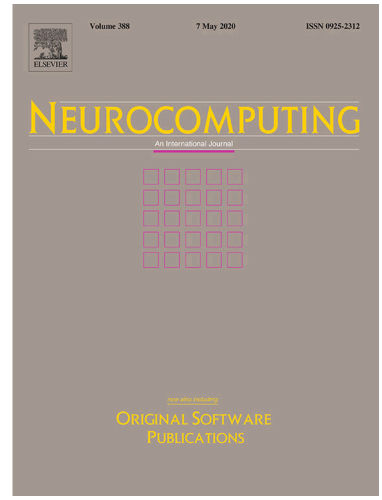

Survey paper

A Survey of Recommender Systems with Multi-Objective Optimization

Yong Zheng, David (Xuejun) Wang

PII: S0925-2312(21)01718-5
DOI: https://doi.org/10.1016/j.neucom.2021.11.041
Reference: NEUCOM 24597

To appear in: *Neurocomputing*

Received Date: 25 September 2021
Revised Date: 1 November 2021
Accepted Date: 14 November 2021

Please cite this article as: Y. Zheng, D. Wang, A Survey of Recommender Systems with Multi-Objective Optimization, *Neurocomputing* (2021), doi: https://doi.org/10.1016/j.neucom.2021.11.041

A Survey of Recommender Systems with Multi-Objective Optimization

Yong Zheng[a]*, David (Xuejun) Wang[b]

[a]College of Computing, Illinois Institute of Technology, Chicago, IL, 60616, USA
[b]Morningstar, Inc., Chicago, IL, 60602, USA

Abstract

Recommender systems have been widely applied to several domains and applications to assist decision making by recommending items tailored to user preferences. One of the popular recommendation algorithms is the model-based approach which optimizes a specific objective to improve the recommendation performance. These traditional recommendation models usually deal with a single objective, such as minimizing the prediction errors or maximizing the ranking quality of the recommendations. In recent years, there is an emerging demand for multi-objective recommender systems in which multiple objectives are considered and the recommendations can be optimized by the multi-objective optimization. For example, a recommendation model may be built by optimizing multiple metrics, such as accuracy, novelty and diversity of the recommendations. The multi-objective optimization methodologies have been well developed and applied to the area of recommender systems. In this article, we provide a comprehensive literature review of the multi-objective recommender systems. Particularly, we identify the circumstances in which a multi-objective recommender system could be useful, summarize the methodologies and evaluation approaches in these systems, point out existing challenges or weaknesses, finally provide the guidelines and suggestions for the development of multi-objective recommender systems.

*Corresponding author
Email address: yzheng66@iit.edu (Yong Zheng[a])

Keywords: recommender systems, multi-objective, optimization, Pareto optimal

1. Introduction

The problem of information overload results in difficulties of getting access to desired information or consuming preferred items in various applications. Recommender systems (RSs) were proposed and developed to alleviate this issue by delivering a list of recommendations tailored to user preferences. RSs have been applied in several domains to assist users' decision making, such as in the area of e-commerce [1, 2], online streaming [3, 4, 5], education [6, 7], social networks [8, 9], and so forth.

Traditional recommendation models usually deal with a single objective, such as minimizing the prediction errors [10] or maximizing the ranking quality [11]. Recently, there is an emerging demand in multi-objective recommendations [12], where the recommendation models can be built by considering multiple objectives through a process of multi-objective optimization (MOO). For example, recommendations may be produced by balancing different quality metrics, such as accuracy, diversity and novelty [13, 14, 15]. Group fairness may be considered to alleviate the conflicts between individuals and the whole group in the context of group recommendations [16, 17]. The objectives associated with multiple tasks may need to be jointly optimized in the multi-task RSs [18, 19, 20]. We use the term multi-objective recommender systems (MORS) to refer to the RSs which produce recommendations by optimizing multiple objectives.

We provide a comprehensive literature review on MORS in this artcile, and aim to help researchers understand the circumstances in which MORS is useful and learn how MOO can be utilized to solve the multi-objective problems in RSs. The major contributions of this article can be summarized as follows:

- There are several surveys in the area of recommender systems. This article provides the first literature review on the multi-objective recommender systems.

- We summarize the circumstances in which MORS is useful, which provides a guidance for the researchers in both recommender systems and the area of multi-objective optimization.
- We introduce and discuss the technologies in multi-objective optimization and their applications in recommender systems. Furthermore, we point out the weaknesses and challenges in the current development in multi-objective recommendations.
- We provide a workflow as guideline for researchers to select appropriate multi-objective optimization methods in their model development and experimental design.

The remainder of the article is organized as follows: Section 2 briefly reviews the recommendation techniques and compares this survey with others in recommender systems; Section 3 introduces and describes different multi-objective optimization techniques in details; Section 4 illustrates the applications and methodologies of multi-objective recommender systems; Section 5 provides a summary for the current development of multi-objective recommendations, points out the weaknesses of existing research, and further delivers a list of guidelines for the future development, followed by the conclusion and future work in Section 6.

2. Recommender Systems and Related Surveys

In this section, we provide an overview of the recommender systems, including the classification of recommendation algorithms, and different new types of recommender systems emerged in recent years. In addition, we compare this article with other surveys in the area of recommender systems, and introduce our survey methods in this article.

2.1. Recommendation Algorithms

RSs can be considered as supervised learning models which can learn from users' preference history and predict the list of items a user may like. The data used for building RSs is usually a composition of users, items, and the user preference information on the items. These preferences could be explicit information (e.g., numerical ratings on the items) or implicit feedback (e.g., browsing/clicking behaviors, user opinions in reviews or comments).

There are two major recommendation tasks in RSs – *rating predictions* and *top-N recommendations*. The rating-prediction task refers to predicting a rating given a user and an item, and thereby the recommendation performance can be examined by prediction errors, such as mean absolute error (MAE) and root-mean-square error (RMSE). By contrast, a list of N recommended items for a user is expected to be produced in the top-N recommendation task, while the performance can be evaluated by relevance metrics (e.g., precision, recall, F_1, etc) and ranking metrics (e.g., normalized discounted cumulative gain (NDCG)). Accordingly, recommendation algorithms can be developed specifically for the rating prediction (e.g., matrix factorization [21]) and top-N recommendation (e.g., BPR [22]) tasks. Note that the predicted ratings can also be used to sort and rank items to produce the recommendation list.

A popular way to categorize the recommendation algorithms is based on the approaches to build the recommendation models. The *content-based RSs* [23] recommend items based on the similarity between the candidate items and the preferred items by a user in terms of the content features. The *collaborative filtering algorithms* [24, 25] can estimate a user's preference on an item based on the preferences shared in common from a user or item neighborhood. The *hybrid RSs* [26] is able to combine different recommendation strategies and models together. Burke [26] added three extra RSs – *demographic information based RSs* which assumes users with similar demographic information may have similar tastes on the items, *the utility-based RSs* which rely on a utility function to sort and rank the items, and *knowledge-based RSs* which infer a match between users and items based on the knowledge information (e.g., association rules).

There are different limitations among these recommendation algorithms. For example, the collaborative filtering can work as long as there are users, items and rating information; however, it may suffer from the rating sparsity issue.

The content-based RSs rely on the quality of content information which require additional efforts to collect or pre-process. The hybrid RS models may work well, but it is difficult to understand or explain the models, and the computational cost may be increased significantly. The assumptions behind demographic information based RSs are simple and straightforward, but they may not always work well, and the collection of demographic information may lead to user privacy concerns. In the utility-based RSs and knowledge-based RSs, the performance depends on the utility function or knowledge rules which are difficult to build or acquire, especially for the utility-based RSs. The content-based RSs and collaborative filtering models are the most popular RSs among these categories, especially with the development of deep learning or neural network based models [27].

2.2. Different New Types of Recommender Systems

The traditional RSs may only rely on users' preferences on the items, as well as the user demographic information and the item content features. In recent decades, different new types of the recommender systems emerged, and researchers made the efforts to adapt the traditional recommendation algorithms to these new RSs, including but not limited to

- *Context-Aware RSs* [28, 29, 30] in which researchers believe that users preferences may vary from contexts to contexts (e.g., time, location, occasion, etc), where RSs should be built to adapt to these contextual situations.
- *Group RSs* [31, 16, 7] in which the system produces recommendations for a group of users instead of a single user, e.g., in group dinner or group trip.
- *Multi-Criteria RSs* [32, 33, 34] in which researchers take advantage of user preferences in different aspects of the items (i.e., criteria) to build better recommendation models. For example, we may consider room cleanliness, room size, location, safety in a hotel recommendation.

- *Cross-Domain RSs* [35, 36] in which the model can assist recommendations in a target domain based on the knowledge learned from a source domain. For example, researchers may take advantage of the preference information in a movie domain to predict users' tastes in a music domain.

- *Multi-Stakeholder RSs* [37, 38, 39] in which we produce the recommendations by considering the utility of an item from the perspective of multiple stakeholders (e.g., the item supplier, the platform, etc) in addition to the end users.

- *Multi-Task RSs* [40, 41, 42] in which researchers produce recommendations by optimizing multiple tasks through a joint learning process. The joint optimization over different tasks is not novel, but multi-task RSs can share common representations (e.g., latent factors, feature space, neural network layers, etc) in the optimization process.

2.3. Related Surveys

Recommender systems have been developed for several decades, and there are many surveys on recommender systems. For example, Lu et al. [43] contributed a survey on recommender systems from a general perspective. By contrast, there are several surveys which focus on different types of recommender systems, such as literature reviews on context-aware RSs [29], multi-criteria RSs [44], multi-stakeholder RSs [37], and so forth. In addition, there are also surveys on recommendation models using specific technologies, such as RSs based on deep learning [27] and swarm intelligence [45].

In this article, we provide a literature review on multi-objective recommender systems, and this topic was never investigated in the current publications. This article aims to help researchers understand the circumstances in which MORS is useful and learn how MOO can be utilized to solve the multi-objective problems in RSs.

The selection process of the relevant publications in MORS began with the help of Google Scholar by using the keywords "multi-objective" and "recommender systems". Moreover, we identified the recent developed multi-task RSs as another practice in MORS, and additionally searched for publications related to multi-task RSs. We collected these publications in reputable journals and conferences (e.g., WWW, SIGIR, RecSys, UMAP, KDD, CIKM, ICDE, etc). This process resulted in a selection of 56 articles (from 2007 to August, 2020) for our literature review. Furthermore, we reviewed related articles in the area of multi-objective optimization in order to discuss it in details in Section 3.

