ISSN 1870-4069

Collaborative Learning Team Formation Considering Team Roles: An Evolutionary Approach based on Adaptive Crossover, Mutation and Simulated Annealing

Virginia Yannibelli[1,2], Analía Amandi[1,2]

[1] ISISTAN Research Institute,
UNCPBA University Campus Universitario,
Argentina

[2] CONICET, National Council of Scientific and Technological Research,
Argentina

{virginia.yannibelli, amandi}@isistan.unicen.edu.ar

Abstract. In this paper, a hybrid evolutionary algorithm is proposed to solve a collaborative learning team formation problem in higher education contexts. This problem involves a grouping criterion evaluated satisfactorily in a great variety of higher education courses as well as training programs. This criterion is based on the team roles of students, and implies forming well-balanced teams respecting the team roles of their members. The hybrid evolutionary algorithm uses adaptive crossover, mutation and simulated annealing processes, in order to improve the performance of the evolutionary search. These processes adapt their behavior regarding the state of the evolutionary search. The performance of the hybrid evolutionary algorithm is exhaustively evaluated on data sets with very different complexity levels, and after that, is compared with those of the algorithms previously reported in the literature to solve the addressed problem. The results obtained from the performance comparison indicate that the hybrid evolutionary algorithm significantly outperforms the algorithms previously reported, in both effectiveness and efficiency.

Keywords: collaborative learning, collaborative learning team formation, team roles, evolutionary algorithms, hybrid evolutionary algorithms, adaptive evolutionary algorithms, simulated annealing algorithms.

1 Introduction

Collaborative learning is a pedagogical approach used frequently in higher education contexts, with the aim of enriching the individual learning of students. Such approach implies organizing students into collaborative learning teams to develop collaborative learning tasks.

These collaborative learning teams must be formed considering that students can gain new knowledge and also develop new skills by interacting with their peers, to improve their individual learning.

In this collaborative learning context, the grouping criterion is extremely important since the composition of the collaborative learning teams influences significantly the learning process and the social behavior of their members as well as the performance of the teams [1, 2]. Besides, the procedure used to apply the grouping criterion (i.e., either a manual or automated procedure), is important since many grouping criteria reported in the literature need a significant amount of knowledge, time and effort to be applied manually [10]. In these cases, an automated procedure could reduce the workload of professors as well as optimize the collaborative learning team formation.

In the literature, many different works have addressed the problem of automatically forming collaborative learning teams from the students [10, 4]. These works have significant differences in many aspects, particularly the grouping criteria applied, and the algorithms used. In this respect, to the best of the authors' knowledge, only few works apply grouping criteria that have been both satisfactorily and widely evaluated in higher education contexts.

The work reported in [5], formally describes the problem of automatically forming collaborative learning teams from the students enrolled in a given course, considering a grouping criterion satisfactorily evaluated in a very great variety of higher education courses as well as training programs. This grouping criterion refers to that defined by Belbin's team role model [3]. Such criterion based on the team roles of students, and implies forming well-balanced teams with respect of the team roles of their members. In this respect, a team role is defined as the way in which a person tends to behave, contribute and interrelate with others throughout a collaborative task. In the literature, many different studies indicate that collaborative learning team formation in higher education contexts considering the Belbin's criterion generates good interactions and discussions during the learning process, improves the social behavior of the students, enhances the learning process of the students, and influences very positively on the learning level of the students and also the performance of the teams [4]. Therefore, the collaborative learning team formation problem described in [5], is very valuable in higher education contexts.

The collaborative learning team formation problem described in [5], is an NP-Hard optimization problem. Because of this reason, as reported in [5], exhaustive search and optimization algorithms only can solve very small instances of the problem in a reasonable period of time. Thus, heuristic search and optimization algorithms have been proposed in the literature to solve the problem: an evolutionary algorithm was proposed in [5], a memetic algorithm was proposed in [9], which incorporates a hill-climbing algorithm into the framework of an evolutionary algorithm, and a hybrid evolutionary algorithm was proposed in [8] that integrates an adaptive simulated annealing algorithm into the framework of an evolutionary algorithm. These three algorithms utilize non-adaptive crossover and mutation processes to develop the evolutionary search.

