binary>
p	155	149	3	2	3	<NULL>	<Binary>	<Binary>
p	156	260	4	5	3	<NULL>	<Binary>	<Binary>
[illegible]	157	15	3	2	4	<NULL>	<Binary>	<Binary>
[illegible]	158	9	3	2	4	<NULL>	<Binary>	<Binary>
[illegible]	159	38	3	2	4	<NULL>	<Binary>	<Binary>
[illegible]	160	7	3	2	4	<NULL>	<Binary>	<Binary>
[illegible]	161	19	3	2	4	<NULL>	<Binary>	<Binary>

Figure 3. Part of data table in the database in SQL Server.

To measure how close the performance of the system is to the original PROMETHEE II, (Simplified PROMETHEE II) we use mean absolute error (MAE), which measures the average of absolute differences between ranks assigned by any pair of MCDM methods. MAE has been used in different works [11, 31, 32] to evaluate rank-based recommender systems.

$$MAE = \frac{\sum_{i=1}^{N} |p_i - r_i|}{N}$$

p_i and r_i are ranks of item i in two different MCDM methods and N is the number of all ranked items.

Response time (RT), in seconds, is one of the evaluation metrics of algorithms and systems [33]. RT shows the time that each method takes to produce a final result (a ranked list of items).

Herein, the precision measure is also adopted to measure accuracy and compare different methods' outputs. The precision (P) is the ratio of erroneous estimates of rating to the correct ratings [34].

$$P = 1-\left(\sum_{k=1}^{N} \frac{|p_k - r_k|}{\max(|R_{\max} - r_k|, |R_{\min} - r_k|)}\right)/N$$

Where $R_{\max}$ and $R_{\min}$ are the maximum and minimum possible ranks for an item, respectively.

B. Experiment settings and data

The main experimental goals were initially to examine the accuracy, and the degree to which the proposed modifications (simplification and estimation) contribute to response time reduction. For this purpose a prototype of the system was developed using C# .NET 2005 and SQL Server 2000 on a Pentium PC platform (1.7 G CPU, 256 M of RAM).

A set of real world data (from an electronics store database) was used in the experiments. A total of 500 records (items) from 15 different categories (such as PCs, scanners, printers) were stored in a table in SQL Server 2000 on the local host. Each item has a unique ID number. The column corresponding to *IDNumber* in the table design was set as the primary key. The other four important columns in the table contain information about price, brand reputation, category number and average of other customers' ratings.

In order to evaluate and compare the performance of the proposed modifications, the main part (core) of the recommender system, which is its MCDM algorithm, was developed in six different scenarios using four MCDM methods (AHP, ELECTRE III, PROMETHEE II and TOPSIS), the proposed simplification and estimation of PROMETHEE II. We have asked 30 users to use the system. In all iterations, the evaluation parameters (Precision, MAE, Response time and Spearman rank correlation) were calculated. The results are presented in the following subsection.

C. Results

As we discussed before, the simplified PROMETHEE II, contrary to the estimated version, produces exactly the same outputs as those of PROMETHEE within a relatively shorter time. There is also a compromise between speed and accuracy in estimated PROMETHEE. Table 1 shows the correlation between the results of the estimation and those of the other MCDM methods.

MAE shows the deviation of the results of each method from the results of the other methods. The MAE of the estimation outputs from those three MCDM methods' outputs is shown in table 3. As it is shown in table 3, the range of the MAE values is below 17%. It is important to note that larger MAE values have been considered acceptable in other studies [11, 31, 32].

To show the level of confidence in the results in tables 1, 3 and 6, each of those tables is followed by a similar table (table 2, table 4 and table 7) that includes the confidence interval (CI) of the elements in its preceding table.

