Journal of Computer Science and Artificial Intelligence
ISSN: 3078-8242 (Print), ISSN: 3078-2309 (Online) | Vol. 5, No. 1, 2025

Knowledge Graph-Based Multi-Objective Optimization Recommendation Model

Hua Pang [1], **Long Wang** [2, *]

[1] Shenyang Normal University, Shenyang 110034, China
[2] Liaoning University, Shenyang 110034, China

Abstract: The current recommendation application scenario is more complex, and the traditional recommendation algorithm can no longer meet the needs of user diversity. To solve this problem, this paper proposes an improved multi-objective optimization recommendation model based on knowledge graph to optimize four recommendation goals of novelty, diversity, accuracy and recall simultaneously. A user preference set is constructed based on user behavioral preferences and item-related characteristics before recommendation, which provides the basis for subsequent recommendations. Two improved algorithms are proposed in this paper: 1) at the bottom layer, a knowledge graph embedding algorithm with variable weighted scoring function is used to transform the association information between items, into the relationship between vectors; 2) for the top layer, a multi-objective evolutionary algorithm is used to optimize the recommendation list. Comprehensive experiments show that the model can effectively improve the evaluation metrics of the four recommendations. And it provides users with a recommendation list of items containing novel and diverse items in a more efficient way while maintaining accuracy.

Keywords: Knowledge graph, Knowledge graph embedding, Multi-objective optimization algorithm, Two-tier recommendation.

1. Introduction

As the number of users who utilize online resources for learning increases, a vast amount of user learning information is also generated accordingly, which provides substantial data support for personalized item recommendation [1]. As an effective approach to address information overload, the core objective of a recommendation system is to mine information that users are interested in from massive datasets by analyzing user behaviors, interests, needs, and other relevant information. Traditional personalized recommendation systems include content-based recommendation algorithms, collaborative filtering-based recommendation algorithms, and hybrid recommendation algorithms. Although collaborative filtering methods are generally effective and universally applicable, they still suffer from several issues, such as the cold start problem [2] and the data sparsity problem [3]. Researchers have attempted to adopt hybrid recommendation algorithms to resolve the aforementioned issues, thereby enhancing recommendation performance.

Since Google released Google Knowledge Graph in 2012, research on knowledge graphs in the academic community has been increasing, and the role of knowledge graphs has gradually gained attention from various sectors [4]. In recent years, research on recommendation systems based on knowledge graphs has developed rapidly. Personalized recommendation systems constructed using knowledge graphs have improved the interpretability of recommendation results to a certain extent; however, they are unable to simultaneously optimize the accuracy and diversity of the recommendation results. To address this gap, we employ a multi-objective optimization algorithm to compensate for the limitation that recommendation systems built on knowledge graphs fail to consider the diversity and accuracy of recommendation results.

In summary, this paper makes three main contributions:

(1) A two-layer recommendation model is proposed. The bottom layer adopts a knowledge graph entity embedding algorithm to provide candidate sets for the top layer, which further optimizes the candidate sets to ultimately generate recommendations.

(2) The entity embedding algorithm (TransD algorithm) is improved to fully consider the influence of vectors in different dimensions.

(3) The traditional multi-objective optimization algorithm (NSGA-III algorithm) is modified to simultaneously optimize accuracy, recall, diversity, and coverage.

2. Related Work

(1) Knowledge Graph-Based Recommendation

To convert high-dimensional data information into computable low-dimensional data, high-dimensional data is embedded into low-dimensional vectors, enabling researchers to better mine potential information in the knowledge graph. Bordes A et al. proposed the classic embedding algorithm TransE, which treats entities as two matrices, performs operations on these two matrices, and thereby achieves the goal of link prediction. However, TransE often exhibits poor performance when dealing with complex correspondence relationships such as one-to-many, many-to-one, and many-to-many. Subsequently, numerous researchers have made considerable efforts to address this issue [5]. To solve the aforementioned problems, Ji Guoliang et al. proposed the TransD algorithm, which projects head entities, tail entities, and relationships into different spaces respectively. This not only greatly improves the accuracy of link prediction but also adopts the method of dynamically constructing mapping matrices, which reduces the substantial computation caused by projection to a large extent [6]. However, the scoring function of TransD does not treat complex entity correspondence relationships such as one-to-many, many-to-one, and many-to-many differently, which is obviously unreasonable. This paper has made modifications to this aspect.

