



HAL
open science

Multi-Objective Recommendations and Promotions at TOTAL

Idir Benouaret, Mohamed Bouadi, Sihem Amer-Yahia

► **To cite this version:**

Idir Benouaret, Mohamed Bouadi, Sihem Amer-Yahia. Multi-Objective Recommendations and Promotions at TOTAL. The 32nd International Conference on Database and Expert Systems Applications (DEXA2021), Sep 2021, Lintz, Austria. hal-03379570

HAL Id: hal-03379570

<https://hal.science/hal-03379570>

Submitted on 15 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-Objective Recommendations and Promotions at TOTAL (Authors' Copy) *

Idir Benouaret¹, Mohamed Bouadi², and Sihem Amer-Yahia¹

¹ CNRS, Univ Grenoble Alpes, France

² SAP Labs, France

{idir.benouaret, sihem.amer-yahia}@univ-grenoble-alpes.fr
mohamed.bouadi@sap.com

Abstract. In this paper, we revisit the semantics of recommendations and promotional offers using multi-objective optimization principles. We investigate two formulations of product recommendation that go beyond traditional settings by optimizing simultaneously two conflicting objectives: *Budget-Reco* optimizes two customer-centric goals, namely utility and budget, and *Business-Reco* optimizes utility, a customer-centric goal, and profit margin, a business-oriented goal. To capture those objectives, we formulate knapsack problems and propose adaptations of exact and approximate algorithms. We also propose *Group-Promo*, the problem of generating product promotions that we model as a group discovery problem with multiple objectives and develop a Pareto-based solution. Our experiments on our TOTAL datasets demonstrate the importance of multi-objective optimization in the retail context, as well as the usefulness of our solutions when compared to their exact baselines. The results are valuable to TOTAL's marketing department that has been improving hand-crafted strategies by launching several promotional campaigns using our algorithms.

Keywords: Product recommendations, promotional offers, case study.

1 Introduction

Traditionally, recommendation systems were designed to learn a user's interest to suggest items that maximize the user's utility, items the user is most likely to appreciate. Collaborative filtering emerged as the most common method to estimate item utilities. In practice, several other dimensions are considered when designing recommendations especially in the context of retail and e-commerce platforms. On one hand, users only purchase items that are within their budget, on the other hand, the business impact of recommendations in terms of revenue

* Supported by TOTAL Data Science Services. We are very grateful to Christiane Kamdem-Kegne and Jalil Chagraoui from the Marketing Department of TOTAL, France.

loss or gain, plays a major role in determining which products to recommend to which customers. Similarly, when designing promotional campaigns to reward their customers, retailers consider the business value of each customer. Our study focuses on product recommendations and promotions that combine customers’ interests with business-oriented goals such as profit margin and revenue.

We formalize optimization problems coming from two real-world application scenarios. We revisit the semantics of recommendations and promotional offers using multi-objective optimization principles. We investigate two formulations of product recommendation that go beyond traditional settings by optimizing simultaneously two conflicting objectives: *Budget-Reco* optimizes both utility and budget, two customer-centric goals, and *Business-Reco* optimizes utility to serve customers’ interests, and profit margin, a business-oriented goal. We formalize knapsack problems and adapt exact and approximate algorithms [10, 6]. We also propose *Group-Promo*, the problem of providing customers with product promotions that we formalize as a group discovery problem with multiple objectives and develop a solution based on Pareto plans ([20]). In summary, we make the following contributions:

1. We formalize three new problems: *Budget-Reco*, *Business-Reco* and *Group-Promo*;
2. We develop exact and approximation algorithms that efficiently solve our optimization problems;
3. We conduct a set of experiments with a real dataset on TOTAL customers in collaboration with our data analysts at the marketing department.

The paper is organized as follows. Section 2 provides the data model and preliminaries. In Section 3, we formalize our problems and discuss the proposed algorithms. In Section 4, we subject our algorithms to a set of experiments. In Section 5, we discuss related works. Finally, Section 6 concludes the paper.

