Resolution to Educational Group Formation Problem Based on Improved Particle Swarm Optimization Using Fuzzy Knowledge

Hasan Bikhtiyar
Higher Institute of Computer Science and Communication Technologies
University of Sousse
Sousse, Tunisia
bikhtiyar.hasan85@gmail.com

Amine Boufaied
Higher Institute of Computer Science and Communication Technologies
University of Sousse
Sousse, Tunisia
Boufaied.amine@gmail.com

***Abstract*— In the educational context, instructors usually partition students into collaborative learning teams to perform collaborative learning tasks. Indeed, one of the grouping criteria most utilized by instructors is based on the students' roles and on forming similar teams according to the roles of their members, which is costly and complex. This paper addresses the optimization problem of forming automatic learning teams by minimizing the knowledge-difference cost among formed teams. The knowledge index of each group depends on the Belbin roles of their students' members in the form of a sum of students' fuzzy rating indexes. The proposed algorithm is called improved particle swarm optimization with multi-parent order crossover (IPSOMPOX). The multi-parent order crossover is used in IPSOMPOX in order to investigate new solutions in the search space and to accelerate the convergence of the proposed algorithm to the best global solution. To evaluate the performance of the proposed algorithm, we apply it to several different experiments with different numbers of teams and students. The results demonstrate the superiority of our proposed performance over the standard PSO.**

Keywords*— *Particle Swarm Optimization; Learning group formation problem; Belbin roles; Multi-Parent Order Crossover; Fuzzy Classification.

I. INTRODUCTION

Nowadays, there is an increasing interest in developing teamwork skills [1] [2]. This growing interest is motivated by its effectiveness and the fact that, in labor contexts, enterprises organize their employees in teams to carry out complex projects [3]. In fact, problems relating to team formation are common across many industrial sectors, including education, sport and general business. It is beyond manual implementation to build near-optimal teams as pools of candidates grow [4]. The team is comparable to the human body, like various organs collaborate to make things happen, and the various individuals collaborate daily to bring success to the project [5].

Recently, an appreciable number of researchers have attempted to solve the problem of team formation without leaders. In 1995, Kennedy and Eberhart designed the particle swarm optimization (PSO) in observations modeling the "social behavior" of schools of birds or fish searching for their nest or food. Kennedy and Eberhart expressed interest in Frank Heppner's model (among the various available models). In [6], a modified PSO algorithm is proposed for solving a team formation optimization problem by minimizing the communication cost among experts. The proposed algorithm is called Improved Particle Swarm Optimization with New Swap Operator (IPSONSO). In IPSONSO, a new swap operator is applied within particle swarm optimization to ensure the consistency of the capabilities and the skills to perform the required project.

In the educational context, one of the grouping criteria most utilized by instructors is based on taking into account the students' roles and forming teams according to the roles of their members [7]. A role is how a person tends to behave, contribute and interrelate with others throughout a collaborative task. Several team role models proposed in the literature recommend this grouping criterion [8]. Students belonging to Belbin teams acknowledge that they attend classes more regularly, need less time to study outside the classes and show a higher interest in the subject at the end of the course. This team forming method allows students to identify their own strengths and weaknesses and understand the roles (behaviors) of their teammates and their strengths and weaknesses [9].

In this paper, we propose a new method addressing the optimization problem of forming automatically working teams and making the teams as similar as possible to each other across the knowledge indexes to get a homogenous working rate. These knowledge indexes depend on the skills of their members in the form of a sum of fuzzy rating indexes. The proposed algorithm is named Improved Particle Swarm Optimization with Multi-Parent Order Crossover (IPSOMPOX). The multi-parent order crossover is used in IPSOMPOX in order to investigate new solutions in the search space and to accelerate the convergence of the proposed algorithm to the best global solution.

The rest of this paper is structured as follows. Section II presents the problem statement. Section III provides important details about the optimization algorithm. Our proposed algorithm (IPSOMPOX) is described in Section IV. Section V evaluates the proposal and compares it with traditional PSO. Section VI concludes this paper.