(2) Multi-Objective Optimization

With the development of recommendation algorithms, researchers have found that there are usually multiple factors influencing recommendation results. Consequently, some researchers have combined multi-objective optimization algorithms with recommendation systems to address the problem of optimizing multiple conflicting recommendation attributes while pursuing the optimal solution. Certain researchers have attempted to solve multi-objective optimization problems by simulating the living and evolutionary laws of organisms in nature. Specifically, Deb et al. addressed multi-objective optimization through genetic iteration: they improved genetic algorithms and proposed multi-objective genetic algorithms, which remain a popular approach for solving multi-objective optimization problems to this day [7].

Some researchers have treated various factors affecting recommendation results as conflicting objectives, and optimized these conflicting objectives via multi-objective optimization to mitigate the impact of different factors on recommendation outcomes. Wang et al. [8] adopted the approach of predicting the unknown item ratings of target users, defined genetic operators and representations in a novel manner, and then integrated these with multi-objective genetic algorithms for recommendation, ultimately achieving favorable results. Jain et al. [9] combined collaborative filtering with multi-objective optimization to propose a new multi-objective optimization model, and redefined the crossover operator. Relevant experiments have proven the effectiveness of this new model. Currently, to achieve better recommendation performance and adapt to complex recommendation environments, an increasing number of researchers have integrated multiple individual recommendation algorithms. Facts have demonstrated that this approach is indeed effective.

3. Two-Layer Recommendation Model

The Two-Layer Recommendation Model proposed in this paper reduces the errors caused by the inconsistency between upper and lower objectives in existing models, and makes up for their limitation of only being able to optimize two objectives [10]. This hybrid recommendation model adopts the knowledge graph-based recommendation as the bottom-layer recommendation model, which is responsible for data supplementation and mining potential factors between relationships. On this basis, a recommendation candidate set is generated. Subsequently, accuracy, novelty, and diversity are taken as key factors for the final recommendation to form the ultimate recommendation set—this is the MaORA-TDA Two-Layer Recommendation Model. The recommendation process is illustrated in Figure 1.

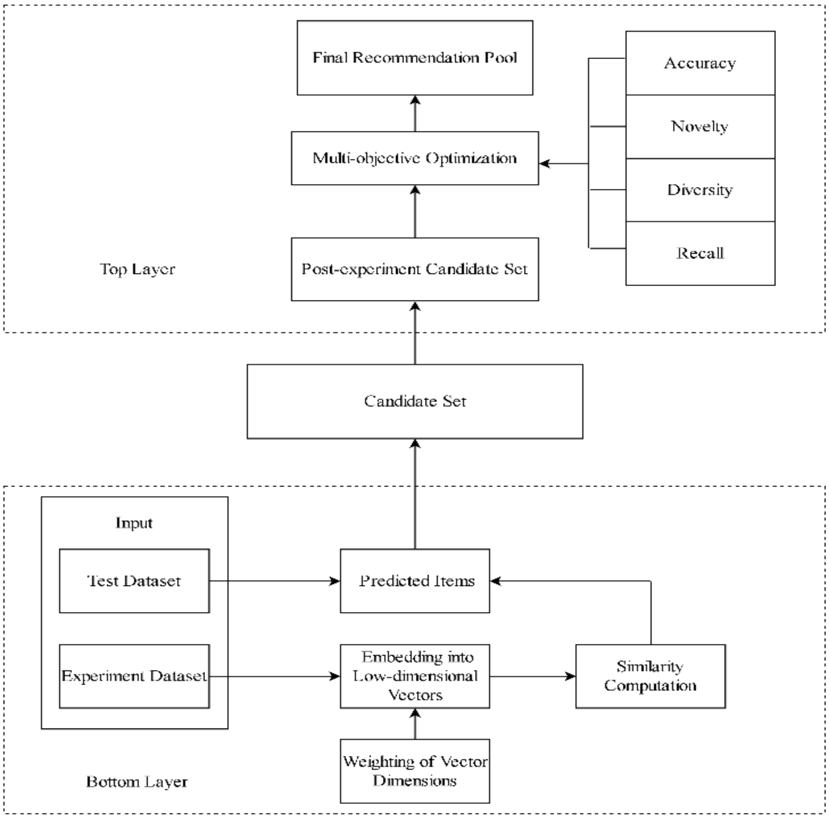

Figure 1. Diagram of the Two-Layer Recommendation Model

(1) Preparation Work

Before conducting recommendations, it is necessary to construct a user behavior profile based on user behavior information. Specifically, whether a user is interested in a certain item is determined by factors such as the time the user spends on the item, whether they view the answer, their score, and their comments—these factors collectively form the user profile.

