A Competition-Oriented Student Team Building Method

Dapeng Qu, Ruiduo Li, Tianqi Yang, Songlin Wu, Yan Pan, Xingwei Wang, and Keqin Li, *Fellow, IEEE*

***Abstract*—There are many important and interesting academic competitions that attract an increasing number of students. However, traditional student team building methods usually have strong randomness or involve only some first-class students. To choose more suitable students to compose a team and improve students' abilities overall, a competition-oriented student team building method is proposed. This would not only lead to better competition results by choosing more suitable students and teams but also improve the overall involvement of students in considering education fairness. First, a Big Data platform is constructed to collect students' various behavior data. Based on that, a competition with a six-tuple attribute and a student with a six-tuple attribute are modeled. Then, a corresponding utility function is designed for each attribute in the student model to denote the student's utility in this attribute for attending a competition. Furthermore, a team utility function is developed for each team to denote the utilities of all involved students. A team building utility function is also developed to denote the utilities of all involved teams. Second, a multiple-objective particle swarm optimization algorithm with dimension by dimension improvement is proposed to build appropriate teams to optimize team building utility maximization and education fairness simultaneously. Finally, extensive experimental results demonstrate that the overall performance of our proposed team building method not only has better performance in terms of team utility and student ability than other current methods, but also has better performance in terms of hyper volume and inverted generational distance than other optimization algorithms.**

***Index Terms*—Academic competition, collaborative learning, computer uses in education, team building.**

Manuscript received 17 December 2021; revised 25 December 2022, 16 May 2023, and 26 October 2023; accepted 29 November 2023. Date of publication 18 December 2023; date of current version 28 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62032013 and Grant 92267206, in part by the Natural Science Basic Research Fund of Liaoning Provincial Education Department under Grant LJC202002, in part by the Postgraduate Education Reform Project of Liaoning Province under Grant LNYJG2022012, in part by the 2021 Annual Project in 14th Five-Year Plan of Liaoning Provincial Education Science under Grant JG21DB242, and in part by the Postgraduate Education Reform Project of Liaoning University under Grant YJG202301055. *(Corresponding author: Xingwei Wang.)*

Dapeng Qu, Ruiduo Li, and Yan Pan are with the College of Information, Liaoning University, Shenyang 110036, China (e-mail: dapengqu@lnu.edu.cn; 1354942870@qq.com).

Tianqi Yang is with the College of Information, Liaoning University, Shenyang 110036, China, and also with Chengfang FinTech Company, Ltd., Beijing 100080, China (e-mail: tianqiyang0@163.com).

Songlin Wu is with the College of Information, Liaoning University, Shenyang 110036, China, and also with the Zhejiang Water Conservancy Information & Publicity Center, Hangzhou 310009, China (e-mail: songlinwu8@163.com).

Xingwei Wang is with the School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China (e-mail: wangxw@mail.neu.edu.cn).

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

Digital Object Identifier 10.1109/TLT.2023.3343525

Nomenclature

Notations in Student Team Building Method

Name	Description
$S_x(s_i)$	Attribute of student s_i in $x \in \{s_{\mathrm{ta}}, s_{\mathrm{pa}}, s_{\mathrm{ce}}, s_{la}, s_{ca}, s_{ae}\}$.
$G_y(g_j)$	Attribute of competition g_j in $y \in \{g_{\mathrm{pf}}, g_{\mathrm{cl}}, g_{\mathrm{ta}}, g_{\mathrm{pa}}, g_{\mathrm{ce}}, g_{\mathrm{re}}\}$.
$T_y(t_j^k)$	Attribute of team t_j^k built for competition g_j in $y \in \{g_{\mathrm{pf}}, g_{\mathrm{cl}}, g_{\mathrm{ta}}, g_{\mathrm{pa}}, g_{\mathrm{ce}}, g_{\mathrm{re}}\}$.
$U_x(s_i, t_j^k)$	Utility of student s_i's attribute for building team t_j^k to attend g_j in $x \in \{s_{\mathrm{ta}}, s_{\mathrm{pa}}, s_{\mathrm{ce}}, s_{la}, s_{ca}, s_{ae}\}$.
$tg(s_i, c_h)$	Student s_i's grade in theoretical course c_h.
$te(s_i, c_h)$	Student s_i's evaluation in theoretical course c_h.
$tc(c_h)$	Credit of theoretical course c_h.
$N_{tc}(s_i)$	Number of theoretical courses student s_i attends.
$pg(s_i, c_h)$	Student s_i's grade in practice course c_h.
$pc(c_h)$	Credit of practice course c_h.
$N_{pc}(s_i)$	Number of practical courses student s_i attends.
$N_g(s_i)$	Number of competitions student s_i had attend.
$N_t(s_i)$	Number of teams student s_i had involved in.
$N_t(g_j)$	Number of teams built for competition g_j.
$N_s(t_j^k)$	Number of students aggregated to build team t_j^k.
$simi(g_{j_1}, g_{j_2})$	Similarity between two different competitions g_{j_1} and g_{j_2}.
$E_z(s_i, s_l)$	Evaluation of student s_i made by student s_l in $z \in \{s_{la}, s_{ca}\}$.
$E_z(t_j^k)$	Evaluation of the whole team t_j^k in $z \in \{s_{la}, s_{ca}\}$.
$\bar{E}_z(s_i, t_j^k)$	Average of the evaluations of student s_i made by other members in team t_j^k in $z \in \{s_{la}, s_{ca}\}$.
$\hat{E}_z(s_i, t_j^k)$	Variance of the evaluations of student s_i made by other members in team t_j^k in $z \in \{s_{la}, s_{ca}\}$.

I. Introduction

Recently, as an essential supplement and important extension of classroom education, academic competition has become a crucial component of current higher education [1]. It can not only attract students' interest in learning but also improve their self-learning ability. Moreover, nearly all academic competitions require that students build a team to attend, which can undoubtedly enhance their abilities as a whole through teamwork between learners [2], [3].

Academic competition provides a natural common learning goal for the whole team. All members achieve this common goal through positive interdependence, considerable interaction, individual accountability, social skills, and group processing [4]. During the whole learning process of attending academic competitions, students can improve their abilities as a whole and prepare for their future work after graduation [5]. Therefore, academic competition has become an efficient way of achieving collaborative learning and attracts much attention from students, teachers, and companies.

At the current stage, different organizations have held hundreds of academic competitions to provide students with collaborative learning opportunities. For example, China "Internet+" College Students Innovation & Entrepreneurship Competition[1] is organized by the Ministry of Education, aiming to improve college students' innovation and entrepreneurship ability in the process of collaborative learning, demonstrate the achievements of global universities' innovation and entrepreneurship education, and build a platform for social investment resources and local economic development policies for global collegiate innovation and entrepreneurship projects [6]. Chinese Collegiate Programming Contest (CCPC)[2] is an annual competition organized by the Organizing Committee for the CCPC. It aims to improve Chinese college students' ability in programming innovation and solve practical problems in the process of collaborative learning.

Most competitions require multiple students to build a team to take part in. Therefore, to optimize the effect of collaborative learning, reasonable team building is essential. That is, a student should choose several appropriate partners to build a team to attend an academic competition [7]. "Appropriate" means that various properties of students should be considered. For example, not only theoretical ability and practical ability, but also leadership ability and cooperation ability should be considered when building a team. Moreover, the competition that can bring the highest profit should be chosen. However, "appropriate" is a complex and fuzzy concept. Not only must various properties of a student be considered, but the purpose of the competition should also be taken into consideration. Furthermore, when several competitions take place at the same time, a student should choose one or several competitions, not all of them, to conserve his or her energy.

Thus, we construct a competition-oriented student team building method. We first build a Big Data platform to collect students' behavior data, for example, students' theoretical behavior in the classroom. We not only model competitions with a six-tuple attribute but also model students with a six-tuple attribute based on the above data platform. Then, we design a corresponding utility function for each attribute in the student's model to denote the student's utility in this attribute for attending a competition. Moreover, we develop a team utility function to denote the utility of all involved students in this team for attending a competition and then a team building utility function to denote the utility of all involved teams for attending current competitions. To consider both team building utility and education fairness, that is, the team building utility maximization and the involved student utility difference minimization, we propose an improved multiple-objective particle swarm optimization (MOPSO) algorithm with dimension by dimension improvement (MOPSO-DDI). It applies a dimension by dimension evaluation and update strategy in the movement of particles in each iteration process, and an updated better value of one dimension with the old values of other dimensions can be combined into a new solution. The experimental results demonstrate that the proposed team building method has better performance than current state-of-the-art methods in terms of team utility, student utility, and student ability. Moreover, it can improve the ability of all students as a whole. Meanwhile, the proposed MOPSO-DDI also can achieve better performance than other multiobjective optimization algorithms, such as nondominated sorting genetic algorithm II (NSGA-II), MOPSO, and self-organized speciation-based MOPSO (SS-MOPSO), in terms of hyper volume (HV), and inverted generational distance (IGD).

The contributions of our work are summarized as follows.

1) We construct a Big Data platform to collect students' behavior data, which consist of theoretical behavior, practical behavior, and competition behavior of the teaching management module, laboratory management module, and competition management module, respectively. The detailed behavior data provide a solid basis for later analysis.

2) We model competitions and students with a six-tuple attribute based on the above data platform and then design a corresponding utility function for each attribute in the student's model to denote the student's profit in this attribute for attending a competition. Moreover, we develop a team utility function to denote all involved students' profit in this team for attending a competition and then a team building utility function to denote all involved teams' profit for attending current competitions.