II. PROBLEM STATEMENT

Let set S comprise n collaborators, $S = \{s_1, s_2, \ldots, s_n\}$. We have to partition the n collaborators into a set GP_k of g

 ISBN: 978-1-68558-098-8

teams, $GP_k = \{G_1, G_2, \ldots, G_g\}$, k is a positive integer. Each team G_i, i=1…g, is made up of a z_i number of member collaborators, and each collaborator can only belong to one team.

Regarding team size, collaborators must be divided so that the g teams have a similar number of collaborators each. Specifically, the difference between a team's size and the other teams' size must not exceed one. The values of the terms n and g are known.

Each team G_i has to accomplish a given task (i.e., a project or a part of the same global project), and a set of m skills $R = \{r_1, r_2, \ldots, r_m\}$ represents the abilities of the collaborators to a given task.

The pair (F, R) is called a Fuzzy Soft Set over S, where F: R → P(S) and where P(S) is the set of all fuzzy subsets of a universal set S relative to a $r_i \in R$ [5] [10].

Every skill in this problem is not in crisp nature. Skills are all in a fuzzy nature and are called linguistic variables. The linguistic variables do not receive any numerical values but some words or sentences of information. The linguistic variables of the fuzzy system for project team selection can be classified into four categories: VG (very good), GD (good), FR (fair), and PR (poor) [11].

Here, we consider that each collaborator has an evaluation for each skill. In this sense, we define four evaluations: VG (very good), GD (good), FR (fair), and PR (poor). The evaluation determines the degree to which a collaborator possesses a particular skill naturally. Therefore, the membership value (MV) of a collaborator to a skill can be calculated as follows.

$$MV = [LB + (UB - LB) * (AV /100)] \tag{1}$$

where LB: Lower Bound, UB: Upper Bound and AV: Actual Value.

Let us define $x_1, x_2 \ldots x_j$ as the membership values of each evaluation. Additionally, let $w_1, w_2 \ldots, w_j$ are the weights of the required skills for a given project respectively, $\sum_{i=1}^{j} w_i = 1$, then the fuzzy rating indexes (FRI) are:

$$(FRI)s_i=\Sigma[x_i*w_i],\ i=1 \text{ to } j \tag{2}$$

$$(DFRI)s_i=(FRI)s_i*100 \tag{3}$$

Where $(DFRI)_{si}$ is the defuzzified or crisp value of $(FRI)_{si}$ of the collaborator s_i. The output can be interpreted based on a fuzzy rating index or its defuzzified value (crisp value).

For each group $G_i \in GP_k$, i=1...g, we calculate the Knowledge Index (KI), which is the sum of the fuzzy rating index of its collaborators.

$$KI(G_i) = \sum_{k=1}^{/G_i/} FRI_{S_k} \tag{4}$$

From equation (4), we calculate the average AV of the different KIs.

$$AV(GP_k) = \frac{\sum_{i=1}^{g} KI(G_i)}{g} \tag{5}$$

Then, we calculate the squared difference between each KI and the average. For instance, for the first KI:

$$(KI(G_1) - AV)^2 \tag{6}$$

The squared deviations of each value are then added:

$$\sum_{i=1}^{g} (KI(G_i) - AV)^2 \tag{7}$$

This sum is then divided by the number of KIs to get the variance, i.e.,

$$\frac{\sum_{i=1}^{g} (KI(G_i) - AV)^2}{g} \tag{8}$$

The standard deviation ST of a team's partition GP_k is given with:

$$ST(GP_k) = \sqrt{\frac{\sum_{i=1}^{g} (KI(G_i) - AV)^2}{g}} \tag{9}$$

$ST(GP_k)$ is zero if all the Knowledge Indexes (KI) of teams in the partition GP_k are the same (because each value is equal to the average).

A set of weighted skills forms a given project P. Each collaborator $s_i \in S$ is associated with a fuzzy rating index FRI_{si} relatively to the given project. A set of possible collaborators' partitions achieving P is denoted GP, GP = $GP_1, GP_2, \ldots, GP_m$, where m is a positive integer. The goal is to find a partition with the least knowledge difference cost among teams of collaborators $ST(GP_k)$, $k \in \{1..m\}$ realizing the same given project according to (9).

