Three-Stage Grouping Optimization for Large-Scale Collaborative *E*-Learning via Knowledge Graph and E-CARGO

Hua Ma, *Senior Member, IEEE*, Xiangru Fu, *Member, IEEE*, Wensheng Tang, Ming Chen, Haibin Zhu, *Fellow, IEEE*, and Keqin Li, *Fellow, IEEE*

***Abstract*— With the popularity of *e*-learning, the number of learners enrolling in an online class has increased dramatically. Facing the huge learner size and diverse learner characteristics, existing research does not deeply explore the relationships among learners' characteristics, making it difficult to accurately form appropriate learning teams to enhance learners' learning efficiency and quality. We propose a three-stage grouping optimization approach for large-scale collaborative *E*-learning via knowledge graph and the environments—classes, agents, roles, groups, and objects (*E*-CARGOs). First, this approach constructs a knowledge graph of large-scale *E*-learning environments to facilitate accurate learner clustering for reducing computational cost. Second, a multidimensional learner model is proposed to comprehensively evaluate learners' competencies for team leader selections. Third, the remaining learners are assigned to different learning teams based on their complementarities with team leaders, and this problem is modeled as a role-based collaboration (RBC) problem by the E-CARGO model. Meanwhile, an effective and efficient solution via an optimization package is presented. Finally, experiments demonstrate that the proposed approach accurately achieves global grouping optimization with good execution performance in large-scale collaborative *e*-learning scenarios.**

***Index Terms*— Environments—classes, agents, roles, groups, and objects (*E*-CARGOs), grouping optimization, knowledge graph, large-scale *e*-learning, learning team.**

I. Introduction

A. Motivations

AT PRESENT, the world is in a critical period of digital education transformation, and *e*-learning platforms have experienced rapid development, leading to a significant increase in the number of online learners and learning resources. As of May 2024, more than 76 800 MOOCs have been released in China, with a total enrollment of 454 million learners. A total of 415 million credit recognitions have been granted to students, and the platform has served 1.277 billion learning sessions across the country.[1] The number of learners in an online course often exceeds that of a traditional in-person class. For example, more than 23 000 students have taken "Computer Organization" lecture by Associate Professor Junlin Lu on the icourse163.org platform. In this case, personalized instruction for each learner becomes difficult for teachers due to their limited energy, while learners are more eager to complete their learning tasks with high-quality and avoid serious learning risks [1], [2].

To alleviate this issue, researchers attempt to propose learning team formation (LTF) methods [3], [4], [5] to motivate collaborative learning effects in learning teams, which is conducive to improving both learners' learning efficiency and abilities. For example, Li et al. [5] propose an optimized heterogeneous grouping method based on a genetic algorithm (GA), which outperforms other methods in terms of student final performance, collaborative processes, and student perceptions. Despite effectiveness, there are still some limitations to be addressed.

1) Existing studies are inadequate to solve the LTF problem in large-scale *e*-learning environments, resulting in limited applicability. Current *e*-learning scenarios include massive learners with diverse characteristics, which significantly increases the complexity of the LTF problem. The existing approaches are mainly divided into three categories: meta-heuristic methods [5], [6], clustering-based methods [7], [8], and optimization package-based methods [9]. All of these methods are computationally expensive when dealing with large-scale data. For instance, the GA [5] and deterministic crowding algorithm (DCA) [6] suffer from slow convergence issues, and the IBM CPLEX optimization package-based method [9] cannot work when the number of learners exceeds 500, which is a common amount in a large-scale *e*-learning environment.
2) Various differences of learners regarding multidimensional characteristics make it difficult to quantitatively

Received 20 August 2025; accepted 1 January 2026. Date of publication 22 January 2026; date of current version 19 March 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62477009 and Grant 62077014, in part by the Major Scientific and Technological Innovation Platform Project of Hunan Province under Grant 2024JC1003, and in part by the Key Project of Scientific Research Fund of Hunan Provincial Education Department under Grant 23A0061. This article was recommended by Associate Editor M. P. Fanti. *(Corresponding author: Xiangru Fu.)*

Hua Ma, Xiangru Fu, Wensheng Tang, and Ming Chen are with the College of Information Science and Engineering, Hunan Normal University, Changsha 410081, China, and also with Hunan Provincial Key Laboratory of Philosophy and Social Sciences of Artificial Intelligence and International Communication, Changsha 410081, China (e-mail: huama@hunnu.edu.cn; xiangrufu@hunnu.edu.cn; tangws@hunnu.edu.cn; chenming@hunnu.edu.cn).

Haibin Zhu is with the Department of Computer Science and Mathematics, Nipissing University, North Bay, ON P1B 8L7, Canada (e-mail: haibinz@nipissingu.ca).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2026.3651349.

Digital Object Identifier 10.1109/TSMC.2026.3651349

[1] http://www.moe.gov.cn/jyb_xwfb/s5147/202405/t20240515_1130643.html

evaluate learners, enhancing the complexity of LTF in large-scale *e*-learning environments. In existing research, learners are usually grouped according to conventional characteristics such as personalities [10], learning styles [11], and knowledge levels [12]. A few studies have considered the characteristics of *e*-learning environments, e.g., learning time [13] and ethnicity [14]. Learners in large-scale learning environments come from diverse regions, resulting in widely distributed multidimensional features in terms of multiple aspects, such as educational background, geographic location, religion, and learning paths. However, existing research often focuses on a single learner characteristic when forming learning teams, neglecting to comprehensively explore and analyze multidimensional characteristics. Consequently, this limitation affects the effectiveness of grouping optimization.

To address the above limitations, this article proposes a novel three-stage grouping optimization approach for large-scale collaborative *e*-learning. In the first stage, a knowledge graph of large-scale *e*-learning environments is constructed, and the Node2vec algorithm [15] is applied on it to obtain informative learner representations for clustering. The clustering results can help us filter irrelevant learners, reducing computational cost. Regarding the second stage, a well-developed evaluation mechanism is designed to comprehensively model learners from a multidimensional perspective, which is conducive to accurately selecting team leaders for each learning team. As for the third stage, the remaining learners are assigned to appropriate learning teams based on the calculated compatibilities between them and the team leaders. This stage aligns with the role-based collaboration (RBC) [16], [17] methodology, and the environments—classes, agents, roles, groups, and objects (E-CARGO) model [18] is utilized to formalize it. Furthermore, benefiting from the clustering in the first stage, we can solve the LTF problem in large-scale *e*-learning environments via the IBM CPLEX optimization package. Finally, extensive simulation experiments verify the effectiveness of the proposed method.

B. Contributions

The main contributions of this article are as follows.

1) Aiming at the limitation of existing LTF methods for large-scale collaborative *e*-learning, this article proposes a novel E-CARGO model-based and knowledge-enhanced LTF approach, where a knowledge graph is innovatively proposed to improve the clustering results for accurately identifying similar learners, which effectively reduces problem complexity. To the best of our knowledge, this is the first study to investigate the application of E-CARGO for large-scale collaborative learning.
2) Aiming at the difficulty in learner evaluation during the process of LTF caused by multidimensional characteristics, this article proposes a multidimensional learner model to comprehensively evaluate learners' competencies for team leader selections and the complementarities between team members and team leaders. Furthermore, this learner model is conducive to identifying potential conflicts to guarantee the rationality of LTF results.

TABLE I
RELATED WORK WITH KEY FEATURES

Scenarios	Key indicators	Data sources	Related work
Small and medium-sized collaborative learning	Learning perspectives and personality	Class notes	[7]
	Personality	Questionnaires	[10]
	Academic, knowledge level, learning style, and social	Questionnaires	[11]
	Learning experience with each topic, pre-test time, and knowledge level	Quiz and logs	[12]
	Extraversion and prior knowledge	Questionnaires and quizzes	[19]
Large-scale collaborative learning	Learning time	Logs	[13]
	Ethnicity	Questionnaires	[14]
	Knowledge level, interest, and leadership	Questionnaires	[20]

The rest of this article is arranged as follows: Section II reviews the related work. Section III introduces the problem statement. Section IV presents the methodology. Section V introduces the learner clustering method. Section VI discusses team leader selection. Section VII describes team member assignment. Section VIII conducts the experiments and analyzes the results. Finally, Section IX concludes the work.

II. RELATED WORK

A. Learners Characteristic Modeling

The learner characteristic model is a key prerequisite to accurately evaluating the compatibility between learners during the process of LTF. In recent years, many scholars have investigated diverse learners' features to solve the LTF problem, which is shown in Table I. The column "Key indicators" refers to the primary learner's characteristics commonly used in the literature. The column "Data sources" indicates the methods used in the literature to collect data for the key indicators. Logs are usually collected from the *e*-learning system.

The analysis of Table I is summarized as follows.

1) Most literature uses conventional indicators such as knowledge level, learning style, and personality as key indicators. Taking knowledge level as an example, cognitive diagnosis [2] and knowledge tracing [21] can be used to measure learners' competence.
2) Most literature acquires key indicators via questionnaires, quizzes, or logs. Taking questionnaires as an example, the Felder–Silverman learning style model can be used to determine learners' learning styles [9].

Existing research provides an important reference framework for modeling learners. To accurately and comprehensively model learners, this article explores additional learners' characteristics based on existing key indicators. Meanwhile, we will obtain learners' characteristic data from multiple sources to enrich learner modeling.

B. Small- and Medium-Sized LTF

Constructing learning teams in traditional courses to help learners improve their learning efficiency is a mainstream

research direction. In traditional courses, the number of learners is usually no more than 200. Meanwhile, each team usually consists of 3–5 people [8], [12], [22].

Chen et al. [23] propose a 3-D Monte Carlo tree search algorithm, where teams that meet the task requirements can be formed by comprehensively considering learners' online time and skill levels. Singh et al. [24] propose a two-stage optimization method for constructing heterogeneous learning teams. By considering three different types of variables, i.e., discrete, multivalued, and continuous variables, teams are formed with diversity, improving the robustness of LTF. Silva et al. [25] propose a homogeneous LTF method based on clustering, extracting features from learners' C code, and clustering them. Zhang et al. [26] propose a GA with partial repair operators to form learning teams for addressing issues of team robustness, fairness, and student conflicts. Qu et al. [27] propose a competition-oriented student team building method, which balances educational equity and ability improvement by maximizing team utility and minimizing differences in students' utilities.

In previous work [9], we proposed an approach to hybrid LTF based on the E-CARGO model. Using a refined model with five characteristics and three constraints to evaluate learners and their interactions as team members, the efficiency and quality of team formation were greatly enhanced. Nevertheless, this work overlooks the fact that the number of learners in *e*-learning environments is continuously growing. When it comes to a large learner size, the difficulty and complexity of LTF increase dramatically, making it challenging to produce effective solutions.