3) Including taking team building utility maximization as one optimizing objective, from the viewpoint of the essence of higher education, we also take the difference of the utilities of these involved students' minimization as the other optimization objective; thus, our model aims at achieving multiobjective optimization about both student profits and education fairness by building appropriate teams to attend current competitions.

4) We propose an improved MOPSO-DDI. It applies a dimension by dimension evaluation and update strategy in particle movement, and an updated better value of one dimension with the old values of other dimensions can be combined into a new solution. It demonstrates a better performance than other state-of-the-art methods, such as NSGA-II, MOPSO, and SS-MOPSO, in terms of HV and IGD.

The rest of this article is organized as follows. Section II reviews the related work about the team building and optimization algorithms. Section III presents the models and formulates the problem. Section IV proposes an improved MOPSO variant (MOPSO-DDI) and applies it to solve the defined problem. Section V evaluates the model and the corresponding algorithm with extensive experimental results. Finally, Section VI concludes this article.

[1] [Online]. Available: https://cy.ncss.org.cn/
[2] [Online]. Available: https://ccpc.io/

II. Related Work

We review the related work about the team building and optimization algorithms.

A. Team Building

Currently, academic competitions attract more and more attention from teachers and students in universities and the corresponding companies. However, team building for academic competitions has not received enough attention. In most cases, only some first-class students with good scores are usually chosen. Thus, there are three usual and classical team-building methods, namely, choosing students with first-class theoretical ability randomly (FCTA), students with first-class practical ability (FCPA), and students with first-class theoretical and practical ability (FCTPA). Obviously, some important issues, for example, leadership and cooperation ability, are ignored by these methods. On the other hand, some students choose partners, coaches (teachers), and competitions with high randomization [8]. For example, they usually choose roommates or classmates to build a team for a close relationship. The close partners may just take a free ride and do not contribute their service to the whole team.

Obviously, team building for academic competitions is a complex problem, not only the involving students that compose the competition team, but also the education fairness that covers all students should be considered. It means that the team building should bring huge benefits to all involving students, and the income for different students should be reduced to achieve education fairness. Thus, the team building can be modeled to be a multiple-objective optimization problem.

B. Optimization Algorithms

Evolutionary algorithms integrate biological information into metaheuristic algorithms and have achieved numerous breakthrough research results in the field of combinatorial optimization and numerical optimization [9]. Among these evolutionary algorithms, particle swarm optimization (PSO) is considered as the most representative and successful one, since there had been over 60 000 publications on the related research just up to 2015 [10]. It was higher than the sum of other six popular intelligence algorithms, namely, differential evolution algorithm, ant colony optimization, artificial bee colony, bat algorithm, bacterial foraging optimization, and glowworm swarm optimization. PSO [11] is proposed by Kennedy and Eberhart in 1995 and inspired by observing the social behaviors of the individuals in bird flocking. Each particle represents a solution of the optimization problem, and its position is randomly generated in the initial procedure and adjusted based on the self-learning ability and social learning ability in solution space. On the one side, PSO owns the advantages of easy realization and high efficiency; on the other side, it has some disadvantages of premature convergence, low convergence rate, and entrapment in local optimum. Thus, there have been a great number of interesting variants, such as orthogonal learning PSO [12], semiautonomous particle swarm optimizer [13], and PSO with an enhanced learning strategy and crossover operator [14], proposed to solve the above issues.

However, the real-world problems usually have multiple conflicting optimization objectives, not a single one, and the best solution for a certain objective cannot optimize and even degrade other objectives [15]. These complex multiobjective optimization problems widely exist in production scheduling, network communication, etc. The traditional multiobjective optimization algorithms usually take the weighting and distance method to convert the multiobjective problem into a single-objective one, and then apply traditional evolutionary algorithms to optimize. This kind of methods relies on prior knowledge, and is heavily limited by the shape of the Pareto front. They usually fail in the nonlinear and high dimension multiobjective problem.

Therefore, various evolutionary algorithms are improved to solve multiobjective optimization problems. For example, Cello and Lechuga [16] proposed MOPSO in 2002. It takes advantage of an adaptive gridding strategy and an external archive to save the nondominated solutions. After that, due to their simplicity and fast convergence, some MOPSO variants have been proposed and are extensively used for solving different real-world problems [17], [18]. For example, pccsAMOPSO is a self-adaptive MOPSO and based on a parallel cell coordinate system (PCCS). PCCS is used to assess the evolutionary environment, including density, rank, and diversity indicators, based on the measurements of parallel cell distance, potential entropy, and distribution entropy, respectively. Moreover, strategies are proposed for selecting global best and personal best, maintaining archive, adjusting flight parameters, and perturbing stagnation and are integrated into pccsAMOPSO [19]. Technique for order of preference by similarity to ideal solution (TOPSIS) Fuzzy MOPSO is proposed to solve the trapezoidal labyrinth weir optimization problem. It utilizes TOPSIS to rank the solutions, while a fuzzy inference system is developed to select the algorithm strategy for finding two leaders among the nondominated solutions [20]. Self-organized speciation-based MOPSO (SS-MOPSO) is proposed to locate multiple Pareto optimal solutions for solving multimodal multiobjective problems. The speciation strategy is used to form stable niches, and these niches/subpopulations are optimized to search and maintain Pareto-optimal solutions in parallel. Moreover, SS-MOPSO is incorporated with the nondominated sorting scheme and special crowding distance techniques to maintain the diversity of the solutions in decision and objective spaces [21]. A novel MOPSO, which combines a balanceable fitness estimation method and a novel velocity update equation, is proposed to tackle many-objective optimization problems. Moreover, an evolutionary search is further run on the external archive in order to provide another search pattern for evolution [22].

On the other hand, multiswarm particle swarm optimization (MSPSO) methods generate multidiverse particle swarms to carry out a specific task [10]. For example, a MSPSO method is adopted to generate multidiverse particle swarms on several cross-training subsets to select effective emotional features, where these swarms are utilized to find the best features by the F-Measure fitness function [23]. Cooperative MSPSO is proposed to divide the entire population into four cooperative subswarms with an adaptive and time-varying inertia weight. The particles of each subswarm share the best overall optimum to ensure the cooperation between the four subswarms. Moreover, the inertia weight is used to create search potential and maintain a balance between exploitation and exploration [24]. A new MSPSO variant is proposed to balance the exploration ability and exploitation ability. It is based on multiple swarms framework cooperating with the dynamic subswarm number strategy, sub-swarm regrouping strategy, and purposeful detecting strategy [25]. A parallel PSO framework is proposed to solve the sparse reconstruction problem based on Compute Unified Device Architecture (CUDA) platform on graphics processing unit (GPU). Each particle is launched by CUDA threads, and the swarm is divided into multiple subswarms in CUDA streams to further utilize potential computing resources in the GPU. Moreover, a local search strategy based on gradient and a particle coding strategy is combined into PSO to achieve better reconstruction accuracy and accelerate convergence [26].

Some other evolutionary algorithms are also improved to tackle multiobjective optimization problems. For example, NSGA-II is proposed to alleviate the difficulties in multiobjective evolutionary algorithms. Specifically, a fast nondominated sorting approach with lower computational complexity and a selection operator are presented. The selection operator created a mating pool by combining the parent and offspring populations and selecting the best solutions [27]. Nondominated sorting whale optimization algorithm collects all nondominated Pareto optimal solutions in the achieve and choose the best solutions from the collection of all Pareto optimal solutions using a crowding distance mechanism based on the coverage of solutions and bubble-net hunting strategy to guide humpback whales toward the dominated regions of multiobjective search spaces [28].

C. Multiobjective Optimization for Team Building

Obviously, based on the above analysis, team building for academic competitions is a multiobjective optimization problem, and an effective multiobjective optimization algorithm should be designed to solve it.

III. Model and Problem Formulation

We develop a Big Data platform to collect and analyze the student's various behaviors [29], namely, theoretical behavior, practical behavior, and competition behavior, to describe their different attributes. As shown in Fig. 1, there are four layers, namely, collection layer, analysis layer, attribute layer, and application layer. Furthermore, there are three modules, namely, teaching management, laboratory management, and competition management, in the collection layer. These modules are used to collect and analyze theoretical behavior data, practical behavior data, and competition behavior data, respectively. The theoretical behavior data and practical behavior data are analyzed and transformed into theoretical ability and practical ability, respectively. The competition behavior data are analyzed and represent competition experience, leadership ability, cooperation ability, and available energy. These four abilities can be used to model students when they build a team to attend competitions. In the application layer, we just focus on our team building, we calculate the student utility and team utility, and then build appropriate teams for the target competitions. The notations and terminologies used in the proposed model are defined in the Nomenclature.

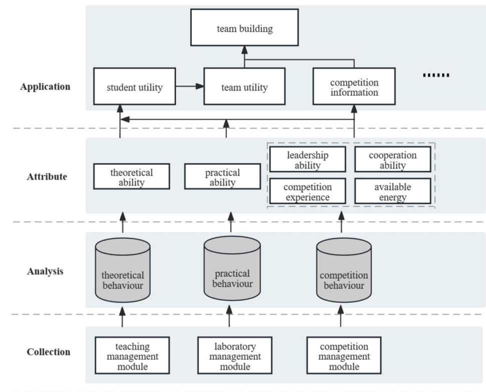

Fig. 1. Big Data platform for team building.

A. Competition Model

For each competition $g_j \in G$ (competition set), where $1 \leq j \leq N_g$ (N_g is the number of competitions, and $N_g = |G|$), we describe it with a six-tuple $\langle pf, cl, ta, pa, ce, re\rangle$, where g_{pf}, g_{cl}, g_{ta}, g_{pa}, g_{ce}, and g_{re} denote the professional field, competition level, theoretical ability, practical ability, competition experience, and required energy of a competition, respectively. These properties can be divided into two types, where g_{pf} and g_{cl} denote the basic characteristics of a competition, and the latter four denote the requirements on the students for attending a competition.