The team formation problem can be considered as an optimization problem by forming a feasible partition GP* among a set of possible collaborators' partitions with minimum knowledge difference cost among formed teams, and GP* can be obtained by the following:

$$Min_{(GP_k \in GP)}\ ST(GP_k) = \sqrt{\frac{\sum_{i=1}^{g} (KI(G_i) - AV)^2}{g}} \tag{10}$$

subject to

$$\forall s_j \in S,\ \sum_{i=1}^{g} P_{ij} = 1 \tag{11}$$

Where P_{ij} is a binary variable, $P_{ij} = 1$ if collaborator s_j belongs to the team G_i and 0 otherwise. A collaborator belongs to one team among a partition GP_k.

The notations of the team formation problem are summarized in Table 1.

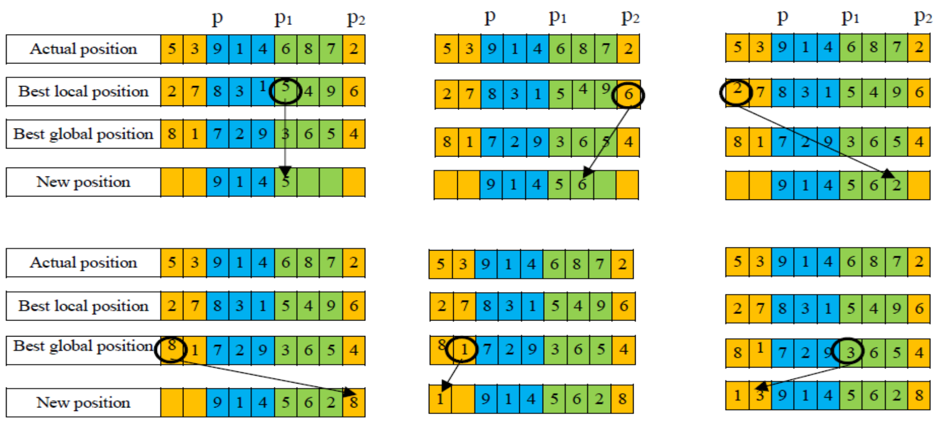

Figure. 1. An example of the implementation of the 3-POX.

TABLE I. NOTATIONS OF TEAM FORMATION PROBLEM

Notation	Definition
S	A set of collaborators
GP_k	A partition of collaborators into g teams
P	A project with weighted skills
R	A set of skills
G_i	collaborators' team
$(FRI)_{si}$	The fuzzy rating index of a collaborator s_i
$KI(G_i)$	Knowledge index of a team G_i

III. OPTIMIZATION ALGORITHM

Each solution in the optimization algorithm population represents a set GP_k of g teams (g is the number of teams), which may be built when the n collaborators in the class are partitioned. Each solution is represented as a list with a length equal to n (i.e., a list with as many positions as students in the class). Specifically, each position p (p = 1…, n) on this list contains a different collaborator (i.e., repeated collaborators are not admitted) (n is the number of collaborators). Besides, each collaborator s_j (j = 1…, n) may be in any position on the list. In short, the list is a permutation of the n collaborators [8].

In this process, the g teams are built, considering the two restrictions as part of the problem. The first restriction is that each student may belong to only one team. While the second restriction holds that the difference between the size of a team and the size of the rest of the teams must not exceed one. The size of the teams is considered to depend on the relationship between the values n and g.

When n>2, there are so many possible permutations to realize from a functional point of view. Therefore, designing an intelligent mechanism to realize a minimum number of permutations is necessary to finally get a collaborator's partition with minimum difference knowledge between its teams. We propose to use a new, improved Particle Swarm Optimization (PSO) method to solve such an optimization problem.

In fact, to solve the stated working group formation problem, we used the Multi-Parent Order Crossover (MPOX). We developed a new, Improved Particle Swarm Optimization with Multi-Parent Order Crossover (IPSOMPOX).