2 Data Model and Preliminaries

Let $\mathcal{U} = \{u_1, u_2, \dots, u_m\}$ be the set of customers and $\mathcal{I} = \{i_1, i_2, \dots, i_n\}$ be the set of products. For a given customer u , $\mathcal{H}_u \subset \mathcal{I}$ denotes the purchase history of u . We note \mathcal{X} the boolean purchase matrix of m customers in \mathcal{U} over the n products in \mathcal{I} . We now provide the definitions that we use in designing our multi-objective recommendation and promotion framework.

1. *Utility*: The function $Utility(u, i)$, measures the interestingness of a product i for a customer u . We use a classical Item-based Collaborative Filtering (IBCF) approach [24] which calculates a similarity $sim(i, j)$ between each pair of products i and j using cosine similarity. Our model accommodates other similarity functions and recommendation strategies such as association rules [3]. We choose IBCF for its better precision on our dataset [1].
2. *Budget*: Different customers have different spending habits. The budget is a customer specific upper-bound on the cumulated cost of multiple products.

For a given customer u , we denote β_u her budget. In our experiments in Section 4, we estimate the budget using the customer’s average spending on past purchases.

3. *Profit margin*: For a given product i , its profit margin $Margin(i)$ is calculated as follows: $Margin(i) = \alpha \times sp(i)$, where $sp(i)$ is the selling price of product i , and α is a value between 0 and 1 that depends on the product category and the location at which it is purchased (in our dataset, different gas stations).

$$Margin(u, i) = (F_u^{COCO} \times \alpha_{COCO} + F_u^{CODO} \times \alpha_{CODO} + F_u^{DODO} \times \alpha_{DODO}) \times sp(i) \quad (1)$$

where F_u^{COCO} , F_u^{CODO} , F_u^{DODO} , correspond to the frequency the customer u visits station types COCO, CODO and DODO respectively. These station types are determined by the Marketing department of TOTAL and depend on the size and location of the station as well as the traffic and number of customers in that station.

4. *Cost price*: For a given product i , $CostPrice(i)$ is defined as the price of product i from which is subtracted its profit margin.
5. *User Generosity*: The generosity of a customer u , denoted Gen_u , is estimated using a business rule that is used by TOTAL’s marketing department to run promotional campaigns. The user generosity consists of a small percentage of the total revenue (usually ranging from 1% to 5%) that is generated by the customer in a given period.

3 Multi-Objective Recommendations and Promotions

3.1 Budget Recommendations

Generally, customers have finite budgets, a.k.a. spending power, and they will spend more on products they prefer buying. It is therefore natural to account for budget in generating recommendations. To do that, we propose, *Budget-Reco*, a new formulation that accounts for both utility and budget. Given a target customer u with purchase history \mathcal{H}_u and a budget β_u , the set of all available products \mathcal{I} , select a set of k items $\mathcal{S} \subset \mathcal{I}$ such that:

$$\begin{aligned} \sum_{i \in \mathcal{S}} Utility(u, i) \text{ is maximized} \\ \text{subject to} \\ \sum_{i \in \mathcal{S}} sp(i) \leq \beta_u \\ |\mathcal{S}| = k \\ \mathcal{S} \cap \mathcal{H}_u = \emptyset \end{aligned} \quad (2)$$

Budget-Reco is a variant of the well-known 0-1 knapsack problem [8] where the values of items correspond to their utilities and the capacity of the knapsack

corresponds to the budget β_u . A naive approach to solving this problem is to generate the set of all possible combinations of k items and then choose the k items achieving the highest utility whose cumulative costs is less than β_u . However, this approach is prohibitively expensive. To address that, we use two exact algorithms (branch-and-bound and dynamic programming) and a greedy heuristic:

- **Branch-and-Bound.** The general algorithmic concept of branch-and-bound is based on an intelligent enumeration of the solution space since in many cases only a small subset of the feasible solutions are enumerated explicitly. It is however guaranteed that the parts of the solution space which are not considered explicitly cannot contain the optimal solution [25, 11].
- **Dynamic Programming.** Another popular method to compute an optimal solution for our problem is dynamic programming. Instead of enumerating the solution space of the problem, this algorithm starts by solving a small subproblem and then extends the solution iteratively until an overall optimal solution is found [11, 15].
- **Greedy Algorithm.** We implemented a greedy item-based collaborative filtering which selects the items that yield the next highest utility as long as the budget constraint β_u is not achieved.