3. Multi-Objective Optimization

Multi-objective optimization becomes a dedicated discipline of scientific research, especially in the area of decision making due to the fact that a set of objectives are usually involved. In this section, we introduce the multi-objective problem (MOP), and discuss different popular MOO methods.

3.1. Problem Definition

Suppose we have multiple objectives to be minimized, the general MOP can be described as follows [46].

$$\min_{x} \left(f_1\left(x\right),\ f_2\left(x\right),\ \ldots,\ f_M\left(x\right)\right) \tag{1}$$

$$\text{subject to } g_k\left(x\right) \le 0,\ k = 1, 2,\ \ldots,\ K;\ x \in\ X \subset\ R^n \tag{2}$$

M is the number of objective functions, and $x = (x_1, x_2, \ldots, x_n)$ is a n-dimension decision vector in space R^n. X defines the lower bound and upper bound of x: $X = \{x \mid a_i \le\ x_i \le\ b_i,\ i = 1,\ 2,\ \ldots,\ n\}$. We define $F(x)$ as $\left(f_1\left(x\right),\ f_2\left(x\right),\ \ldots,\ f_M\left(x\right)\right)$ which is a vector function in space objective R^M. $g_k\left(x\right)$ is used to indicate possible constraints in MOP. A *feasible solution* is defined as any solutions that can satisfy Equation 2.

In order to determine which solution is better by given two feasible solutions x and x^*, the notion of *dominance* [47] is introduced to MOP: x is dominated by x^* if and only if

$$f_m\left(x^*\right) \leq \ f_m\left(x\right) \ \ for \ all \ m = 1,\ 2,\ \ldots,\ M,\ \exists m' : f_{m'}(x^*) < f_{m'}(x) \tag{3}$$

Accordingly, the feasible solution x^* is usually named as a *non-dominated solution* or *Pareto optimal solution*, if there are no other feasible solutions that dominates x^* [47]. The group of all non-dominated solutions is known as the *Pareto optimal set* or *Pareto set*. The range of objective function $F(x)$ over the Pareto set is referred to *Pareto optimal front* or *Pareto front* [47].

Different objectives may have various importance in the MOO process. The humans who have preferences on these objectives are referred as "*decision makers*" (DM). According to the engagement of DM's preferences, the optimization can be classified into priori (i.e., DM's preferences as inputs before the optimization), posteriori (i.e., DM's preferences used to select solutions after the optimization) and interactive methods (i.e., DM may change preferences during the optimization) [48].

3.2. Classification of Multi-Objective Optimization Methods

There are many ways to classify MOO algorithms. Here we classify MOO based on its underlying optimization strategy:

- *Scalarization Methods*: it is used to transform a MOP to a single-objective problem (SOP), so that most of the existing optimization methods for SOP can be reused to solve the problem.
- *Population-Based Heuristic Methods*: in this category, different population-based heuristic search techniques can be developed and applied to solve MOP directly. Most of these methods are known as the multi-objective evolutionary algorithms (MOEAs) which are the evolutionary algorithms (EAs) or swarm intelligence methods (SIs) that were modified and adapted to MOP. We can produce the Pareto optimal set as the output by using MOEAs.

In Section 3.3 and Section 3.4, we discuss the popular scalarization and MOEA methods in details.

3.3. Scalarization Methods

There are several ways to transform a MOP into a SOP, such as the weighting methods [49, 50], the ϵ-constraint method [51], the global criterion method [49], goal programming [52, 53], lexicographic method [54], the Tchebycheff approach [55], etc. We discuss the weighting methods and the ϵ-constraint method in this Section, since they are two most popular MOO methods in MORS.

The *weighted sum method* is one of the most popular and straightforward methods. We can simply assign a weighting vector w=$(w_1, w_2, \ldots, w_M)$ in which each weight is associated with an objective to represent the importance of the objective. Consequently, the weighted sum, $\min_x \sum_{i=1}^{M} w_i f_i(x)$, is utilized to represent a single objective to be optimized. The weighted sum method can produce a single Pareto optimal solution by given a set of weights, as long as $w_i > 0$ and $\sum_{i=1}^{M} w_i = 1$ [56].

In addition to the weighted sum method, there are many variations of the weighting methods, such as weighted exponential sum method [49, 57], weighted product [50], weighted metric method [58, 59], weighted chebyshev method [60] and exponential weighted criterion [61]. They just apply the weights to different aggregations of the objectives.

The *ϵ-constraint method* [51] is another popular scalarization approach. It optimizes one objective and treats other objectives as constraints.

$$\min_x f_l(x) \tag{4}$$

subject to:

$$f_i\,(x) \leq\ \epsilon_i, \text{ for all } i\text{=1,2,}\ldots\text{,}K\text{, and } i \neq l,\ x \in S \tag{5}$$

In the example above, we optimize the l^{th} objective, and set others as constraints, where S refers to the set of all feasible solutions. ϵ_i is used to represent the upper bound of each objective $f_i(x)$, and considered as parameters to be tuned up in the optimization process.

3.4. Population-Based Heuristic Methods

The population-based heuristic methods are well known as MOEAs which are a family of approaches based on EAs or SIs inspired by natural evolution or optimization processes [62], such as the genetic algorithms (GA) [63, 64], particle swam optimization (PSO) [65, 66, 67], artificial immune system (AIS) [68, 69], and so forth.

Traditional EAs or SIs were developed to solve SOPs. Theses methods usually start with an initialization of a population P which is a set of random feasible solutions. Each solution $x \in P$ will be evaluated by a fitness function (i.e., the objective function) in the iterative learning process. In GA, a Parent set $P_s \subset P$ will be selected based on these fitness values computed from the fitness functions. A crossover and mutation are applied to P_s to produce a new solution set as the child set P_c. An elitism process is applied to select best solution from the original population P and the newly generated solution P_c, which results in a new generation of the population. By this way, each generation of population is better than the previous generation according to the fitness values. SI follows a similar process to obtain a better population in each iteration. The iterative learning is expected to produce optimal solutions eventually.

These EA and SI methods can be modified and adapted to MOPs, which results in different MOEAs. More specifically, the MOEAs at the early age, such as VEGA [70], define multiple fitness functions, while each function is associated with an objective. Afterwards, they can select and blend solutions from the population by using multiple fitness functions. Thereby the set of solutions in the process of crossover and mutation is composed by the solutions optimized by different objectives. By contrast, the modern and more popular way to develop MOEAs is defining the fitness function by the non-dominance relation directly, such as the popular MOGA [71] and NGSA [72] methods. By this way, MOEAs can converge faster and find better Pareto optimal solutions.

In addition to the GA-based MOEAs discussed above, there are other MOEAs based on different evolutionary algorithms, such as multi-objective PSO [73, 74, 75], multi-objective AIS [76, 77, 78], and so forth. We can incorporate the notion of Pareto dominance into these heuristic methods too. Take MOPSO [73] for example, we can select the global optimal solution by the Pareto dominance in the swarm. Zeng et al. [75] recently adopted differential evolution [79] to diversify the population so that they can achieve a better Pareto set. The advances in single-objective PSO can also be adopted to improve multi-objective PSO, such as the methods to enhance the convergence rate [80] or perturb the acceleration coefficients [81, 82] in order to have a more thorough search in the problem space.

The search ability of the evolutionary algorithms depends on the variation among the individuals or candidate solutions in the population [83]. There are usually two solutions to enhance the search ability. On one hand, researchers may develop different evolutionary operations, e.g., developing new mutation operations [84] in GA or differential evolution in multi-objective PSO [75]. On the other hand, the attention was paid to how to increase the diversity of the solutions in MOEAs by the niche method which is also known as fitness sharing [85, 86, 87]. The idea behind fitness sharing is to degrade the fitness of similar solutions that causes population diversity pressure [87]. For example, a niche method can be developed to penalize the fitness value of a solution in a more crowded neighborhood [86]. Namely, a solution in a high-density neighborhood has a higher niche count. As a result, the solution with higher niche count will less likely be selected for the next generation in the evolution process.

3.5. Selecting the Best Solution from Pareto Optimal Set

Both scalarization methods and population-based heuristic methods can generate a Pareto set, while the scalarization methods need to be executed multiple times with different parameters (e.g., the weights in the weighted sum method) in order to obtain a Pareto set. It is worth mentioning that there are specific requirements to get all Pareto solutions for the scalarization methods, e.g., the problem needs to be convex in the weighted sum method [88].

Once we have the Pareto set, it is important to select a single optimal solution for specific applications, such as recommender systems. The criteria or methods that can be used for the selection can be summarized as follows.

- *Knee point.* The "knee point" of the Pareto front is the position at which small improvement in either objective will cause a large deterioration in the other objectives [89, 90]. Researchers proposed different approaches to identify this knee point, such as the angle-based method [91], marginal utility [91, 90], expected marginal utility [90] and the hyperplane normal vector method [92].

- *Hypervolume.* The hypervolume [93] of a solution in two-dimensional space is the area measurement of the rectangle with opposite vertex as this point and the origin. The point of a solution on the Pareto front with the maximum hypervolume can be treated as the optimal solution. Hypervolume can also be applied to the whole Pareto front by measuring volumes covered by hypervolumes of all solution points. The hypervolume of Pareto front can be used to compare the quality of different Pareto sets.

- *Multiple-criteria decision-making (MCDM) methods.* The multiple-criteria decision-making is a dedicated discipline that evaluates conflicting criteria in the decision making process. Given a set of Pareto solutions, the selection of an optimal solutions can be considered as a multi-criteria decision-making process. The most popular MCDM methods include but not limited to: the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) methods [94, 95] and Preference Ranking Organization METHod for Enrichment of Evaluations (PROMETHEE) method [96], and so forth. One of the limitations in these methods is that they usually need the known importance or weights of the objectives. Without these pre-defined weights, researchers have to figure out additional methods to estimate the importance, or simply consider equal weights in the selection process.

3.6. Comparison Between Scalarization and MOEAs

Both Scalarization and MOEAs can be applied to solve MOPs. However, they do have their own advantages and disadvantages.

Scalarization methods are popular due to its simplicity since they transform MOPs to SOPs and traditional single-objective optimizer can also be applied. However, scalarization methods may not be able to handle non-convex problems in contrast to the MOEAs, though researchers may be able to use a relaxed problem to approximate the non-convex problems. Furthermore, specific requirements are necessary in order to get Pareto optimal solutions, e.g., a feasible solution is Pareto optimal if it solves the ϵ-constraint problem and it is unique in the ϵ-constraint method. In addition, it is difficult to find the optimal parameters (e.g., the weights in the weight sum method) in the scalarization methods, unless the preferences by the decision makers are known in advance.