Table 1. THE AVERAGES OF SPEARMAN RANK CORRELATION COEFFICIENTS (SRCC) BETWEEN THE OUTPUTS OF DIFFERENT MCDM METHODS

Method	AHP	TOPSIS	ELECTRE	Simplified	Estimation
Estimation	0.7620	0.7358	0.8027	0.7414	1

Table 2. 95% CONFIDENCE INTERVAL (CI) OF SPEARMAN RANK CORRELATION COEFFICIENT(SRCC) AVERAGES BETWEEN THE OUTPUTS OF DIFFERENT MCDM METHODS

Method	AHP	TOPSIS	ELECTRE	Simplified	Estimation
Estimation	± 0.03	± 0.04	± 0.02	± 0.03	0

Table 3. AVERAGES OF MAE (%) BETWEEN THE OUTPUTS OF DIFFERENT MCDM METHODS

Method	Estimation	Simplified	TOPSIS	ELECTRE	AHP
Estimation	1	15.491	15.037	12.971	14.539
Simplified		1	9.319	16.650	12.776
TOPSIS			1	15.841	13.388
ELECTRE				1	13.331
AHP					1

Table 4. 90% CONFIDENCE INTERVAL(CI) OF AVERAGES OF MAE BETWEEN THE OUTPUTS OF DIFFERENT MCDM METHODS

Method	Estimation	Simplified	TOPSIS	ELECTRE	AHP
Estimation	0	± 0.92	± 1.23	± 0.7	± 1.04
Simplified		0	± 0.98	± 1.29	± 0.68
TOPSIS			0	± 1.52	± 0.89
ELECTRE				0	± 0.72
AHP					0

Beside the lower complexity class to which the estimation of (simplified) PROMETHEE II belongs, the results in table 5 indicate a shorter response time for both the estimated and simplified versions compared to PROMETHEE II. The estimation and TOPSIS rank first and second with regards to response time.

Finally, to show the competitive accuracy of the estimated and simplified versions, a pair wise comparison has been made between all methods with respect to the precision measure. The results are shown in table 6.

In general, there are two parameters affecting the complexity of these algorithms: the number of criteria k and the number of items (alternatives) n . Fig. 4 illustrates the number of operations from two opposite points of view for PROMETHEE II, simplified PROMETHEE, and the estimated version, respectively. The uppermost surface represents the number of operations in PROMETHEE II ($3kn^2$). The surface in the middle and the one under that represent the number of operations in simplified (kn^2) and estimation ($kn\log n + 3kn + kl$) versions, respectively.

Table 5. AVERAGES OF RESPONSE TIMES (SECONDS)

Method	Avg. response time
Estimation	1.2578
Simplified	694.467
ELECTRE	1393.097395
TOPSIS	1.8692
AHP	691.618

Table 6. PRECISION AVERAGES BETWEEN THE OUTPUTS OF DIFFERENT MCDM METHODS

Method	Estimation	Simplified	TOPSIS	ELECTRE	AHP
Estimation	1	0.7790	0.7828	0.8117	0.7903
Simplified		1	0.8726	0.7546	0.8153
TOPSIS			1	0.7671	0.8057
ELECTRE				1	0.8064
AHP					1

Table 7. 90%CONFIDENCE INTERVAL(CI) OF PRECISION AVERAGES BETWEEN THE OUTPUTS OF DIFFERENT MCDM METHODS

Method	Estimation	Simplified	TOPSIS	ELECTRE	AHP
Estimation	0	± 0.01	± 0.02	± 0.01	± 0.01
Simplified		0	± 0.01	± 0.02	± 0.01
TOPSIS			0	± 0.02	± 0.01
ELECTRE				0	± 0.01
AHP					0