To describe g_{pf} fairly, we divide the fields of disciplines of conferring academic degrees with equally spaced in (0, 1), and then further divide each field with equally spaced. Thus, two competitions with a similar professional field will have a near distance in g_{pf}.

There are so many competitions; thus, we apply the analytic hierarchy process (AHP) [30], which is a structured technique for organizing and analyzing complex decisions to define g_{cl}. AHP is a simple, flexible, and practical multicriteria decision-making method for quantitative analysis of qualitative problems. Its characteristic is to organize various factors in complex problems by dividing them into interrelated and orderly levels. According to the subjective judgment structure of certain objective reality (mainly pairwise comparison), it directly and effectively combines the expert opinions and the objective judgment results of analysts and quantitatively describes the importance of pairwise comparison of elements at one level. Then, the weight reflecting the relative importance order of elements at each level is calculated by a mathematical method for organizing and analyzing complex decisions to define g_{cl}.

TABLE I
COMPETITION LEVEL HIERARCHY TABLE

Goal Level	Factor Level	Project Level
Competition level	Organization hierarchy	Class A: Internationality level
		Class B: Nation level
		Class C: State level
		Class D: University level
	Competition mode	limitation and selection
		limitation and no selection
		no limitation and selection
		no limitation and no selection
	Competition history	[0, 5) year
		[5,10] year
		(10, +∞) year

Specially, we make a decision with three levels, namely, *Goal Level*, *Factor Level*, *Project Level*. The *Factor Level* consists of three-part, namely, organizational hierarchy, competition mode, and competition history. The *Project Level* consists of different components under each *Factor Level*. The competition level hierarchy table is shown in Table I.

The latter four properties are determined by the Delphi method [31]. It is a structured method and relies on a panel of experts. These experts answer questionnaires in two or more rounds. After each round, we provide an anonymous summary of the experts' forecasts from the previous round and the reasons they provided for their judgments. Thus, experts are encouraged to revise their earlier answers in light of the replies of other members. During this process, the range of the answers is expected to decrease, and the group is believed to converge toward the correct answer.

When a competition is determined, the requirements of its involved teams are determined. Thus, a team t_j^k obviously inherits the six-tuple from the corresponding competition g_j. Moreover, the competition g_j implies $N_t(g_j)$, which is the number of teams built for g_j, and the team t_j^k implies $N_s(t_j^k)$, which is the number of students aggregated for t_j^k.

In order to help students quickly understand the new competition, we define a similarity function of two competitions g_{j_1} and g_{j_2} as (1) shown at the bottom of the this page, where $G_y(g_{j_h})$ ($y \in \{g_{\text{pf}}, g_{\text{cl}}, g_{\text{ta}}, g_{\text{pa}}, g_{\text{ce}}, g_{\text{re}}\}$, $h \in \{1, 2\}$) denotes the value of an attribute of competition g_{j_h}. The form of coordinate distance will give a high similarity to two competitions with a nearer value in each attribute.

B. Student Model

For each student $s_i \in S$ (student set), where $1 \leq i \leq N_s$ (N_s is the number of students, and $N_s = |S|$), we describe it with a six-tuple $\langle s_{\text{ta}}, s_{\text{pa}}, s_{\text{ce}}, s_{la}, s_{ca}, s_{ae}\rangle$, where s_{ta}, s_{pa}, s_{ce}, s_{la}, s_{ca}, and s_{ae} denote the theoretical ability, practical ability, competition experience, leadership ability, cooperation ability, and available energy of a student, respectively. Thus, $S_x(s_i)$ ($x \in \{s_{\text{ta}}, s_{\text{pa}}, s_{\text{ce}}, s_{la}, s_{ca}, s_{ae}\}$) denotes the student s_i's one attribute value.

1) Attribute of Models: We use the data from the Big Data platform to calculate the value of each attribute. The theoretical ability reflects a student's ability to solve a problem in theory, and it can be calculated by the theoretical grade and the evaluation in the course, as shown in the following:

$$S_{s_{\text{ta}}}(s_i) = \frac{\sum_{h=1}^{N_{tc}(s_i)} (\alpha_{tg} \times tg(s_i, c_h) + \alpha_{te} \times te(s_i, c_h)) \times tc(c_h)}{\sum_{h=1}^{N_{tc}(s_i)} tc(c_h)} \tag{2}$$

where $tg(s_i, c_h)$ denotes student s_i's grade in theoretical course c_h, $te(s_i, c_h)$ denotes student s_i's evaluation in theoretical course c_h, $tc(c_h)$ denotes the credit of theoretical course c_h, $N_{tc}(s_i)$ denotes the number of theoretical courses student s_i attended, and α_{tg} and α_{te} ($0 \leq \alpha_{tg}, \alpha_{te} \leq 1$, $\alpha_{tg} + \alpha_{te} = 1$) are the weight factors and determined by whether we attach more importance to students' grade or students' evaluation in the theoretical course.

Practical ability reflects a student's ability to solve a problem in practice. It is very important for students to attend a competition and calculated as follows:

$$S_{s_{\text{pa}}}(s_i) = \frac{\sum_{h=1}^{N_{pc}(s_i)} pg(s_i, c_h) \times \text{pc}(c_h)}{\sum_{h=1}^{N_{pc}(s_i)} \text{pc}(c_h)} \tag{3}$$

where $pg(s_i, c_h)$ denotes student s_i's grade in a practical course c_h, $\text{pc}(c_h)$ denotes the credit of practical course c_h, and $N_{\text{pc}}(s_i)$ denotes the number of practical courses student s_i attend.

When there are multiple competitions to be chosen, a student will have a high probability to attend and get a high grade in a competition that he or she had experienced or is similar to the competitions he or she had experienced. Thus, the experience of student s_i in a new competition g_{j_1} is calculated as follows:

$$S_{s_{\text{ce}}}(s_i, g_{j_1}) = \frac{\sum_{j2=1}^{N_g(s_i)} (G_{g_{\text{ce}}}(g_{j_2}) \times \text{simi}(g_{j_1}, g_{j_2}))}{N_g(s_i)} \tag{4}$$

where $N_g(s_i)$ denotes the number of competitions student s_i had attended, and $G_{g_{\text{ce}}}(g_{j_2})$ denotes the experience of competition g_{j_2}. Obviously, the more competitions a student had attended, the more similar the new competition and the old competitions are, and the more experience the student will have for the new one.

After finishing a competition, the competitors (students) and coaches (teachers) will evaluate the leadership ability and cooperation ability of other members in the same team built for this competition. Thus, student s_i's leadership ability and cooperation ability are calculated by the following equations,

$$\text{simi}(g_{j_1}, g_{j_2}) = 1 - \sqrt{\begin{aligned}&(G_{g_{\text{pf}}}(g_{j_1}) - G_{g_{\text{pf}}}(g_{j_2}))^2 + (G_{g_{\text{cl}}}(g_{j_1}) - G_{g_{\text{cl}}}(g_{j_2}))^2 + (G_{g_{\text{ta}}}(g_{j_1}) - G_{g_{cta}}(g_{j_2}))^2 \\ &+(G_{g_{\text{pa}}}(g_{j_1}) - G_{g_{\text{pa}}}(g_{j_2}))^2 + (G_{g_{\text{ce}}}(g_{j_1}) - G_{g_{\text{ce}}}(g_{j_2}))^2 + (G_{g_{\text{re}}}(g_{j_1}) - G_{g_{\text{re}}}(g_{j_2}))^2\end{aligned}} \tag{1}$$

respectively:

$$S_{s_{la}}(s_i) = \frac{\sum_{k=1}^{N_t(s_i)} \bar{E}_{s_{la}}(s_i, t_j^k)}{N_t(s_i)} \tag{5}$$

$$S_{s_{ca}}(s_i) = \frac{\sum_{k=1}^{N_t(s_i)} \bar{E}_{s_{ca}}(s_i, t_j^k)}{N_t(s_i)} \tag{6}$$

where $N_t(s_i)$ denotes the number of teams involved by student s_i, and $\bar{E}_{s_{la}}(s_i, t_j^k)$ and $\bar{E}_{s_{ca}}(s_i, t_j^k)$ are the average of the evaluations of student s_i made by other members in team t_j^k in leadership ability and cooperation ability, respectively. They can be calculated directly based on the evaluations of other team members, just as the following, respectively:

$$\bar{E}_{s_{la}}(s_i, t_j^k) = \frac{\sum_{s_l \in t_j^k, s_l \neq s_i} E_{s_{la}}(s_i, s_l)}{N_s(t_j^k)} \tag{7}$$

$$\bar{E}_{s_{ca}}(s_i, t_j^k) = \frac{\sum_{s_l \in t_j^k, s_l \neq s_i} E_{s_{ca}}(s_i, s_l)}{N_s(t_j^k)}. \tag{8}$$

Attending competitions is so important that it will cost plenty of students' time. Thus, to assure that a student has enough energy to attend competitions, we introduce the concept of "available energy" to denote the status of a student in an academic year. In the beginning, a student is assumed to have full available energy, for example 1, when he or she attends a competition, he or she will consume some energy that is required by the competition. Thus, the available energy of student s_i is calculated as follows:

$$S_{s_{ae}}(s_i) = 1 - \sum_{j=1}^{N_g(s_i)} G_{g_{\text{re}}}(g_j). \tag{9}$$

2) Utility Function: Based on the above analysis, we can get the six properties of a student. To denote a student's profit for building a team to attend a competition in each attribute, we introduce the concept of "utility" to define a utility function for each attribute [32].