By contrast, MOEAs can handle both convex and concave problems. The solutions in the Pareto set produced by MOEAs are guaranteed to be Pareto optimal. Moreover, there are several open-source libraries available for MOEAs, such as MOEA framework[1], pymoo[2], PyGMO[3], inspyred[4], and so forth. However, MOEAs may converge at local optima while the diversity of the solution set is always one of the challenges as mentioned in Section 3.4. In addition, the efficiency of MOEAs may be another concern if the data is a large or MOEAs need more agents (e.g., particles in PSO) in the search process.

4. Multi-Objective Recommender Systems

In this section, we illustrate the applications and methodologies of multi-objective recommender systems.

As mentioned previously, MORS refer to the recommender systems which produce recommendations optimized by multiple objectives. It is considered as

[1] http://moeaframework.org/
[2] https://pymoo.org/
[3] https://esa.github.io/pygmo/
[4] https://aarongarrett.github.io/inspyred/

an application of MOO in the area of recommender systems. In comparison with the MOO applications in other areas, MORS may be different from two perspectives. On one hand, DM's preferences on the objectives can be engaged through the priori, posteriori or interactive methods in most of the MOO applications. However, DM's preferences are not always available in recommender systems. Take the RSs by considering the accuracy, novelty and diversity of the recommendation list [13] for example, the user preferences (e.g., the weights or trade-off preferences) on accuracy, novelty and diversity are usually not available. As a result, researchers have to build the models first, and find the optimal trade-off by performing post-experiment user studies. On the other hand, we usually need a single optimal solution in MORS, while other areas may acquire the Pareto front or more than one solutions as the final outputs or presentations to the end users. Take the investment portfolio optimization [97] for an example, the investment managers would like to maximize the expected portfolio return and minimize the portfolio risk, but they may not have clear preferences on these two objectives. They can acquire the Pareto front and present it to the investors, so that investors can select the trade-off based on the Pareto front.

The major challenges in MORS include but not limited to: choosing the appropriate MOO techniques to solve the MOP in RSs; selecting a singe optimal solution from the Pareto set as the output in RSs; balancing multiple objectives, especially when there are conflicting interests.

In the following discussions, we first identify the circumstances or contexts in which a multi-objective recommender system could be useful. These circumstances can be classified into five categories, as shown by Table 1. We classify these research work by not only the circumstances, but also the categories of the MOO methods adopted. If the scalarization methods were applied, we further identify its subcategory as described in Section 3.3. If MOEAs were adopted, we present the EA or SI method which was extended to develop MOEA in the table.

Table 1: Categories of the Circumstances Using MORS

	MOEA or Scalarization	**Specific MOO Methods**	**List of References**
RSs with Multiple Quality Metrics	Scalarization	Weighted Sum	[98, 99, 100, 101, 102]
		Weighted Chebyshev	[103]
		Tchebycheff Approach	[104]
	MOEA	Genetic Algorithm	[105, 106, 107, 108, 109, 110] [111, 112, 113, 114, 115]
		Immune Algorithm	[116, 117, 118, 119]
		Teaching- Learning Based SI	[120]
Group RS	Scalarization	Weighted Sum	[16, 17]
Multi-Stakeholder RSs	Scalarization	Weighted Sum	[121, 122, 123, 124, 125, 126] [127, 128]
		Weighted Product	[129]
		Weighted Average	[130]
		ϵ-Constraint Method	[131, 38]
	MOEA	Genetic Algorithm	[39, 132]
Multi-Task RSs	Scalarization	Weighted Sum	[42, 133, 41, 134, 135] [136, 137, 20, 138, 139] [140, 19, 18, 141]
Clustering & Rule Based RSs	Scalarization	Weighted Sum	[142, 143]
	MOEA	Genetic Algorithm	[144]

For each of the five categories above, we have the discussions from four perspectives:

- *Objectives.* We explore the motivation of applying MOO in each circumstance, and collect the definition of the objectives in each category.
- *MOO Methods.* We summarize the multi-objective optimization methods with reference to the introductions in Section 3. If MOEAs are adopted, we further discuss the method for selecting the single optimal solution. Otherwise, we introduce how the parameters were found (e.g., pre-defined, grid search, etc.) in the scalarization methods.
- *RS Models with MOO.* We investigate how the MOO methods can be integrated with the recommendation models. For example, some research work may consider it as a joint learning process in the recommendation

models, while some others may adopt MOEAs to learn a list of recommendations directly.

- *Effectiveness*. There are no general or specific metrics for MOO evaluations, since the purposes or objectives in different MORS may vary from categories to categories. Therefore, we focus on the demonstration of the effectiveness, i.e., how the researchers demonstrated that the proposed MOO solutions in RSs were better than others.

4.1. Recommender Systems Balanced by Multiple Quality Metrics

Beyond recommendation accuracy, a wider perspective towards the recommendation utility may include other quality metrics [13], such as novelty and diversity [13], serendipity [145], popularity [146], and so forth. Maximizing one of these metrics may hurt others. Therefore, there is a demand in MOO so that recommendations can be produced by balancing these metrics.

Objectives. Most of existing research in this category took accuracy, diversity, and novelty into consideration. Some work [107, 110, 113, 114] added coverage as one of their concerns. Xie et al. [102] considered click-through rate and dwell time (i.e., the length of time that a visitor spends on a page) as two representations of recommendation accuracy without considering other metrics (e.g., diversity, novelty, etc). It is worth noting that there are no uniform measures for accuracy, diversity and novelty, and researchers may use different representations. For example, Pang et al. [110] utilized prediction errors as the representation for recommendation accuracy, while Geng et al. [116] measured accuracy by using the similarity between the item candidates and items in user profiles. Di Noia et al. defined diversity from the perspective of item catalogs, and used catalog coverage together with Gini coefficient to measure the diversity metric [98]. Fortes et al. [109] computed diversity by using the popular EILD metric [14] which is a doubly rank-sensitive and rank-aware expected intra-list diversity measure.

Moreover, researchers may consider special metrics which vary from applications to applications. For example, Patil et al. [101] focused on the compatibility and versatility of capsule wardrobes in fashion recommendations so that they can help consumers buy minimal fashion items that produce a maximum number of compatible and versatile outfit combinations. Paul et al. [100] raised the security concerns in the recommendation models, and built a joint learning based model by considering recommendation accuracy and the model vulnerability (i.e., adversarial perturbations on image features).

MOO Methods. Based on the information in Table 1, we can observe that MOEA is the most popular MOO method for the research in this category. There are only limited research work using scalarization. More specifically, Di Noia et al. [98], Patil et al. [101] and Karabadji et al. [99] used the weighted sum described in Section 3.3 as the scalarization method for the optimization process. But they used different ways to find the best weights. Di Noia et al. [98] used a grid search which iterates the weight from 0 to 1 with a step of 0.1, since there is only one weight parameter (i.e., w and 1-w) in their work. Patil et al. [101] tried five different sets of the weights without a complete grid search in the space. Lacerda et al. [103] adopted the weighted Chebyshev as the scalarization, since it encounters all solutions in a non-convex Pareto set. Wang et al. [104] used the Tchebycheff approach [55] as the scalarization method. By contrast, other research work that utilized scalarization optimize multiple objectives by a joint learning process through the recommendation models, such as the multi-arm bandit algorithm [103], Pareto-oriented reinforcement learning [102] or defense modeling [100]. In this case, the weights will be considered as hyperparameters to be tuned up in the optimization process.

Most of the work using MOEA adopted GA-based and AIS-based MOEAs, as shown in Table 1. Zou [120] utilized a teaching-learning based optimizer which is a multi-objective swarm intelligence approach. By using MOEA as the optimization method, a subsequent challenge is the selection of the optimal solution from the Pareto optimal set. The selection methods in these work can be summarized as follows.

- Using hypervolume discussed in Section 3.5 to select the optimal solution such as the work by Zuo et al. [107] or compare the quality of Pareto fronts among different MOEA approaches [114].
- Adopting the multi-criteria decision making methods. Chai et al. [119] applied the PROMETHEE method to select the optimal solution from the Pareto set. PROMETHEE [96] is one of the strategies used in multi-criteria decision making, and it is able to produce a ranking score for the solutions based on the comparisons, e.g., pairwise ranking. Fortes et al. [115] selected the optimal solution which minimizes the distance between the learned objective weights and the computed weights from sample data, which can be considered as a TOPSIS method.
- Selecting the optimal solution based on the weighted mean of multiple objectives. More specifically, Di Noia et al. [98] and Wang et al. [104] simply considered equal weights and used the average value of various objectives to select the optimal solution. Ribeiro et al. [105] defined different sets of the weights and produced the optimal solution by using each set of the weights for the purpose of comparisons. Fortes et al. [109] figured out one method to compute the individual user's weight for each objective based on user preferences data.
- Selecting the optimal solution by each objective. These work [104, 108] selected the solution with best single objective (e.g., accuracy, novelty, or diversity) and observe how the values in other metrics changed in comparison with other solutions.
- Presenting the descriptive statistics (e.g., minimal, maximal, and mean value) of the objectives based on the Pareto optimal set or subset, without selecting a single optimal solution. These work [116, 106, 117, 110, 111, 112, 118, 114, 120] can only demonstrate that they were able to produce desired solutions in the Pareto set, but leaved the challenge to select

the single best solution. The work by Ribeiro et al. [105] proposed to use a weighted sum of multiple objectives as the metric to select the best solution. However, they only tried four sets of weights to observe the experimental results without performing a search to find the optimal solution. Pang et al. [110] learned weights in the similarity function associated with collaborative filtering, and proposed to use the average weights from the solutions in the Pareto set as the final solution. However, there are no fundamental evidences showing that the average solution from the Pareto set is the optimal one.

RS Models with MOO. The way to integrate MOO with RS models depends on how the RS models work, i.e., how the models produce recommendations. First of all, most of these work can directly produce the recommendation list through MOEA, since the recommendation list for a user can be encoded in MOEAs (e.g., gene encoding in GAs or positions in PSOs). However, the encoding may be very long if there are large scale of the items as recommendation candidates. Some research [104, 111] produced a list of K item candidates for each user by a traditional recommendation model, and further learned the top-N item recommendations from these K candidates ($N << K$). Moreover, some other work [105, 109, 114] built a hybrid recommendation model which predicts a rating for a user on an item by a weighted sum of prediction ratings by mutliple recommendation algorithms. In this case, they can utilize MOEAs to learn these weights directly. In addition, the work by Di Noia [98] produce the recommendation list first, then re-rank the items based on the scalarization optimization. Cao et al. [113] tried to produce recommendations by using tensor factorization. They did not utilize the tensor decomposition as the optimizer, but adopted MOEA to learn the tensor representations by considering the loss function in tensor factorization as one of the objectives. By contrast, Lacerda et al. [103], Xie et al. [102] and Paul et al. [100] utilized a joint learning model which delivers a joint loss function by the weighted sum of the losses associated with multiple objectives.