C. Large-Scale LTF

With the rapid development of MOOCs, learners now have the flexibility to study anytime and anywhere. This flexibility has led to steadily increasing numbers of learners in individual courses, often reaching into the thousands or even tens of thousands. However, the massive scale of participation poses challenges for learner engagement, personalized support, and effective collaboration. In response to these challenges, learning teams are formed to help learners complete their tasks more efficiently and enhance their overall learning outcomes through collaboration and peer support.

To achieve the goal of LTF in a large-scale environment, existing research usually follows a two-stage paradigm: 1) reducing the scale of the LTF problem via certain algorithms such as clustering [13]; and 2) designing LTF algorithms. For example, Qi et al. [13] proposed a collaborative learning grouping method based on clustering. First, pre-grouping is construed according to learners' topic willingness. Then, to improve communication efficiency among team members, clustering algorithms are applied multiple times to iteratively group learners according to their available time. To build inclusive and flexible learning groups, Kohli et al. [14] collect learners' information, e.g., current time zone and grade, and use this information to perform stepwise partitioning to reduce the scale of the grouping problem. At the same time, discrimination is avoided as much as possible by checking the uniqueness of gender or ethnicity. Liao et al. [28] present a context-flow-driven heterogeneous dynamic grouping method for large-scale *e*-learning. To ensure grouping accuracy and flexibility, the fuzzy C-means algorithm and incremental stream processing are utilized for dynamic clustering.

TABLE II
RELATED WORK WITH LTF

Scenarios	Optimization methods	Datasets	Related work
Small and medium-sized collaborative learning	Deterministic crowding evolutionary algorithm	Simulated data	[6]
	Cplex optimization package	Simulated data and self-collected data	[9]
	Three-dimensional monte carlo tree search algorithm	The MOOC user activity dataset and the Web tracking dataset	[23]
	Surrogate optimization method	Simulated data	[24]
	Clustering algorithm	Self-collected data	[25]
	Genetic algorithm	Simulated data	[26]
	Multiple-objective particle swarm optimization algorithm	Self-collected data	[27]
Large-scale collaborative learning	Clustering algorithm	The MOOC user activity dataset and simulated data	[13]
	Partitioning-based algorithm	Self-collected data	[14]
	Particle swarm optimization algorithm	Simulated data	[20]
	Fuzzy C-means algorithm	Self-collected data	[28]
	Linear programming	Self-collected data	[29]

Differently, some research directly groups students by analyzing their characteristics. Ullmann et al. [20] put forward a particle swarm optimization algorithm built on three dimensions (i.e., knowledge level, interests, and leadership) to enhance collaborative learning outcomes and encourage knowledge sharing. Ma et al. [29] develop a personas-based student grouping approach, where students are assigned personas based on their behavioral characteristics, and reinforcement learning is applied to develop appropriate grouping rules, followed by linear programming to produce the most effective grouping strategy.

However, existing research failed to fully explore the connection among learners' multidimensional characteristics in large-scale *e*-learning environments. It is not conducive to modeling learners and thereby affects the effectiveness of LTF.

Moreover, Table II summarizes the solving methods and datasets used in existing studies for a clearer overview.

Specifically, the column "Optimization methods" outlines the algorithms applied to address the group formation problem, while the column "Datasets" indicates the types of data sources employed in each study. Table II indicates that traditional algorithms, such as clustering and GAs, are still used to address the LTF problem. Moreover, as shown in Table II, most existing studies rely on simulated data or self-collected datasets. Although publicly available datasets such as the MOOC user activity dataset and the Web tracking dataset are utilized in some works, they fall short of meeting the requirements for learner modeling in this article. Specifically, the MOOC user activity dataset includes limited profile information (i.e., user ID, gender, education level, and year of birth)

and lacks rich individual characteristics such as learning styles. The Web tracking dataset contains web browsing records and basic demographic attributes (i.e., gender and age). Clearly, no dataset provides sufficient information to support the comprehensive learner modeling required in this study.

By summarizing the datasets and methodologies adopted in prior research, we can better identify current limitations and derive valuable insights that inform both the data selection and methodological design of this work.

D. RBC Methodology and the E-CARGO Model

RBC offers a systematic method for modeling organizational structures by assigning roles to members, enabling effective interaction and coordination to achieve overall optimization. To enhance the efficiency of resource utilization, the responsibilities of each role are clearly defined. Meanwhile, RBC helps maximize the overall benefit of the organization by coordinating individual and collective utilities. As an implementation of RBC, E-CARGO provides support for modeling systems, processes, and behaviors, thereby facilitating the analysis, design, and simulation of collaborative activities in complex systems. Currently, this model has been applied in various domains, including resource recommendation [30], travel recommendation [31], service computing [32], personalized learning [9], and intelligent logistics [33], demonstrating strong generality and adaptability.

To form effective learning teams in large-scale *e*-learning environments, we first evaluate the compatibility among learners and then assign them to appropriate teams based on specific rules to optimize overall learning performance. This process reflects the core characteristics of an RBC problem and can therefore be regarded as a typical instance of RBC. Specifically, each learner is modeled as an agent of the E-CARGO model. There are two types of roles, i.e., leader and member. By assigning appropriate learners to corresponding roles, optimal learning teams can be formed, thereby enhancing the overall learning outcomes.

Therefore, RBC theory is introduced, and E-CARGO is utilized to formally model the LTF problem in the large-scale *e*-learning environment, providing support for role assignment and resource optimization.

III. Problem Statement

Compared to leaderless learning teams, leader-centered learning teams tend to achieve better learning outcomes [9], [34]. This is because the presence of a leader not only facilitates cooperation and communication among team members but also helps maintain information consistency in the team. More learners can engage in team learning, enhancing their learning quality. Existing research has predominantly focused on leader-centered LTF [9], [35]. Accordingly, this article also investigates leader-centered LTF.

In large-scale *e*-learning environments, the problem scenario of LTF is described as follows: Suppose there are I ($I \geq 1000$) students registered for course c. Their characteristics (e.g., learning style, personality, and leadership) are obtained by conducting questionnaires. By combining these characteristics with an analysis of historical learning data from prerequisite courses, we could create comprehensive profiles for learners and further evaluate their compatibility. By assigning learners to the right teams based on the evaluation results, their individual performance and learning outcomes can be significantly improved. The problem of forming learning teams in large-scale *e*-learning environments is depicted in Fig. 1. In Fig. 1, the LTF problem is divided into three parts, including input, output, and process. Their characteristics are as follows:

1) *Characteristics of the Input:* 1) as the number of learners increases, it presents a challenge to the efficiency of LTF; 2) the characteristics of learners in large-scale *e*-learning environments are more diverse, including massive factors such as educational background, geographic location, etc. Comprehensively considering learners' multidimensional characteristics helps form suitable teams and improve the rationale for grouping; 3) learners' historical learning data on *e*-learning platforms plays a vital role in quantifying their abilities; and 4) in large-scale *e*-learning environments, differences in the number of prerequisite courses completed lead to variations in learners' prior knowledge.
2) *Characteristics of the Output:* 1) A suitable team leader could guide collaboration among members and boost the team's overall learning performance [9]. Thus, we assume that a learning team consists of one leader and a certain number of members; 2) learners who enroll in the same course are geographically dispersed, which may make them feel lonely while studying. It is essential to alleviate this issue during the grouping process; and 3) the primary goal of LTF is to foster more effective collaboration. Hence, it is crucial to maximize the comprehensive complementarity between the leader and the members to achieve optimal collaboration.
3) *Characteristics of the Process:* 1) more constraints need to be considered, such as conflicts in learners' available learning time; 2) the importance of a learner's characteristics varies in different scenarios. For example, theoretical courses emphasize learners' analytical abilities, while practical courses stress the importance of applying theoretical knowledge; and 3) in practice, LTF not only focuses on the similarities between learners but also considers the compatibility among them. This dual consideration fosters effective communication within the group while stimulating individual learning potential, thus improving grouping effectiveness.

In general, this article focuses on the following key issues: 1) how can learners' characteristics and historical data from the *e*-learning platform be used to analyze their abilities and form effective teams; 2) how can we decrease the complexity of LTF and boost the efficiency of the formation; and 3) how can we select appropriate leaders and assign the remaining learners to suitable teams, while maximizing the comprehensive compatibility between leaders and their members, meeting the constraints, and reducing learners' loneliness?

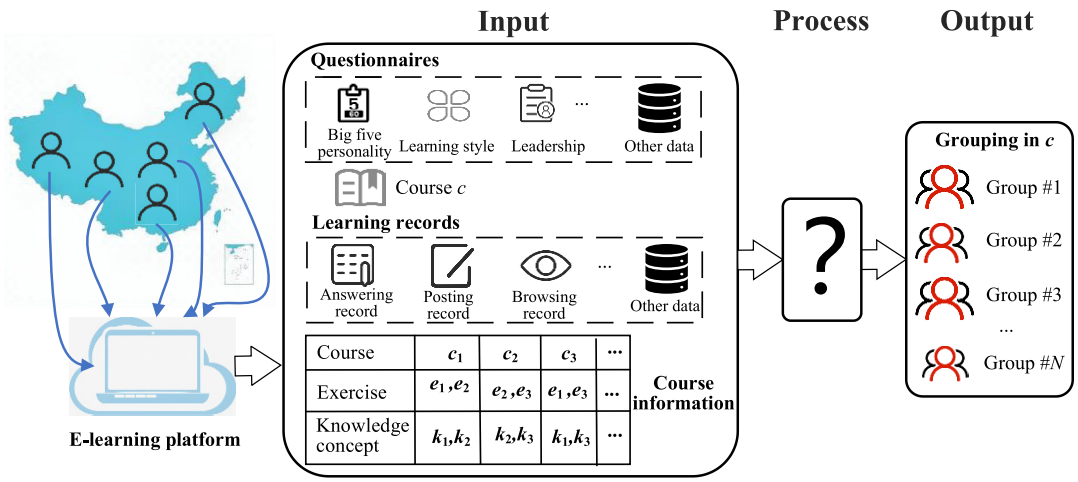

Fig. 1. Problem description. (Input: It exhibits the input data, including learners' personal information obtained by questionnaires and their historical learning records in prerequisite courses; Process: It represents the processing of input data by utilizing a specific algorithm to generate output results; Output: It means the grouping solution obtained for course c.)