IV. THE IMPROVED PARTICLE SWARM OPTIMIZATION WITH MULTI-PARENT ORDER CROSSOVER (IPSOMPOX)

IPSOMPOX starts with an initial population containing a specific number of feasible particles. Each particle consists of a permutation of the n collaborators and is a list. A random method has been designed to generate each of the particles of this population. This kind of method guarantees a good level of diversity in the initial population and, therefore, helps prevent the premature convergence of the algorithm. Then, *itmax* iterations are executed to define the content of the positions of the particles. All the swarm particles are updated in each iteration m (m =1…, itmax). The position of a particle is updated using the multi-parent order crossover (MPOX) [12]. In our resolution method, we used three parents: the actual position, the best local position and the best global position.

The following algorithm 1 presents the pseudocode of the 3-POX. Notice that moving through the actual position, the best local position, the best global position, or the new position can be circular) (see Figure 1).

Algorithm 1 (3-parent order crossover (3-POX)):

1. ***Parents selection***: select the actual position, the best local and global positions.
   // Three segments selection
2. ***Crossover points generation***: randomly generate the first crossover points p and calculate p_1 and p_2 to divide the actual position into almost equal three segments' widths. The three segments' widths are equal if the actual position width is divisible by 3.
   // Segment copy
3. For i = p to p_1
   Copy the elements from actual_position[i] into the new position[i].

// Elements' selection

4. Pick up the elements from the best local position, starting at p_1, which do not exist in the new position, and copy them into the new position starting at p_1 until reaching p_2.
5. Pick up the elements from the best global position, starting at p_2, which does not exist in the new position, and copy them into the new position starting at p_2 until reaching p.

V. SIMULATIONS AND PERFORMANCE EVALUATION

In this section, we study the performance of the IPSOMPOX algorithm using a numerical simulation of several student's permutations in order to verify the effectiveness of our team formation approach. In fact, we conducted a series of experiments to evaluate the performance of our proposed algorithm.

A. Preliminaries

We started by creating a collaborator's vector noted *collaborators* whose size is the number of collaborators noted *nCollaborators*. The value of *nCollaborators* is initialized at the start of the simulation execution. Each component of the collaborator's vector is a student defined by rank and fuzzy rating index (FRI).

Knowing the number of groups *g*, we calculate the number of collaborators per group and create a group vector noted *groups*. The *groups* vector, whose size is the number of collaborators, contains the instances of groups.

The MPSO (3-POX) is applied when the number of collaborators exceeds 2. It is noted that the variable containing the maximum number of iterations of *MaxIt*. We also chose the population number of the swarm equal to *nPop*. Thus, a random population of *nPop* particles is created (These are particles collaborator's permutations). Each particle has a position whose size is *nCollaborators*, where a particle is a possible solution for our optimization problem. The cost of its position is the best position. The cost function returns the standard deviation of the knowledge indexes of all teams, Equation (12).

$$Cost(particle_i.position) = ST(KI(GP_k), k=1..g, in\ particle_i)$$

$$i = 1..\ nPop \qquad (12)$$

When IPSOMPSO is launched, the positions of the different particles are updated, and their costs are calculated, as shown previously. Also, the best personal solution and the best global solution are updated. The particles are moving and approaching the optimal solution from iteration to iteration. Once the stop condition is verified, execution stops, and we get a student's partition into g groups with the minimum standard deviation between the group's knowledge indexes.

B. Evaluation

We conducted a series of experiments to evaluate the performance of our proposed algorithm.

In the first simulation, we considered 50 students to partition into nine learning groups. We got the nine weights w_i of the Belbin roles relatively to the project P to accomplish from the instructor (i.e., P = {w_1*PL, w_2*RI, w_3*CO, w_4*SH, w_5*ME, w_6*TW, w_7*IM, w_8*FI, w_9*SP}), $\sum_{i=1}^{9} w_i = 1$. With the results of the Belbin Team Role Self-Perception Inventory (BTRSPI), we obtain Table II, showing the FRI of each student.

TABLE II. THE FRI OF EACH STUDENT

Student	FRI	Student	FRI	Student	FRI
1	0.0966	18	0.4769	35	0.8325
2	0.2654	19	0.1296	36	0.1751
3	0.7919	20	0.2281	37	0.8798
4	0.9369	21	0.1811	38	0.2796
5	0.5683	22	0.6720	39	0.8495
6	0.9380	23	0.3258	40	0.8067
7	0.5972	24	0.7906	41	0.1891
8	0.9880	25	0.5460	42	0.9786
9	0.7623	26	0.2306	43	0.2537
10	0.1856	27	0.6868	44	0.7785
11	0.5168	28	0.0491	45	0.8108
12	0.6402	29	0.5726	46	0.4980
13	0.8690	30	0.9938	47	0.9157
14	0.6001	31	0.3630	48	0.1010
15	0.9997	32	0.5587	49	0.1358
16	0.1292	33	0.9273	50	0.8583
17	0.2934	34	0.2599		

Table III shows le number of students in each group.