3.2 Business Recommendations

Given a target customer u with purchase history \mathcal{H}_u , the set of all available products \mathcal{I} , an integer constant k , *Business-Reco* selects all sets $\mathcal{S} \subset \mathcal{I}$ satisfying:

$$\begin{aligned}
 & \sum_{i \in \mathcal{S}} \text{Utility}(u, i) \text{ is maximized} \\
 & \sum_{i \in \mathcal{S}} \text{Margin}(u, i) \text{ is maximized} \\
 & \text{subject to} \\
 & |\mathcal{S}| = k \\
 & \mathcal{S} \cap \mathcal{H}_u = \emptyset
 \end{aligned} \tag{3}$$

The output is k -sets of product recommendations, where each set \mathcal{S} satisfies the conditions above.

The main challenge in designing an algorithm for *Business-Reco* is the conflicting nature of the two objectives (utility and margin). In our recent work, we proposed a bi-objective approach based on dynamic programming to generate the set of all non-dominated k -sets [2]. However, computing all k -sets in polynomial time is not feasible. In real world applications, it is not necessary to find an optimal solution but a “good” solution which can be computed in reasonable time. As a consequence, we formulated *Business-Reco* as a multi-objective knapsack problem and implemented a Fully Polynomial Time Approximation Scheme (FPTAS) [7] algorithm with a performance guarantee that returns a suboptimal solution that is close within a factor of $(1 - \epsilon)$ to the optimal solution but with a much faster response time.

3.3 Promotional Offers

Business experts at the marketing department of TOTAL frequently run promotional offers to reward the fidelity of their customers. The set of products that are targets of a promotional campaign are determined by experts. The retailer's goal could be to sell out the stock of some products or to promote new products that are not popular yet. To address that, we propose to formalize *Group-Promo* whose goal is to find which group of customers to match with which subset of products so that: (1) the number of targeted customers is maximized, (2) overlap between group members is minimized, (3) the average utility of products across group members is maximized, and (4) the cost price of the promoted products is closest to the generosity of customers.

Problem Definition Given k sets of items in \mathcal{I} and the set of customers \mathcal{U} , *Group-Promo* forms k pairs (S, G) , where $G \subseteq \mathcal{U}$ is a group, for each set $S \subseteq \mathcal{I}$ such that:

$$\begin{aligned}
 & \frac{|\bigcup G|}{|\mathcal{U}|} \text{ is maximized;} \\
 & |\bigcap G| \text{ is minimized;} \\
 & \max_{i \in S} \max_{u \in G} \text{Utility}(u, i) \text{ is maximized;} \\
 & \max_{i \in S} \max_{u \in G} (|\text{Gen}(u) - \text{CostPrice}(i)|) \text{ is minimized;}
 \end{aligned} \tag{4}$$

Pareto Formulation. When optimizing more than one objective, there may be many incomparable group-sets. For instance, for a target set of items, we can form two groups, each group has its own advantage: the first one has higher utility and the second optimizes better for overlap between group members. In this section, we borrow the terminology of Multi-Objective Optimization introduced in [20] and define these concepts.

Definition 1. Local Plan. A local plan p , associated with a group G for a target set of items S , is a tuple $\langle \text{Utility}(G, S), \text{Convergence}(G, S) \rangle$ where

$$\begin{aligned}
 \text{Utility}(G, S) &= \max_{i \in S} \max_{u \in G} \text{Utility}(u, i) \\
 \text{Convergence}(G, S) &= \max_{i \in S} \max_{u \in G} (|\text{Gen}(u) - \text{CostPrice}(i)|)
 \end{aligned}$$

A local plan consists of objectives that have to be satisfied within each group.

Definition 2. Global Plan. A global plan p , associated with a set of k groups is a tuple $\langle \text{Coverage}, \text{Intersection} \rangle$ where

$$\begin{aligned}
 \text{Coverage} &= \frac{|\bigcup G|}{|\mathcal{U}|} \\
 \text{Intersection} &= |\bigcap G|
 \end{aligned}$$

A global plan consists of objectives that have to be satisfied across all groups.