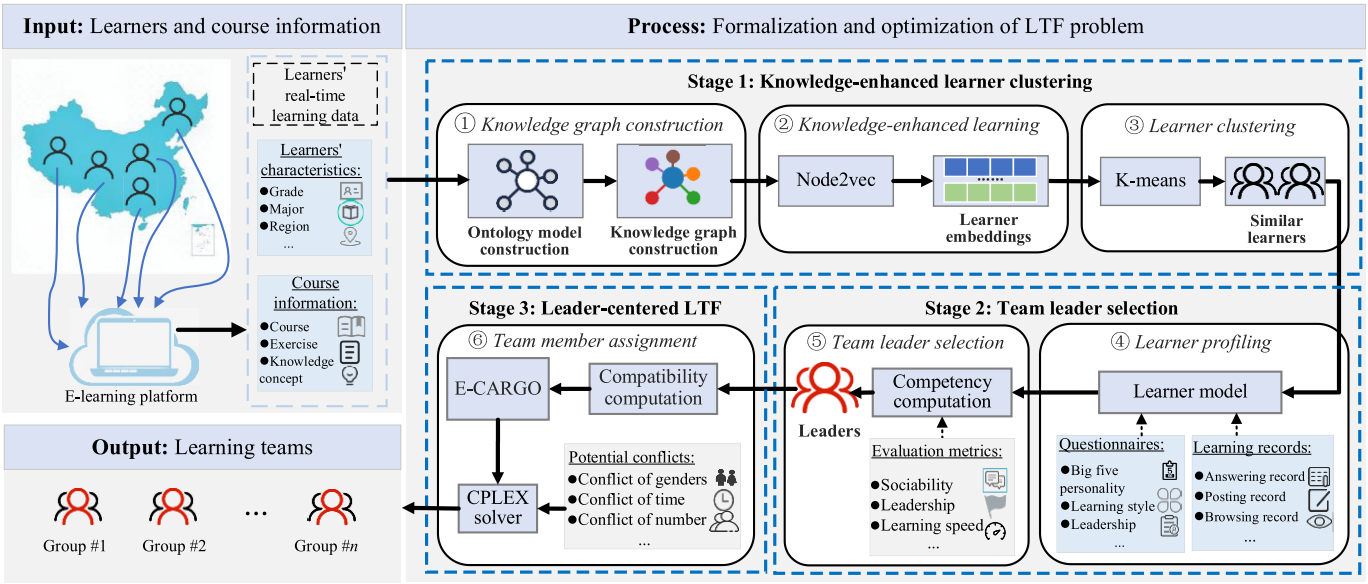

Fig. 2. Framework of TGo4LCE. (Input: The initial input data includes learners' personal characteristics, such as grade, major, together with course information, and learners' real-time learning data; Process: It outlines a three-stage approach to solving LTF problem, consisting of knowledge-enhanced learner clustering, team leader selection, and leader-centered LTF; Output: It shows learning teams that combine homogeneity for effective collaboration with heterogeneity in skills.)

IV. Grouping Optimization for Large-Scale Collaborative *E*-Learning

This article proposes a novel approach to three-stage grouping optimization for large-scale collaborative *e*-learning via knowledge graph and E-CARGO, denoted as TGo4LCE. The framework of TGo4LCE is shown in Fig. 2 and consists of the following three stages.

1) *Stage 1:* Knowledge-enhanced learner clustering. In this stage, potentially similar learners are identified by uncovering the inherent semantic relationships among entities in a large-scale *e*-learning environment. Meanwhile, a clustering algorithm is used to reduce the scale of the grouping problem and maintain similarity among members in the same team.
2) *Stage 2:* Team leader selection. In this stage, analyze learners' competency by modeling their profiles. Based on analysis results, appropriate leaders are selected to facilitate collaborative learning, thereby improving the overall learning outcomes of the learning team.
3) *Stage 3:* Leader-centered LTF. In this stage, the compatibilities between leaders and learners are calculated to assess the collaborative learning effects when assigning them to the same team. Then, the problem is formulated via the E-CARGO model and solved by the IBM CPLEX optimization package to obtain the optimal grouping solution.

The 3 stages are discussed in detail in Sections V–VII.

In real-world *e*-learning environments, learners' characteristics and behaviors are not static but evolve over time. These changes directly influence the construction of learner profiles. Generally, two types of situations may lead to such changes. First, as learners progress through courses, the accumulation

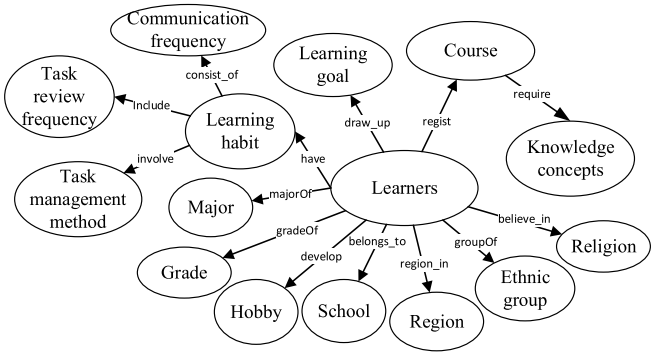

Fig. 3. Ontology model of large-scale *e*-learning environments.

of learning data can gradually alter their characteristics and behaviors. Second, unexpected events may occur, such as dropping out of a course or enrolling in a new one. When these happen, learner profiles should be updated according to real-time data. Furthermore, TGo4LCE needs to be reexecuted to obtain the latest grouping result.

V. Knowledge-Enhanced Learner Clustering

In this section, we detail the knowledge-enhanced learner clustering method for reducing problem complexity, which consists of three parts, i.e., knowledge graph construction, knowledge-enhanced learning, and learner clustering.

A. Knowledge Graph Construction

To capture the rich information in large-scale *e*-learning environments, key factors are identified for the ontology model, thereby constructing the knowledge graph, as shown in Fig. 3.

The ontology model consists of 15 types of entities and 14 types of entity relationships, mainly involving course information, learners' background information, and learning characteristics. In the model, hobbies are divided into nine main categories, i.e., sports, movies, reading, games, music, food, dance, history, and art. Learning goals refer to what the learner expects to achieve at the end of a course, including mainly understanding, familiarity, mastery, application, integration, and innovation. Communication frequency refers to how often learners interact when working with team members. Task review frequency refers to the frequency of reviews preferred by learners when working in teams. Task management method refers to the time management strategies employed by learners when completing group tasks, including structured planning and concentrated effort as the deadline approaches.

To construct the knowledge graph, it is essential to identify relevant entities, their categories, and the relationships between them in the *e*-learning environment. Simultaneously, to better facilitate knowledge extraction, it is necessary to transform raw, unstructured data into structured and interpretable data. Due to the limited information in publicly available datasets, this article utilizes simulated data to construct a knowledge graph.

Definition 1: *E*-learning knowledge graph. The knowledge graph can be denoted as $G = \{V, E, R\}$, where V denotes the set of nodes, E is the set of edges, and R is the set of relations. The relationship between two different nodes in G can be represented by the triplet (v_i, r_{ij}, v_j), indicating that node v_i and v_j have the relationship r_{ij}. For example, in Fig. 3, there is a relationship r_{ug} established between learner v_u and learning goal v_g, forming the triple (v_u, r_{ug}, v_g).

B. Knowledge-Enhanced Learning

To fully leverage the rich structural and semantic information within a knowledge graph for similar learner identification, the Node2vec algorithm based on a random walk [15] is applied to obtain the knowledge-enhanced learner embeddings. It involves two sampling strategies, i.e., breadth-first sampling (BFS) and depth-first sampling (DFS). Balancing the two sampling strategies helps account for both structural equivalence and homophily of the network, leading to higher quality embeddings.

To further clarify the process of learning node representations, the related parameters are given as follows: G denotes the constructed knowledge graph, W is the set of weights of the edges in the graph, d is the dimension of entity representations, *r_walks* indicates the quantity of random walk produced by each source vertex, l represents the length of each random walk, with a default value of 1, and *k_context* is the context size. Node2vec learns node embeddings via the three steps outlined below.

First, the transition probabilities from each node to its neighbors in G are calculated, and a transition probability matrix is then formed. Based on the transition probability matrix, G is updated, i.e., obtain G'. Now, let us take a simple example to illustrate the calculation of transition probability. As shown in Fig. 4, given a random walk that just moved along the edge e_{rs}, it is now located at node s. The next node is determined by calculating the transition probability. Suppose the next node is t, the probability of t being walked to is described in the following equation:

$$P(c_i = t | c_{i-1} = s) = \begin{cases} \dfrac{\pi_{st}}{Z}, & \text{if } (s,t) \in E \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where c_i denotes the ith node in the walk, Z is the normalizing constant, and π_{st} denotes the nonnormalized transition probability from node s to node t, i.e., $\pi_{st} = \alpha_{pq}(s,t) \times w_{st}$. Typically, the weight of the edge is 1, hence $\pi_{st} = \alpha_{pq}(s,t)$. $\alpha_{pq}(s,t)$ is obtained by

$$\alpha_{pq}(s,t) = \begin{cases} \dfrac{1}{p}, & \text{if } d_{st} = 0 \\ 1, & \text{if } d_{st} = 1 \\ \dfrac{1}{q}, & \text{if } d_{st} = 2 \end{cases} \tag{2}$$

where d_{st} denotes the shortest distance from node s to node t; p is a return parameter; q is an in-out parameter. p controls the likelihood of immediately revisiting a node during traversal, thereby minimizing the probability of sampling previously visited nodes in the next two steps. q could control the direction of the search, i.e., inward or outward.

Then, random walks will be generated starting from each node in the graph. The set of random walks is defined

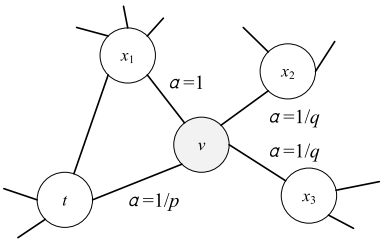

Fig. 4. Mock example of sampling in Novde2vec.

by *walks*. Initially, *walks* are null. Following this, *r_walks* iterations are executed to generate a walk. In each iteration, all nodes in the graph serve as a starting vertex for a random walk of length l. Eventually, the walk generated by the nodes is added to the *walks*.

Finally, stochastic gradient descent (SGD) is used as the optimizer to update the node embeddings based on the following objective:

$$\max_{f} = \sum_{u \in V} \log \Pr(Ns(u)|f(u)) \tag{3}$$

where $f \in \mathbb{R}^{|V| \times d}$, $f(u) \in \mathbb{R}^d$ denotes the uth row of the embedding matrix f, corresponding to learner u's representation, and $N_S(u)$ is the set of u's neighborhoods, created with the sampling strategy S. To improve the overall scalability of Node2vec, each stage is designed to support parallel and asynchronous execution.