TABLE III. NUMBER OF STUDENTS IN EACH GROUP

Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9
6	6	6	6	6	5	5	5	5

Once the execution of the IPSOMPOX is finished, it is possible to calculate the partition of the students (as listed in Table IV). Partition of students into groups.

TABLE IV. PARTITION OF STUDENTS IN EACH GROUP

Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	Group 8	Group 9
36	4	42	46	9	50	47	6	5
29	48	28	26	41	8	44	25	15
2	22	37	17	33	12	11	21	49
40	16	19	10	1	18	43	3	20
32	35	24	30	13	38	27	14	45
7	34	23	39	31				

Accordingly, as listed in Table V, we calculate the KI of each group G_k.

TABLE V. KNOWLEDGE INDEX OF EACH GROUP

KI(G_1)	KI(G_2)	KI(G_3)	KI(G_4)	KI(G_5)	KI(G_6)	KI(G_7)	KI(G_8)	KI(G_9)
2.9758	2.9314	3.1534	3.0507	3.2073	3.2429	3.1515	3.0572	2.7427

The minimal standard deviation we got is equal to 0,1478, Figure 2.

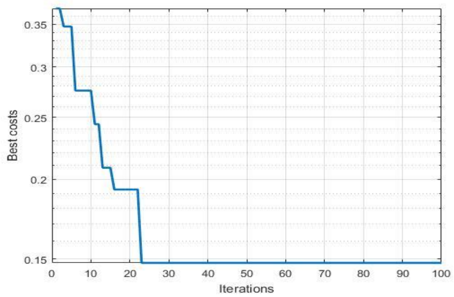

Figure 2. The Best Costs of the example of Table II.

Table VII presents the costing intervals of the knowledge difference costs for three experiments on randomly generated data. The results of IPSOMPOX decrease iteratively to the number of iterations than PSO, achieving better performance ranging from 31% in the fourth iteration to 14% in the last iteration for experiment 1. In comparison, the percentage of the improved results ranged from 2.4% in the second iteration to 16.7% in the fourth iteration when compared with PSO in experiment 2. Also, the results of IPSOMPOX are better and more efficient than PSO, with knowledge difference cost going down from 8.3% better performance in the tenth iteration to 56.4% in the third

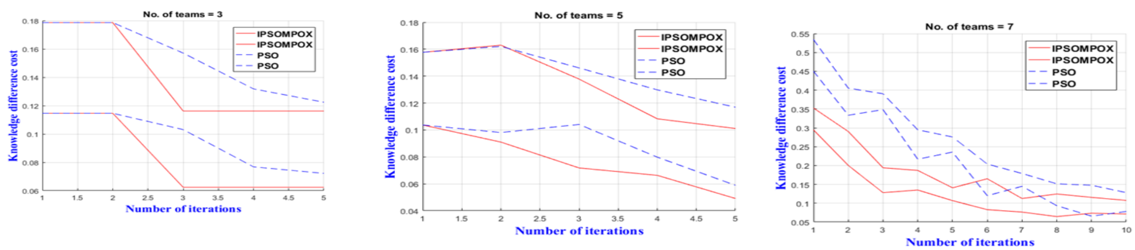

Figure.3. Comparison between PSO and IPSOMPOX on random data.

In a second simulation, three experiments are performed on the random dataset with different teams and student numbers to evaluate the performance of the proposed algorithm that focuses on iteratively minimizing the knowledge difference cost among teams. The average results are taken over 50 runs. The parameters are reported in Table VI. The proposed algorithm is compared with the standard PSO to verify its efficiency.