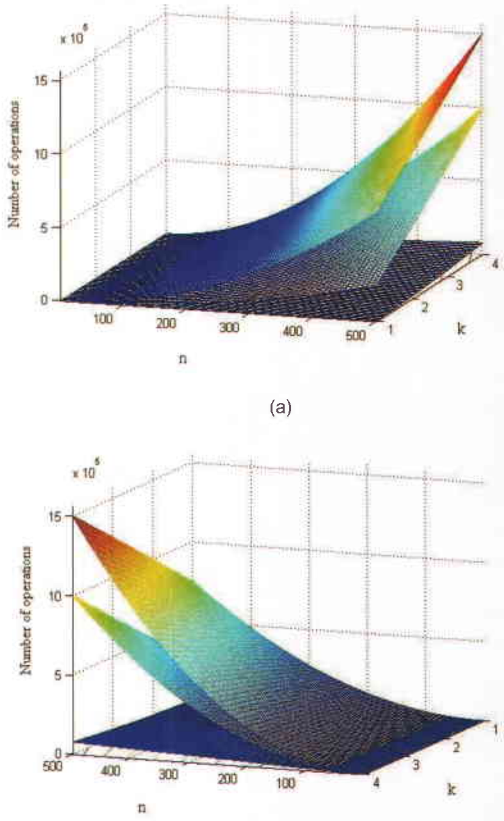
V. CONCLUSIONS AND FUTURE WORK

Although PROMETHEE II is popular for solving MCDM problems, it is not recommended for problems with a large number of criteria and alternatives (e.g. a problem with more than a total number of 100 alternatives and criteria). PROMETHEE's response time drastically increases with any increase in the number of criteria and (especially) alternatives. Its complexity class may discourage system designers from using this method in applications such as decision support and recommender systems despite its advantageous.

Some application-specific advantages of PROMETHEE II in recommender systems, which motivated us to modify and use the method, are:

(a) From a seller's point of view, recommending an item can be seen as a decision-making problem which requires a MCDA (Multi Criteria Decision Analysis) method with the following two criteria [35]: (1) the method does not allow for trade-offs between criteria, (2) it is simple enough to be understood by non-specialist users. PROMETHEE is identified as a suitable method for these purposes and it satisfies both of those criteria.

(b) As discussed in [7], most of the rank-based recommender systems require user item ratings to collect user preferences. Using the alternative approach, the user can express preference through multiple lists of ranked items. However, it is not feasible to ask a customer to rate all items in a store. Moreover, an underlying assumption of the previous rating scheme is that the user can compare the preference between any two rated items, i.e., the user either rates the two as equal or one better than the other. Imagine a situation in which the user knows that $\tau_a > \tau_b > \tau_c$ and $\tau_a > \tau_e > \tau_f$ but cannot differentiate the interest between items τ_b and τ_e or items τ_b and τ_f. PROMETHEE automatically makes the pair wise comparison helping the user out of this time consuming process while it can also compare any two items consistently—avoiding human errors—and without difficulty.



(a)

Figure 4. Comparing the number of operation with respect to parameters k and n. k and n are the number of criteria and the number of items respectively.

(c) As indicated in [33], one of the major concern in recommender systems is the issue of trust in the recommender. This trust is provided by means of transparency. Transparency includes explaining why specific recommendations appear and helping user understand the recommendation process [36, 37]. As suggested in [35, 38, 39] the PROMETHEE method provides this transparency and increases the customer's trust in the recommender by being simple and understandable.

The advantages of applying PROMETHEE II in decision support and recommender systems motivated us to work on reducing its complexity. In this paper, first a simplified version of PROMETHEE II was proposed, which worked faster than the original version. The simplified PROMETHEE II does not alter accuracy; it produces the same outputs as the original method does. Then, an estimation of the simplified method was presented which works much faster than the simplified version and belongs to a lower complexity class.

It was not our goal to find the most robust methods but to verify the reliability of the proposed methods. Therefore, to measure the deviation of the estimation outputs from that of the PROMETHEE II, the MAE between those two methods has been found, while the precision and the Spearman rank correlation are used to show the degree of relevance between their outputs. Calculating the same parameters for other MCDM methods was mainly done in order to prove the competency of two proposed modified versions (simplified and estimation) through comparison. However, a pair wise comparison of performance between all of the MCDM methods is also provided.

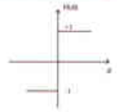
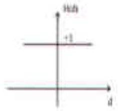
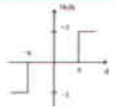
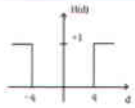
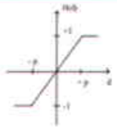
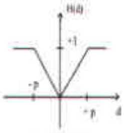
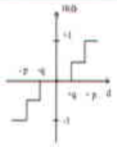
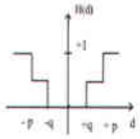
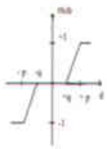
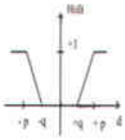
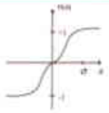
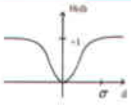
Fig. 4 shows that for a small number of alternatives ($n<100$), the number of required operations in all three versions is very close when compared to more complex situations $(n>100)$. It can also be inferred from Fig. 4 that the distance between the upper surface (the number of operations in PROMETHEE II) and the middle one (The number of operations in the simplified PROMETHEE II) gets bigger for a small number of criteria and a large number of alternatives, which is an indication of an even faster performance for the latter in such situations.