Take the utility of theoretical ability as an example, we design a utility function $U_{s_{\text{ta}}}(s_i, t_j^k)$ of theoretical ability for student s_i for building team t_j^k to attend competition g_j as the following equation to denote his profit in the theoretical ability for doing that:

$$U_{s_{\text{ta}}}(s_i, t_j^k) = S_{s_{\text{ta}}}(s_i) \times T_{g_{\text{ta}}}(t_j^k). \tag{10}$$

Similarly, the utility functions of practical ability, competition experience for the student s_i for building team t_j^k to attending competition g_j as the following equations respectively:

$$U_{s_{\text{pa}}}(s_i, t_j^k) = S_{s_{\text{pa}}}(s_i) \times T_{g_{\text{pa}}}(t_j^k) \tag{11}$$

$$U_{s_{\text{ce}}}(s_i, t_j^k) = S_{s_{\text{ce}}}(s_i) \times T_{g_{\text{ce}}}(t_j^k) \tag{12}$$

where $S_{s_{\text{ce}}}(s_i, t_j^k)$ denotes the experience of student s_i in building team t_j^k to attend competition g_j, and $T_{g_{\text{ce}}}(t_j^k)$ denotes the competition experience of team t_j^k, which can be got from the corresponding competition g_j.

The utility function of a student's leadership ability describes the profit of a student in leadership ability for building a team to attend a corresponding competition. Thus, we define the utility function of student s_i in leadership ability for building team t_j^k as follows:

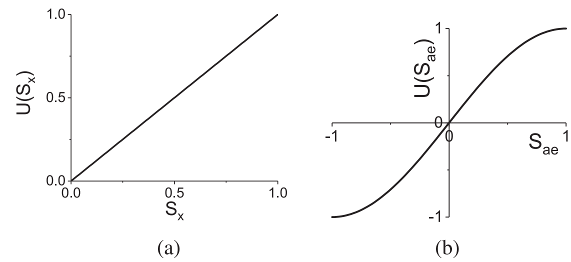

Fig. 2. Illustration of the utility function of different attributes of a student. (a) Utility function of the former five properties. (b) Utility function of the available energy attribute.

$$U_{s_{la}}(s_i, t_j^k) = S_{s_{la}}(s_i) \times E_{s_{la}}(t_j^k) \tag{13}$$

where $E_{s_{la}}(t_j^k)$ denotes the evaluation of leadership of the whole team t_j^k and is calculated as follows:

$$E_{s_{la}}(t_j^k) = \frac{\bar{E}_{s_{la}}(s_i, t_j^k)}{1 + \frac{1}{\hat{E}_{s_{la}}(s_i, t_j^k)}}, s_i \in t_j^k \tag{14}$$

where $\bar{E}_{s_{la}}(s_i, t_j^k)$ and $\hat{E}_{s_{la}}(s_i, t_j^k)$ denote the average and variance of the leadership ability of all members s_i in team t_j^k. This division form will give a high evaluation to the team with a high average and variance of the leadership of all members. This is because higher average leadership means that all members have good leadership and higher variance implies members with different leadership will build a harmonious team conveniently.

In the same way, we can calculate the utility function of student s_i in cooperation ability for building team t_j^k as follows:

$$U_{s_{ca}}(s_i, t_j^k) = S_{s_{ca}}(s_i) \times E_{s_{ca}}(t_j^k) \tag{15}$$

where $E_{s_{ca}}(t_j^k)$ denotes the evaluation of cooperation of the whole team t_j^k and is calculated as follows:

$$E_{s_{ca}}(t_j^k) = \frac{\bar{E}_{s_{ca}}(s_i, t_j^k)}{1 + \hat{E}_{s_{ca}}(s_i, t_j^k)}, s_i \in t_j^k \tag{16}$$

where $\bar{E}_{s_{ca}}(s_i, t_j^k)$ and $\hat{E}_{s_{ca}}(s_i, t_j^k)$ denote the average and variance of the cooperation ability of all members s_i in team t_j^k. This division form will give a high evaluation to the team with a high average and variance of the cooperation of all members. This is because higher average cooperation means that all members have good cooperation and will build a harmonious team conveniently. The utility function of student's available energy describes the profit of student s_i's available energy in building team t_j^k to attend competition g_j, and is defined as follows:

$$U_{s_{ae}}(s_i, t_j^k) = sin\left(S_{s_{ae}}(s_i, t_j^k) \times \frac{\pi}{2}\right). \tag{17}$$

To understand the above utility functions easily, two different figures as shown in Fig. 2. Fig. 2(a) demonstrates that the multiplication form of the utility function is appropriate for the former five properties, and it considers both properties of a student and a competition. It gives a higher utility to the students and competitions with the better attribute. Fig. 2(b) shows that the trigonometric form of the utility function is appropriate for the available energy, and it considers both the available energy attribute of students and competitions. It gives a higher utility to the students with higher available energy and considers the negative influence when the available energy is lower than 0, that is, the student attends too many competitions to get negative influence.

Based on the above utility function in each attribute, we define a total utility function in student s_i's six properties for building team t_j^k to attend competition g_j as (18) shown at the bottom of the this page, where $\omega_{s_{\mathrm{ta}}}, \omega_{s_{\mathrm{pa}}}, \omega_{s_{\mathrm{ce}}}$, and $\omega_{s_{ae}}$ are the weight factor of four properties, namely, theoretical ability, practical ability, competition experience, and available energy, respectively, and $0 \leq \omega_{s_{\mathrm{ta}}}, \omega_{s_{\mathrm{pa}}}, \omega_{s_{\mathrm{ce}}}, \omega_{s_{ae}} \leq 1, \omega_{s_{\mathrm{ta}}}+\omega_{s_{\mathrm{pa}}}+\omega_{s_{\mathrm{ce}}}+\omega_{s_{ae}}=1$. They are determined by whether we put a higher weight on one attribute. In a similar way, $\omega_{s_{la}}$ and $\omega_{s_{ca}}$ are the weight factors of leadership ability and cooperation ability, and $0 \leq \omega_{s_{la}}, \omega_{s_{ca}} \leq 1, \omega_{s_{la}} + \omega_{s_{ca}}=1$. It is also determined by whether we put a higher weight on leadership or cooperation ability. The exponential form will give a high total utility to the students with the high attribute. The leadership and cooperation ability reflect the attribute of a student in a team, thus they are in the exponent.

Furthermore, the utility of team t_j^k for attending competition g_j is the summary of the utilities of all members in this team and can be calculated as follows:

$$U(t_j^k) = \sum_{s_i \in t_j^k} U(s_i, t_j^k). \tag{19}$$

In the same way, the utility of team building TB for current competition set G from the whole student set S is the summary of the utilities of all built teams $U(t_j^k)$ and can be calculated as follows:

$$U(\mathrm{TB}) = \sum_{t_j^k \in \mathrm{TB}} U(t_j^k). \tag{20}$$

C. Illustration Example

To help understand our proposed method, we give a simple example for illustration. We assume that there are two academic competitions, namely, International Collegiate Programming Contest (ICPC) and Chinese College Students Computer Design Competition (CCSCDC), and ten students. Moreover, ICPC needs two teams and CCSCDC requires only one team, and each team consists of three students.

Take ICPC as an example, its six-tuple can be denoted as $\langle 0.32, 0.3238, 0.42, 0.53, 0.36, 0.23\rangle$. These values are obtained as defined in Section II-A. Similarly, CCSCDC is denoted as $\langle 0.32, 0.2206, 0.38, 0.5, 0.3, 0.2\rangle$. Moreover, their similarity is 0.8671, as calculated as (1).

TABLE II
PROPERTIES OF TEN STUDENTS IN THE EXAMPLE

Name	Ability
student1	0.9263, 0.8684, 0.2658, 0.3600, 0.7300, 1
student2	0.8886, 0.8763, 0.6819, 0.6819, 0.1648, 1
student3	0.8748, 0.9096, 0.5746, 0.5746, 0.5560, 1
student4	0.8731, 0.8886, 0.3766, 0.3766, 0.6045, 1
student5	0.8721, 0.7634, 0.6530, 0.6530, 0.0593, 1
student6	0.8699, 0.8963, 0.4541, 0.4541, 0.3836, 1
student7	0.7458, 0.7985, 0.0819, 0.0819, 0.4907, 1
student8	0.7452, 0.7633, 0.6064, 0.6064, 0.4437, 1
student9	0.6984, 0.6832, 0.4475, 0.4475, 0.0706, 1
student10	0.6963, 0.6532, 0.0516, 0.0516, 0.5415, 1

We name the first student student1 and can get her different properties based on (2)–(9), respectively. Thus, we can denote student1 as $\langle 0.9263, 0.8684, 0.2658, 0.36, 0.73, 1\rangle$. Similarly, we can get the properties of the other nine students, as given in Table II.