TABLE VI. PARAMETER SETTING

Exp. No.	No. of iterations	No. of the initial population	No. of teams	No. of students
1	5	5	3	10
2	5	10	5	20
3	10	20	7	30

In Table VII, the average ($\bar{x}$), the standard deviation (s) and the mean (μ) of the results sample are reported over 50 random runs. The mean μ is also reported with a confidence level of 95%.

The performance (%) between the compared algorithms can be computed in Eq. (13).

$$\text{Performance}(\%) = \frac{(\mu_{(PSO)} - \mu_{(IPSOMPOX)})}{\mu_{(PSO)}} * 100 \quad (13)$$

where $\mu_{(PSO)}$ and $\mu_{(IPSOMPOX)}$ are the mean results obtained from SPSO and IPSOMPOX algorithms, respectively.

iteration of experiment 3. In Figure 3, the costing intervals of the proposed algorithm are presented against the standard PSO for different team numbers by plotting the number of iterations against the costing intervals on knowledge difference costs. The results in Figure 3 show that the proposed algorithm is better than the standard PSO.

TABLE VII. COMPARISON BETWEEN PSO AND IPSOMPOX ON RANDOM DATA

Exp. No.	Iteration no.		PSO	IPSOMPOX
1	1	$\bar{x}$	0,1488	0,1488
		s	0,0125	0,0125
		μ ± 2,5*σ	0,1466±0,032	0,1466±0,032
	2	$\bar{x}$	0,1488	0,1488
		s	0,0125	0,0125
		μ ± 2,5*σ	0,1466±0,032	0,1466±0,032
	3	$\bar{x}$	0,1321	0,0815
		s	0,0114	0,0109
		μ ± 2,5*σ	0,1301±0,027	0,0893±0,0269
	4	$\bar{x}$	0,1190	0,0815
		s	0,0108	0,0109
		μ ± 2,5*σ	0,1043±0,0275	0,0893±0,0269
	5	$\bar{x}$	0,0935	0,0815
		s	0,0099	0,0109
		μ ± 2,5*σ	0,0973±0,0251	0,0893±0,0269
2	1	μ ± 2,5*σ	0,1307±0,027	0,1307±0,027
	2	μ ± 2,5*σ	0,1301±0,032	0,1270±0,036
	3	μ ± 2,5*σ	0,1251±0,021	0,1048±0,033
	4	μ ± 2,5*σ	0,1048±0,025	0,0873±0,021
	5	μ ± 2,5*σ	0,0879±0,029	0,0751±0,026

3	1	μ ± 2,5*σ	0,4923±0,042	0,3235±0,029
	2	μ ± 2,5*σ	0,3693±0,036	0,2453±0,045
	3	μ ± 2,5*σ	0,3693±0,021	0,1610±0,033
	4	μ ± 2,5*σ	0,2557±0,039	0,1610±0,026
	5	μ ± 2,5*σ	0,2557±0,02	0,1239±0,017
	6	μ ± 2,5*σ	0,1619±0,042	0,1239±0,041
	7	μ ± 2,5*σ	0,1619±0,017	0,0945±0,018
	8	μ ± 2,5*σ	0,1224±0,029	0,0945±0,030
	9	μ ± 2,5*σ	0,1067±0,041	0,0945±0,021
	10	μ ± 2,5*σ	0,1031±0,025	0,0945±0,023

Finally, based on [13], the average processing times (in seconds) reported in Figure 4 over 50 runs for each experiment. We consider the 90 students and vary the number of working groups (10 experiments): 3, 5, 7, 9, 11, 15, 19, 21, 25, 30.

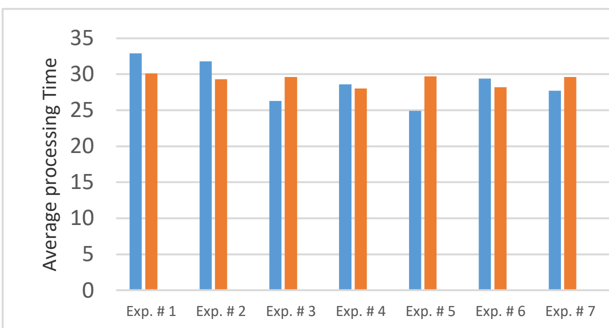

Figure 4. The average processing times (in seconds) of the PSO and IPSOMPOX

As shown in Figure 4, the time for forming a working group using the proposed algorithm IPSOMPOX varies slightly from one experiment to another and is around a mean of 29,4s. Also, the average processing time of PSO has a mean of 28,9s. Thus, the average processing time of PSO is better than that of IPSOMOX with 1,8%. This is due to additional processing time using the multi parent order crossover (MPOX).