Recommending new items is a challenging issue in recommender systems. Our future work includes proposing a solution to the *new item* problem through manipulation of preference functions and $H(d)$ threshold parameters, which are the most flexible parts of PROMETHEE II. It will be also interesting to take a fuzzy solution approach towards this sort of recommendation problems because of the fuzzy nature of customers' interests and their uncertainties. Using Fuzzy MCDM is also another interesting topic that we are considering as a potential future work.

Finally, it is definitely worth mentioning that the application of multi-criteria decision making is not restricted to a specific field. Decision making is an important aspect of our daily life in different areas and information technology and communication is not exempt. The use of MCDM methods in ICT decision makings has received more attention in the recent years. For instance in [40] the application of MCDM methods in development of information technology industry has been studied. [41] uses these methods for evaluating mobile alternatives in the communication services while [42] uses them in information technology allocation problems. These are just a few examples but we hope that our research results can contribute to faster MCDM processes with less computational complexities in all fields of applications.

APPENDIX A

$H(d)$ returns a value within the range [0, 1]. The following table shows both the original six functions introduced in [19, 39] and the proposed function definitions in the simplified version. An index (j) is used to denote the criterion to which a specific type of function $H(d)$ is assigned (H_j , $j=$ criterion index). [19] suggests that these six types cover most of the cases occurring in practice.

$H(d)$ in the simplified and estimated versions	$H(d)$ in the original version of PROMETHEE II	
		Type I : Usual criterion $H(d)=\begin{cases}0 & \text{d}=0\\1 & \lvert d\rvert>0\end{cases}$
		Type II: U-shape criterion $H(d)=\begin{cases}0 & \lvert d\rvert\le q\\1 & \lvert d\rvert>q\end{cases}$
		Type III: V-shape criterion $H(d)=\begin{cases}\frac{\lvert d\rvert}{p} & \lvert d\rvert\le q\\1 & \lvert d\rvert>q\end{cases}$
		Type IV: Level criterion $H(d)=\begin{cases}0 & \lvert d\rvert\le q\\\frac{1}{2} & q<\lvert d\rvert\le p\\1 & \lvert d\rvert>p\end{cases}$
		Type V: V-shape criterion with indifference criterion $H(d)=\begin{cases}0 & \lvert d\rvert\le q\\\frac{\lvert d\rvert-q}{p-q} & q<\lvert d\rvert\le p\\1 & \lvert d\rvert>p\end{cases}$
		Type VI: Gaussian criterion $H(d)=1-e^{-d^2/2\sigma^2}$

REFERENCES

[1] S. D. Pohekar and M. Ramachandran, "Application of multi-criteria decision making to sustainable energy planning—A review," Renewable and Sustainable Energy Reviews, vol. 8, pp. 365-381, 2004.

[2] C. Macharis, J. Springael, K. D. Brucker and A. Verbeke, "PROMETHEE and AHP: The design of operational synergies in multi criteria analysis. Strengthening PROMETHEE with ideas of AHP," European J. of Operational Research, vol. 153, pp. 307-317, 2004.

[3] L. C. Dias, J. P. Costa and J. N. Climaco, "A parallel implementation of the PROMETHEE method," European J. of Operational Research, vol. 104, pp. 521-531, 1998.

[4] K. C. Lee and N. Chung, "A web DSS approach to building an intelligent internet shopping mall by integrating virtual reality and avatar," Expert systems with applications, vol. 28, pp. 333-346, 2005.