Effectiveness. The work in this category tried to produce the recommendation list by balancing multiple quality metrics. Researchers evaluate the recommendation models by comparing multiple objectives (i.e., different quality metrics in this case). It is worth mentioning that not all of the objectives are conflicting in these research. For example, Xie et al. [102] considered click-through rate and dwell time as two accuracy metrics, therefore they demonstrate that their models were able to improve these two metrics in their experiments.

In the research where there are conflicting objectives, researchers believe that the proposed approaches are effective as long as they can improve some metrics (e.g., novelty and diversity) without a significant loss in the recommendation accuracy. However, there are no standards to define a "significant" loss, and there are no user studies which can explore the tolerance of the loss from the perspective of the end users. For example, the proposed method by Zhang et al. [108] was demonstrated to improve both accuracy (measured by precision) and diversity by comparing to the models without considering multiple objectives. However, the model increased precision at the loss of diversity in comparison with another multi-objective RS model. Zhang et al. believed that the proposed model was still effective, since the improvement over precision is much larger than the decline in diversity.

4.2. Group Recommender Systems

It is not surprising that MOO is also helpful in group recommendations. Due to the nature of group recommender systems, the conflicts between individual preferences and group satisfaction are involved in the recommendation challenge. Take the group dinner for example, an individual in the group may prefer spicy food, while several other group members may not.

Objectives. Xiao et al. [16] utilized MOO in the group recommendation model by incorporating three objectives, including individual preferences, group satisfactions and group fairness. Particularly, they defined 4 different group fairness measures – Variance Fairness and Jain's Fairness [147] encourage the group members to achieve close utilities between each other; while Least Misery Fairness and Min-Max Ratio emphasize the gap between the least and highest utilities of group members. Wu et al. [17] considered group satisfaction, social relationship density (i.e., the social closeness between group members with respect to an item) and group fairness in their work, while they adopted a variance-based fairness measure [16].

MOO Methods. Xiao et al. [16] combined multiple objectives into a single one by using the weighted sum method, and proposed to apply the greedy search and integer programming to find the optimal solutions. Wu et al. [17] adopted the weighted sum method and used exhaustive grid search to find the best weights.

RS Models with MOO. Wu et al. [17] adopted joint learning and used gradient descent as the optimizer in their recommendation models. Xiao et al. [16] proposed two optimizer (i.e., greedy search and integer programming) which can help them learn the recommendation list directly.

Effectiveness. These work assumed that the group recommendations could be improved by considering multiple objectives, especially the group fairness. Therefore, they did not compare multiple objectives in their experiments, and only evaluated the performance of group recommendations directly. For example, both Wu et al. [17] and Xiao et al. [16] used F_1 measure and NDCG to evaluate the quality of top-N recommendations, without presenting or comparing other objectives defined in their work, e.g., group fairness.

4.3. Multi-Stakeholder Recommender Systems

Multi-stakeholder RSs deliver recommendations by considering the perspective of multiple stakeholders. Take e-commerce for example, not only the user preferences, but also the utility of the item in view of sellers, the marketplace, as well as the shipping companies may also be taken into account in the recommendation process. Consequently, there could be at least one objective which is associated with each stakeholder. Multi-stakeholder RSs are expected to achieve a balance among these stakeholders.

Objectives. The definition of stakeholders and the related objectives may vary from cases to cases. We summarize them as follows.

- The *reciprocal recommendation* models usually take advantage of the bidirectional preferences and perform a process of user matching to recommend a user to the target user, such as dating [124] or job-seeking [131, 127]. Rodriguez et al. [131] used job-seeking as a case study, set a job-seeking intent score for the job seekers and a semantic matching score for the recruiters, so that they can utilize MOO for the joint optimization. Zheng et al. [124] applied MOO in an online-dating context in which users were asked to describe their desired partners or expectations by rating different criteria. As a result, the distance between a user's multi-criteria rating and his or her expectation ratings can be used to estimate the utility values.

- Applications are more popular in the *e-commerce* [122, 125] or *marketplace* [121] areas. Both Lin et al. [122] and Louca et al. [125] considered the customers preferences (e.g., click-through rate or purchase likelihood) and platform revenue in their applications, while Nguyen et al. [121] additionally took the profits of item suppliers into account.

- *Mobile application* is another interesting case study in multi-stakeholder RSs. Xia et al. [129, 123] made their attempts to apply MOO in mobile app recommendations, while they considered the objectives for users (e.g., relevance), app market (e.g., revenue) and the recommender system (e.g., diversity, robustness), respectively.

- *Provider fairness.* Surer et al. [38] proposed a general framework for recommender systems which take the end users and item providers into consideration. They proposed to maximize user preferences, and also make sure that the items by each provider or retailer could be equally likely to be recommended. Kermany et al. [132] considered the provider fairness too, in additional to the recommendation accuracy and diversity.

- *Others.* Malthouse et al. [126] described their work in sponsored recommendations in which they consider user preferences and the Ad revenue in the optimization process. Mehrotra et al. [130] considered the user interests and clicks, diversity of the singers and the platform promotions in the context of music streaming. Zheng et al. [39] presented a case study in educations, and built different utility functions from the perspective of students and instructors as the objectives. Unger et al. [128] investigated a case study in music listening in which they took the listening behaviors of the end users and the number of fans for the artists into consideration.

MOO Methods. Most of the research in this category adopted the scalarization as the optimization method. More specifically, weighted sum is still the most popular scalarization approach, while Nguyen et al. [121], Malthouse et al. [126] and Zheng et al. [124] used grid search to find the optimal weights, Yıldırım et al. [127] and Louca et al. [125] considered the weights as hyperparameters to be tuned up, and Xia et al. [123] simply used equal weights as their assumptions. All of these work produced a single solution by using the scalarization method, except the work by Lin et al. [122] in which they tried different weights to produce a Pareto optimal set and select the optimal solution from this set. By contrast, Xia et al. [129] used a weighted product method, and Mehrotra et al. [130] utilized a generalizes Gini function [148] which is a special case of the ordered weighted averaging approach as the scalarization method. In addition to the weighting methods, Surer et al.[38] utilized the ϵ-constraints as the scalarization method and adopted grid search for some of the constraints while others were pre-defined. Rodriguez et al. [131] framed the job recommendation as a ϵ-constraints problem but actually used a greedy search as the solution in the experimental evaluations.

Zheng et al. [39] adopted MOEA and used a method similar to TOPSIS described in Section 3.5 to select the optimal solution. More specifically, the upper and lower bounds of the objectives were calculated from the experimental results, so that the gain and loss score for the MOO solution can be computed to sort and rank the solutions in the Pareto set. Kermany et al. [132] used MOEA too, but they did not provide any information related to the selection of the optimal solution in the Pareto optimal set. In addition, Lin et al. [122] suggested to use a least misery or the marginal utility strategy to select the single optimal solution which can be viewed as different methods to find the knee point as described in Section 3.5.

RS Models with MOO. These work usually fuse MOO with RS models in three options – joint learning, learning recommendation list directly, or learning the importance of objectives to further rank items and produce recommendations. The joint learning is the most straightforward method in which researchers just add more objectives into a single loss function, such as the joint learning by learning-to-rank [121, 122, 129], bandit based recommendation model [130], or joint deep learning frameworks (i.e., neural network models) [127, 128]. Some other research proposed to learn the recommendation list directly by using MOEA [132] or integer programming [131, 38, 126]. The work by Zheng et al. [39, 124] utilized MOEA to learn the importance of the objectives in order to rank the items by a utility function to produce the recommendations by considering multiple objectives in MOEA.

Effectiveness. Note that maximizing the utility of the items from the perspective of one stakeholder may hurt the one in view of other stakeholders. Therefore, the researchers would like to demonstrate that multi-stakeholder RSs can enhance the benefits of other stakeholders without or with a small loss for the end users who are the receivers of the recommendations. The potential issue is similar to the one in the category mentioned in Section 4.1 – there are no guidelines to indicate the degree of the "small" loss that can be accepted. For example, Zheng et al. [39] built three types of the models – models which maximize the utility function for students (M_s), models which maximize the utility function for instructors (M_t), and multi-stakeholder models which maximize the utilities for both students and instructors (M_{st}). The optimal model M_{st} was demonstrated to improve students' satisfaction[5] by 37.9% and instructors' satisfaction by 114% in comparison with M_s which is a traditional recommendation model that only considers the end users, though there is a decline in instructors' satisfaction by 17% in comparison with M_t. The improvement of 114% in comparison with M_s is much more important than the 17% decline in comparison with M_t, since students are the receivers of the recommendations instead of the instructors. Therefore M_{st} was still considered as a successful model. However, the optimal "trade-off" should be examined by user studies. Very limited number of research conducted or present user studies to examine whether the trade-off by their offline experiments can be accepted in user studies. Zheng et al. [149] additionally performed a user study to collect the tolerance of each stakeholder. By this way, the bottom line of the recommendation satisfaction can be identified, which can further assist the researchers to find the optimal "trade-off" among different objectives.

4.4. Multi-Task Recommender Systems

Multi-Task Learning (MTL) is an inductive transfer process that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias [150]. A MTL problem can be solved by MOO in which the objectives of each task will be optimized jointly. However, MOO is not the only solution for MTL, since each task can also be optimized independently [151]. Multi-task RSs aims to perform or optimize multiple tasks by a joint optimization process that shares common representations (e.g., latent factors, feature space, neural network layers, etc) among multiple tasks. MOO is widely adopted as the optimizer to solve the MTL problem in multi-task RSs.

Objectives. The definition of tasks may vary from applications to applications in the multi-task RSs, while each task is usually associated with at least one objective. We summarize these tasks and objectives as follows.

[5] The student and instructor satisfaction is measured by a utility function which computes a distance between user expectations and experiences [39].

- Researchers may want to optimize multiple user behaviors or reactions together. By this way, the joint-optimization model in these work [137, 18, 42] can improve the click-through, view-through, comment rate or the probability of purchases in their systems.

- Some other work proposed to improve different recommendation tasks (e.g., rating prediction and top-N recommendations) simultaneously. Hu et al. [141], Hadash et al. [41] and Shi et al. [140] considered both the rating prediction and ranking tasks in a joint learning process.