Algorithm 1: Promotional Offer Algorithm to solve *Group-Promo*

Input: k sets of items in \mathcal{I} , a set of customers \mathcal{U}
Output: Pair (S, G) for each set $S \subseteq \mathcal{I}$

```

1  $\mathcal{P} \leftarrow \emptyset$ 
2 foreach itemset  $S$  do
3    $G_s \leftarrow$  Construct user groups with LCM
4    $P_G \leftarrow \emptyset$ 
5   for all user groups  $g$  in  $G_s$  do
6      $p \leftarrow$  Local-Plan( $g, S$ )
7     if  $p$  is not dominated by any other plan in  $P_G$  then
8        $P_G.add(p)$ 
9     end
10  end
11 end
12 for all  $k$ -groupset combinations from  $P_G$  do
13   Keep non-dominated plans and add them to  $\mathcal{P}$ 
14 end
15 return( $\mathcal{P}$ )
```

Definition 3. Dominance. A plan p_1 dominates a plan p_2 if p_1 has better or equivalent values as p_2 for every objective. The term “better” is equivalent to “greater than” for maximization objectives, and “lower than” for minimization objectives. Furthermore, plan p_1 strictly dominates p_2 if p_1 has better values than p_2 for every objective.

Definition 4. Pareto Plan. A plan p is Pareto if no other plan strictly dominates p . The set of all Pareto plans is denoted as \mathcal{P} .

Promotional Offer Algorithm. The algorithm described in 1 first constructs a set of possible groups for each set of items, based on the description of the frequent items purchased by customers. We use the LCM [28] mining algorithm for this task (Line 3). This step produces a large number of candidate groups that share common buying patterns. We compute, for each group, a local plan and compare the plans to discard all the dominated groups (Lines 5-11). After that, we generate different combinations of k groups where each group is associated with one of the set of items in the input. We keep the group-sets that are not dominated by any other (Lines 12-14).

4 Experiments

The purpose of our experiments is to study the balance between recommendation accuracy and response time for our three problem formulations. While our datasets are proprietary, our code is made available on GitHub³

³ github.com/multiobjective-recos/Biobjective_RecSys

4.1 Experimental Protocol

We split our dataset into training and test sets using the temporal global strategy [17]. We used purchase records from January 2017 to December 2018 for training and records from January 2019 to December 2019 for testing. We discarded cold start customers having fewer than 10 purchases.

4.2 Evaluation Measures

The goal is to evaluate the effectiveness of our recommendations with respect to accuracy and the generated profit margin in the case of product recommendations. In the case of promotions, we measure the induced costs and the average utility. We use the following evaluation measures and report results as averages over all test customers.

- *Precision*: measures the percentage of relevant recommendations among the top- k recommendations.

$$precision_u@k = \frac{card(S_u@k \cap test_u)}{k}$$

where $S_u@k$ is the top- k recommendations and $test_u$ is the target test set of customer u

- *Margin*: measures the average profit margins generated by the top- k recommendations.

$$margin_u@k = \frac{\sum_{i \in S_u@k} Margin(u, i)}{k}$$

where $S_u@k$ is the set of top- k recommendations.

- *Utility* measures the average utility generated by the products offered to the customer.

$$Utility_u = \frac{\sum_{i \in S} Utility(u, i)}{card(S)}$$

where, given a customer u , S is the set of product promotions for u .

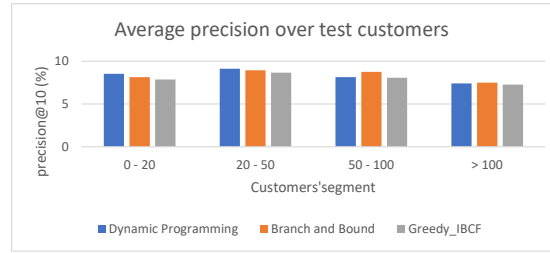
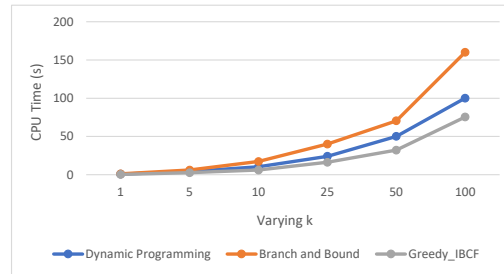
- *Costs* measures the average costs generated by the products offered to a customer.

$$Costs_u = \frac{\sum_{i \in S} CostPrice(i)}{card(S)}$$

where, given a customer u , S is the set of product promotions for u .