D. Optimization Problem Definition

Based on the above analysis, for the given competitions, we need to build appropriate teams for each competition based on its requirements from the whole students. Because there are usually some limitations on the number of teams for each competition, and the numbers of members in each team, not all students must be chosen to attend competitions. Just like those athletic contests, for example, the Olympic games, not only the students but also the colleges try to chase a good result. On the other side, with the rapid development of higher education, fairness attracts more and more attention [33]. More and more students hope to build a team to attend competitions to improve their various abilities. Moreover, it should not be just a few top students who are chosen to attend competitions, those students who attend competitions should make progress together, not further widen their gaps. Therefore, we seek to maximize the utility of the team building TB by choosing the appropriate students to build teams to attend corresponding competitions, at the same time, minimize the difference value of the utilities of these involved students in the TB. The mathematical description is defined as follows:

$$\begin{aligned} & \max \quad U(\mathrm{TB}) \\ & \min \quad U_{\max}(S, \mathrm{TB}) - U_{\min}(S, \mathrm{TB}) \\ \text{s.t.} \quad & s_i \in t_j^k, \quad t_j^k \in \mathrm{TB}, \quad s_i \in S \\ & 0 \leq i \leq N_s \\ & 0 \leq k \leq N_t \\ & 0 \leq j \leq N_g \end{aligned} \tag{21}$$

$$U(s_i, t_j^k) = \left(\omega_{s_{\mathrm{ta}}} \times U_{s_{\mathrm{ta}}}(s_i, t_j^k) + \omega_{s_{\mathrm{pa}}} \times U_{s_{\mathrm{pa}}}(s_i, t_j^k) + \omega_{s_{\mathrm{ce}}} \times U_{s_{\mathrm{ce}}}(s_i, t_j^k) + \omega_{s_{ae}} \times U_{s_{ae}}(s_i, t_j^k)\right)^{\omega_{s_{la}} \times U_{s_{la}}(s_i, t_j^k) + \omega_{s_{ca}} \times U_{s_{ca}}(s_i, t_j^k)}. \tag{18}$$

where $U_{\min}(S, \mathrm{TB})$ and $U_{\max}(S, \mathrm{TB})$ are the minimum value and the maximum value of one student in S in the current team building TB, and are calculated as follows:

$$U_{\max}(S, \mathrm{TB}) = \max(U(s_i, t_j^k)|s_i \in S, t_j^k \in \mathrm{TB})$$
$$U_{\min}(S, \mathrm{TB}) = \min(U(s_i, t_j^k)|s_i \in S, t_j^k \in \mathrm{TB}). \tag{22}$$

Obviously, the above optimization problem is a multiconstrained multiobjective optimization one, and thus, we can apply some intelligence or heuristic algorithms to find feasible solutions [34].

IV. Optimization Algorithm

We design an improved MOPSO-DDI and apply it to solve the above modeled multiobjective problem as defined in (21).

A. MOPSO With Dimension by Dimension Improvement (MOPSO-DDI)

PSO is inspired by the social behaviors of the individuals in bird flocking. Each particle represents a solution of the optimization problem, and its position is randomly generated in the initial procedure and adjusted based on the self-learning ability and social learning ability in solution space. The specific movement is given as follows: where $v_i(t)$ and $x_i(t)$ are the velocity and position of particle i in the tth iteration, respectively. w is the inertia weight controlling the effect of the previous velocity on the current velocity. $c1$ and $c2$ are referred to the self-learning ability and social learning ability, respectively. $r1$ and $r2$ are mutually independent pseudorandom numbers, subject to an uniform distribution on [0,1].

Based on the classical PSO, MOPSO uses the concept of Pareto dominance to determine the flight direction of a particle and maintains previously found nondominated vectors in a global repository that is later used by other particles to guide their flight. The advantage of fast convergence in turn makes MOPSO easily trap into local optimization; thus, we design MOPSO-DDI, which applies a dimension by dimension evaluation and update strategy, that is, an updated better value of one dimension with the old values of other dimensions can combined into a new solution [35]. Specially, after calculating the velocity of each particle, when calculating the position, we compare the new position of the current particle with the global best particle by dimension, and the better one will chosen in each dimension. Moreover, to further avoid trapping into local optimization, we choose a particle randomly as the candidate solution for each dimension in position update.

Since MOPSO is a popular multiobjective optimization algorithm, we just provide the proposed dimension by dimension improvement method, and its procedure is described as Algorithm 1.

Algorithm 1: Dimension by Dimension Improvement.

Input: $v_i(t), x(t-1), x_{gbest}$
Output: $x_i(t)$

1 ***Update*** *the position* $x_i(t)$ *of particle* i *as Eq. 23;*
2 **for** *each dimension* **do**
3   ***Get*** *a new solution* $x_i'^d(t)$ *by replacing the current dimension with the corresponding dimension in gbest;*
4   **if** *the new solution* $x_i'^d(t)$ *is dominant to the current solution* $x_i^d(t)$ **then**
5   ***Replace*** *the current dimension in* $x_i^d(t)$ *with the corresponding dimension in gbest to get a better solution* $x_i^d(t)$*;*
6   ***Get*** *a new solution* $x''^d_i(t)$ *by replacing the current dimension with the corresponding dimension in a random particle;*
7   **if** *the new solution* $x_i''^d(t)$ *is dominant to the current solution* $x_i^d(t)$ **then**
8   ***Replace*** *the current dimension in* $x_i^d(t)$ *with the corresponding dimension in the random particle to get a better solution* $x_i^d(t)$*;*
9 ***Return*** $x_i(t)$*;*

In Algorithm 1, $v_i(t)$ is the velocity of particle i in (t)th iteration, $x_i(t)$ is the position of particle i in (t)th iteration, and $x(t-1)$ is the position of all particles in $(t-1)$th iteration. Line 1 is to update the position of particle i as (23) shown at the bottom of the this page. Lines 2–8 are to take dimension by dimension improvement operations. Among them, for each dimension, lines 3–5 are to try to get a new solution by replacing each dimension with the corresponding one in gbest, and lines 6–8 are to try to get a new solution by replacing each dimension with the corresponding one in a random destination particle. Line 9 returns the new position of particle i as the solution.

B. CTB With MOPSO-DDI

We apply the proposed MOPSO-DDI to solve the above modeled multiobjective problems, namely, the competition-oriented student team building. The procedure of competition-oriented student team building with the MOPSO-DDI algorithm is described as Algorithm 2.

In Algorithm 2, N_p is the number of particles, and N_{it} is the number of iterations. $p$best and $g$best are a local optimal particle and global optimal particle, respectively. Line 1 is to initiate particle with random position and velocity, and the position of a particle is one team building, which contains some built teams. Lines 2–14 are to optimize the whole team building

$$v_i(t) = w \times v_i(t-1) + c_1 \times r_1 \times (x_{p\mathrm{best}} - x_i(t)) + c_2 \times r_2 \times (x_{g\mathrm{best}} - x_i(t))$$
$$x_i(t) = x_i(t-1) + v_i(t) \tag{23}$$

Algorithm 2: Competition Team Building (CTB) With MOPSO-DDI.

Input: S, G, N_{it}
Output: A_N
1 ***Initiate*** *the velocity and position of each particle randomly; it=1;* A_N*=∅;*
2 **while** $it \leq N_{it}$ **do**
3 **for** *each particle* **do**
4 ***Calculate*** *the fitness of particle as Eq.* (21)*;*
5 **if** *the current solution* A_C *is non-dominated solution* **then**
6 A_N **=** $A_N \cup \boldsymbol{A}_C$**;**
7 ***Sort*** *all solutions in* A_N *by the descending order of crowding distance*;
8 ***Update*** A_N *based on the dominance relationship*;
9 ***Update*** *pbest and gbest*;
10 **for** *each particle* **do**
11 ***Update*** *its velocity as Eq.* (23);
12 ***Update*** *its position as Algorithm 1*;
13 it++;
14 ***Return*** A_N;

with MOPSO-DDI. Among them, lines 4–6 are to record all nondominated solutions; lines 7–9 are to calculate the quarterly distance of all particles in A_N, sort them in descending order, and update $p$best and $g$best; lines 10–12 are to update the velocity and position of each particle with dimension by dimension improvement; and line 14 returns A_N as the solution.

As stated above, the time complexity of the proposed CTB with MOPSO-DDI is dependent on the maximum number of iterations N_{it}, the number of particles N_p, and the number of dimensions N_d. Therefore, the complexity is $\mathbf{O}(N_{it} \times N_p \times N_d)$.

Recall the illustration example in Section II-C, we should build two teams for ICPC and one team for CCSCDC from the ten students based on their properties, which are described in Section II-C. Based on the defined problem and the applied optimization algorithms, we can obtain a set of feasible Pareto solutions. In this example, there are four feasible solutions in the obtained Pareto solutions set, namely, [(5,1,6),(3,8,9),(9,4,5)], [(5,2,7),(3,8,9),(9,6,5)], [(5,2,6),(3,8,9),(9,7,5)], and [(5,2,7),(3,8,9),(9,6,4)]. Furthermore, we can further choose the final solution based on different conditions; for example, we can give more important weight to the first optimization objective than the second one in (21), that is, to pursue excellent results in attending academic competitions. On the other hand, we can increase the weight of the second optimization objective to emphasize education fairness. Here, we just take the first solution with the maximum $U(\text{TB})$ as an example, students 1, 5, and 6 build team 1, and students 3, 8, and 9 build team 2, which attend ICPC, and students 4, 5, and 9 build team 3, which attend CCSCDC.

TABLE III
Parameter Settings

Parameter	Symbol	Value
Number of particle	N_p	40
Number of iteration	N_g	1000
Weight factor in Eq. (18)	$\omega_{s_{ta}}$ $\omega_{s_{pa}}$ $\omega_{s_{ce}}$ $\omega_{s_{ae}}$	0.25
Weight factor in Eq. (18)	$\omega_{s_{la}}$ $\omega_{s_{ca}}$	0.5

V. Performance Evaluation

We will first show the illustration results about the team building method, and then show the student utility and team utility through simulation experiments and real experiments, respectively. Finally, we further demonstrate our proposed MOPSO-DDI compared with other multiple objective optimization algorithms.