First, the average time spent on an item by all users is calculated, excluding excessively high and low time data. Rules are then defined as follows: if a user's time spent on the item is longer than the average time, the project is considered to be of great help to them; if the time spent is around this average time, the project is also considered to be highly helpful; if the time spent is shorter than the average time, the project is considered to be of little help to them.

Based on the constructed user profile, the set of items that the user is interested in is filtered out. By embedding relationships and entities into low-dimensional vectors, the similar set of the user-interested item set is obtained through the operation of these low-dimensional vectors.

(2) Knowledge Graph Entity Embedding

A knowledge graph consists of numerous triples denoted as S = {h, r, t} [11], where h and t represent the head node and tail node, respectively, and r denotes the relationship between the head node and the tail node. By constructing such relationships, massive amounts of information can be managed and utilized easily. Moreover, potential connections between different entities can be discovered through the knowledge graph, providing additional information for recommendation tasks. Based on the set of items that users are interested in, projection matrices for entities and relationships are constructed. Each entity and relationship are represented by two vectors (an entity vector and a projection vector): one vector represents the embedding of the entity/relationship itself, and the other is used to construct the projection matrix. The projection matrix used for each "entity-relationship" pair is distinct, and the head node and tail node are projected separately [12]. Thus, the formula for the projection matrix is:

$$M_{rh} = r_p h_p^T + I^{m \times n} \tag{1}$$

$$M_{rt} = r_p t_p^T + I^{m \times n} \tag{2}$$

Where Mrh and Mrt denote the projection matrices obtained by projecting the head entity and tail entity into different relational spaces, respectively; hp and tp denote the projection vectors obtained by projecting the head entity and tail entity into the relational space, respectively; n denotes the dimension of the head entity and tail entity; and m denotes the dimension of the relationship.

Projection Operation:

$$h_{\perp} = M_{rh} h, \ t_{\perp} = M_{rt} t \tag{3}$$

Due to the excessive simplicity of the distance metric in the original loss function, it adopts the same scoring function for scenarios such as one-to-N, N-to-one, and N-to-N. It fails to fully consider that each scenario exerts distinct impacts on projection relationships and the accuracy of results. When dealing with diverse data, the uniqueness of the data is not adequately accounted for. Instead, it is necessary to flexibly handle the relationship between entity and relation vectors based on the importance of each dimension in these vectors, thereby reducing the adverse impact of certain dimensions on the algorithm. Therefore, the algorithm is improved.

To fully consider the influence of vectors across different dimensions without treating each dimension equally, a dynamic weight assignment method is adopted. This method helps mitigate the impact of any single dimension on the overall performance of the algorithm. On this basis, we improve TransD and modify its scoring function as follows:

$$f_r(h,t) = \ w_r \|h_{\perp} + r - t_{\perp}\|_2^2 \tag{4}$$

Where ω_r denotes:

$$\omega_r = \frac{h_p^{\perp} h r_p + t_p^{\perp} h r_p}{t_p^{\perp} t r_p \times h_p^{\perp} h r_p} \tag{5}$$

(3) Similarity Calculation

After embedding all items into the low-dimensional space, the vectors of all items are obtained. Subsequently, the similarity between unanswered items and learners' preference sets is calculated using cosine similarity. The higher the similarity between the two, the closer the distance; the lower the similarity, the greater the distance between them. This paper adopts cosine similarity to calculate item similarity, and the formula is as follows:

$$Sim(I,I') = \frac{\sum_{i=1}^{d}(I_i \times I_i')}{\sqrt{\sum_{i=1}^{d} I_i^2} \times \sqrt{\sum_{i=1}^{d} I_i'^2}} \tag{6}$$

Here, d denotes the dimensionality of the item vector. I represents the item vector, and I′ denotes the item vectors within the preference set. By calculating the similarity between the unanswered items and the learner’s preference set, and then ranking them, the top 100 items are selected as the candidate recommendation set, which serves as the basis for subsequent multi-objective optimization. Moreover, the user’s rating for similar items is obtained by multiplying the similarity value with the original rating.