- Other research try to integrate the recommendation task with a non-recommendation task. For example, researchers can deliver recommendations together with a process of text productions, such as user reviews [134] or opinions [133], recommendation explanations [136] or tips [19], and so forth. Note that some of these non-recommendation tasks may be used to assist the recommendation task. For example, Huang et al. [138] performed different classification tasks and Wang et al. [20] added a process of graph embedding in the joint optimization which can finally improve the recommendations. By taking the rating prediction task into account, Bansal et al. [135] and Wang et al. [139] additionally integrate the process with a tag prediction and trust prediction task, respectively.

MOO Methods. All the work above used the weighted sum method to transform the objectives associated with multiple tasks into a single objective, so that they can be optimized through a joint learning process. Regarding the weighting parameters, the most common method is treating weights as hyper-parameters to be tuned up in order to get the best performance [141, 134, 135, 20, 152, 138, 140, 18]. Others define the weights based on the importance of each task [41, 139, 19] or simply assigned equal weights [133]. Particularly, Tang et al. [42] defined the weights at the initialization process and used a decay function to adjust the weights in the training iterations, to obtain a better solution. Chen et al [136] and Shi et al. [140] adopted grid search to find the optimal weights in the experiments.

RS Models with MOO. All of these research built the recommendation models and optimized the objectives through a joint learning process. More specifically, the recommendation models were built based on specific machine learning methods, such as machine factorization [141], tensor factorization [133], multilayer neural network models which integrate several structures (e.g., multi-layer perceptron models with a gated recurrent neural network by Li et al. [19]). These models usually allow adding additional components (i.e., loss function by each objective) into the weighted aggregation so that multiple objectives can be transformed into a single one.

Effectiveness. In most of the work [42, 141, 41, 134, 135, 136, 139, 19, 18], researchers evaluate the objectives of each task in multi-task RSs, and demonstrate that their solutions are able to improve all these objectives. For example, Li et al. [19] incorporated the rating prediction task and tip generation task in a joint-learning process. The proposed multi-task RS was able to reduce prediction errors in the rating prediction task (e.g., MAE and RMSE), and improve the accuracy of generated tips measured by the ROUGE-N score [153] which is a metric in natural language processing.

Some work assumes that the recommendations can be improved by additionally performing other tasks [137, 140], while some others considered the intermediate processing(e.g., graph embedding [20], classifications [138]) as additional tasks. These work believe that their multi-task RSs are effective as long as they can improve the recommendation performance (e.g., precision, recall, etc), without evaluating or presenting the objectives in these additional tasks.

4.5. Clustering and Rule Based RSs

Clustering-based [154] and rule-based [155, 156] recommendation models rely on the quality of clusters or association rules. MOO can be applied to these unsupervised learning process (i.e., clustering or association rule mining) to produce better outputs.

Objectives. Tyagi et al. [142] and Neysiani et al. [143] developed rule-based recommender systems, and considered support and confidence as two objectives to be optimized in the rule mining, in order to produce high-quality rules. Demir et al. [144] generated clusters for Web sessions by taking the overall deviation and connectivity of the clusters, in order to assist Web based recommendations.

MOO Methods. Tyagi et al. [142] and Neysiani et al. [143] utilized the weighted sum method and selected the weights empirically. By contrast, Demir et al. [144] adopted MOEA as the solution and manually selected the optimal solution by an expert based on the plot of Pareto front.

RS Models with MOO. These work were generally applied to the unsupervised learning process (i.e., clustering or association rule mining). The outputs, such as clusters or rules can be further used to assist the recommendations. For example, Tyagi et al. [142] and Neysiani et al. [143] produce rules like $\{t_1, t_2\} \rightarrow \{t_3\}$, so that they can recommend item t_3 to a user if he or she likes t_1 and t_2 before.

Effectiveness. Researchers believe that the outputs by the unsupervised learning through a MOO process are able to improve recommendations. Therefore, they usually evaluate the recommendation quality without comparing different objectives in the unsupervised learning process. Tyagi et al. [142] and Neysiani et al. [143] demonstrated the effectiveness by evaluating the models based on the recommendation performance (e.g., precision, recall and F_1 measure) only, without comparing multiple objectives. Demir et al. [144] examined the quality of the clusters without further experiments on clustering-based recommendations.

5. Summary, Weaknesses and Guidelines

In this section, we deliver a summary of multi-objective recommendations, point out the weekness in the current development of MORS, and finally provide a guideline for the application of MOO in the area of recommender systems.

5.1. Summary: Multi-Objective Recommendations

In this article, we identify five circumstances in which a MORS could be useful. MOP is a common and general problem in RSs as long as we consider multiple stakeholders in RSs or would like to balance the recommendations among different quality metrics. Among these five circumstances, we can observe that there are three applications which are most popular – the RSs balancing multiple quality metrics (23/56), multi-stakeholder (14/56) and multi-task RSs (14/56). We can observe that the number of publications related to MORS surged due to the development of multi-stakeholder and multi-task RSs recently. The topic of RSs balancing multiple quality metrics actually was well-studies in the past decades before.

Among these 56 relevant research, 37 of them utilized the scalarization method to solve MOP, while 19 publications adopted MOEAs as the optimizer. The weighted sum is the most popular scalarization method, while GA-based MOEA is more popular than other MOEAs (e.g., PSO-based approaches). By using MOEAs as the optimizer in recommender systems, researchers need to well design the encoding (e.g., genes in GA or positions in PSO). Take the gene encoding for example, we can use binary encoding (e.g., 1 or 0 to indicate whether an item is recommended or not) [111] or permutation encoding (e.g., integers as item ID in the recommendations list) [112, 118], if the MOEA is developed to learn a top-N recommendation list for each user. Other parameters are associated with the specific evolutionary algorithm, such as the population size, the number of particles, learning rate, etc. Researchers can empirically tune up these parameters to find the optimal results. By contrast, the process of parameter tuning is more complicated if we use scalarization. In addition to the parameters in the single-objective optimizer (e.g., learning rate, regularization rate, etc), researchers have to assign parameters which are associated with the scalarization method, such as the weights in the weighted sum methods, or the constraints in the ϵ-constraint method. There are usually two ways to setup these parameters. First of all, researchers can treat these parameters as hyper-parmeters to be tuned up – as the same as the way to adjust other parameters (e.g., learning rate). This is a common approach, especially in the joint learning by multi-task RSs [42, 20]. Alternatively, researchers can perform a grid search to iterate different values for a parameter, e.g., vary the weight from 0 to 1 with a step of 0.1 in the weighted sum method [98, 17, 143].

Researchers take multiple objectives into account in their research, but they may have different purposes. As a result, the effectiveness can be examined and demonstrated in different ways which can be summarized as follows.

- Researchers would like to improve the quality of recommendations by considering multiple objectives. For example, the research on group recommendations considered group fairness as one additional objective, and assumed that they could improve the group recommendations by using MOO. In the category of clustering-based and rule-based RSs, researchers aimed to improve the recommendations by considering multiple objectives in the clustering or rule mining to generate better clusters or association rules. As a result, the research in these two circumstances only examine the improvement over the recommendation qualities, without comparing other objectives in their experiments.

- Researchers may build multi-task RSs to combine different tasks together, in order to improve the performance of all tasks. In this case, multi-task RSs are expected to demonstrate the improvement on multiple tasks. At current stage, we did not observe conflicting objectives in these multiple tasks, thereby a trade-off is actually not required. However, it is possible to have conflicting objectives in multiple tasks in the future deveopment of multi-task RSs.

- Researchers would like to balance multiple objectives in two circumstances – the multi-stakeholder RSs and the RSs with multiple quality metrics. There are possible conflicting objectives involved in these RSs in which a balance or trade-off is expected to be achieved by the optimal solution.

We can use the methods in Section 3.5 to select a single and optimal solution from the Pareto set which is considered a solution with trade-off in the offline experiments. However, the actual "trade-off" depends on DM's preferences on the objectives (e.g., accuracy and diversity in RSs). It should be achieved by examining these solutions based on user studies, if DM's preferences are not available during the process of system development. However, most of the research compared different models through offline experiments only, without performing or presenting post-experiment user studies.

5.2. Weaknesses

Based on the review above, we have identified several weaknesses in the current development of MORS.

- First of all, some researchers are satisfied with a MOO solution as long as it is better than the baselines without proving that the MOO solution is Pareto optimal. The MOO methods, especially the scalarization methods, can produce Pareto optimal solutions under specific conditions. Without validating these requirements, it is not guaranteed that researchers can find the optimal solution, though they are able to find a better solution than the baselines.
- The weighted sum is the most popular scalarization method, but it is necessary to normalize the objective values before assigning weights to them. Otherwise, the results may be overwhelmed by the objective with larger scales. Many research work did not exploit the scales of the objectives or did not mention normalization in their work.
- By using MOEAs as the optimizer, a single best solution is usually required in the area of recommender systems. Some of the existing work either did not select the single best solution, or did not provide any information about the selection method. For example, most of the work in the category

of RSs with mutliple metrics compared the descriptive statistics (e.g., min, max, mean) among the Pareto optimal solutions without selecting a single optimal solution from the Pareto set. Actually, it is better to compare different selection strategies to observe which one is better.

- It is well-known that the evolutionary algorithms are easily to converge in local optima. Therefore, it is important to tune up the parameters related to the niche method which can improve the diversity of the Pareto optimal solutions and enhance search ability in MOEAs.

- Some research assume that they are able to improve the recommendation performance by taking multiple objectives into account. However, they only evaluated the recommendation performance (e.g., precision, recall, etc) without comparing multiple objectives (e.g., group fairness in the group RSs) in their experimental analysis. It is difficult to validate the dependency between these objectives and the improvement on recommendations, without presenting and comparing multiple objectives in their experiments.

- Both scalarization and MOEAs can be applied as the optimizer if DM's preferences are not available. There are no existing research which compare these two types of the optimizer in the research work.

5.3. Guidelines

Finally, we deliver our guidelines based on the flowchart described in Figure 1 .