VI. Conclusion and futur works

This study investigates a new particle swarm optimization algorithm to solve the team formation problem. The proposed algorithm is called Improved Particle Swarm Optimization with Multi-Parent Order Crossover (IPSOMPOX). In the IPSOMPOX algorithm, exploiting the multi-parent order crossover in the proposed algorithm has accelerated its convergence to the global best solution. The performance of the proposed algorithm is investigated in five experiments with different numbers of teams and students. The results of the proposed algorithm show that it can obtain a promising result in a reasonable time compared to the standard PSO.

As a future work, we propose to compare our method's performance when using the adjacency-based crossover (ABC) or the multi-parent partially mapped crossover (MPPMX) instead of the 3-POX. Also, it is worthwhile to test our proposed algorithm over various benchmark problems of nonlinear mixed integer programming problems.

References

[1] D. Mburasek, and M. Odon, «Development of a tool for team formation in engineering education,» *International Journal Of Engineering And Management Research*, vol. 11, n°16, pp. 62-69, 2021.

[2] O. Zvereva, and E. Milovidova, «Engineering Effective Teams: An Example From Educational Domain,» Chez *International Conference on Applied Mathematics & Computational Science (ICAMCS. NET)*, 2018.

[3] J. Alberola, E. Del Val, V. Sanchez-Anguix, A. Palomares and M. Dolores Teruel «An artificial intelligence tool for heterogeneous team formation in the classroom,» *Knowledge-Based Systems*, vol. 101, pp. 1-14, 2016.

[4] J. Flores-Parra, M. Castañón-Puga, R. Evans, R. Rosales-Cisneros and C. Gaxiola-Pacheco «Towards team formation using Belbin role types and a social networks analysis approach,» chez *IEEE Technology and Engineering Management Conference (TEMSCON)*, 2018.

[5] S. Kalayathankal, J. Kureethara and S. Narayanamoorthy, «A modified fuzzy approach to project team selection,» *Soft Computing Letters 3*, vol. 3, p. 100012, 2021.

[6] W. El-Ashmawi, A. Ali and M. Tawhid, «An improved particle swarm optimization with a new swap operator for team formation problem,» *J Ind Eng Int*, vol. 15, p. 53–71, 2019.

[7] M. Winter, Developing a group model for student software, Master's thesis, University of Saskatchewan, 2004.

[8] V. Yannibelli and A. Amandi, «Forming well-balanced collaborative learning teams according to the roles of their members: An evolutionary approach,» Chez *12th International Symposium on Computational Intelligence and Informatics (CINTI)*, 2011.

[9] A. Aranzabal, E. Epelde and M. Artetxe, «Team formation on the basis of Belbin's roles to enhance students' performance in project based learning,» *Education for Chemical Engineers*, vol. 38, pp. 22-37, 2022.

[10] Z. Chen, S. Kosari, S. Kaarmukilan, C. Yuvapriya. «A video processing algorithm using temporal intuitionistic fuzzy sets,» *Journal of Intelligent & Fuzzy Systems*, vol. 43, pp. 8057-8072, 2022.

[11] L. Zhou, Y. Wang and Y. Jiang, «Investment Project Assessment by a MAGDM Method Based on the Ranking of Interval Type-2 Fuzzy Sets,» *Journal of Intelligent & Fuzzy Systems*, vol. 35, n° 12, p. 1875 – 1888, 2018.

[12] A. Arram and M. Ayob, «A novel multi-parent order crossover in genetic algorithm for combinatorial optimization problems,» *Computers & Industrial Engineering*, vol. 133, pp. 267-274, 2019.

[13] A. Boufaied, «A Diagnostic Approach for Advanced Tracking of Commercial Vehicles With Time Window Constraints,» *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, n° 13, pp. 1470-1479, 2013.