(4) Multi-Objective Optimization

Before performing multi-objective optimization, the candidate set generated based on the knowledge graph needs to be encoded. In this paper, the dimensionality of each individual is set to 10, and each dimension of the individual represents the ID of an item within the candidate set. Through optimization, a set of recommendation lists is obtained, that is, the optimal individual is identified. Each individual is represented as follows:

$$X = \{x_1, x_2, x_3 x_4 ...... x_{10}\} \tag{7}$$

Here, X denotes the representation of an individual, where every 10 items constitute one individual, and x_i represents the ID of a particular item.

Multi-objective optimization has been widely applied in personalized recommendation systems, among which NSGA-III is one of the most classical algorithms. To better meet user requirements and account for item-specific characteristics, this paper makes appropriate modifications to the individual representation, genetic operators, and objective functions of the NSGA-III algorithm. The optimization is performed based on four metrics—accuracy, recall, diversity, and novelty—to filter the candidate set and obtain the final recommendation list.

Therefore, the three commonly used evaluation metrics in recommendation algorithms, along with a newly defined measure of novelty, are adopted as objective functions. Through multi-objective optimization, these four objectives are optimized simultaneously, thereby directly improving recommendation performance across all four aspects [13].

Accuracy is one of the four optimization objectives. It indicates the proportion of positive (i.e., accepted) items within the recommendation list. An item is considered accepted if its rating exceeds 4 or its preference weight is greater than or equal to 0.7. The accuracy is defined as follows:

$$\frac{|L_u \cap T_u|}{|L_u|} \tag{8}$$

Here, L_u denotes the recommendation list for user u, T_u represents the set of items accepted by the user, and $|\cdot|$ is a function that computes the size of the input data list. A higher accuracy value indicates that more items in the recommendation list are accepted by the learner. The accuracy ranges from 0 to 1, where 1 represents the best possible performance of the algorithm. Since the algorithm returns only the top 10 recommended items, accuracy is calculated based solely on these 10 items.

Recall reflects the model's ability to retrieve items from the set of acceptable items. It is defined as follows:

$$\frac{|L_u \cap T_u|}{|T_u|} \tag{9}$$

In addition, a recommendation list with a rich variety of items is more appealing to users than a monotonous one. Therefore, diversity is one of the objectives to be optimized. It reflects the richness of items within the recommendation list. The formula is defined as follows:

$$\frac{\sum_{i\in L_u}\sum_{j\in L_u}(1-sim(i,j))}{|L_u|\,(|L_u|-1)} \tag{10}$$

Here, i and j represent different items within the recommendation list. A higher diversity value indicates a broader (more varied) range of items in the recommendation list.

If an item has been rated by a larger number of users, its novelty decreases. Therefore, when more popular items appear in the recommendation results, the overall novelty value will be lower. Conversely, higher novelty indicates that the system recommends more uncommon or less popular items to the target user. The novelty is defined as follows:

$$\frac{\sum_{i\in I}\log_2\left(\frac{n}{|I_i|}\right)}{|I_u|} \tag{11}$$

Here, I_u denotes the set of items in the user's recommendation list, n represents the total number of ratings, and I_i refers to a specific item I_u within the recommendation list.

Since the optimization involves four objectives, genetic operators are introduced during the optimization process to search for the best individuals. These operators determine both the convergence speed of the algorithm and the quality of the obtained solutions.

(4.1) Crossover

In multi-objective optimization algorithms, the commonly used simulated binary crossover (SBX) operates on real-valued encodings; however, it is not suitable for the discrete encoding method used in this algorithm. Therefore, a one-point uniform crossover is implemented instead.

The crossover process is as follows: first, two parent individuals A and B are randomly selected. Then, a random breakpoint i is chosen in A, and the same breakpoint position is selected in B. The items between the breakpoint i and the endpoint j in A and B are then exchanged. After this operation, two offspring individuals are generated.

Since one-point crossover may result in duplicate items within the offspring individuals, any duplicates are replaced by randomly selected items from the candidate set. The crossover operation is then performed again to produce the next generation of individuals, and this process is repeated iteratively.

(4.2) Mutation

Single-point mutation is used to generate new individuals. Specifically, a mutation point is randomly selected, and the item at that position is replaced with another item that does not appear in the parent individual. This operation is simple and has low computational complexity, yet it can efficiently help discover more robust individuals. Therefore, mutation is widely used in multi-objective optimization algorithms.