Due to storing the interconnections among node neighbors, Node2vec has a space complexity of $O(\alpha^2|V|)$. Here, α is the graph's average degree, which tends to be very small in practical networks. By simulating a random walk of length l ($l >$ *k_context*), *k_context* samples can be generated for *l–k_context* nodes based on the Markov property of the random walk. The time complexity for generating each sample is $O = (l/k(l-k))$.

C. Learner Clustering

Referring to existing research [28], we assume that learners with high similarities tend to cooperate better and offer more support to each other when they're in the same team. Following this assumption, we leverage the K-means clustering algorithm to divide I learners in the dataset into h clusters based on the learner embeddings obtained by Node2vec. Each cluster obtained by K-means can be regarded as a set of learners for LTF, which effectively reduces the complexity of the LTF problem in a large-scale e-learning environment. Formally, the objective function of K-means and the cluster center μ_i are calculated by the following equation:

$$J(H, \mu_1, \ldots, \mu_h) = \sum_{i=1}^{h} \sum_{j=1}^{n_i} \left\| x_j - \mu_i \right\|_2^2 \tag{4}$$

$$\mu_i = \frac{1}{n_i} \sum_{j=1}^{n_i} x_j \tag{5}$$

where h is the number of clusters, n_i refers to the number of learners in the ith cluster, $x_j \in \mathbb{R}^d$ denotes the embedding of the jth learner, $\mu_i \in \mathbb{R}^d$ denotes the embedding of the ith cluster center, and $\|\cdot\|_2^2$ denotes the Euclidean distance between x_j and μ_i. u_{ij} indicates whether x_j belongs to the ith cluster. $u_{ij} = 1$ indicates that x_j belongs to the ith cluster; otherwise, $u_{ij} = 0$. $i \in \{1, \ldots, h\}$, u_{ij} is obtained by the following equation:

$$u_{ij} = \begin{cases} 1, & \text{if } i = \underset{h}{\operatorname{argmin}} \left\| x_j - \mu_h \right\|_2^2 \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

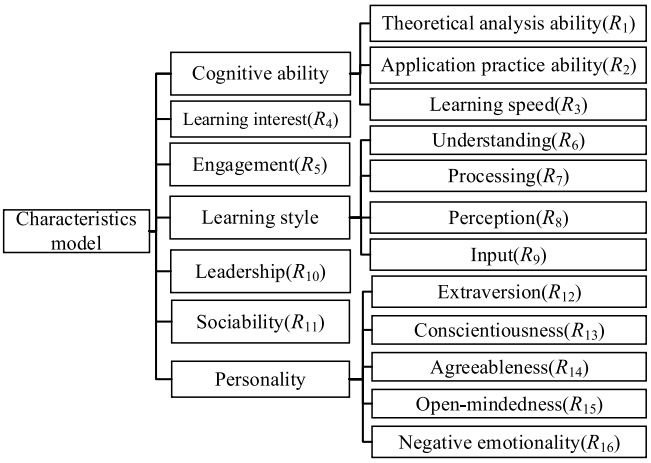

Fig. 5. Characteristics model of learners.

VI. Team Leader Selection

In this section, we will introduce the process of team leader selection to provide support for the following leader-centered LTF.

A. Learner Profiling via Multidimensional Features

To comprehensively profile learners, we extend [9] to model learners from seven aspects, including sixteen features. The overall learner model is shown in Fig. 5. Specifically, three new features, namely learning speed (LS), learning interest, and engagement, are added to improve learner modeling, and we will introduce their concepts and ways to evaluate them.

1) LS: It reflects the speed at which learners acquire new knowledge and is a common characteristic of learners. Here, the LS of learners in a single course is evaluated based on their performance in answering questions. In practical applications, the accuracy of learners' answers, the number of attempts per exercise, and the time spent on responses collectively indicate their LS. Therefore, the above factors are utilized to compute the LS. Additionally, (7) and (8) are applied to normalize the number of times learner u submits the answer to each question (attempt_u) and the total answering time for each question (attTime_u), respectively

$$\text{attempt}_{u\text{avg}} = \frac{\text{attempt}_u}{\text{attempt}_j} \tag{7}$$

$$\text{attTime}_{u\text{avg}} = \frac{\text{attTime}_u}{\text{attTime}_j} \tag{8}$$

where attempt_j corresponds to the average number of times for the learners to have answered exercise j, and attTime_j

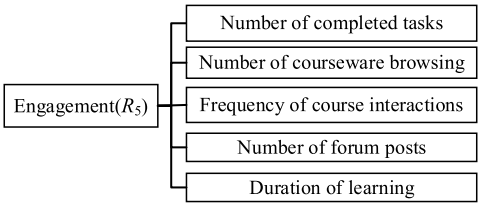

Fig. 6. Components of engagement.

represents the average answering time for all learners on exercise j.

Subsequently, $\text{attempt}_{u\text{avg}}$ and $\text{attTime}_{u\text{avg}}$ are adjusted to ensure their values are limited to the 0–1 range. The adjusted values of $\text{attempt}_{u\text{avg}}$ and $\text{attTime}_{u\text{avg}}$ are denoted as $\text{attempt}'_u$ and $\text{attTime}'_u$, respectively. The formulas for adjustment are given by the following equation:

$$\text{attempt}'_u = \max\left(0, \min\left(1 - \alpha \times \left(\text{attempt}_{u\text{avg}} - 1\right)\right)\right) \quad (9)$$

$$\text{attTime}'_u = \max\left(0, \min\left(1 - \beta \times \left(\text{attTime}_{u\text{avg}} - 1\right)\right)\right) \quad (10)$$

where α and β serve as the weights used to modify the effect of $\text{attempt}_{u\text{avg}}$ and $\text{attTime}_{u\text{avg}}$ on LS, respectively.

Based on this, along with u's score of x on the exercise, u's LS can be calculated by the following equation:

$$\text{LS} = \frac{1}{n}\sum_{i=1}^{n} w(x) \times \text{attempt}'_u \times \text{attTime}'_u \quad (11)$$

where n denotes the total number of exercises included in the course, and $w(x)$ refers to the weight function for answering exercises. It is clear that the LS differs based on whether the learner answers the exercise correctly or incorrectly. Specifically, the value of 1 represents that the learner answered the exercise correctly; otherwise, the value is 0. $w(x)$ is defined as follows:

$$w(x) = \begin{cases} 1, & \text{if } x = 1 \\ 0.5, & \text{if } x = 0. \end{cases} \quad (12)$$

2) Learning Interest: It reflects the learners' interest in the course content. Learners with a strong interest in their studies can not only improve their own learning outcomes but also motivate their team members, boosting the overall motivation of the team. Specifically, it is assessed based on the academic interest scale for adolescents [36].

3) Engagement: It reflects the learners' participation and involvement in the course, including, but not limited to, their completion of assignments and task engagement. In this article, the evaluation of engagement is carried out from five perspectives, i.e., number of completed tasks, number of courseware browses, frequency of course interactions, number of forum posts, and duration of learning, as illustrated in Fig. 6.

The number of completed tasks is defined as the ratio of the number of completed course tasks to the total number of tasks. The number of courseware browsed is calculated similarly to the number of completed tasks. The frequency of course interactions refers to how often learners engage with the course, reflecting whether they are actively learning it. The number of forum posts refers to the total count of various actions, such as posting, commenting, and other interactions, made by learners in the course forum. The duration of learning represents the time learners spend on the course. Similar to LS, these five metrics are first normalized, then weighted and summed, with the resulting value representing the engagement. The weights can be customized, and their sum should equal 1.

Other characteristics follow the same method as in [9]. After obtaining the values of learners' characteristics, the fuzzy values corresponding to all characteristics are calculated. We use interval numbers to represent the fuzzy values of learners' characteristics, and the interval number O is calculated by the following equation:

$$O = \left[O^{\text{L}}, O^{\text{H}}\right] \quad (13)$$

where O^{L} represents the lower bound of O, and O^{H} is the upper bound of O.

Take the learner u's LS (R_3) as an example. It is defined as follows: 1) when there is only one prerequisite course, and u has not taken it, $S_{u3} = 0$. Here, S_{u3} denotes u's evaluation value on R_3. If u has taken the prerequisite course, u has only one performance value s_u, $S_{u3} = [s_u, s_u]$; 2) when there are fewer than six prerequisite courses, and u has not taken any course, $S_{u3} = 0$. If u has taken some courses, $S_{u3} = [s_{u1}, s_{u2}]$, where s_{u1} and s_{u2} correspond to the minimum and maximum performance values, respectively. 3) when there are more than six prerequisite courses, and u has not taken any course, $S_{u3} = 0$. If u has taken some courses, $S_{u3} = [s_{u1}, s_{u2}]$. If u has completed all courses, the reverse cloud generator [30] is employed to compute three numerical features of S_{u3}, i.e., expectation E_u^x, entropy E_u^n, and hyperentropy H_u^E. Learners usually have different LSs when learning different courses. Let $s_{u1}^{R_3}$ denote the LS of learner u in prerequisite course #1, and let $T_{R_3} = [s_{u1}^{R_3}, s_{u2}^{R_3}, \ldots, s_{uD}^{R_3}]$ represent the set of LS values of learner u across all prerequisite courses, where D denotes the number of prerequisite courses. The three numerical features of S_{u3} are calculated by the following equations:

$$\begin{cases} E_u^x = \dfrac{1}{D} \times \displaystyle\sum_{d=1}^{D} s_{ud}^{R_3} \\ E_u^n = \sqrt{\dfrac{\pi}{2}} \times \dfrac{1}{D} \displaystyle\sum_{d=1}^{D} \left| s_{ud}^{R_3} - E_u^x \right| \\ H_u^E = \sqrt{\left| \dfrac{1}{D-1} \displaystyle\sum_{d=1}^{D} \left(s_{ud}^{R_3} - E_u^x\right)^2 - \left(E_u^n\right)^2 \right|} \end{cases} \quad (14)$$

where $s_{ud}^{R_3}$ indicates u's LS in course d. E_u^x, E_u^n, and H_u^E, respectively, represent the expectation, entropy, and hyperentropy of T_{R3}, corresponding to the most representative value, first-order uncertainty, and second-order uncertainty of the qualitative concept. Based on the E_u^x, E_u^n, and H_u^E, the upper bound S_{u3}^{H} and the lower bound S_{u3}^{L} of S_{u3} are obtained by the following equations:

$$\begin{cases} S_{u3}^{\text{H}} = E_u^x + E_u^n + H_u^E \times \gamma \\ S_{u3}^{\text{L}} = E_u^x - E_u^n - H_u^E \times \gamma \end{cases} \quad (15)$$

where γ represents the influence coefficient related to H_u^E, and its value lies within the range of 0.1–0.2 [37].