A. Parameter Setting and Illustration Results

Based on the results from plenty of experiments and to calculate easily, some important simulation parameters are set as shown in Table III, and their values are determined based on the best experimental results from multiple experiments. The CPU is Intel Core i5-6300HQ at 2.30 GHz 2.30 GHz, and the memory capacity is 8 GB.

Recall the illustration example in Section II-C, we have built two teams for ICPC and one team for CCSCDC from the ten students based on their properties with MOPSO-DDI. The three built teams are shown in Fig. 3(a). We can further choose any one team to view the details of the members, as shown in Fig. 3(b), and we use radar map to show the six attributes for each chosen student.

B. Simulation Experiments

1) Experimental Settings: Based on the built platform, we collect the data of students in our university. There are 88 college students in the major of computer science and technology. We simulate the condition of attending the competition in an academic year. There are five related competitions in an academic year, and each competition can accommodate three teams, which consist of five students, respectively. These competitions are sequenced by the required energy from the lowest to the highest. The student attends a competition and gets a utility improvement in some corresponding attribute.

We select students with first-class theoretical ability randomly (FCTA), randomly select students with first-class practical ability (FCPA), and randomly select students with first-class theoretical and practical ability (FCTPA) to compare the performance of three team methods with our proposed CTB method. We set the threshold of the first class at 0.75 to assure that there are enough students to build teams. Therefore, there are 57% of students having the first-class theoretical ability, 47% of students having the first-class practical ability, and 38% having the first-class theoretical and practical one, respectively.

We use the following metrics to do performance evaluations and comparisons.

1) *Average team utility (ATU):* The average value of utilities of all teams involved in current competitions.

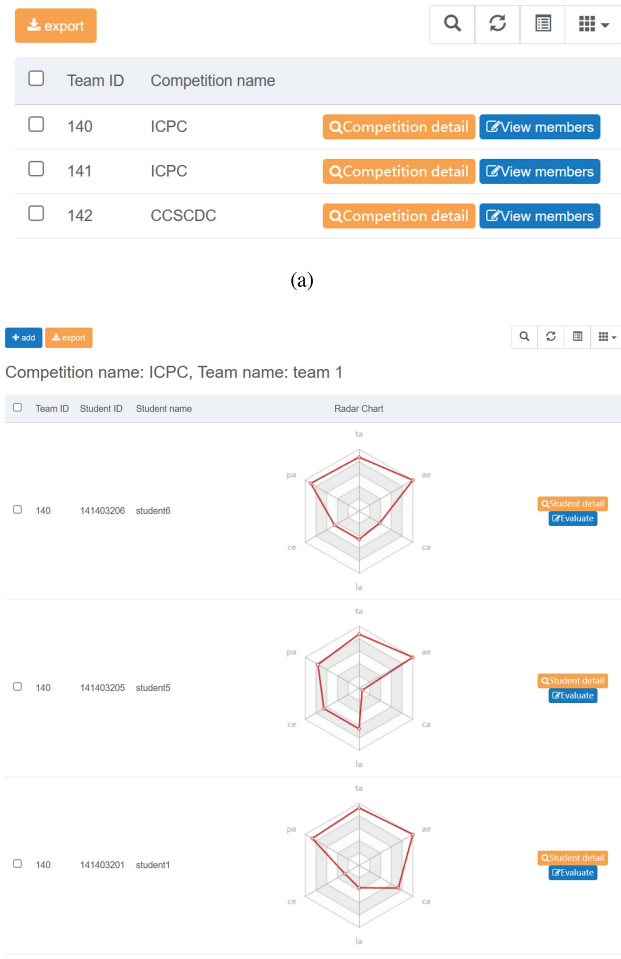

Fig. 3. Illustration results for example. (a) Built teams for example. (b) Details of students in team1.

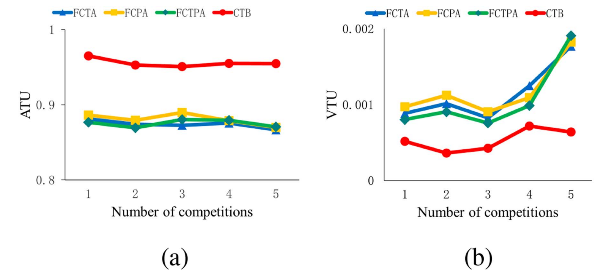

Fig. 4. Team utility comparison. (a) ATU. (b) VTU.

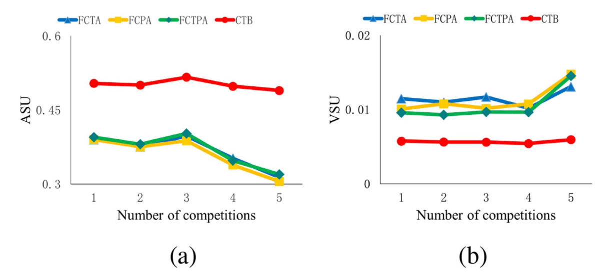

Fig. 5. Student utility comparison. (a) ASU. (b) VSU.

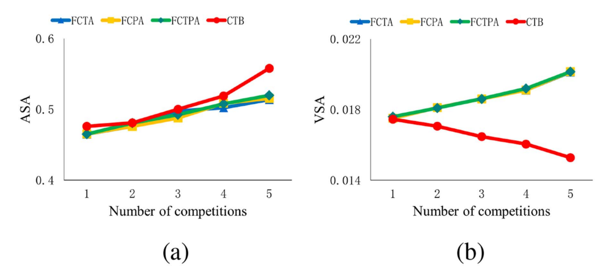

Fig. 6. Student ability comparison. (a) ASA. (b) VSA.

2) *Variance of team utility (VTU):* The variance value of utilities of all teams involved in current competitions.
3) *Average student utility (ASU):* The average value of utilities of all students involved in current competitions.
4) *Variance of student utility (VSU):* The variance value of utilities of all students involved in current competitions.
5) *Average student ability (ASA):* The average value of all students' ability.
6) *Variance of student ability (VSA):* The variance value of all students' ability.

These metrics reflect the improvement of students in building teams and attending competitions. *ATU* and *VTU* measure the profit of teams, and the others measure the profit of students.

2) Results Analysis: Fig. 4 compares the team utility of the four team building methods. It can be clearly seen that the ATU of CTB is much higher than the other three methods. This is because CTB considers the utility value of each student's six attributes in building team for the current competitions, which tries to average the utilities of all built teams. The other three methods are to select top students in different aspects to make their ATU curves similar. With the increment of the number of competitions, the advantage of CTB over the other three methods becomes larger and larger. Especially, the *VTU*s of FCTA, FCPA, and FCTPA increase sharply, since the choice of first-class students will limit the scope of students, and the difference among these built teams will increase.

As shown in Fig. 5(a), FCTA, FCPA, and FCTPA decreased with the increase in the number of competitions, while the *ASU* curve of CTB remained stable, because CTB comprehensively considers a student's six attributes to build teams and other team building methods only choose first-class students from different viewpoints. If a student is chosen by multiple competitions, his energy will be consumed too much. Therefore, when the number of competitions increases to 5, their performance degrades sharply, namely, *ASU* decreases and *VSU* increases.

Finally, we test the whole students' ability to verify the influence of team building methods on the whole students. A student's ability is defined as the cumulative sum of his six properties. As shown in Fig. 6, the *ASA* of all methods increases smoothly, and when there are five competitions, the increment extent of CTB remains stable, and that of other methods decreases because they cannot provide more appropriate students.

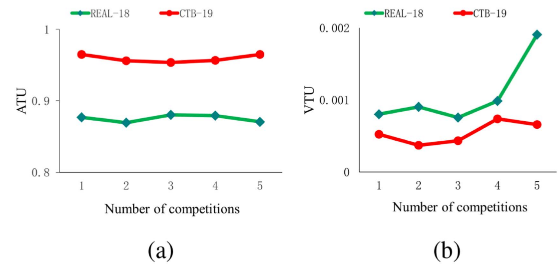

Fig. 7. Team utility comparison. (a) ATU. (b) VTU.

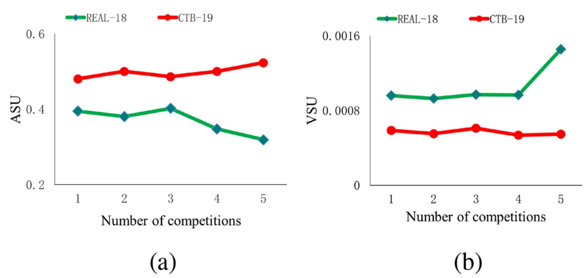

Fig. 8. Student utility comparison. (a) ASU. (b) VSU.

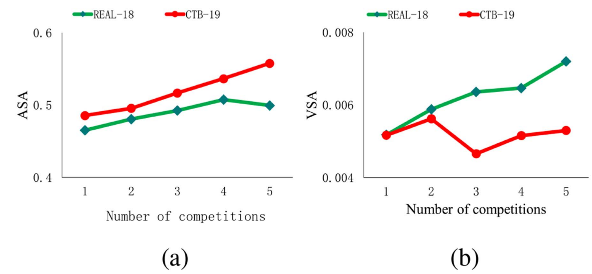

Fig. 9. Student ability comparison. (a) ASA. (b) VSA.

It is worth noting that with the increment of the number of competitions, *VSA* of CTB keeps decreasing, and that of other methods increases. The reason is that CTB considers all students and tries to reduce the difference among students, while the other methods only consider the first-class students and increase in the difference of the whole students.