4. Experiments

This section first explains the data, comparison models, etc., used in the experiments, and then conducts a comparative analysis of the experimental results.

(1) Introduction to Experimental Dataset and Comparison Models

This paper adopts the MOOper dataset, which is constructed by extracting the interaction data of users participating in practical exercises on the EduCoder platform from 2018 to 2019. The attribute information of courses, practical exercises, levels, and knowledge points, as well as the relationships between them, are modeled as a knowledge graph to build a large-scale practice-oriented online learning dataset (MOOper). This dataset includes interaction data and a knowledge graph, and the relationships in the knowledge graph are illustrated in Figure 2:

To verify the effectiveness of the proposed model, this paper conducts comparative experiments between the proposed two-layer recommendation model (MaORA-TDA) and other recommendation models. MORS [14], NNIA [15], NSGAIII-GBFE-RS [16], and MaORA-IMF [17]are used as comparison models for MaORA-TDA. MORS takes novelty and accuracy as optimization objectives to improve the visibility of niche items. NNIA optimizes diversity and accuracy and uses a multi-objective optimization algorithm to improve item diversity. NSGAIII-GBFE-RS is a rating-based multi-objective hybrid recommendation model that optimizes five objectives: accuracy, recall, diversity, novelty, and coverage. MaORA-IMF is a two-layer recommendation model whose bottom layer adopts an improved matrix factorization algorithm to provide a candidate set for multi-objective optimization, optimizing accuracy, recall, diversity, and novelty.

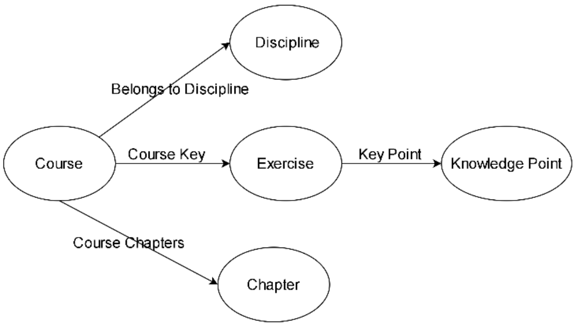

Figure 2. Diagram of the Knowledge Graph

(2) Parameter Setting

The parameters of the multi-objective optimization algorithm in this paper are shown in Table 1:

Table 1. Model Parameter Settings

Parameter	Value
CrossoverProbability	1
Individual Size(n)	10
MutationProbability	1/n
Maximum Generations	5000
Population Size	100

(3) Analysis of Experimental Results

First, the MOOper dataset is divided into 10 equal parts, with 8 parts used for algorithm training and the remaining portion reserved as test data. Accuracy and diversity are used as evaluation metrics to assess the performance of MaORA-TDA. The performance of each algorithm under these metrics is shown below:

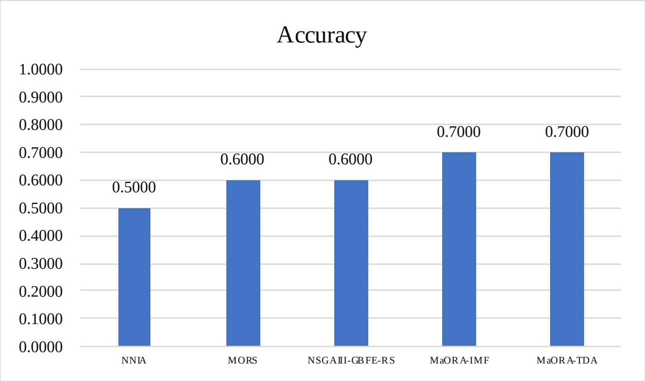

Figure 3. Accuracy performance of each model

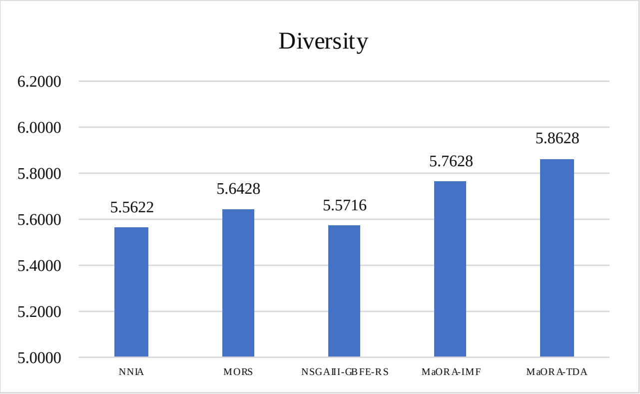

Figure 4. Diversity performance of each model

As can be seen, MaORA-TDA demonstrates good performance in both accuracy and diversity. This is because the knowledge graph contains many potential relationships between entities. By using entity embedding, these latent relationships are converted into similarities between vectors, thereby improving both accuracy and diversity. Its performance in novelty and recall is slightly lower, possibly due to setting the similarity threshold too high when constructing the candidate set.