[5] A. Niknafs, N. M. Charkari and A. A. Niknafs, "A PROMETHEE-based recommender system for multi-sort recommendations in on-line stores," In the proceedings of the 3rd IEEE International conference of digital information management, pp: 399-404, 2008.

[6] T. P. Liang, P. J. H. Hu, Y. R. Kuo and D. N. Chen, "A web-based recommendation system for mobile phone selection," In the proceedings of the 11th Pacific-Asia conference on information systems Auckland, 2007.

[7] Y. -L. Chen and L. -C. Cheng, "A novel collaborative filtering approach for recommending ranked items," Expert systems with applications, vol. 34, pp. 2396-2405, 2008.

[8] A. Felfernig, G. Friedrich and and L. Schmidt-Thieme, "Recommender systems," IEEE Intelligent systems, vol. 22, pp. 18-21, 2007.

[9] G. Adomavicius and A. Tuzhilin, "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, vol. 17, pp. 734–749, 2005.

[10] M. Balabanovic and Y. Shoham, "Content based collaborative recommendation," Communications of the ACM, vol. 40, pp. 66–72, 1997.

[11] L. M. D. Campos, J. M. Fernández-Luna and J. F. Huete, "A collaborative recommender system based on probabilistic inference from fuzzy observations," Fuzzy sets and systems, vol. 159, pp. 1559-1576, 2008.

[12] T. Q. Lee, Y. Park and Y. -T. Park, "A time-based approach to effective recommender systems using implicit feedback," Expert Systems with Applications, vol. 34, pp. 3055-3062, 2008.

[13] H. J. Ahn, "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," Information sciences, vol. 178, pp. 37-51, 2008.

[14] R. Bruke, "Hybrid recommender systems: survey and experiments," User Modeling and User-Adapted Interaction, vol. 12, pp. 331-370, 2002.

[15] T. L. Saaty, Decision making for leaders. Wadsworth: Lifetime learning publications, 1982.

[16] B. Roy, "The Outranking Approach and the Foundation of ELECTRE Methods," Theory and decision, vol. 31, pp. 49-73, 1991.

[17] H. -F. Li and J. -J. Wang, "An Improved Ranking Method for ELECTRE III," In the proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, pp 6659-6662, 2007.

[18] K. Yoon, "System Selection by Multiple Attribute Decision Making," Ph.D. Thesis. Kansas State University, 1980.

[19] J. P. Brans and P. Vincke, "A preference ranking organization method (The PROMETHEE method for multiple criteria decision making)," Management Sciences, vol. 31, pp. 647-656, 1985.

[20] E. Georgopoulou, Y. S. Fidis and D. Diakoulaki, "Design and implementation of a group DSS for sustaining renewable energies exploitation," European J. of Operational Research, vol. 109, pp. 483-500, 1998.

[21] J. L. Méndez, J. Oubiña and N. Rubio, "Expert quality evaluation and price of store vs. manufacturer brands: An analysis of the Spanish mass market," J. Retailing and Consumer Services, vol. 15, pp. 144-155, 2008.

[22] J. Sweeney and J. Swait, "The effects of brand credibility on customer loyalty," J. Retailing and Consumer Services, vol. 15, pp. 179-193, 2008.

[23] S. Ba, J. Stallaert and Z. Zhang, "Price competition in e-tailing under service and recognition differentiation," Electronic Commerce Research and Applications, vol. 6, pp. 322-331, 2007.

[24] A. E. Cretu and R. J. Brodie, "The influence of brand image and company reputation where manufacturers market to small firms: A customer value perspective," Industrial Marketing Management, vol. 36, pp. 230-240, 2007.

[25] H. Oh, "The Effect of Brand Class, Brand Awareness, and Price on Customer Value and Behavioral Intentions," J. of Hospitality & Tourism Research, vol. 24, pp. 136-162, 2000.

[26] R. W. Ralston, "The effects of customer service, branding, and price on the perceived value of local telephone service," J. Business Research, vol. 56, pp. 201-213, 2003.

[27] W. Müller, "Gaining competitive advantage through customer satisfaction," European Management J., vol. 9, pp. 201-211, 1991.

[28] C. Spearman, "The proof and measurement of association between two things," American J. Psychology, vol. 15, pp. 72-101, 1904.