C. Real Experiments

1) Experimental Settings: In a real experiment, there are 88 and 80 college students from two grades 2018 and 2019, and in the major of computer science and technology. Students in the class of 2018 are a control group, namely, REAL-18, and they use the team building method without reference to CTB results to build a team in the real environment. A total of 80 students in the class of 2019 are experience group, and they build a team with CTB results as a reference, namely, CTB-19. There are five major-related competitions in an academic year, and each competition sets the number of teams and team members based on their actual participation; teams and members of each competition are counted according to actual participation. The parameters and metrics are the same.

2) Results Analysis: As shown in Fig. 7, CTB-19 achieves a higher *ATU* and a lower *VTU* than REAL-18 with different numbers of competitions. Moreover, when the number of competitions goes to 5, REAL-18 gets a sharp increment in *VTU*, while CBT-19 still gets a similar and stable one, because it takes the utilities of teams into full consideration and REAL-18 just builds teams with more randomness.

Fig. 8 compares the changes in student utilities between two grades in the real environment. As shown in Fig. 8(a), with the increment of the number of competitions, the *ASU* of REAL-18 shows a descending trend and that of CTB-19 demonstrates an increased one. While, as shown in Fig. 8(b), with the increment of the number of competitions, REAL-18 and CTB-19 show a stable and similar performance in terms of *VSU*, except for a sharp increment in REAL-18 when there are five competitions. The main reason is that CTB-19 considers the utility of students in building teams. In general, compared with the REAL-18 method, the CTB-19 method improves *ASU* and tries to lessen *VSU*.

Finally, we test the whole students' ability to verify the influence of team building methods on whole students. As shown in Fig. 9(a), with the increase in the number of competitions, the *ASA* of CTB-19 is higher than REAL-18. As shown in Fig. 9(b), with the increase in the number of competitions, the *VSA* of REAL-18 shows an overall upward trend, which is higher than CTB-19. Fig. 9 compares the changes of *ASA* and *VSA* between the REAL-18 and the CTB-19 in the real environment. In general, compared with the REAL-18, the CTB-19 method improves students' average ability and reduces the gap between students in terms of students' ability.

D. Performance Comparison With Other Algorithms

We finally compare the performance of our proposed MOPSO-DDI with other optimization algorithms, namely, NSGA-II, MOPSO, and SS-MOPSO. Recall that MOPSO [16] and SS-MOPSO [21] are closely related work to MOPSO-DDI. MOPSO is the basic multiobjective optimization algorithm and the basis of MOPSO-DDI, and SS-MOPSO is also an improvement of MOPSO and forms stable niches, which are optimized to search and maintain Pareto-optimal solutions in parallel. NSGA-II [27] is a basic and popular nondominated sorting genetic algorithm. To ensure fairness, we use the same population size and the number of iterations in the four compared algorithms.

We take HV and IGD as performance metrics [36]. HV metric evaluates the performance of an algorithm by computing the size of the area covered by the nondominated solution set in the objective space. It has good mathematical properties in theory, namely, among all univariate metrics, HV is a method that can judge that nondominated solution set X is not worse than another nondominated solution set, and it can also maintain consistency with Pareto dominance. Generally, a larger HV value indicates a higher quality of the solution set, and better convergence and diversity of the solution set.

Table IV shows that MOPSO-DDI wins the other three compared algorithms in all the best, the average, and the worst

TABLE IV
COMPARISON OF FOUR ALGORITHMS IN HV

	NSGA-II	MOPSO	SS-MOPSO	MOPSO-DDI
Best	0.45372	0.53459	0.552790	0.573164
Average	0.43250	0.52406	0.549137	0.561611
Worst	0.40628	0.51231	0.546372	0.558242

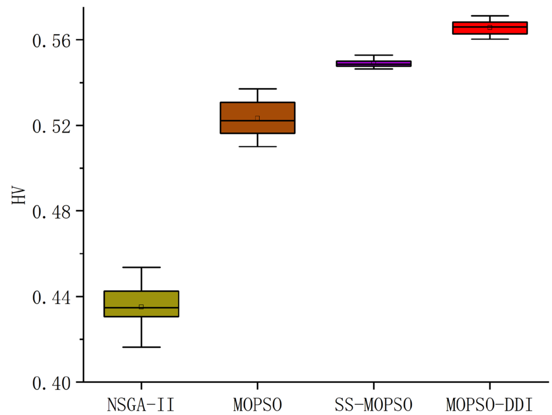

Fig. 10. Boxplot for HV values of four compared algorithms.

TABLE V
COMPARISON OF FOUR ALGORITHMS IN IGD

	NSGA-II	MOPSO	SS-MOPSO	MOPSO-DDI
Average	6.18E-02	2.76E-02	2.35E-02	2.21E-02
Variance	3.87E-04	7.83E-05	1.55E-07	1.42E-07

HV, suggesting that MOPSO-DDI exhibits better convergence and diversity. A more clear comparison is shown in Fig. 10. MOPSO-DDI has a higher HV index, indicating that it is better in terms of convergence and diversity. SS-MOPSO achieves the second best performance for its searching and maintaining Pareto-optimal solution in parallel. As two basic evolutionary algorithms, MOPSO and NSGA-II get the third and fourth best one, respectively. However, SS-MOPSO has the lowest distribution range of HV value, which indicates that it has the best stability and all got solutions have more similarities for its parallel operation, while MOPSO-DDI has the second one, which shows that it has better stability than MOPSO and NSGA-II.

IGD is used to evaluate the degree of approximation of non-dominated solutions to the true Pareto-optimal solution set in multiobjective optimization algorithms. A smaller IGD value indicates a higher recommendation accuracy, as well as better convergence and distribution performance of the algorithm. The IGD index can be calculated as follows:

$$\mathrm{IGD}(P, Q) = \frac{1}{|P|} \sum_{v \in P} d(v, Q). \tag{24}$$

Here, P is the optimal solution set on the real Pareto surface, $|P|$ is the number of individuals in the optimal solution set on the real Pareto surface, Q is the nondominated solution set obtained by the algorithm, and $d(v, Q)$ is the minimum Euclidean distance from individual v to population Q. Table V shows that MOPSO-DDI achieves the best average and variance of IGD than other three compared algorithms, which means that it outperforms them in terms of convergence and distribution uniformity, while SS-MOPSO, MOPSO, and NSGA-II get the second, third, and fourth performances in IGD, respectively.

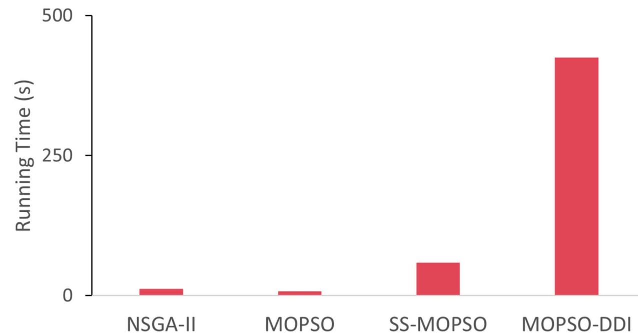

Fig. 11. Running time of four compared algorithms.

Finally, we compare the running time of the four optimization algorithms. As shown in Fig. 11, MOPSO costs the least time for its simplicity, NSGA-II needs a little more time than MOPSO, and SS-MOPSO spends more time than MOSPO and NSGA-II for its parallel operation, while MOPSO-DDI consumes the longest time for its dimension by dimension search. Considering its excellent performance in HV and IGD, the necessary overhead is worth. Moreover, the student team building is not a real-time task, and we have enough time to choose appropriate students to build teams to attend current academic competitions.

VI. CONCLUSION

This article compared traditional team building methods, which just choose first-class students to attend competitions. We build a Big Data platform to collect and analyze students' various behavior data, and then develop a comprehensive model for students and competitions based on six different properties, respectively. Furthermore, a utility function is defined for each property of students to denote their gain in this property for building a team to attend a competition. Based on the above analysis, we formulate competition-oriented team building as a multiple-objective optimization problem, which tries to maximize the utilities of all built teams and minimize the utility difference among all involved students simultaneously. Moreover, we design MOPSO-DDI, which updates the positions of particles with dimension by dimension improvement in MOPSO to solve it. The detailed search improves the local search ability of MOPSO and avoids to trap in local optima. The simulation results demonstrate the performance of our team building method, in terms of student utility and team utility. Moreover, it decreases the difference in the whole students' abilities. MOSPO-DDI also shows better performance, in terms of HV and IGD than other optimization algorithms.

The proposed competition-oriented student team building method has two important roles. On the one hand, it chooses the more suitable students to build different teams to attend various competitions, which can bring better competition results, enhance students' ability, and promote the development of major and university. On the other hand, by considering education fairness, the study does not only focus on first-class students, but also takes into account overall involved students, which has an important meaning in current high education. Both the good competition results and the consideration of education fairness can exert a far-reaching influence on the students, the universities, and then the whole of higher education.

ACKNOWLEDGMENT

The authors would like to thank Prof. Min Huang in Northeastern University, China, for her very valuable suggestions.

REFERENCES

[1] Y. B. Ma and Z. Bai, "Research and practice mode of China's innovation and entrepreneurship education," *Tsinghua. J. Educ.*, vol. 14, no. 6, pp. 99–103, Dec. 2015.

[2] L. Wang, D. W. Hu, and M. Yan, "On interaction of discipline contest and practice teaching for contemporary college students," *J. Higher. Educ.*, vol. 4, no. 17, pp. 79–85, Sep. 2018.

[3] F. Zeng, C. Li, and J. Li, "Promote the reform of analytical chemistry teaching mode with "Internet discipline competition" as the guide," in *Proc. Int. Conf. Comput. Inf. Process. Adv. Educ.*, Paris, France, 2021, pp. 69–72.