B. Team Leader Selection via Competence Evaluation

After modeling the learners, we select one appropriate team leader for each learning team based on the score in various evaluation metrics. These leaders can guide team members to improve collaboration efficiency, thereby enhancing the overall learning performance of the team. We employ nine key indicators in the process of team leader selection, i.e., theoretical analysis ability, application practice ability, LS, learning interest, engagement, leadership, sociability, conscientiousness, and negative emotionality.

First, since the leader's primary responsibility is to assist group members in their learning, leaders excelling in the first four indicators are more capable of offering effective assistance. For example, a leader with a strong interest in learning actively organizes study activities and helps team members understand the course content. Therefore, the team leader's competency of a learner is determined by the weighted sum of their evaluation scores on R_1–R_4. Second, the team leader with a high level of course engagement could contribute to guiding team members to complete the designated tasks more effectively. Accordingly, it is necessary to establish a minimum threshold for the evaluation of a team leader's engagement. Any value falling below this threshold will be revised to 0. Third, leadership, sociability, and conscientiousness are crucial for team leaders to positively influence the team and keep group activities on track. Thus, each feature requires its own threshold, and values below it are reset to 0. Fourth, a team leader's emotional instability can affect group harmony and work efficiency. Thus, the negative emotionality values above the threshold are corrected to 0.

To select suitable team leaders, the possibility degree of interval numbers is employed to assess learners' team leader competency. The formula is given by the following equations:

$$P(O_1 \geq O_2) = \frac{\min\{l_1 + l_2, \max\{o_1^{\mathrm{H}} - o_2^{\mathrm{L}}, 0\}\}}{l_1 + l_2} \tag{16}$$

where $P(O_1 \geq O_2)$ is the probability of $O_1 \geq O_2$, $O_1 = [o_1^{\mathrm{L}}, o_1^{\mathrm{H}}]$, $O_2 = [o_2^{\mathrm{L}}, o_2^{\mathrm{H}}]$, $l_1 = o_1^{\mathrm{H}} - o_1^{\mathrm{L}}$, and $l_2 = o_2^{\mathrm{H}} - o_2^{\mathrm{L}}$.

Then, u's team leader competency is computed by the following equations:

$$\begin{cases} Y_u = \mathrm{sgn}(2^{\alpha_u} - 0.5) \times \sum_{j=1}^{4} w_j^G \times P\left(S_{uj} \geq O_3\right) \\ \alpha_u = \min\left(\mathrm{sgn}\left(S_{u5}^{\mathrm{L}} - f_1\right), \mathrm{sgn}\left(S_{u10}^{\mathrm{L}} - f_2\right), \mathrm{sgn}\left(S_{u11}^{\mathrm{L}} - f_3\right), \right. \\ \qquad \left. \mathrm{sgn}\left(S_{u13}^{\mathrm{L}} - f_4\right), \mathrm{sgn}\left(f_5 - S_{u16}^{\mathrm{H}}\right)\right). \end{cases} \tag{17}$$

In (17), Y_u denotes the competency value of learner u as a team leader; α_u shows whether the learner meets the thresholds; the thresholds are comprehensively suggested as $f_1 = 0.45$, $f_2 = 0.5$, $f_3 = 0.5$, $f_4 = 0.6$, and $f_5 = 0.7$; S_{u5}^{L}, S_{u10}^{L}, S_{u11}^{L}, and S_{u13}^{L} represent the lower bounds of the interval numbers for u's engagement (i.e., R_5), leadership (i.e., R_{10}), sociability (i.e., R_{11}), and conscientiousness (i.e., R_{13}), respectively. S_{u16}^{H} denotes the upper bound of the interval numbers for u's negative emotionality (i.e., R_{16}); O_3 stands for an interval number that ranges from 0 to 1. $\mathrm{sgn}(y)$ is defined by the following equations:

$$\mathrm{sgn}(y) = \begin{cases} 1, & y > 0 \\ 0, & y = 0 \\ -1, & y < 0. \end{cases} \tag{18}$$

In (17), w_j^G is the weight of the jth characteristic (i.e., R_j). The value of w_j^G changes with the application scenario. Taking the computer science major as an example, application scenarios can be categorized into the following three scenarios.

1) *Scenario #1:* In this scenario, courses have longer lecture time than lab time, for instance, algorithm design and analysis. In view of this, w_1^G, w_2^G, w_3^G, and w_4^G are recommended to take values of 0.4, 0.2, 0.2, and 0.2.
2) *Scenario #2:* In this scenario, courses have longer lab time than lecture time, for instance, software development practice. For this course, w_1^G, w_2^G, w_3^G, and w_4^G are recommended to take values of 0.2, 0.4, 0.2, and 0.2.
3) *Scenario #3:* In this scenario, courses with equal teaching and lab hours are considered comprehensive courses. In view of this, w_1^G, w_2^G, w_3^G, and w_4^G are recommended to take values of 0.3, 0.3, 0.2, and 0.2.

VII. Leader-Centered LTF

Based on the selected team leaders, the LTF problem can be solved by following the paradigm of leader-centered LTF [9]. This problem aligns with the RBC problem, which inspires us to utilize E-CARGO to formalize it. The detailed process of leader-centered LTF includes complementarity evaluation, problem formalization via E-CARGO, and problem solving via CPLEX.

A. Complementarity Evaluation

To achieve efficient teamwork and improve learning effectiveness, it is crucial to maximize the compatibility between a team member and the leader.

The comprehensive complementarity (CQ_{uEk}) between learner u and leader (E_k) is calculated by the following equations:

$$CQ_{uE_k} = \sum_{j=1}^{16} w_j \times \left(\left|S_{uj}^{\mathrm{L}} - S_{E_kj}^{\mathrm{L}}\right| + \left|S_{uj}^{\mathrm{H}} - S_{E_kj}^{\mathrm{H}}\right|\right) \times 0.5 \tag{19}$$

where S_{uj}^{L} denotes the lower bound of the fuzzy value of learner u's jth characteristic, and other symbols are defined similarly. w_j representing the importance of R_j (i.e., jth characteristic) in comprehensive complementarity, is computed using the fuzzy analytic hierarchy process (FAHP). FAHP can not only minimize the influence of subjective factors but also prevent the problems of large computation and low precision [37].

B. Problem Formalization via E-CARGO

Similar to [9], we formalize the leader-centered LTF problem through E-CARGO with the consideration of potential constraints. The E-CARGO model is described as follows:

$$\sum ::= \left(\mathcal{C}, \mathcal{O}, \mathcal{A}, \mathcal{M}, \mathcal{R}, \mathcal{E}, \mathcal{G}, s_0, \mathcal{H}\right) \tag{20}$$

where $\mathcal{C}$ is the set of classes, which represents abstract concepts; $\mathcal{O}$ is the set of concrete objects associated with $\mathcal{C}$; $\mathcal{A}$ is a set of agents, and a learner is an agent; $\mathcal{M}$ is a set of messages; $\mathcal{R}$ is a set of roles, and a role corresponds to a learning team consisting of some learners; $\mathcal{E}$ is a set of problem environments involving multiple learning teams and learners; $\mathcal{G}$ is a set of learners assigned to learning teams, and $\mathcal{G}_j$ is a learner group assigned to the jth learning team; s_0 is the initial state of the problem; and $\mathcal{H}$ is a set of users.

Based on the above definitions, the objective function of LTF (σ) is defined as follows:

$$\max \sigma = \sum_{u=1}^{I} \sum_{k=1}^{J} CQ_{uE_k} \times T_{uk} \tag{21}$$

where I and J represent the number of learners and learning teams, respectively. E_k is the leader of the kth learning team, T_{uk} represents whether learner u is assigned to team k, subject to the following constraints:

$$T_{E_k k} = 1, \quad 1 \le k \le J \tag{22}$$

$$\sum_{k=1}^{J} T_{uk} = 1, \quad 1 \le u \le I \tag{23}$$

$$B_1 \le \sum_{u=1}^{I} T_{uk} \le B_2, \quad 1 \le k \le J \tag{24}$$

$$M_{ua} \times T_{uk} \times T_{ak} \le 0, \quad 1 \le u \le I, \quad 1 \le a \le I, \quad 1 \le k \le J \tag{25}$$

$$C_{uv} = \begin{cases} 1, & \text{if } \sum_{n=1}^{N} \sum_{q=1}^{Q} L_{unq} \times L_{vnq} \le 1, \ T_{uk} = 1, \ T_{vk} = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

Equation (22) requires that every learning team has just one leader. Equation (23) indicates that each learner can be a member of only one learning team. Equation (24) limits the size of each learning team, with $B_1 = [I/J]$ as the minimum number of members and $B_2 = B_1 + 1$ as the maximum. Equation (25) indicates that two learners who have subjective conflicts cannot be assigned to the same team. $M_{ua} = 0$ means that learner u is willing to join the same learning team with learner a; otherwise, they are unwilling. $C_{uv} = 1$ means that there is a conflict related to learning time between learners u and v; otherwise, there is none. N designates the duration of group tasks, which ranges from 1 to 7 days. Specifically, N is defined as a week. Q defines how a day is divided into Q individual time blocks. L_{unq} indicates learner u's availability during the qth time slot on the day n. $L_{unq} = 1$ means available; otherwise, unavailable. Equation (26) reflects that learners who have insufficient overlapping available time (less than two days) should not be assigned in the same team.

In addition, the design of the gender conflict formulas depends on the number of male and female learners. When the number of females is insufficient, (27) is used to handle the conflict. Similarly, if the number of males is insufficient, (28) applies. In cases where both genders are adequately represented, (29) is employed to model the conflict

$$1 \ge \sum_{u=1}^{I} T_{uk} \times G_u, \quad 1 \le k \le J \tag{27}$$

$$1 \ge \sum_{u=1}^{I} T_{uk} \times (1 - G_u), \quad 1 \le k \le J \tag{28}$$

$$\begin{cases} 1 \le \sum_{u=1}^{I} T_{uk} \times G_u, & 1 \le k \le J \\ 1 \le \sum_{u=1}^{I} T_{uk} \times (1 - G_u), & 1 \le k \le J \end{cases} \tag{29}$$

where $G_u = 0$ means that learner u is male, and $G_u = 1$ means that u is female.

C. Problem Solving via IBM CPLEX

Based on the above formalization, we can exploit the IBM CPLEX optimization package to solve the leader-centered LTF problem by the following steps.