In this paper, the collaborative learning team formation problem described in [5], is addressed with the aim of proposing a better heuristic search and optimization

algorithm to solve it. In this regards, a hybrid evolutionary algorithm is proposed that uses adaptive crossover, mutation and simulated annealing processes.

The behavior of these processes is adaptive in accordance with the state of the evolutionary search. Such adaptive crossover, mutation and simulated annealing processes are utilized in order to enhance the performance of the evolutionary search [6, 12, 13].

The above-mentioned hybrid evolutionary algorithm is proposed mainly because of the following reason. Evolutionary algorithms with adaptive crossover and mutation processes have been proven to be much more effective than evolutionary algorithms with non-adaptive crossover and mutation processes in respect of the resolution of a great variety of NP-Hard optimization problems [6, 12, 13]. Thus, the proposed hybrid evolutionary algorithm could outperform the heuristic search and optimization algorithms previously proposed to solve the problem.

The remainder of the paper is organized as follows. Section 2, describes in detail the problem addressed in this paper. Section 3, presents the hybrid evolutionary algorithm proposed. Section 4, presents the computational experiments developed to evaluate the performance of the hybrid evolutionary algorithm, and an analysis of the obtained results. Section 5, presents related works. Finally, Section 6, presents the conclusions of the present work.

2 Description of the Collaborative Learning Team Formation Problem

In this paper, the collaborative learning team formation problem described in [5] is addressed. A description of this problem is presented below.

Assume that a course S has a number of n students enrolled, $S = \{s_1, s_2, \ldots, s_n\}$, and the professor must organize these n students into g teams, $G = \{G_1, G_2, \ldots, G_g\}$. Each team G_i is composed of a number z_i of students, and each student can only belong to one team. In relation to team size, students must be organized considering the g teams have a similar number of students each. In particular, the difference among the sizes of the teams must not exceed one. The values of the terms S, n and g are known.

Regarding the students, it is assumed that they naturally play different team roles when participating in a collaborative task. A team role is the way in which a person tends to behave, contribute and interrelate with others throughout a collaborative task. In relation to the team roles which can be played by the students, the nine team roles defined in the Belbin's model [3], are considered. Table 1 shows these nine roles, with a brief description of the features of each one.

Based on the Belbin's model [3], it is assumed that each student naturally plays one or several of the nine roles presented in Table 1. In this respect, the roles naturally played by each student are known data. Such roles may be obtained by using the Belbin Team-Role Self-Perception Inventory (BTRSPI) developed by Belbin [3]. The BTRSPI determines the team roles of the persons by giving them self-evaluation tests [3].

As part of the problem, teams must be formed considering that the balance among the team roles of their members is maximized.

Table 1. Belbin's team role characteristics.

Role	Characteristics
Plant (PL)	Creative, imaginative, unorthodox. Solves difficult problems.
Resource Investigator (RI)	Extrovert, enthusiastic, communicative. Explore opportunities. Develops contacts.
Co-ordinator (CO)	Mature, confident, a good chairperson. Clarifies goals, promotes decision-making, delegates well.
Shaper (SH)	Challenging, dynamic, thrives on pressure. Has the drive and courage to overcome obstacles.
Monitor Evaluator (ME)	Sober, strategic and discerning. Sees all options. Judges accurately.
Teamworker (TW)	Co-operative, mild, perceptive and diplomatic. Listens, builds, averts friction.
Implementer (IM)	Disciplined, reliable, conservative and efficient. Turns ideas into practical actions.
Completer/Finisher (CF)	Painstaking, conscientious, anxious. Searches out errors and omissions. Polishes and perfects.
Specialist (SP)	Single-minded, self-starting, dedicated. Provides knowledge and skills in key areas.

This grouping criterion requires analyzing the balance level of the formed teams. In order to analyze such balance level, the balance conditions defined by Belbin are considered [3]. In relation to these conditions, Belbin [3], states that a team is balanced if each role included in his model is played naturally by at least one team member. Thus, in a balanced team, all team roles are naturally played. Besides, Belbin states that each role should be naturally played by only one team member [3]. Belbin states that a team is unbalanced if some roles are not played naturally, or if several of its members play the same role naturally (i.e., duplicate role) [3].