[29] K. S. Raju and C. R. S. Pillai, "Multi criterion decision making in river basin planning and development," European J. Operational Research, vol. 112, pp. 249-257, 1999.

[30] T. D. Gauthier, "Detecting trends using spearman's rank correlation coefficient," Enviromental forensics, vol. 2, pp. 359-362, 2001.

[31] Y. Li, L. Lu and L. Xuefeng, "A hybrid collaborative filtering method for multiple-interests and multiple-content recommendation in E-Commerce," Expert systems with applications, vol. 28, pp. 67-77, 2005.

[32] P. Symeonidis, A. Nanopoulos, A. N. Papadopoulos and Y. Manolopoulos, "Collaborative recommender systems: Combining effectiveness and efficiency," Expert systems with applications, vol. 34, pp. 2995-3013, 2008.

[33] S. M. McNee, "Meeting user information needs in recommender systems," in Department of computer science. PhD Thesis. University of Minnesota, 2006.

[34] Y. -L. Chen, L. -C. Cheng and C. -N. Chuang, "A group recommendation system with consideration of interactions among group members," Expert systems with applications, vol. 34, pp. 2082-2090, 2008.

[35] J. F. L. Teno and B. Mareschal, "An interval version of PROMETHEE for the comparison of building products' design with ill-defined data on environmental quality," European J. of Operational Research, vol. 109, pp. 522-529, 1998.

[36] S. M. McNee, S. K. Lam, C. Guetzlaff, J. A. Konstan and J. Riedl, "Confidence displays and training in recommender systems," In the proceedings of the International conference on human-computer interaction, pp: 176-183, 2003.

[37] J. L. Herlocker, J. A. Konstant and J. Riedle, "Explaining collaborative filtering recommendations," In the proceedings of the ACM conference on computer supported cooperative work, pp: 241-250, 2000.

[38] O. Marinoni, "A stochastic spatial decision support system based on PROMETHEE," International J. Geographical Information Science, vol. 19, pp. 51-58, 2005.

[39] J. P. Brans, B. Mareschal and P. Vincke, "PROMETHEE: A new family of outranking methods in multi criteria analysis," Operational Research, vol. 84, pp. 477-490, 1984.

[40] H. Fazlollahtabar, “Applying Multiple-Criteria Decision Making methods for developing Information Technology industry,” International Journal of Information and Decision Sciences, vol. 1, pp. 115 - 131, 2008.

[41] G. Işıklar, Büyüközkan, G., “Using a multi-criteria decision making approach to evaluate mobile phone alternatives,” Computer Standards & Interfaces, vol. 29, pp. 265-274, 2007.

[42] C. E. Escobar-Toledo, L6pez-Garcia, B., “The use of multicriteria decision aid system in the information technology (IT) allocation problem,” Operational Research, An International Journal, vol. 5, pp. 223-240, 2005.

Arash Niknafs is currently a Ph.D. student at the Department of Computer Science in the University of Calgary, Canada . He got his M.Sc. from Tarbiat Modares University and his B.Sc. from Shahid Bahonar University of Kerman, respectively. His research interests include decision support, machine learning, and multi-agent systems.

Nasrollah Moghaddam Charkari received his B.Sc. in computer science from Shahid Beheshti University in Tehran, Iran in 1986 and his M.Sc. and Ph.D. degrees in computer engineering and information systems engineering from Yamanashi University, Japan in 1992 and 1995 respectively. He is currently an assistant professor in faculty of electrical & computer engineering, in Tarbiat Modares university, Tehran, Iran. His research interests include parallel algorithms, AI, and image understanding.

Ali Akbar Niknafs is an assistant professor at the Department of Computer Engineering in Shahid Bahonar University of Kerman (SBUK). He is also the head of the department in SBUK and head of Information Tech. Research Group of ICST (International Centre for Science, High Technology and Environmental Sciences). He got his Ph.D. in Applied Mathematics from SBUK. His research interests include: Data Mining, DSS, Business Intelligence, Quantum Logic circuits and computers, e-Commerce, MIS, Expert Systems.