Step 1: Set the core data structures for interfacing with CPLEX. The input parameters, such as the objective function coefficients, constraint coefficients, and bounds, must be specified. The linear programming problem in CPLEX is constructed with CQ, B, and T. In this context, CQ represents the objective function coefficients, i.e., the comprehensive compatibility between the leader and the members. T is a variable that determines whether a learner is assigned to a team. B is the constraint on the number of team members, with both upper and lower limits.

Step 2: Establish the objective function and constraint expressions. Initially, define the decision variable T: mdl = docplex.mp.model.Model(), T_vars = {(i, j): mdl.binary_var[name = "T_{0}_{1}."format(u, k)] for u in range(0, I) for k in range(0, J)}. Subsequently, add the constraint expressions iteratively. Finally, introduce the optimization objective and use the following method: objective = mdl.sum[T_vars[u, k] ∗ CQ[u, k] for u in range(0, I) for k in range(0, J)], mdl.maximize(objective).

Step 3: Execute the mdl.solve() method of CPLEX. Based on the objective function and constraint expressions, calculate the maximized σ value to derive the final grouping solution.

VIII. Simulation Experiments

To verify the effectiveness of TGo4LCE, simulation data are employed for experimental analysis, and experiments are conducted to answer the following research questions (RQs):

1) *RQ1:* What is an appropriate cluster size for achieving the best performance in forming learning teams?
2) *RQ2:* Is the clustering performance of the K-means algorithm optimal during the first-stage optimization?
3) *RQ3:* How about the performance of TGo4LCE in terms of objective function values compared with existing research?
4) *RQ4:* How about the performance of TGo4LCE in terms of execution efficiency compared with existing research?

TABLE III
COMPARISON OF CLUSTERING PERFORMANCE WITH DIFFERENT k_{mcs}

(a)

Max cluster size	Metrics		
	SC index	CH index	Execution time
100	0.0619	16.6072	0.58
200	0.0568(8.24% ↓)	21.7320(30.86% ↑)	0.37(36.21% ↑)
300	0.0489(21.00% ↓)	22.5052(35.52% ↑)	0.42(27.59% ↑)
400	0.0567(8.40% ↓)	21.7469(30.95% ↑)	0.35(39.66% ↑)
500	0.0506(18.26% ↓)	22.5597(35.84% ↑)	0.36(37.93% ↑)

(b)

Max cluster size	Metrics		
	SC index	CH index	Execution time
100	0.0123	16.4117	2.12
200	0.0523(325.20% ↑)	27.1502(65.43% ↑)	1.08(49.05% ↑)
300	0.0486(295.12% ↑)	36.7418(123.88% ↑)	1.03(51.42% ↑)
400	0.0482(291.87% ↑)	36.7833(124.13% ↑)	0.81(61.79% ↑)
500	0.0502(308.13% ↑)	38.1033(132.17% ↑)	1.23(41.98% ↑)

TABLE IV
COMPARISON OF SOLUTION PERFORMANCE WITH DIFFERENT k_{mcs}

(a)

Max cluster size	Metrics	
	Objective function value	Execution time
100	237.69	4.10
200	245.27(3.19% ↑)	13.22(222.44% ↓)
300	245.74(3.39% ↑)	13.97(240.73% ↓)
400	245.22(3.17% ↑)	15.19(270.49% ↓)
500	246.51(3.71% ↑)	17.03(315.37% ↓)

(b)

Max cluster size	Metrics	
	Objective function value	Execution time
100	469.15	8.53
200	486.68(3.74% ↑)	23.98(181.13% ↓)
300	499.77(6.53% ↑)	89.56(949.94% ↓)
400	499.30(6.43% ↑)	90.56(961.66% ↓)
500	501.58(6.91% ↑)	132.09(1448.53% ↓)

5) *RQ5:* Does TGo4LCE ensure the homogeneity among learners, where the intragroup similarity is not zero?

All experiments are implemented on a 64-bit Windows 10 operating system with an Intel Core i5-13500H CPU and 16 GB of memory.

A. Experiment Setup

When training the embedded model of the knowledge graph, the relevant hyperparameters are listed as follows: the number of walks is set as 10, the length of walks is set as 80, the number of iterations is set as 10, and the dimension of entity representations d is set as 64.

B. Analysis of Maximum Cluster Size (RQ1)

To investigate the appropriate maximum cluster size k_{mcs}, we compare TGo4LCE with several variants using sizes ranging from 100 to 500. The Silhouette coefficient (SC), Calinski–Harabasz (CH) index, objective function value, and execution time are used as evaluation metrics. Higher SC, CH, and objective function values indicate better performance, while lower execution time reflects higher efficiency. In the tables, an upward arrow (↑) denotes improvement, whereas a downward arrow (↓) indicates degradation. Taking 1000 and 2000 learners as an example, the experimental results are presented in Tables III and IV.

As shown in Table IV, the objective function value increases with the maximum cluster size, but the execution time also grows accordingly. Although the setting of 100 yields favorable results in terms of efficiency, Table III indicates that its clustering performance is not satisfactory. For a balance between clustering quality, objective function value, and execution time, the maximum cluster size is comprehensively suggested to be set to 200 in the following experiments.

C. Analysis of Knowledge-Enhanced Learner Clustering (RQ2)

We compare the clustering effectiveness of K-means, spectral clustering, and BIRCH [38]. Moreover, C-LO, D-LO, and Min-D-LO, proposed by [39], are also used for comparison. The K-means algorithm is widely used in existing research. After eigenvector decomposition, spectral clustering could easily identify natural clustering structures in a low-dimensional space. The BIRCH algorithm can achieve high-quality clustering of large datasets with limited memory resources. The methods proposed in [39] address the critical issue that the standard K-means algorithm may converge to a solution that is not locally optimal. Learner clustering will be performed using six methods, denoted as *LC*_Kmeans, *LC*_Spectral, *LC*_Birch, *LC*_C_LO, *LC*_D_LO, and *LC*_Min_D_LO, respectively.

SC and CH indices are selected as metrics. The SC score lies within the range of $[-1, 1]$, with higher values indicating better cohesion and separation of the clusters. The CH score ranges from 0 to infinity, and it evaluates clustering performance by comparing the dispersion within and between clusters. Meanwhile, execution time is selected as an evaluation metric. Fig. 7 shows the clustering results for 1000 and 2000 learners, with the results being the best from ten repeated experiments.

Since the number of learners in small and medium-sized courses is typically under 200, and based on the results in Section VIII-B, the cluster size is restricted to 200. Taking 1000 learners as an example, the initial number of clusters is set between 5 and 10, inclusive. Since some clusters in the initial division contain more than 200 learners, we perform secondary clustering on those that exceed 200 learners. In Fig. 7, Initial_K is the original number of clusters. *LC*_Kmeans_K, *LC*_Birch_K, *LC*_Spectral_K, *LC*_C_LO_K, *LC*_D_LO_K, and *LC*_Min_D_LO_K represent the number of clusters after secondary clustering using the K-means, Birch, spectral, C_LO, D_LO, and Min_D_LO clustering, respectively.

In the experiment, the same algorithm is employed for both the initial and secondary partitions. As shown in Fig. 7, the *LC*_Kmeans outperforms *LC*_Spectral and *LC*_Birch in both the CH and SC indices. Although *LC*_Kmeans shows slight numerical differences compared with *LC*_D_LO, *LC*_C_LO, and *LC*_Min_D_LO in terms of SC and CH, it achieves better clustering performance within a shorter execution time. Thus, the K-means is chosen as the clustering method in this article.

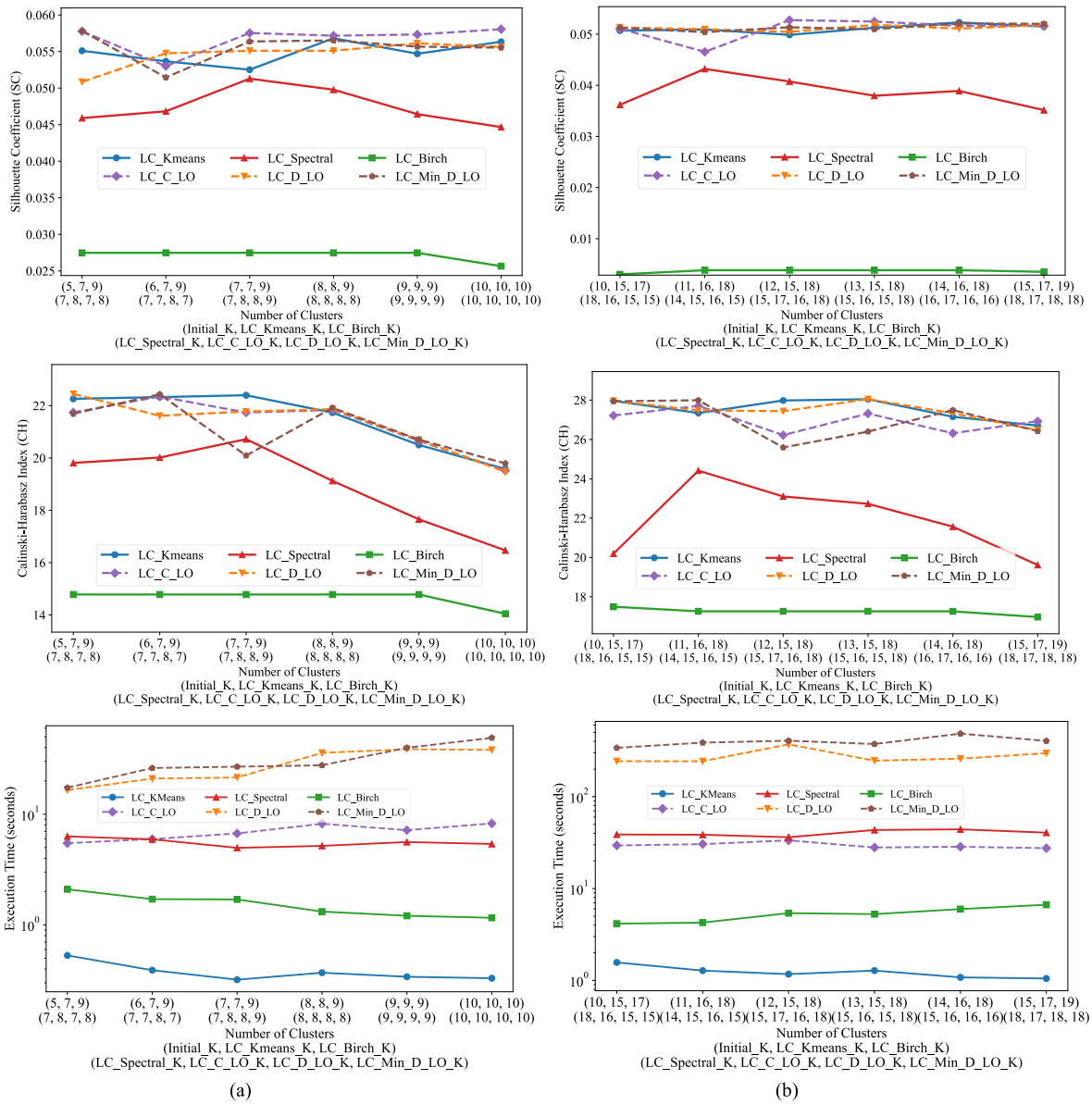

Fig. 7. Comparison of clustering performance. (a) 1000 Learners. (b) 2000 Learners.