4.3 Recommendation Experiments

Budget recommendations. Since users generally have different spending habits, we customize each customer’s budget to better reflect the value of the budget constraint β_u . We calculate the average spending of each customer in

Fig. 1: Precision@10 for all algorithms and all customer segments (*Budget-Reco*)Fig. 2: Response time of all algorithms for solving *Budget-Reco* as a function of recommendation size k

a single transaction based on past purchases. We segment them into 4 different subsets: $[0, 20[$, $[20, 50[$, $[50, 100[$ and ≥ 100 . For example, the segment $[0, 20[$ groups together customers who spend between 0 and 20 euros on average. Recommendations are then tested on each segment independently. Figure 1 shows the achieved $precision@10$ values of all algorithms and for each customer segment. Precision results for different values of k were similar and are omitted due to lack of space.

Results show that for all customer segments the greedy-IBCF performs very closely to the two exact algorithms (branch-and-bound and dynamic programming) with a marginally small decrease on precision. Greedy-IBCF is much faster than exact algorithms especially when the number of recommendations k increases.

Business recommendations. We use the same experimental setting to evaluate business recommendations. We report the achieved results on precision and profit margin with different values of k . We vary the approximation parameter ϵ in $\{0.1, 0.3, 0.5\}$ and compare the results against a baseline *Pareto* which is an exact algorithm based on dynamic programming [2].

Results are reported in Figure 3. Without much surprise the approximation algorithm performs worse than *Pareto* and the values of precision and profit margin decrease as ϵ increases. However, in terms of response time, the approx-

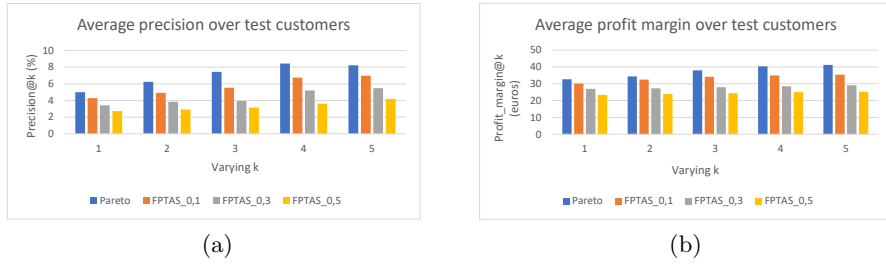


Fig. 3: Average precision (a) and profit margin (b) for different values of k (*Business-Reco*)

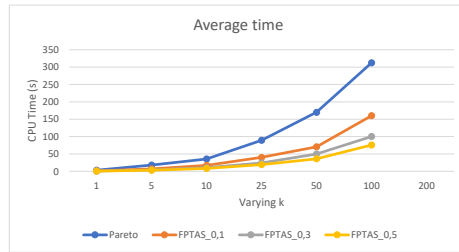


Fig. 4: Response time of all algorithms for solving *Business-Reco* as a function of recommendation size k

imation method is always faster as is shown in Figure 4. The gap between the response time needed by *Pareto* and the approximation algorithm increases with the number of recommendations. These results have pushed our collaborators at TOTAL to consider using the exact algorithm for testing their models offline and to use the approximation algorithm that yields an acceptable response time without hurting much precision and profit margin.

4.4 Promotion Experiments

To validate our solution for *Group-Promo*, we measure the average utility of the promoted products and the average costs across all customers. We compare our solution with one of the hand-crafted promotional campaigns that was deployed by analysts at TOTAL. In such campaigns, marketers select a set of items for each customer independently and then determine the subset of products that minimize the difference between the price of the products and the generosity of the customer. Our results in Table 1 for different values of k , show that our solution provides customers with much more interesting products (higher utility scores). While having an average cost per customer that is very close to the average cost induced by applying our *Group-Promo* algorithm.