First, researchers should clearly define the MOP and multiple objectives at the beginning. It is better to declare whether the MOP is convex or not. Afterwards, scalarization and/or MOEAs should be adopted as the optimization methods. With explicit and known DM's preferences (e.g., weights), the MOP can be converted and treated as a SOP directly. Otherwise, we can select scalarization and/or MOEAs as the MOO methods. In most cases, researchers do not know the DM's preferences in RSs unless researchers collect their preferences

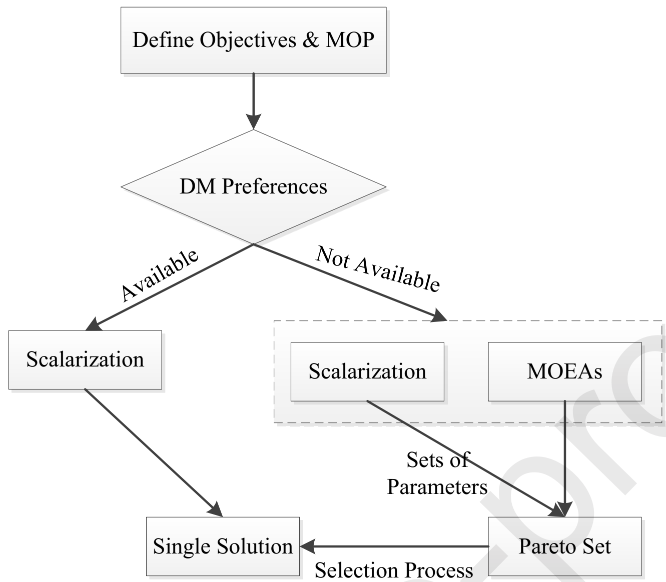

Figure 1: Suggested Workflow for MORS

before. For example, researchers may be able to collect how the end users prefer the recommendations in terms of the accuracy, diversity and novelty in advance. Otherwise, DM preferences are usually not available in the applications, and we have to produce a Pareto set and select the optimal solution from the set. We can apply MOEAs directly, or try different parameters and run the scalarization methods for several times, in order to obtain a Pareto set. Consequently, we can adopt some strategies (described in Section 3.5) to select a single optimal solution from the Pareto set.

In the experimental evaluations, researchers are suggested to compare the models based on different objectives, in addition to evaluating the quality of recommendations. It enables researchers to discover more insights about the correlations between the objectives and the recommendation quality. Furthermore, a trade-off is required when there are conflicting objectives. Some research claim that the MOO solution is effective since it improves some objectives at an acceptable loss at other objectives. The post-experiment user studies may be necessary to examine whether the trade-off is acceptable, if the DM's preferences are not available at the moment. User studies are also beneficial for researchers to figure out the degree of the tolerance with respect to different objectives.

6. Conclusions and Future Work

Multi-objective optimization becomes an emerging concern and demand in the area of recommender systems. In this article, we summarize the circumstances in which a MORS could be useful, discuss the multi-objective optimization methods, point out the weaknesses in the current research, and provide a guideline for the future development of MORS.

We identify the following challenges which could be considered as future work.

- Without DM's preferences, both scalarization and MOEAs can be selected as the MOO methods. However, there are no existing work which compare these two categories of the optimization approaches.
- There are different ways to select the single optimal solution from the Pareto set. It is interesting to compare different selection strategies to figure out more insights, e.g., which one is better or more efficient.
- User-centric studies are necessary to deliver more reliable evaluations, especially when there are conflicting objectives.
- It is interesting to exploit other circumstances or applications (e.g., cross-domain or multi-criteria RSs) where MOO can help build better recommender systems.

References

[1] J. B. Schafer, J. Konstan, J. Riedl, Recommender systems in e-commerce, in: Proceedings of the 1st ACM conference on Electronic commerce, 1999, pp. 158–166.

[2] B. Smith, G. Linden, Two decades of recommender systems at amazon. com, Ieee internet computing 21 (3) (2017) 12–18.

[3] S. Chang, Y. Zhang, J. Tang, D. Yin, Y. Chang, M. A. Hasegawa-Johnson, T. S. Huang, Streaming recommender systems, in: Proceedings of the 26th international conference on world wide web, 2017, pp. 381–389.

[4] C. A. Gomez-Uribe, N. Hunt, The netflix recommender system: Algorithms, business value, and innovation, ACM Transactions on Management Information Systems (TMIS) 6 (4) (2015) 1–19.

[5] M. Schedl, P. Knees, B. McFee, D. Bogdanov, M. Kaminskas, Music recommender systems, in: Recommender systems handbook, Springer, 2015, pp. 453–492.

[6] N. Manouselis, H. Drachsler, R. Vuorikari, H. Hummel, R. Koper, Recommender systems in technology enhanced learning, in: Recommender systems handbook, Springer, 2011, pp. 387–415.

[7] Y. Zheng, Identifying dominators and followers in group decision making based on the personality traits., in: IUI Workshops, 2018.

[8] I. Guy, Social recommender systems, in: Recommender systems handbook, Springer, 2015, pp. 511–543.

[9] J. He, W. W. Chu, A social network-based recommender system (snrs), in: Data mining for social network data, Springer, 2010, pp. 47–74.

[10] G. Shani, A. Gunawardana, Evaluating recommendation systems, in: Recommender systems handbook, Springer, 2011, pp. 257–297.

[11] A. Karatzoglou, L. Baltrunas, Y. Shi, Learning to rank for recommender systems, in: Proceedings of the 7th ACM Conference on Recommender Systems, 2013, pp. 493–494.

[12] Y. Zheng, D. Wang, Multi-objective recommendations, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 4098–4099.

[13] P. Castells, N. J. Hurley, S. Vargas, Novelty and diversity in recommender systems, in: Recommender systems handbook, Springer, 2015, pp. 881–918.

[14] S. Vargas, P. Castells, Rank and relevance in novelty and diversity metrics for recommender systems, in: Proceedings of the fifth ACM conference on Recommender systems, 2011, pp. 109–116.

[15] M. Kunaver, T. Požrl, Diversity in recommender systems–a survey, Knowledge-based systems 123 (2017) 154–162.

[16] L. Xiao, Z. Min, Z. Yongfeng, G. Zhaoquan, L. Yiqun, M. Shaoping, Fairness-aware group recommendation with pareto-efficiency, in: Proceedings of the Eleventh ACM Conference on Recommender Systems, 2017, pp. 107–115.

[17] Y. Wu, N. Yang, H. Luo, Unified group recommendation towards multiple criteria, in: Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, Springer, 2019, pp. 137–151.

[18] Y. Gu, Z. Ding, S. Wang, L. Zou, Y. Liu, D. Yin, Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 2493–2500.

[19] P. Li, Z. Wang, Z. Ren, L. Bing, W. Lam, Neural rating regression with abstractive tips generation for recommendation, in: Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval, 2017, pp. 345–354.

[20] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, M. Guo, Multi-task feature learning for knowledge graph enhanced recommendation, in: The World Wide Web Conference, 2019, pp. 2000–2010.

[21] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.

[22] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, Bpr: Bayesian personalized ranking from implicit feedback, arXiv preprint arXiv:1205.2618.

[23] P. Lops, M. De Gemmis, G. Semeraro, Content-based recommender systems: State of the art and trends, Recommender systems handbook (2011) 73–105.

[24] J. B. Schafer, D. Frankowski, J. Herlocker, S. Sen, Collaborative filtering recommender systems, in: The adaptive web, Springer, 2007, pp. 291–324.

[25] X. Su, T. M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in artificial intelligence 2009.

[26] R. Burke, Hybrid recommender systems: Survey and experiments, User modeling and user-adapted interaction 12 (4) (2002) 331–370.

[27] S. Zhang, L. Yao, A. Sun, Y. Tay, Deep learning based recommender system: A survey and new perspectives, ACM Computing Surveys (CSUR) 52 (1) (2019) 1–38.

[28] G. Adomavicius, A. Tuzhilin, Context-aware recommender systems, in: Recommender systems handbook, Springer, 2011, pp. 217–253.

[29] S. Raza, C. Ding, Progress in context-aware recommender systems—an overview, Computer Science Review 31 (2019) 84–97.

[30] Y. Zheng, B. Mobasher, R. Burke, Cslim: Contextual slim recommendation algorithms, in: Proceedings of the 8th ACM Conference on Recommender Systems, 2014, pp. 301–304.

[31] J. Masthoff, Group recommender systems: Combining individual models, in: Recommender systems handbook, Springer, 2011, pp. 677–702.

[32] G. Adomavicius, N. Manouselis, Y. Kwon, Multi-criteria recommender systems, in: Recommender systems handbook, Springer, 2011, pp. 769–803.

[33] Y. Zheng, Criteria chains: a novel multi-criteria recommendation approach, in: Proceedings of the 22nd International Conference on Intelligent User Interfaces, 2017, pp. 29–33.

[34] D. Monti, G. Rizzo, M. Morisio, A systematic literature review of multicriteria recommender systems, Artificial Intelligence Review 54 (2021) 427–468.

[35] I. Cantador, I. Fernández-Tobías, S. Berkovsky, P. Cremonesi, Cross-domain recommender systems, in: Recommender systems handbook, Springer, 2015, pp. 919–959.

[36] I. Fernández-Tobías, I. Cantador, M. Kaminskas, F. Ricci, Cross-domain recommender systems: A survey of the state of the art, in: Spanish conference on information retrieval, sn, 2012, pp. 1–12.

[37] H. Abdollahpouri, G. Adomavicius, R. Burke, I. Guy, D. Jannach, T. Kamishima, J. Krasnodebski, L. Pizzato, Multistakeholder recommendation: Survey and research directions, User Modeling and User-Adapted Interaction 30 (1) (2020) 127–158.

[38] Ö. Sürer, R. Burke, E. C. Malthouse, Multistakeholder recommendation with provider constraints, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 54–62.

[39] Y. Zheng, N. Ghane, M. Sabouri, Personalized educational learning with multi-stakeholder optimizations, in: Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization, 2019, pp. 283–289.

[40] X. Ning, G. Karypis, Multi-task learning for recommender system, in: Proceedings of 2nd Asian Conference on Machine Learning, JMLR Workshop and Conference Proceedings, 2010, pp. 269–284.

[41] G. Hadash, O. S. Shalom, R. Osadchy, Rank and rate: multi-task learning for recommender systems, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018, pp. 451–454.

[42] H. Tang, J. Liu, M. Zhao, X. Gong, Progressive layered extraction (ple): A novel multi-task learning (mtl) model for personalized recommendations, in: Fourteenth ACM Conference on Recommender Systems, 2020, pp. 269–278.

[43] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender system application developments: a survey, Decision Support Systems 74 (2015) 12–32.

[44] S. M. Al-Ghuribi, S. A. M. Noah, Multi-criteria review-based recommender system–the state of the art, IEEE Access 7 (2019) 169446–169468.

[45] L. Peška, T. M. Tashu, T. Horváth, Swarm intelligence techniques in recommender systems-a review of recent research, Swarm and Evolutionary Computation 48 (2019) 201–219.