D. Analysis of Objective Function Values (RQ3)

Particle swarm optimization algorithm [27], GA [5], and DCA [6] have been applied to the LTF problem. To evaluate the performance of TGo4LCE, we compare it with PSO-based LTF(LTF_PSOA), GA-based LTF(LTF_GA), and DCA-based LTF(LTF_DCA) in terms of the objective function value.

Taking Scenario #1 as an example, the experiment is conducted with the number of teams $J = \lfloor I/5 \rfloor$, where the number of learners I ranges from 1000 to 10 000 in increments of 1000. The learners' subjective conflict probability is set to 0.1, and the probability of a learner being female is 0.5. The experiment is repeated 30 times, and the results are shown in Table V. Here, N/A indicates that the optimal solution is not available in one day.

From Table V, it is evident that TGo4LCE achieves higher objective function values than LTF_PSOA, LTF_GA, and LTF_DCA. Moreover, as the number of learners increases, the gap between the obtained objective function values becomes larger. This shows that TGo4LCE consistently outperforms LTF_PSOA, LTF_GA, and LTF_DCA as the number of learners increases.

E. Analysis of Execution Time (RQ4)

Similar to the experimental settings in Section VIII-D, Scenario #1 is used as an example. The experiment is conducted 30 times to evaluate the execution time of TGo4LCE for grouping 1000 to 10 000 learners. The results are shown in Fig. 8. The x-axis denotes the number of learners, and the y-axis represents the execution time of TGo4LCE (in seconds). Due to execution times exceeding 30 min for grouping 1000 learners, both LTF_PSOA, LTF_GA, and LTF_DCA were excluded from execution time comparisons.

From Fig. 8, the execution time of TGo4LCE increases progressively with the number of learners but remains within an

TABLE V
COMPARISON OF OBJECTIVE FUNCTION VALUES

Methods	I									
	1000	2000	3000	4000	5000	6000	7000	8000	9000	10000
LTF_PSOA	199.07	397.44	599.02	807.26	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
LTF_GA	206.32	414.62	620.10	842.02	1024.80	1262.24	1460.30	1665.34	1871.31	2080.52
LTF_DCA	214.07	419.39	635.54	857.25	1055.30	*N/A*	*N/A*	*N/A*	*N/A*	*N/A*
TGo4LCE	**244.89**	**487.32**	**737.99**	**980.61**	**1227.24**	**1470.10**	**1722.88**	**1968.72**	**2220.72**	**2454.74**

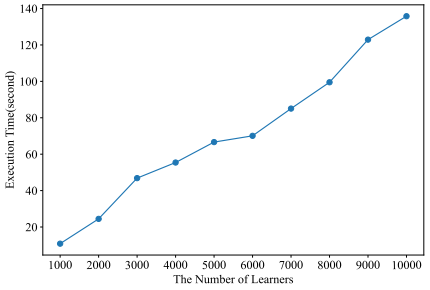

Fig. 8. Analysis of execution time.

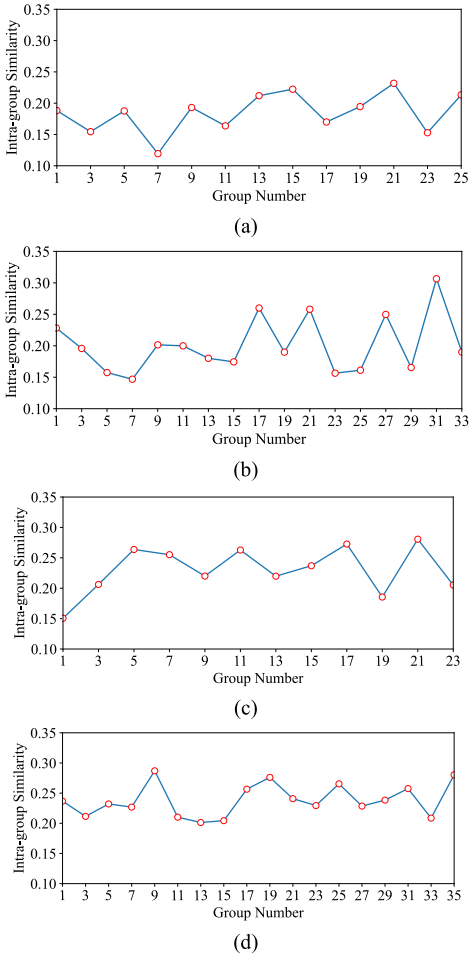

Fig. 9. Analysis of intragroup similarity. (a) 1000 Learners. (b) 4000 Learners. (c) 7000 Learners. (d) 9000 Learners.

acceptable limit. TGo4LCE requires only about two minutes to group 10 000 learners. It demonstrates TGo4LCE's efficiency in providing solutions for large-scale LTF problems.

F. Analysis of Intragroup Similarity (RQ5)

Referring to [28], we evaluate the similarity among group members using intragroup similarity by (30), which is calculated by averaging the cosine similarity scores of all learner pairs within the team

$$Gsim = \frac{2}{F(F-1)} \times \sum_{u=1}^{F-1} \sum_{v=u+1}^{F} \cos\left(e^u, e^v\right) \tag{30}$$

where F denotes the number of learners in the learning team, e_u and e_v are representations of learners u and v, respectively.

As shown in Fig. 9, the intragroup similarity of certain groups is presented for learner numbers of 1000, 4000, 7000, and 9000. The x-axis denotes group number, and the y-axis represents intragroup similarity.

From Fig. 9, the intragroup similarity of the groups produced by TGo4LCE is around 0.1–0.3, demonstrating that TGo4LCE guarantees a certain level of homogeneity among learners. Thus, group members can collaborate more effectively, which improves e-learning outcomes and alleviates learning loneliness.

IX. CONCLUSION

In large-scale e-learning environments, to better achieve LTF, we propose an approach to three-stage grouping optimization via a knowledge graph and E-CARGO. First, we design an ontology model to construct a knowledge graph by analyzing learners' characteristics. Then, a graph embedded model is trained via the Node2vec algorithm to effectively preserve both the structural and semantic information of the graph and improve the accuracy of entity representations. In addition, to ensure homogeneity among learners and reduce the complexity of the grouping problem, the K-means algorithm is employed to cluster learners based on their learned representations. Second, to comprehensively model learners, a multidimensional learner model is constructed. Based on the learner model, leaders are selected according to multiple evaluation metrics. Third, to evaluate the collaboration effectiveness, the comprehensive compatibility between the leaders and the remaining learners is calculated. Afterward, to obtain optimal grouping, the grouping problem is modeled based on the E-CARGO model and solved via an optimization package. Finally, experiments demonstrate that the proposed approach can accurately achieve global grouping optimization in large-scale collaborative e-learning scenarios and exhibit excellent execution performance.

However, the proposed approach still has considerable room for improvement in the future.

1) The real-world data from *e*-learning platforms will be collected to validate the effectiveness of the proposed approach.
2) Aiming at the individual differences and diverse learning needs among learners, we will incorporate considerations of educational equity into the grouping mechanism to enhance both the effectiveness and fairness of collaborative learning.
3) In real-world learning scenarios, learners' characteristics and behaviors change over time during the course. Therefore, to better meet real demands, we will explore a dynamic learner profiling approach that adapts to changes in learners' abilities and characteristics.
4) The dynamic formation strategies need to be explored for supporting LTF at different learning stages.

REFERENCES

[1] H. Ma et al., "Personalized early warning of learning performance for college students: A multilevel approach via cognitive ability and learning state modeling," *IEEE Trans. Learn. Technol.*, vol. 17, pp. 1414–1427, Mar. 2024.

[2] H. Ma, Z. Huang, W. Tang, H. Zhu, H. Zhang, and J. Li, "Predicting student performance in future exams via neutrosophic cognitive diagnosis in personalized *e*-learning environment," *IEEE Trans. Learn. Technol.*, vol. 16, no. 5, pp. 680–693, Oct. 2023.

[3] A. Aranzabal, E. Epelde, and M. Artetxe, "Team formation on the basis of Belbin's roles to enhance students' performance in project based learning," *Educ. Chem. Engineers*, vol. 38, pp. 22–37, Jan. 2022.

[4] V. Sanchez-Anguix, J. M. Alberola, E. Del Val, A. Palomares, and M. D. Teruel, "Comparing computational algorithms for team formation in the classroom: A classroom experience," *Int. J. Speech Technol.*, vol. 53, no. 20, pp. 23883–23904, Jul. 2023.

[5] X. Li, F. Ouyang, and W. Chen, "Examining the effect of a genetic algorithm-enabled grouping method on collaborative performances, processes, and perceptions," *J. Comput. Higher Educ.*, vol. 34, no. 3, pp. 790–819, May 2022.

[6] V. Yannibelli and A. Amandi, "A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 8584–8592, Aug. 2012.

[7] Kanika, S. Chakraverty, P. Chakraborty, and M. Madan, "Effect of different grouping arrangements on students' achievement and experience in collaborative learning environment," *Interact. Learn. Environments*, vol. 31, no. 10, pp. 6366–6378, Dec. 2023.

[8] I. M. M. Ramos, D. B. Ramos, B. F. Gadelha, and E. H. T. de Oliveira, "An approach to group formation in collaborative learning using learning paths in learning management systems," *IEEE Trans. Learn. Technol.*, vol. 14, no. 5, pp. 555–567, Oct. 2021.

[9] H. Ma, J. Li, H. Zhu, W. Tang, Z. Huang, and Y. Tang, "Collaborative optimization of learning team formation based on multidimensional characteristics and constraints modeling: A team leader-centered approach via E-CARGO," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 1, pp. 184–196, Feb. 2024.

[10] O. R. Sánchez, C. A. Collazos Ordóñez, M. Á. Redondo Duque, and I. Ibert Bittencourt Santana Pinto, "Homogeneous group formation in collaborative learning scenarios: An approach based on personality traits and genetic algorithms," *IEEE Trans. Learn. Technol.*, vol. 14, no. 4, pp. 486–499, Aug. 2021.