The grouping criterion above-mentioned is modeled by Formulas (1)-(3). In this respect, Formulas (1)-(2), model the balance conditions defined by Belbin [3].

Formula (1), analyzes the way in which a given role r is played within a given team G_i, and after that gives a score accordingly. If role r is naturally played by only one member of team G_i, then 1 point is awarded to G_i. Otherwise, if role r is not naturally played by any member of G_i, or role r is naturally played by several members of G_i, then 2 points and p points are taken off respectively.

Formula (2), defines the balance level of a given team G_i. This balance level is defined based on the scores obtained by G_i, through Formula (1), regarding the nine roles. Thus, the greater the number of non-duplicate roles (i.e., roles played naturally by only one member of G_i), the greater the balance level assigned to G_i. On contrary, the fewer the number of roles played naturally, or the more duplicate roles, the lower the balance level assigned to G_i. Note that the balance conditions defined by Belbin [3] can be seen in Formula (2).

By this formula, a perfectly balanced team (i.e., a team in which each one of the nine roles is played naturally by only one team member), will obtain a level equal to 9.

Formula (3) maximizes the average balance level of *g* teams defined from the *n* students of the course. Specifically, this formula aims to find a solution (i.e., set of *g* teams), that maximizes the average balance level of *g* teams. Such solution is the optimal solution to the addressed problem. In Formula (3), set *C* contains all the sets of *g* teams that may be defined from the *n* students. The term *G* represents a set of *g* teams belonging to *C*. The term *b(G),* represents the average balance level of the *g* teams belonging to set *G*. Note that in the case of a set *G* of perfectly balanced *g* teams, the value of the term *b(G),* is equal to 9.

For a more detailed discussion of Formulas (1)-(3), readers are referred to [5].

$$nr(G_i, r) = \begin{cases} 1 & \text{if } r \text{ is naturally played by only one member of } G_i, \\ -2 & \text{if } r \text{ is not naturally played in } G_i, \\ -p & \text{if } r \text{ is naturally played by } p \text{ members of } G_i. \end{cases} \quad (1)$$

$$nb(G_i) = \sum_{r=1}^{9} nr(G_i, r), \quad (2)$$

$$\max_{\forall G \in C} \left(b(G) = \frac{\sum_{i=1}^{g} nb(G_i)}{g} \right). \quad (3)$$

3 Description of the Hybrid Evolutionary Algorithm

To solve the problem, a hybrid evolutionary algorithm is proposed. This algorithm uses adaptive crossover, mutation and simulated annealing processes. The behavior of these processes is adaptive regarding the state of the evolutionary search. The use of adaptive crossover, mutation and simulated annealing processes is meant to enhance the performance of the evolutionary search, in both exploration and exploitation of the search space [6, 12, 13].

The general behavior of this hybrid evolutionary algorithm is described as follows. Assuming a course with *n* students enrolled who should be organized into *g* teams, the algorithm creates a random initial population of feasible solutions. In such population, each one of the solutions encodes a feasible set of *g* teams that may be defined from the *n* students. Once the initial population is created, the algorithm both decodes and evaluates each solution of this population by a fitness function. Specifically, the set of *g* teams represented by each solution is built, and after that evaluated in relation to the optimization objective of the problem. As was mentioned previously in Section 2, this objective is to maximize the balance level of the *g* teams formed from the *n* students. Therefore, the fitness function analyzes the balance level of the *g* teams represented by

each solution, and then defines a fitness level for each solution. In order to develop such analysis, the fitness function based on knowledge of the students' team roles.

After each one of the solutions of the population is evaluated, a parent selection process is utilized to determine which solutions of the population will compose the mating pool. In this respect, the highest fitness solutions will have more likelihood of being selected. Once the mating pool is composed, the solutions within the mating pool are organized in pairs. After that, a crossover process is applied to each pair of solutions with an adaptive probability AP_c, in order to generate new feasible solutions. After that, a mutation process is applied to each one of the solutions obtained by the crossover process, with an adaptive probability AP_m. After that, a survival selection process is applied to determine which solutions from the solutions into the population and the solutions generated from the mating pool will compose the new population. Finally, an adaptive simulated annealing algorithm is applied to each solution within the new population, except to the highest fitness solution of this population which is maintained.