[46] R. T. Marler, J. S. Arora, Survey of multi-objective optimization methods for engineering, Structural and multidisciplinary optimization 26 (6) (2004) 369–395.

[47] C. Von Lücken, B. Barán, C. Brizuela, A survey on multi-objective evolutionary algorithms for many-objective problems, Computational optimization and applications 58 (3) (2014) 707–756.

[48] H. Wang, M. Olhofer, Y. Jin, A mini-review on preference modeling and articulation in multi-objective optimization: current status and challenges, Complex & Intelligent Systems 3 (4) (2017) 233–245.

[49] P.-L. Yu, A class of solutions for group decision problems, Management science 19 (8) (1973) 936–946.

[50] E. Gerasimov, V. Repko, Multicriterial optimization, Soviet applied mechanics 14 (11) (1978) 1179–1184.

[51] Y. Haimes, On a bicriterion formulation of the problems of integrated system identification and system optimization, IEEE transactions on systems, man, and cybernetics 1 (3) (1971) 296–297.

[52] A. Charnes, W. W. Cooper, R. O. Ferguson, Optimal estimation of executive compensation by linear programming, Management science 1 (2) (1955) 138–151.

[53] A. Charnes, W. W. Cooper, Goal programming and multiple objective optimizations: Part 1, European journal of operational research 1 (1) (1977) 39–54.

[54] P. C. Fishburn, Exceptional paper—lexicographic orders, utilities and decision rules: A survey, Management science 20 (11) (1974) 1442–1471.

[55] R. E. Steuer, E.-U. Choo, An interactive weighted tchebycheff procedure for multiple objective programming, Mathematical programming 26 (3) (1983) 326–344.

[56] K. Miettinen, Nonlinear multiobjective optimization, Vol. 12, Springer Science & Business Media, 2012.

[57] A. Messac, C. Puemi-Sukam, E. Melachrinoudis, Aggregate objective functions and pareto frontiers: required relationships and practical implications, Optimization and Engineering 1 (2) (2000) 171–188.

[58] V. Chankong, Y. Y. Haimes, Multiobjective decision making: theory and methodology, Courier Dover Publications, 2008.

[59] M. Zeleny, Multiple criteria decision making Kyoto 1975, Vol. 123, Springer Science & Business Media, 2012.

[60] M. Lightner, S. Director, Multiple criterion optimization for the design of electronic circuits, IEEE Transactions on Circuits and Systems 28 (3) (1981) 169–179.

[61] T. W. Athan, P. Y. Papalambros, A note on weighted criteria methods for compromise solutions in multi-objective optimization, Engineering optimization 27 (2) (1996) 155–176.

[62] O. B. Augusto, F. Bennis, S. Caro, A new method for decision making in multi-objective optimization problems, Pesquisa Operacional 32 (2) (2012) 331–369.

[63] J. H. Holland, Genetic algorithms, Scientific american 267 (1) (1992) 66–73.

[64] M. Z. Ali, N. H. Awad, P. N. Suganthan, A. M. Shatnawi, R. G. Reynolds, An improved class of real-coded genetic algorithms for numerical optimization, Neurocomputing 275 (2018) 155–166.

[65] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, Vol. 4, IEEE, 1995, pp. 1942–1948.

[66] X. Luo, Y. Yuan, S. Chen, N. Zeng, Z. Wang, Position-transitional particle swarm optimization-incorporated latent factor analysis, IEEE Transactions on Knowledge and Data Engineering.

[67] W. Liu, Z. Wang, X. Liu, N. Zeng, D. Bell, A novel particle swarm optimization approach for patient clustering from emergency departments, IEEE Transactions on Evolutionary Computation 23 (4) (2018) 632–644.

[68] J. E. Hunt, D. E. Cooke, Learning using an artificial immune system, Journal of network and computer applications 19 (2) (1996) 189–212.

[69] J. Li, Z.-M. Liu, C. Li, Z. Zheng, Improved artificial immune system algorithm for type-2 fuzzy flexible job shop scheduling problem, IEEE Transactions on Fuzzy Systems.

[70] J. D. Schaffer, Multiple objective optimization with vector evaluated genetic algorithms, in: Proceedings of the first international conference on genetic algorithms and their applications, 1985, Lawrence Erlbaum Associates. Inc., Publishers, 1985.

[71] C. M. Fonseca, P. J. Fleming, Multiobjective genetic algorithms, in: IEE colloquium on genetic algorithms for control systems engineering, IET, 1993, pp. 6–1.

[72] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evolutionary computation 2 (3) (1994) 221–248.

[73] C. C. Coello, M. S. Lechuga, Mopso: A proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Vol. 2, IEEE, 2002, pp. 1051–1056.

[74] A. J. Nebro, J. J. Durillo, J. Garcia-Nieto, C. C. Coello, F. Luna, E. Alba, Smpso: A new pso-based metaheuristic for multi-objective optimization, in: 2009 IEEE Symposium on computational intelligence in multi-criteria decision-making (MCDM), IEEE, 2009, pp. 66–73.

[75] N. Zeng, D. Song, H. Li, Y. You, Y. Liu, F. E. Alsaadi, A competitive mechanism integrated multi-objective whale optimization algorithm with differential evolution, Neurocomputing 432 (2021) 170–182.

[76] G.-C. Luh, C.-H. Chueh, W.-W. Liu, Moia: multi-objective immune algorithm, Engineering Optimization 35 (2) (2003) 143–164.

[77] K. C. Tan, C. K. Goh, A. Mamun, E. Ei, An evolutionary artificial immune system for multi-objective optimization, European Journal of Operational Research 187 (2) (2008) 371–392.

[78] Y. Qi, F. Liu, M. Liu, M. Gong, L. Jiao, Multi-objective immune algorithm with baldwinian learning, Applied Soft Computing 12 (8) (2012) 2654–2674.

[79] S. Das, P. N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE transactions on evolutionary computation 15 (1) (2010) 4–31.

[80] W. Liu, Z. Wang, Y. Yuan, N. Zeng, K. Hone, X. Liu, A novel sigmoid-function-based adaptive weighted particle swarm optimizer, IEEE transactions on cybernetics.

[81] N. Zeng, Z. Wang, W. Liu, H. Zhang, K. Hone, X. Liu, A dynamic neighborhood-based switching particle swarm optimization algorithm, IEEE Transactions on Cybernetics.

[82] W. Liu, Z. Wang, N. Zeng, Y. Yuan, F. E. Alsaadi, X. Liu, A novel randomised particle swarm optimizer, International Journal of Machine Learning and Cybernetics 12 (2) (2021) 529–540.

[83] M. Bhattacharya, Diversity handling in evolutionary landscape, arXiv preprint arXiv:1411.4148.

[84] S. Chauhan, M. Singh, A. K. Aggarwal, Diversity driven multi-parent evolutionary algorithm with adaptive non-uniform mutation, Journal of Experimental & Theoretical Artificial Intelligence (2020) 1–32.

[85] J. Horn, N. Nafpliotis, D. E. Goldberg, A niched pareto genetic algorithm for multiobjective optimization, in: Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence, Ieee, 1994, pp. 82–87.

[86] A. Konak, D. W. Coit, A. E. Smith, Multi-objective optimization using genetic algorithms: A tutorial, Reliability engineering & system safety 91 (9) (2006) 992–1007.

[87] J.-H. Chen, D. E. Goldberg, S.-Y. Ho, K. Sastry, Fitness inheritance in multi-objective optimization, in: Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation, GECCO'02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002, p. 319–326.

[88] Y. Censor, Pareto optimality in multiobjective problems, Applied Mathematics and Optimization 4 (1) (1977) 41–59.

[89] F. Waltz, An engineering approach: hierarchical optimization criteria, IEEE Transactions on Automatic Control 12 (2) (1967) 179–180.

[90] J. Branke, K. Deb, H. Dierolf, M. Osswald, Finding knees in multi-objective optimization, in: International conference on parallel problem solving from nature, Springer, 2004, pp. 722–731.

[91] K. Deb, S. Gupta, Understanding knee points in bicriteria problems and their implications as preferred solution principles, Engineering optimization 43 (11) (2011) 1175–1204.

[92] G. Yu, Y. Jin, M. Olhofer, A method for a posteriori identification of knee points based on solution density, in: 2018 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2018, pp. 1–8.

[93] E. Zitzler, D. Brockhoff, L. Thiele, The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration, in: International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2007, pp. 862–876.

[94] K. Yoon, A reconciliation among discrete compromise solutions, Journal of the Operational Research Society 38 (3) (1987) 277–286.

[95] C.-L. Hwang, Y.-J. Lai, T.-Y. Liu, A new approach for multiple objective decision making, Computers & operations research 20 (8) (1993) 889–899.

[96] J. Figueira, S. Greco, M. Ehrgott, Multiple criteria decision analysis: state of the art surveys.

[97] M. G. C. Tapia, C. A. C. Coello, Applications of multi-objective evolutionary algorithms in economics and finance: A survey, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 532–539.

[98] T. Di Noia, J. Rosati, P. Tomeo, E. Di Sciascio, Adaptive multi-attribute diversity for recommender systems, Information Sciences 382 (2017) 234–253.

[99] N. E. I. Karabadji, S. Beldjoudi, H. Seridi, S. Aridhi, W. Dhifli, Improving memory-based user collaborative filtering with evolutionary multi-objective optimization, Expert Systems with Applications 98 (2018) 153–165.

[100] A. Paul, Z. Wu, K. Liu, S. Gong, Robust multi-objective visual bayesian personalized ranking for multimedia recommendation, Applied Intelligence (2021) 1–12.

[101] S. Patil, D. Banerjee, S. Sural, A graph theoretic approach for multi-objective budget constrained capsule wardrobe recommendation, ACM Transactions on Information Systems (TOIS) 40 (1) (2021) 1–33.

[102] R. Xie, Y. Liu, S. Zhang, R. Wang, F. Xia, L. Lin, Personalized approximate pareto-efficient recommendation, in: Proceedings of the Web Conference 2021, 2021, pp. 3839–3849.

[103] A. Lacerda, Multi-objective ranked bandits for recommender systems, Neurocomputing 246 (2017) 12–24.

[104] S. Wang, M. Gong, H. Li, J. Yang, Multi-objective optimization for long tail recommendation, Knowledge-Based Systems 104 (2016) 145–155.

[105] M. T. Ribeiro, A. Lacerda, A. Veloso, N. Ziviani, Pareto-efficient hybridization for multi-objective recommender systems, in: Proceedings of the sixth ACM conference on Recommender systems, 2012, pp. 19–26.