5. Conclusion

This paper proposes a new two-layer recommendation model, which can significantly improve diversity and novelty while maintaining high accuracy. Through the analysis of experimental results, the effectiveness of the model is verified, and the recommendation list of the model shows superiority in the four evaluation metrics. Although the proposed two-layer recommendation model outperforms state-of-the-art models, there are still many parameters that need to be adjusted, which have a significant impact on the final results. In future work, we will further optimize its parameters and incorporate other related technologies for recommendation.

References

[1] Wang H, Zhang F, Xie X, et al. DKN: deep knowledge-aware network for news recommendation. In: Proceedings of the 2018 World Wide WebConference, 2018. 1835–1844.

[2] Huang Z, Liu Q, Zhai C, et al. Exploring multi-objective exercise recommendations in online education systems. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 2019. 1261–1270

[3] Xinhua Wang, Wenyun Ma, Lei Guo, Haoran Jiang, Fangai Liu, ChangdiXu, HGNN: Hyperedge-based graph neural network for MOOC Course Recommendation, Information Processing & Management, Volume 59, Issue3, 2022, 102938, ISSN 0306-4573.

[4] Adomavicius G, Tuzhilin A. Toward the next generation of recommendersystems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng, 2005, 17: 734–749.

[5] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data [C]//Advances in neural information processing systems. 2013: 2787-2795.

[6] Ji, Guoliang, He, Shizhu, Xu, Liheng, Liu, Kang, Zhao, Jun,et al. Knowledge Graph Embedding via Dynamic Mapping Matrix. Association for Computational Linguistics, 2015.

[7] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2):182-197.

[8] J. Luzuriaga, E. Muñoz, H. Rosales-Méndez and A. Hogan, "Merging Web Tables for Relation Extraction With Knowledge Graphs," in IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 2, pp. 1803-1816, 1 Feb. 2023.

[9] Ahmad, H.K., Qi, C., Wu, Z. et al. ABiNE-CRS: course recommender system in online education using attributed bipartite network embedding. Appl Intell 53, 4665–4684 (2023).

[10] Huanyu Zhang, Xiaoxuan Shen, Baolin Yi, Wei Wang, Yong Feng, KGAN: Knowledge Grouping Aggregation Network for course recommendation in MOOCs, Expert Systems with Applications, Volume 211, 2023, 118344, ISSN 0957-4174.

[11] Anna Y.Q. Huang, Owen H.T. Lu, Stephen J.H. Yang, Effects of artificialIntelligence–Enabled personalized recommendations on learners' learning engagement, motivation, and outcomes in a flipped classroom, Computers & Education, Volume 194, 2023, 104684, ISSN 0360-1315.

[12] Thanh Le, Nam Le, Bac Le, Knowledge graph embedding by relational rotation and complex convolution for link prediction, Expert Systems with Applications, Volume 214, 2023, 119122, ISSN 0957-4174.

[13] Cui Z, Zhang J, Wu D, Cai X, Wang H, Zhang W, Chen J (2020) Hybrid many-objective particle swarm optimization algorithm for green coal production problem. Inf Sci 518:256–271.

[14] Xie. L., Hu. Z., Cai, X.et al. Explainable recommendation based on knowledge graph and multi-objective optimization. Complex Intell. Syst.7, 1241–1252 (2021).

[15] B. Jiang, J. Yang, Y. Qin, T. Wang, M. Wang and W. Pan, "A Service Recommendation Algorithm Based on Knowledge Graph and Collaborative Filtering," in IEEE Access, vol. 9, pp. 50880-50892, 2021.

[16] Kalyanmoy Deb, Himanshu Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sortingapproach, part I: solving problems with box constraints, IEEE Trans. Evol. Comput.18 (4) (2014) 577–601.

[17] Zhang F, Yuan N J, Lian D, et al. Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 353–362.