The described process is repeated until a given number of generations is achieved.

a. Encoding of Solutions

The representation proposed in [5], is used in order to encode the solutions. Thus, each solution is encoded as a list with as many positions as students enrolled in the course (i.e., n positions). Each position j ($j = 1,\ldots, n$), on this list contains a different student (i.e., repeated students are not admitted). Besides, each student s_k ($k = 1,\ldots, n$), may be in any position on the list. This list is a permutation of the n students.

To decode the set G of g teams from the list, the decoding process proposed in [5], is utilized. By using this process, the list is divided into g segments, considering that the difference among the sizes of the segments must not exceed one. Each segment represents to a different team.

b. Fitness Function

To evaluate the encoded solutions, a specially designed fitness function is used. Given an encoded solution, this function decodes the set G of g teams represented by the solution. The decoding is developed by the process mentioned in Section 3.1. Then, the function calculates the value of the term $b(G)$, corresponding to G (Formulas (1)-(3)). This value represents the average balance level of the g teams composing the set G, and thus, determines the fitness level of the encoded solution.

c. Parent Selection Process

To develop the parent selection, the well-known roulette wheel selection process [6], is utilized. In this process, a selection probability is defined for each solution of the population. The probability of each solution is proportional to its fitness level. Thus, the highest fitness solutions have more probability of being selected for the mating pool.

d. Adaptive Crossover and Adaptive Mutation Processes

In relation to the crossover and mutation processes, processes feasible for solutions encoded as permutations of n students are utilized.

To develop the crossover, the process named partially mapped crossover [6], is applied. This process generates two new feasible encoded solutions (i.e., two new permutations of the n students) from a given pair of encoded solutions. This crossover process is one of the most applied for permutations of n elements in the literature [6].

In order to develop the mutation, the process named insert mutation [6], is applied. This process generates a new feasible encoded solution (i.e., a new permutation of the n students), from a given encoded solution. This mutation process is one of the most used for permutations of n elements in the literature [6].

These crossover and mutation processes are applied with adaptive crossover and mutation probabilities, respectively. In this regards, an adaptive crossover probability named AP_c and an adaptive mutation probability named AP_m are defined by Formulas (4)-(7). In these formulas, the term S refers to the state of the evolutionary search, and the term S_W refers to the widest possible state of the evolutionary search. In Formula (4), C^L and C^H refer to the lower and upper bounds for the crossover probability, respectively. In Formula (5), M^L and M^H refer to the lower and upper bounds for the mutation probability, respectively. The term f_{max} refers to the maximal fitness within the population, f_{min} refers to the minimal fitness within the population, and f refers to the fitness of the solution to be mutated.

The term S is defined by Formula (6). In this formula, f_{max} refers to the maximal fitness within the population, f_{avg} refers to the average fitness of the population, and the term $(f_{max} - f_{avg})$ refers to a well-known measure of the state of the evolutionary search [7, 6].

The term S_W is defined by Formula (7). In this formula, the term f_{MAX} represents to the maximum fitness value possible (i.e., the upper bound of the fitness function), and the term f_{MIN} represents to the minimum fitness value possible (i.e., the lower bound of the fitness function).

Through Formulas (4)-(7), probabilities AP_c and AP_m are adaptive according to the state of the evolutionary search. In this respect, when the evolutionary search starts to converge, probabilities AP_c and AP_m are increased, to promote the exploration of new regions of the search space, and therefore, to avoid the premature convergence of the evolutionary search. In contrast, when the evolutionary search is well scattered in the search space, probabilities AP_c and AP_m are reduced, to promote the exploitation of known regions of the search space. Thus, probabilities AP_c and AP_m are adaptive to promote either the exploration or exploitation of the search space, according to the state of the evolutionary search.