Table 1: Promotions Results

	<i>Group-Promo</i>		TOTAL's offers	
	Utility(%)	Costs (Euros)	Utility(%)	Costs (Euros)
k = 10	11.61	3.191	5.19	2.657
k = 20	19.191	5.779	8.72	6.155
k = 50	28.77	9.550	10.27	10.450
k = 100	32.91	11.789	11.85	11.053

5 Related Work

Our work is related to several others in its aim and optimization mechanism. We survey the closest literature and emphasize similarities and differences.

Multi-objective optimization. There exist different approaches to solve a multi-objective problem [26, 27]. Scalarization, where all objectives are combined into a single objective. Another popular method is ϵ -constraints where one objective is optimized and others are constrained [21]. This formulation could be used in our context in cases where a specific profit amount is desired and products must be chosen accordingly. Another approach is Multi-Level Optimization [18] which needs a meaningful hierarchy between objectives. In our case, all objectives are independent and conflicting, hence using this mechanism is not feasible. While we focus on recommendations, it overlaps with a well-known problem in combinatorial optimization which is the knapsack problem [9, 11] where the goal is to optimize an objective function given some capacity constraints.

Recommendations and promotions. Our work is obviously related to the rich area of recommendations [22]. Recommendation approaches can be broadly classified into collaborative filtering (CF) and content-based (CB) [23]. CF approaches started with user-based and then more attention was paid to item-based to address the sparsity and scalability challenges in user-based CF [24].

Promotions play a central role in shopping [12, 13]. Instead of focusing solely on determining relevant recommendations, we aim to identify promotions that are also considered interesting [14, 19]. Among other things, customers use promotions for budgeting and planning, whereas retailers use promotions to accelerate purchase cycles, stimulate sales of complementary products and attract new customers [5, 16]. The motivation for focusing on existing promotions instead of product recommendations is that results from a user study that investigated customers' preferences regarding features in an intelligent mobile grocery assistant indicate that customers are more interested in information about relevant and actual special offers than suggestions for additional products [4].

6 Conclusion

In this paper, we revisited multi-objective optimization and applied it in the context of retail to generate product recommendations and promotional offers.

We proposed new formulations and appropriate solutions that adapt existing algorithms. Our empirical validation on real datasets confronts exact and approximate solutions as well as a hand-crafted promotional campaign against our results, showing the importance of multi-objective recommendation approaches in the retail world.

Our work lays the ground for a number of short-term and medium-term directions. Our immediate line of work is to launch a large-scale experiments and analyses study on a variety of publicly-available retail datasets such as Amazon. Our second immediate course of action is to gather the results of the recommendations and promotional offers that were deployed recently at our partner’s premises and confront the results to our empirical evaluation.

In the medium-term, we would like to characterize users’ generosity more finely for different product categories. That would require to formulate new problems that combine different generosityes for the same customer. The problem of promotional offers also needs to be revisited to combine products with different constraints in the same promotional package. We would also like to study the applicability of our framework in a changing context where for instance, customers’ generosity evolves and users may migrate from one group to another. This will necessitate the design of adaptive algorithms to handle such changes.

References

1. Benouaret, I., Amer-Yahia, S.: A comparative evaluation of top-n recommendation algorithms: Case study with total customers. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 4499–4508. IEEE (2020)
2. Benouaret, I., Amer-Yahia, S., Kamdem-Kengne, C., Chagraoui, J.: A bi-objective approach for product recommendations. In: 2019 IEEE International Conference on Big Data (Big Data). pp. 2159–2168. IEEE (2019)
3. Benouaret, I., Amer-Yahia, S., Roy, S.B., Kamdem-Kengne, C., Chagraoui, J.: Enabling decision support through ranking and summarization of association rules for total customers. In: Transactions on Large-Scale Data-and Knowledge-Centered Systems XLIV, pp. 160–193. Springer (2020)
4. Bhattacharya, S., Floréen, P., Forsblom, A., Hemminki, S., Myllymäki, P., Nurmi, P., Pulkkinen, T., Salovaara, A.: Ma \$\$ ive—an intelligent mobile grocery assistant. In: 2012 Eighth International Conference on Intelligent Environments. pp. 165–172. IEEE (2012)
5. Blattberg, R.C., Briesch, R., Fox, E.J.: How promotions work. *Marketing science* **14**(3-supplement), G122–G132 (1995)
6. Chekuri, C., Khanna, S.: A polynomial time approximation scheme for the multiple knapsack problem. *SIAM J. COMPUT* **38**(3), 1 (2006)
7. Erlebach, T., Kellerer, H., Pferschy, U.: Approximating multiobjective knapsack problems. *Management Science* **48**(12), 1603–1612 (2002)
8. Kellerer, H., Pferschy, U., Pisinger, D.: Introduction to np-completeness of knapsack problems. In: *Knapsack problems*, pp. 483–493. Springer (2004)
9. Kellerer, H., Pferschy, U., Pisinger, D.: *knapsack problems*. Springer (2004)
10. Kellerer, H., Pferschy, U., Pisinger, D.: Other knapsack problems. In: *Knapsack Problems*, pp. 389–424. Springer (2004)

11. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack problems with 33 tables. Springer Berlin (2010)
12. Lal, R., Matutes, C.: Retail pricing and advertising strategies. *Journal of business* pp. 345–370 (1994)
13. Lohse, G.L., Spiller, P.: Electronic shopping. *Communications of the ACM* **41**(7), 81–87 (1998)
14. Makhijani, R., Chakrabarti, S., Struble, D., Liu, Y.: Lore: a large-scale offer recommendation engine with eligibility and capacity constraints. In: *Proceedings of the 13th ACM Conference on Recommender Systems*. pp. 160–168 (2019)
15. Martello, S., Pisinger, D., Toth, P.: Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science* **45**(3), 414–424 (1999)
16. Mauri, C.: Card loyalty. a new emerging issue in grocery retailing. *Journal of Retailing and Consumer Services* **10**(1), 13–25 (2003)
17. Meng, Z., McCreddie, R., Macdonald, C., Ounis, I.: Exploring data splitting strategies for the evaluation of recommendation models. In: *Fourteenth ACM Conference on Recommender Systems*. pp. 681–686 (2020)
18. Migdalas, A., Pardalos, P.M., Värbrand, P.: *Multilevel optimization: algorithms and applications*, vol. 20. Springer Science & Business Media (2013)
19. Nurmi, P., Salovaara, A., Forsblom, A., Bohnert, F., Floréen, P.: Promotionrank: ranking and recommending grocery product promotions using personal shopping lists. *ACM Transactions on Interactive Intelligent Systems (TiiS)* **4**(1), 1–23 (2014)
20. Omidvar-Tehrani, B., Amer-Yahia, S., Dutot, P.F., Trystram, D.: Multi-objective group discovery on the social web. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. pp. 296–312. Springer (2016)
21. Papadimitriou, C.H., Yannakakis, M.: On the approximability of trade-offs and optimal access of web sources. In: *Proceedings 41st Annual Symposium on Foundations of Computer Science*. pp. 86–92. IEEE (2000)
22. Park, D.H., Kim, H.K., Choi, I.Y., Kim, J.K.: A literature review and classification of recommender systems research. *Expert systems with applications* **39**(11), 10059–10072 (2012)
23. Park, D.H., Kim, H.K., Choi, I.Y., Kim, J.K.: A literature review and classification of recommender systems research. *Expert systems with applications* **39**(11), 10059–10072 (2012)
24. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international conference on World Wide Web*. pp. 285–295 (2001)
25. Shih, W.: A branch and bound method for the multiconstraint zero-one knapsack problem. *Journal of the Operational Research Society* **30**(4), 369–378 (1979)
26. Trummer, I., Koch, C.: Approximation schemes for many-objective query optimization. In: *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. pp. 1299–1310 (2014)
27. Tsagouris, G., Zaroliagis, C.: Multiobjective optimization: Improved fptas for shortest paths and non-linear objectives with applications. *Theory of Computing Systems* **45**(1), 162–186 (2009)
28. Uno, T., Asai, T., Uchida, Y., Arimura, H.: Lcm: An efficient algorithm for enumerating frequent closed item sets. In: *Fimi*. vol. 90. Citeseer (2003)