[106] L. Cui, P. Ou, X. Fu, Z. Wen, N. Lu, A novel multi-objective evolutionary algorithm for recommendation systems, Journal of Parallel and Distributed Computing 103 (2017) 53–63.

[107] Y. Zuo, M. Gong, J. Zeng, L. Ma, L. Jiao, Personalized recommendation based on evolutionary multi-objective optimization [research frontier], IEEE Computational Intelligence Magazine 10 (1) (2015) 52–62.

[108] L. Zhang, X. Zhang, F. Cheng, X. Sun, H. Zhao, Personalized recommendation for crowdfunding platform: A multi-objective approach, in: 2019 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2019, pp. 3316–3324.

[109] R. S. Fortes, A. Lacerda, A. Freitas, C. Bruckner, D. Coelho, M. Gonçalves, User-oriented objective prioritization for meta-featured multi-objective recommender systems, in: Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization, 2018, pp. 311–316.

[110] J. Pang, J. Guo, W. Zhang, Using multi-objective optimization to solve the long tail problem in recommender system, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2019, pp. 302–313.

[111] C. Xu, A. S. Ding, S. S. Liao, A privacy-preserving recommendation method based on multi-objective optimisation for mobile users, International Journal of Bio-Inspired Computation 16 (1) (2020) 23–32.

[112] A. Jain, P. K. Singh, J. Dhar, Multi-objective item evaluation for diverse as well as novel item recommendations, Expert Systems with Applications 139 (2020) 112857.

[113] B. Cao, J. Zhao, Z. Lv, P. Yang, Diversified personalized recommendation optimization based on mobile data, IEEE Transactions on Intelligent Transportation Systems.

[114] X. Cai, Z. Hu, P. Zhao, W. Zhang, J. Chen, A hybrid recommendation system with many-objective evolutionary algorithm, Expert Systems with Applications 159 (2020) 113648.

[115] R. S. Fortes, D. X. de Sousa, D. G. Coelho, A. M. Lacerda, M. A. Gonçalves, Individualized extreme dominance (inded): A new preference-based method for multi-objective recommender systems, Information Sciences 572 (2021) 558–573.

[116] B. Geng, L. Li, L. Jiao, M. Gong, Q. Cai, Y. Wu, Nnia-rs: A multi-objective optimization based recommender system, Physica A: Statistical Mechanics and its Applications 424 (2015) 383–397.

[117] Z.-Y. Chai, Y.-L. Li, Y.-M. Han, S.-F. Zhu, Recommendation system based on singular value decomposition and multi-objective immune optimization, IEEE Access 7 (2018) 6060–6071.

[118] T.-m. Ma, X. Wang, F.-c. Zhou, S. Wang, Research on diversity and accuracy of the recommendation system based on multi-objective optimization, Neural Computing and Applications (2020) 1–9.

[119] Z. Chai, Y. Li, S. Zhu, P-moia-rs: a multi-objective optimization and decision-making algorithm for recommendation systems, Journal of Ambient Intelligence and Humanized Computing 12 (2021) 443–454.

[120] F. Zou, D. Chen, Q. Xu, Z. Jiang, J. Kang, A two-stage personalized recommendation based on multi-objective teaching–learning-based optimization with decomposition, Neurocomputing 452 (2021) 716–727.

[121] P. Nguyen, J. Dines, J. Krasnodebski, A multi-objective learning to re-rank approach to optimize online marketplaces for multiple stakeholders, arXiv preprint arXiv:1708.00651.

[122] X. Lin, H. Chen, C. Pei, F. Sun, X. Xiao, H. Sun, Y. Zhang, W. Ou, P. Jiang, A pareto-efficient algorithm for multiple objective optimization

in e-commerce recommendation, in: Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 20–28.

[123] X. Xia, X. Wang, J. Li, X. Zhou, Multi-objective mobile app recommendation: A system-level collaboration approach, Computers & Electrical Engineering 40 (1) (2014) 203–215.

[124] Y. Zheng, T. Dave, N. Mishra, H. Kumar, Fairness in reciprocal recommendations: A speed-dating study, in: Adjunct publication of the 26th conference on user modeling, adaptation and personalization, 2018, pp. 29–34.

[125] R. Louca, M. Bhattacharya, D. Hu, L. Hong, Joint optimization of profit and relevance for recommendation systems in e-commerce., in: RMSE Workshop@ ACM RecSys, 2019.

[126] E. C. Malthouse, Y. K. Hessary, K. A. Vakeel, R. Burke, M. Fudurić, An algorithm for allocating sponsored recommendations and content: Unifying programmatic advertising and recommender systems, Journal of Advertising 48 (4) (2019) 366–379.

[127] E. Yıldırım, P. Azad, Ş. G. Öğüdücü, bideepfm: A multi-objective deep factorization machine for reciprocal recommendation, Engineering Science and Technology, an International Journal.

[128] M. Unger, M. C. Cohen, B. Brost, P. Li, A. Tuzhilin, Deep multi-objective multi-stakeholder music recommendation, NYU Stern School of Business Forthcoming.

[129] X. Xia, X. Wang, X. Zhou, B. Liu, Evolving mobile app recommender systems: An incremental multi-objective approach, in: Future Information Technology, Springer, 2014, pp. 21–27.

[130] R. Mehrotra, N. Xue, M. Lalmas, Bandit based optimization of multiple objectives on a music streaming platform, in: Proceedings of the 26th

ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3224–3233.

[131] M. Rodriguez, C. Posse, E. Zhang, Multiple objective optimization in recommender systems, in: Proceedings of the sixth ACM conference on Recommender systems, 2012, pp. 11–18.

[132] N. R. Kermany, W. Zhao, J. Yang, J. Wu, L. Pizzato, An ethical multi-stakeholder recommender system based on evolutionary multi-objective optimization, in: 2020 IEEE International Conference on Services Computing (SCC), IEEE, 2020, pp. 478–480.

[133] N. Wang, H. Wang, Y. Jia, Y. Yin, Explainable recommendation via multi-task learning in opinionated text data, in: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 165–174.

[134] P. Avinesh, Y. Ren, C. M. Meyer, J. Chan, Z. Bao, M. Sanderson, J3r: Joint multi-task learning of ratings and review summaries for explainable recommendation, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2019, pp. 339–355.

[135] T. Bansal, D. Belanger, A. McCallum, Ask the gru: Multi-task learning for deep text recommendations, in: proceedings of the 10th ACM Conference on Recommender Systems, 2016, pp. 107–114.

[136] Z. Chen, X. Wang, X. Xie, T. Wu, G. Bu, Y. Wang, E. Chen, Co-attentive multi-task learning for explainable recommendation., in: IJCAI, 2019, pp. 2137–2143.

[137] C. Gao, X. He, D. Gan, X. Chen, F. Feng, Y. Li, T.-S. Chua, D. Jin, Neural multi-task recommendation from multi-behavior data, in: 2019 IEEE 35th International Conference on Data Engineering (ICDE), IEEE, 2019, pp. 1554–1557.

[138] Z. Huang, J. Tang, G. Shan, J. Ni, Y. Chen, C. Wang, An efficient passenger-hunting recommendation framework with multitask deep learning, IEEE Internet of Things Journal 6 (5) (2019) 7713–7721.

[139] S. Wang, L. Zhang, M. Yu, Y. Wang, Z. Ma, Y. Zhao, Attribute-aware multi-task recommendation, The Journal of Supercomputing 77 (5) (2021) 4419–4437.

[140] Y. Shi, M. Larson, A. Hanjalic, Unifying rating-oriented and ranking-oriented collaborative filtering for improved recommendation, Information Sciences 229 (2013) 29–39.

[141] J. Hu, P. Li, Collaborative multi-objective ranking, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 1363–1372.

[142] S. Tyagi, K. K. Bharadwaj, Enhancing collaborative filtering recommendations by utilizing multi-objective particle swarm optimization embedded association rule mining, Swarm and Evolutionary Computation 13 (2013) 1–12.

[143] B. S. Neysiani, N. Soltani, R. Mofidi, M. H. Nadimi-Shahraki, Improve performance of association rule-based collaborative filtering recommendation systems using genetic algorithm, Int. J. Inf Technol. Comput. Sci 2 (2019) 48–55.

[144] G. N. Demir, A. S. Uyar, S. G. Ögüdücü, Graph-based sequence clustering through multiobjective evolutionary algorithms for web recommender systems, in: Proceedings of the 9th annual conference on Genetic and evolutionary computation, 2007, pp. 1943–1950.

[145] T. Murakami, K. Mori, R. Orihara, Metrics for evaluating the serendipity of recommendation lists, in: Annual conference of the Japanese society for artificial intelligence, Springer, 2007, pp. 40–46.

[146] D. Jannach, L. Lerche, F. Gedikli, G. Bonnin, What recommenders recommend–an analysis of accuracy, popularity, and sales diversity effects, in: International conference on user modeling, adaptation, and personalization, Springer, 2013, pp. 25–37.

[147] R. K. Jain, D.-M. W. Chiu, W. R. Hawe, et al., A quantitative measure of fairness and discrimination, Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA.

[148] J. A. Weymark, Generalized gini inequality indices, Mathematical Social Sciences 1 (4) (1981) 409–430.

[149] Y. Zheng, J. Ruiz Toribio, The role of transparency in multi-stakeholder educational recommendations, User modeling and user-adapted interaction 31.

[150] R. Caruana, Multitask learning, Machine learning 28 (1) (1997) 41–75.

[151] O. Sener, V. Koltun, Multi-task learning as multi-objective optimization, NIPS'18, Curran Associates Inc., Red Hook, NY, USA, 2018, p. 525–536.

[152] X. Zhao, G. Li, M. Wang, J. Yuan, Z.-J. Zha, Z. Li, T.-S. Chua, Integrating rich information for video recommendation with multi-task rank aggregation, in: Proceedings of the 19th ACM international conference on Multimedia, 2011, pp. 1521–1524.

[153] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Text summarization branches out, 2004, pp. 74–81.

[154] S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem, A. Prugel-Bennett, Novel centroid selection approaches for kmeans-clustering based recommender systems, Information sciences 320 (2015) 156–189.

[155] F. Abel, I. I. Bittencourt, N. Henze, D. Krause, J. Vassileva, A rule-based recommender system for online discussion forums, in: International

Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Springer, 2008, pp. 12–21.

[156] M. K. Swamy, P. K. Reddy, Improving diversity performance of association rule based recommender systems, in: Database and Expert Systems Applications, Springer, 2015, pp. 499–508.