Through Formula (5), probability AP_m is also adaptive regarding the fitness of the solution to be mutated. In this sense, lower values of AP_m are assigned to high-fitness solutions, whereas higher values of AP_m are assigned to low-fitness solutions. This adaptation has the aim of preserving high-fitness solutions, when the exploration of the search space is promoted.

$$AP_c = \left(\frac{S_W - S}{S_W} \right) * \left(C^H - C^L \right) + C^L, \tag{4}$$

$$AP_m = \left(\frac{f_{\max} - f}{f_{\max} - f_{\min}} \right) * \left(\frac{S_W - S}{S_W} \right) * \left(M^H - M^L \right) + M^L, \tag{5}$$

$$S = \left(f_{max} - f_{avg} \right), \tag{6}$$

$$S_W = \left(f_{MAX} - f_{MIN} \right). \tag{7}$$

e. Survival Selection Process

To develop the survival selection, the classical fitness-based steady-state selection process [6], is utilized. In this process, the worst λ solutions of the current population are replaced by the best λ solutions generated from the mating pool. This process preserves the highest fitness solutions reached by the hybrid evolutionary algorithm [6].

f. Adaptive Simulated Annealing Algorithm

The general behavior of the applied adaptive simulated annealing algorithm, which is a variation of the algorithm presented in [8], is described as follows.

This adaptive simulated annealing algorithm is mainly an iterative process. This process begins considering a given encoded solution *s*, and a given initial value T_0 for the parameter named temperature. In each of the iterations, a new encoded solution *s'* is created from the current encoded solution *s*, by applying a move operator. After that, the algorithm analyzes if solution *s* should be replaced or not by solution *s'*. When the fitness value of solution *s'* is better than that of solution *s*, the algorithm replaces to solution *s* by solution *s'*. In contrast, when the fitness value of solution *s'* is worse than or equal to that of solution *s*, the algorithm replaces to solution *s* by solution *s'* based on an acceptance probability which is $exp(-\Delta/T_c)$. In this probability, term T_c is the current value of the temperature parameter, and Δ is the difference between the fitness values of solutions *s* and *s'*. Thus, the acceptance probability is directly proportional to the current value of the temperature parameter.

The above-described process is repeated until a given number *I* of iterations is reached. It is necessary to mention that, at the end of each of the iterations, the value of the temperature parameter is reduced by a given cooling factor α.

In relation to the initial value T_0 of the temperature parameter, this value is defined based on the evolutionary search state S_P reached after the survival selection process, considering that such state is measured by calculating Formula (6), on the population obtained by the survival selection process. Specifically, the value T_0 is calculated by using the next formula: $T_0 = 1 / S_P$.

By this formula, when the evolutionary search is scattered in the search space, the value T_0 is low, and thus the acceptance probability of the algorithm is also low.

Consequently, the algorithm promotes the exploitation of known regions of the search space. When the evolutionary search starts to converge, the value T_0 increases, and therefore the acceptance probability of the algorithm also increases. Consequently, the algorithm promotes the exploration of new regions of the search space. Based on the mentioned, the algorithm is adaptive according to the state of the evolutionary search, in order to promote either the exploitation or exploration of the search space.

In relation to the move operator of the simulated annealing algorithm, an operator feasible for solutions encoded as permutations of *n* students is applied. Specifically, the operator named swap mutation [6], is applied. This operator creates a new encoded solution (i.e., new permutation of the *n* students), from a given encoded solution. This operator is one of the most applied for permutations of *n* elements in the literature [6].

4 Computational Experiments

To evaluate the performance of the hybrid evolutionary algorithm, the ten data sets introduced in [5], were used. Each data set contains a number *n* of students, and details a *g* number of teams to be built from the *n* students. In addition, each data set details the team roles of each of its *n* students, considering that these team roles belong to the Belbin's model [3]. The main characteristics of these ten data sets are shown in Table 2. For a description of the team roles of the students in each of the data sets, readers are referred to [5].

Each one of the ten data sets has a known optimal solution with a fitness level of 9. Note that a solution with a fitness level of 9 contains a set of perfectly balanced *g* teams, regarding the balance conditions defined by Belbin [3]. These known optimal solutions are considered here as references to evaluate the performance of the hybrid evolutionary algorithm.

The hybrid evolutionary algorithm was run 30 times on each of the data sets. After each of the runs, the algorithm provided the best solution found. In order to carry out

Table 2. Main characteristics of the ten data sets used.