[4] C. W. Nam and R. D. Zellner, "The relative effects of positive interdependence and group processing on student achievement and attitude in online cooperative learning," *Comput. Educ.*, vol. 56, no. 3, pp. 680–688, Jan. 2011.

[5] M. G. Burke, J. D. Carter, and A. W. Hughey, "The use of case study competitions to prepare students for the world of work," *Ind. High. Educ.*, vol. 27, no. 3, pp. 157–162, Jun. 2013.

[6] M. Keane, "Internet China: Unleashing the innovative nation strategy," *Int. J. Cult. Creative. Ind.*, vol. 3, no. 2, pp. 68–74, Mar. 2016.

[7] H. Chen and J. Rao, "Research on the computer basics experiment teaching with synergistic effect and team competition," *Comput. Era.*, vol. 17, no. 4, pp. 75–78, Apr. 2018.

[8] S. Deng, X. Yuan, and D. U. Xuan, "Research on the selection mechanism of academic competition in universities," *Guide. Sci. Educ.*, vol. 10, no. 13, pp. 64–68, May 2019.

[9] Y. Hua, Q. Liu, K. Hao, and Y. Jin, "A survey of evolutionary algorithms for multi-objective optimization problems with irregular Pareto fronts," *IEEE/CAA J. Automatica Sinica*, vol. 8, no. 2, pp. 303–318, Feb. 2021.

[10] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *IEEE Access*, vol. 10, pp. 10031–10061, 2022.

[11] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," *Math. Problems Eng.*, vol. 2015, Oct. 2015, Art. no. 931256.

[12] Z. Zhan, J. Zhang, Y. Li, and Y. Shi, "Orthogonal learning particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 6, pp. 832–847, Dec. 2011.

[13] R. Santos, G. Borges, A. D. F. dos Santos, M. Silva, C. Sales, and J. C. W. A. Costa, "A semi-autonomous particle swarm optimizer based on gradient information and diversity control for global optimization," *Appl. Soft Comput.*, vol. 69, pp. 330–343, Apr. 2018..

[14] S. Molaei, H. Moazen, S. Najjar-Ghabel, and L. Farzinvash, "Particle swarm optimization with an enhanced learning strategy and crossover operator," *Knowl. Based Syst.*, vol. 215, pp. 1–21, Mar. 2021.

[15] K. Taha, "Methods that optimize multi-objective problems: A survey and experimental evaluation," *IEEE Access*, vol. 8, pp. 80855–80878, 2020.

[16] C. A. Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proc. Congr. Evol. Comput.*, Honolulu, HI, USA, Aug. 2022, pp. 1051–1056.

[17] N. Padhye, J. Branke, and S. Mostaghim, "Empirical comparison of MOPSO methods - Guide selection and diversity preservation," in *Proc. IEEE Congress Evol. Comput.*, 2009, pp. 1–8.

[18] S. Lalwani, S. Singhal, R. Kumar, and N. Gupta, "A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm," *Trans. Combinatorics.*, vol. 2, no. 1, pp. 39–101, Aug. 2013.

[19] W. Hu and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," *IEEE Trans. Evol.*, vol. 19, no. 1, pp. 1–18, Feb. 2015.

[20] A. Mahmoud, X. Yuan, M. Kheimi, M. A. Almadani, T. Hajilounezhad, and Y. Yuan, "An improved multi-objective particle swarm optimization with TOPSIS and fuzzy logic for optimizing trapezoidal Labyrinth Weir," *IEEE Access*, vol. 9, pp. 25458–25472, 2021.

[21] B. Qu, C. Li, J. Liang, L. Yan, K. Yu, and Y. Zhu, "A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems," *Appl. Soft Comput. J.*, vol. 86, pp. 1–13, Jan. 2020.

[22] Q. Lin et al., "Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems," *IEEE Trans. Evol.*, vol. 22, no. 1, pp. 32–46, Feb. 2018.

[23] Z. Liu, S. Liu, L. Liu, J. Sun, X. Peng, and T. Wang, "Sentiment recognition of online course reviews using multi-swarm optimization-based selected features," *Neurocomputing*, vol. 185, pp. 11–20, Apr. 2016.

[24] S. Zdiri, J. Chrouta, and A. Zaafouri, "Cooperative multi-swarm particle swarm optimization based on adaptive and time-varying inertia weights," in *2021 IEEE 2nd Int. Conf. Signal Control Commun. (SCC)*, 2021, pp. 200–207.

[25] X. Xia, L. Gui, and Z. Zhan, "A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful detecting," *Appl. Soft Comput.*, vol. 67, pp. 126–140, Jun. 2018.

[26] W. Han, H. Li, M. Gong, J. Li, Y. Liu, and Z. Wang, "Multi-swarm particle swarm optimization based on CUDA for sparse reconstruction," *Swarm Evol. Comput.*, vol. 75, Dec. 2022, Art. no. 101153.

[27] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[28] P. Jangir and N. Jangir, "Non-dominated sorting whale optimization algorithm (NSWOA): A multi-objective optimization algorithm for solving engineering design problems," *Glob. J. Researches Eng.: Elect. Electron. Eng.*, vol. 17, no. 4, pp. 1–29, 2017.

[29] Y. Fan and W. B. Frederick, "Study on student performance estimation, student progress analysis, and student potential prediction based on data mining," *Comput. Educ.*, vol. 123, pp. 97–108, Aug. 2018.

[30] P. R. Drake, "Using the analytic hierarchy process in engineering education," *Int. J. Eng. Educ.*, vol. 14, no. 3, pp. 191–196, Jan. 1998.

[31] V. Tsyganok, "Investigation of the aggregation effectiveness of expert estimates obtained by the pairwise comparison method," *Math. Comput. Model.*, vol. 52, no. 3/4, pp. 538–544, Mar. 2010.

[32] G. Yang and L. Wang, "Quality evaluation algorithm based on AHP and utility theory," in *Proc. Int. Conf. Syst. Inf.*, Shanghai, China, 2019, pp. 573–578.

[33] X. Li, "Research on education process fairness from the perspective of differential education," in *Proc. Int. Conf. Educ. Lang. Art. Intercult.*, 2017, pp. 95–98.

[34] M. Mavrovouniotis, C. Li, and S. Yang, "A survey of swarm intelligence for dynamic optimization: Algorithms and applications," *Swarm Evol. Comput.*, vol. 33, pp. 1–17, Apr. 2017.

[35] D. Qu, S. Liu, D. Zhang, J. Wang, and C. Gao, "Teaching-learning based optimization algorithm based on course by course improvement," in *Proc. 11th Int. Conf. Comput. Intell. Secur.*, 2018, pp. 48–52.

[36] K. Li, R. Wang, T. Zhang, and H. Ishibuchi, "Evolutionary many-objective optimization: A comparative study of the state-of-the-art," *IEEE Access*, vol. 6, pp. 26194–26214, 2018.

Dapeng Qu received the B.S. degree in applied mathematics and the M.S. degree in computer science from Central South University, Changsha, China, in 2003 and 2006, respectively, and the Ph.D. degree in computer science from Northeastern University, Shenyang, China, in 2012.

He is currently an Associate Professor and an Advisor for Master's candidates with the College of Information, Liaoning University, China. He has authored or coauthored more than 50 academic articles published in several journals and conference proceedings. His research interests include future Internet architecture and computer networks.

Ruiduo Li is currently working toward the M.S. degree in computer science with Liaoning University, Shenyang, China.

His research focuses on Internet education.

Tianqi Yang received the M.S. degree in computer science from Liaoning University, Shenyang, China, in 2023.

He is currently an Engineer with Chengfang Fin-Tech Company, Ltd., Beijing, China. His research focuses on Internet education.

Songlin Wu received the M.S. degree in computer science from Liaoning University, Shenyang, China, in 2023.

He is currently an Engineer with the Zhejiang Water Conservancy Information and Publicity Center, Hangzhou, China. His research focuses on Internet education.

Yan Pan received the M.S. degree in computer science from Liaoning University, Shenyang, China, in 2022.

Her research focuses on Internet education.

Xingwei Wang received the B.S., M.S., and Ph.D. degrees in computer science from Northeastern University, Shenyang, China, in 1989, 1992, and 1998, respectively.

He is currently a Professor with the College of Computer Science and Engineering, Northeastern University. He has authored more than 100 journal articles, books and book chapters, and refereed conference papers. His research interests include cloud computing and the future Internet.

Dr. Wang was the recipient of several best paper awards.

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 currently a SUNY Distinguished Professor with the State University of New York, New Paltz, NY, USA, and a National Distinguished Professor with Hunan University, Changsha, China. He has authored or coauthored more than 960 journal articles, book chapters, and refereed conference papers, and holds nearly 70 patents announced or authorized by the Chinese National Intellectual Property Administration.

Dr. Li was the recipient of several best paper awards from international conferences, including PDPTA-1996, NAECON-1997, IPDPS-2000, ISPA-2016, NPC-2019, ISPA-2019, and CPSCom-2022, and 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 twenty consecutive years in 2017, the Distinguished Alumnus Award from the Computer Science Department at the University of Houston in 2018, 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 was a winner of the IEEE Region 1 Technological Innovation Award (Academic) in 2023. He is among the world's top five most influential scientists in parallel and distributed computing in terms of single-year and career-long impacts based on a composite indicator of the Scopus citation database. He is a Member of the SUNY Distinguished Academy. He is an AAAS Fellow, an AAIA Fellow, and an ACIS Founding Fellow. He is a Member of Academia Europaea (Academician of the Academy of Europe).