[11] A. Krouska and M. Virvou, "An enhanced genetic algorithm for heterogeneous group formation based on multi-characteristics in social-networking-based learning," *IEEE Trans. Learn. Technol.*, vol. 13, no. 3, pp. 465–476, Jul. 2020.

[12] N. Gavrilovic, T. Šibalija, and D. Domazet, "Design and implementation of discrete jaya and discrete PSO algorithms for automatic collaborative learning group composition in an *e*-learning system," *Appl. Soft Comput.*, vol. 129, Nov. 2022, Art. no. 109611.

[13] T. Qi, M. Ren, J. Zhao, and l. Guo, "Collaborative learning grouping method based on clustering," *Comput. Technol. Develop.*, vol. 33, no. 6, pp. 189–193, Jun. 2023.

[14] S. Kohli, N. Ramachandran, A. Tudor, G. Tumushabe, O. Hsu, and G. Ranade, "Inclusive study group formation at scale," in *Proc. 54th ACM Tech. Symp. Comput. Sci. Educ.*, Mar. 2023, pp. 11–17.

[15] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 855–864.

[16] H. Zhu and M. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Trans. Syst., Man, Cybern., C*, vol. 36, no. 4, pp. 578–589, Jul. 2006.

[17] H. Zhu, *E-CARGO and Role-Based Collaboration: Modeling and Solving Problems in the Complex World*. Hoboken, NJ, USA: Wiley, 2021.

[18] H. Zhu, D. Liu, H. Ma, Y. Sheng, L. Zhang, and Q. Jiang, "E-CARGO/RBC research guide: A road map for researchers," *IEEE Syst., Man, Cybern., Mag.*, vol. 10, no. 3, pp. 64–75, Jul. 2024.

[19] A. Mueller, J. Konert, R. Röpke, Ö. Genc, and H. Bellhäuser, "Group formation based on extraversion and prior knowledge: A randomized controlled study in higher education online," *J. Comput. Higher Educ.*, vol. 37, no. 3, pp. 780–808, Aug. 2024.

[20] M. R. D. Ullmann, D. F. James, C. G. Camilo-Júnior, and T. D. C. Nogueira, "Group composition for collaborative learning with distributed leadership in MOOCs using particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 2374–2383.

[21] X. Sun, X. Zhao, B. Li, Y. Ma, R. Sutcliffe, and J. Feng, "Dynamic key-value memory networks with rich features for knowledge tracing," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8239–8245, Aug. 2022.

[22] M. Kalantzi, A. Polyzou, and G. Karypis, "FERN: Fair team formation for mutually beneficial collaborative learning," *IEEE Trans. Learn. Technol.*, vol. 15, no. 6, pp. 757–770, Dec. 2022.

[23] Y. Chen, L. Zhang, Y. Ding, L. Guo, and K. Bian, "Multi-task oriented team formation in online collaborative learning," *Expert Syst. Appl.*, vol. 259, Jan. 2024, Art. no. 125289.

[24] P. Singh, P. K. Huynh, D. L. T. Nguyen, T. Le, and W. Moreno, "Leveraging multi-criteria integer programming optimization for effective team formation," *IEEE Trans. Learn. Technol.*, pp. 72–84, May 2023.

[25] D. B. Silva, D. R. Carvalho, and C. N. Silla, "A clustering-based computational model to group students with similar programming skills from automatic source code analysis using novel features," *IEEE Trans. Learn. Technol.*, vol. 17, pp. 428–444, May 2024.

[26] L. Zhang, C. Li, T. Li, and Z. Lu, "Forming a robust team in educational scenarios using genetic algorithm with partial repair operators," *Educ. Inf. Technol.*, vol. 30, no. 9, pp. 11523–11548, Jan. 2025.

[27] D. Qu et al., "A competition-oriented student team building method," *IEEE Trans. Learn. Technol.*, vol. 17, pp. 1966–1979, Dec. 2024.

[28] H. Liao, Z. Qu, W. Zhao, and J. Chen, "A context-flow-driven dynamic grouping method for large-scale online learning," *Modern Educ. Technol.*, vol. 33, no. 3, pp. 118–126, Mar. 2023.

[29] S. Ma, Y. Luo, and Y. Yang, "Personas-based student grouping using reinforcement learning and linear programming," *Knowl.-Based Syst.*, vol. 281, Dec. 2023, Art. no. 111071.

[30] H. Ma, C. Xiong, X. Fu, H. Zhu, Y. Tang, and K. Li, "Collaborative recommendation of national image resources for targeted international communication via multidimensional features and E-CARGO modeling," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 55, no. 2, pp. 1549–1563, Feb. 2025.

[31] H. Zhang, Z. Huang, Z. Jiang, H. Ma, and H. Zhu, "Collaborative route planning of road trips in regional central cities of China: An approach based on E-CARGO model," *IEEE Trans. Computat. Social Syst.*, vol. 11, no. 5, pp. 6347–6365, Oct. 2024.

[32] H. Ma, H. Zhu, K. Li, and W. Tang, "Collaborative optimization of service composition for data-intensive applications in a hybrid cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 5, pp. 1022–1035, May 2019.

[33] R. Ding, Y. Zhu, X. Feng, C. Zhang, and H. Zhu, "Continuous charging assignment algorithm for heterogeneous robot clusters based on E-CARGO," *Expert Syst. Appl.*, vol. 259, Jan. 2025, Art. no. 125175.

[34] M. Saqr, S. López-Pernas, and K. Murphy, "How group structure, members' interactions and teacher facilitation explain the emergence of roles in collaborative learning," *Learn. Individual Differences*, vol. 112, May 2024, Art. no. 102463.

[35] H. Ma, J. Li, Y. Tang, H. Zhu, Z. Huang, and W. Tang, "Universal optimization framework: Leader-centered learning team formation based on fuzzy evaluations of learners and E-CARGO," *IEEE Syst., Man, Cybern., Mag.*, vol. 9, no. 2, pp. 6–17, Apr. 2023.

[36] Z. Luo, Y. Dang, and W. Xu, "Academic interest scale for adolescents: Development, validation, and measurement invariance with Chinese students," *Frontiers Psychol.*, vol. 10, Oct. 2019, Art. no. 2301.
[37] H. Ma and Z. Hu, "Recommend trustworthy services using interval numbers of four parameters via cloud model for potential users," *Frontiers Comput. Sci.*, vol. 9, no. 6, pp. 887–903, Sep. 2015.
[38] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Rec.*, vol. 25, no. 2, pp. 103–114, Jun. 1996.
[39] M. Li, M. R. Metel, and A. Takeda, "Modified K-means algorithm with local optimality guarantees," in *Proc. 42th Int. Conf. Mach. Learn. (ICML)*, Jul. 2025, pp. 35656–35684.

Hua Ma (Senior Member, IEEE) received the B.S. degree in computer science and technology, the M.S. degree in computer application technology, and the Ph.D. degree in software engineering from Central South University, Changsha, China, in 2003, 2006, and 2016, respectively.

He is currently a Professor with the College of Information Science and Engineering, Hunan Normal University, Changsha. His research interests include personalized learning and services computing.

Xiangru Fu (Member, IEEE) received the B.S. degree in software engineering from Hunan Normal University, Changsha, China, in 2023, where she is currently pursuing the Ph.D. degree in computer science and technology.

Her research interests include personalized learning and big data mining.

Wensheng Tang received the B.S. degree in electronic information science and technology from Hunan Normal University, Changsha, China, in 1992, and received the M.S. and Ph.D. degrees in computer science and technology from the National University of Defense Technology, Changsha, in 1997 and 2009, respectively.

His research interests include distributed computing and cloud computing. His research interests include personalized learning and distributed computing.

Ming Chen received the B.S. degree in computer science and technology and M.S. degree in computer software and theory from Hunan Normal University, Changsha, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer software and theory from Wuhan University, Wuhan, China, in 2012.

She is currently an Associate Professor with the College of Information Science and Engineering, Hunan Normal University. Her current research interests include graph deep learning and its applications.

Haibin Zhu (Fellow, IEEE) received the B.S. degree in computer engineering from the Institute of Engineering and Technology, Zhengzhou, China in 1983, the M.S. degree in computer science and technology from the National University of Defense Technology, China in 1988, and the Ph.D. degree in computer science and technology from the National University of Defense Technology, Changsha, China in 1997.

He is a Full Professor with Nipissing University, North Bay, Canada. He has accomplished (published or in press) more than 300 research works, including more than 70 IEEE Transactions articles.

Dr. Zhu is a fellow of the Asia-Pacific Artificial Intelligence Association (AAIA) and International Institute of Cognitive Informatics and Cognitive Computing (I2CICC). He is the Founding Researcher of Role-Based Collaboration and Adaptive Collaboration and the creator of the E-CARGO model. He has offered 38 keynote speeches for international conferences and 96 invited talks internationally. His research has been sponsored by SSHRC, NSERC, IBM, DNDC, DRDC, and OPIC.

Keqin Li (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1985, and the Ph.D. degree in computer science from the University of Houston, Houston, TX, USA, in 1990.

He is a SUNY Distinguished Professor with The State University of New York, Buffalo, NY, USA, and a National Distinguished Professor with Hunan University, Changsha, China. He has authored or co-authored more than 1230 journal articles, book chapters, and refereed conference papers. He holds over 80 patents announced or authorized by the Chinese National Intellectual Property Administration. Since 2020, he has been among the world's top few most influential scientists in parallel and distributed computing, regarding single-year impact (ranked #2) and career-long impact (ranked #3) based on a composite indicator of the Scopus citation database. He is listed in Scilit Top Cited Scholars from 2023 to 2024.

Dr. Li is a member of the SUNY Distinguished Academy. He is an AAAS Fellow, an AAIA Fellow, an ACIS Fellow, and an AIIA Fellow. He is a member of the European Academy of Sciences and Arts. He is a Member of Academician of the Academy of Europe (Academia Europaea). He was a 2017 recipient of the Albert Nelson Marquis Lifetime Achievement Award for being listed in Marquis *Who's Who in Science and Engineering*, *Who's Who in America*, *Who's Who in the World*, and *Who's Who in American Education* for over 20 consecutive years. He received the Distinguished Alumnus Award from the Computer Science Department at the University of Houston in 2018. He received the IEEE TCCLD Research Impact Award from the IEEE CS Technical Committee on Cloud Computing in 2022 and the IEEE TCSVC Research Innovation Award from the IEEE CS Technical Community on Services Computing in 2023. He won the IEEE Region 1 Technological Innovation Award (Academic) in 2023. He was a recipient of the 2022-2023 International Science and Technology Cooperation Award and the 2023 Xiaoxiang Friendship Award of Hunan Province, China.