Data set	Number of students enrolled (*n*)	Number of teams to be formed (*g*)
1	18	3
2	24	4
3	60	10
4	120	20
5	360	60
6	600	100
7	1200	200
8	1800	300
9	2400	400
10	3000	500

these runs, the algorithm parameters were set as follows: size of the population = 80; generations = 200; crossover process: $C^L = 0.5$ and $C^H = 0.9$; mutation process: $M^L = 0.01$ and $M^H = 0.2$; survival selection process: $\lambda = 40$; simulated annealing algorithm: $I = 20$ and $\alpha = 0.9$. It is necessary to mention that the algorithm parameters were set with these values based on exhaustive preliminary experiments.

By these preliminary experiments, many different settings were considered for the algorithm parameters, and then the best of these settings was selected for the algorithm parameters.

Table 3, presents the results obtained for each data set. Column 1 presents the name of each data set. Column 2 presents the average fitness value of the solutions reached for each of the data sets. Column 3 presents the average computation time of the runs performed on each data set. The experiments were performed on a personal computer Intel Core 2 Duo at 3.00 GHz and 3 GB RAM under Windows XP Professional Version 2002. The algorithm was implemented in Java.

The results in Table 3, were analyzed considering that each data set has a known optimal solution with a fitness level of 9. For the first seven data sets (i.e., the seven less complex data sets), the algorithm reached an optimal average fitness value. This means that the algorithm found an optimal solution (i.e., a set of perfectly balanced g teams), in each run. For the last three data sets (i.e., the three more complex data sets), the algorithm reached an average fitness value higher than 8.75. This means that the algorithm found near-optimal solutions for each of the data sets.

The composition of such solutions was exhaustively analyzed, noting that these solutions contain a very high percentage of perfectly balanced teams. Based on these results, the algorithm reached very high-quality solutions for the problem instances represented by the data sets.

Table 3. Results obtained by the hybrid evolutionary algorithm for each data set.

Data Set	Fitness Value	Time (seconds)
1	9	0.18
2	9	0.46
3	9	3.78
4	9	6.01
5	9	14.9
6	9	19.03
7	9	72.4
8	8.87	133.07
9	8.81	211.18
10	8.76	303.84

Table 4. Results obtained by the algorithms previously proposed for the addressed problem.

Data set	Evolutionary algorithm [5]		Memetic algorithm [9]		Hybrid algorithm [8]	
	fitness value	**time (s)**	**fitness value**	**time (s)**	**fitness value**	**time (s)**
1	9	0.5537	9	0.42	9	0.29
2	9	1.3741	9	1.03	9	0.721
3	9	11.0669	9	8.30	9	5.81
4	9	17.5976	9	13.20	9	9.24
5	8.8	40.8722	8.92	30.65	9	21.46
6	8.76	55.7548	8.86	41.82	8.97	29.27
7	8.7	196.9964	8.78	147.75	8.86	103.43
8	8.64	362.0328	8.68	271.52	8.77	190.1
9	8.61	574.6589	8.65	430.994	8.74	301.69
10	8.592	771.6553	8.62	578.74	8.7	405.118

As regards the average computation time required by the algorithm, the following may be mentioned. For the first six data sets (i.e., the six less complex data sets), the average time required was lower than 20 seconds. For the last four data sets (i.e., the four more complex data sets), the average time required was higher than 72 seconds and lower than 304 seconds.

Taking into account the complexity level of the problem instances inherent to the ten data sets, particularly the complexity level of the problem instances inherent to the four more complex data sets, it is considered that the average times required by the algorithm are acceptable.

a. Comparative Analysis with Competing Algorithms

To the best of the authors' knowledge, only three heuristic search and optimization algorithms have been previously proposed in the literature for solving the addressed problem: a traditional evolutionary algorithm [5], a traditional memetic algorithm [9], which incorporates a hill-climbing algorithm into the framework of an evolutionary algorithm, and a hybrid evolutionary algorithm [8], which incorporates an adaptive simulated annealing algorithm within the framework of an evolutionary algorithm. These three algorithms use non-adaptive crossover and mutation processes to develop the evolutionary search.

Based on the computational experiments reported in [8, 5, 9], the three algorithms have been evaluated on the ten data sets presented in Table 2, and have obtained the results that are shown in Table 4.

These experiments have been performed on a personal computer Intel Core 2 Duo at 3.00 GHz and 3 GB RAM under Windows XP Professional Version 2002. The algorithms have been implemented in Java.

The results in Table 4 indicate that the algorithm proposed in [8] is the best of the three algorithms. Below, the performance of this algorithm is compared with that of the hybrid evolutionary algorithm proposed here. For sake of simplicity, the algorithm proposed in [8] will be referred as algorithm H.

The results in Tables 3-4 indicate that the hybrid evolutionary algorithm proposed here and the algorithm H reached the same average fitness value (i.e., an optimal average fitness value), for the first five data sets (i.e., the five less complex data sets). Nevertheless, the average fitness value reached by the hybrid evolutionary algorithm for each of the last five data sets (i.e., the five more complex data sets), is much higher than that reached by the algorithm H. In particular, the hybrid evolutionary algorithm reached optimal average fitness values for the data sets 6-7. These results mean that the quality of the solutions achieved by the hybrid evolutionary algorithm for the five more complex data sets is much better than that of the solutions achieved by the algorithm H. Furthermore, the average computation time of the hybrid evolutionary algorithm for each data set is significantly lower than that of the algorithm H.

Based on these results, the performance of the hybrid evolutionary algorithm on the five more complex data sets is much better than that of the algorithm H, regarding the quality of the solutions and also the computation time. This is mainly because of the following reasons.

The hybrid evolutionary algorithm proposed here integrates adaptive crossover and mutation processes. The behavior of these processes is adaptive regarding the state of the evolutionary search. This adaptation is meant with the aim of promoting either the exploitation or exploration of the search space, and thus, to enhance the evolutionary search. In contrast with the hybrid evolutionary algorithm, the algorithm H utilizes non-adaptive crossover and mutation processes. These processes disregard the state of the evolutionary search, and therefore, do not have the possibility of enhancing the evolutionary search.

5 Related Work

Many different studies in the literature indicate that collaborative learning in higher education environments significantly benefits from the application of the Belbin's model [10, 4]. However, to the best of our knowledge, only few works in the literature address the problem of automatically forming collaborative learning teams based on the Belbin's model. These works differ in several aspects, including the modelling of this collaborative learning team formation problem, and the algorithms used to form the collaborative learning teams. In this section, we review related works reported in the literature, focusing the attention on analysing the aspects above-mentioned

Ounnas et al. [11], proposed a framework which utilizes an ontology to describe students' characteristics including Belbin's team roles. This framework provides a list with grouping criteria that includes forming teams based on the Belbin's model.

In this framework, the team formation problem is modeled as a constraint satisfaction problem. The weak constraints of the problem refer to the grouping criteria selected by the professor from the provided list, and the optimization objective of the problem is to find the set of teams that minimizes the number of violated weak constraints. The problem is solved by a DLV constraint satisfaction solver (i.e., an exhaustive search algorithm).

Alberola et al. [4], proposed a tool based on the Belbin's team role model. This tool aims to build well-balanced teams regarding the team roles of their student members. In this case, the team formation problem is modeled as a coalition structure generation problem and is solved by means a linear programming method (i.e., an exhaustive search algorithm). In this tool, the team roles of students are estimated from the feedback given by the other students, by using Bayesian learning. Although this is meant to avoid the drawbacks of the Team Role Self-Perception Inventory, the estimation of the students' roles could be negatively affected by biased feedback.

The above-mentioned works utilize different exhaustive search algorithms to form the teams. However, this kind of algorithms only can solve very small instances of the problem in a reasonable period of time. Therefore, heuristic search and optimization algorithms are required to solve the problem.

Yannibelli and Amandi [5, 9, 8], proposed very different evolutionary algorithms in order to solve problem instances with different complexity levels. These algorithms, in particular the algorithm proposed in [8], reached promising results regarding both effectiveness and efficiency. Nevertheless, these algorithms based on non-adaptive crossover and mutation processes to develop the evolutionary search. These processes disregard the evolutionary search state, and therefore, do not have the possibility of enhancing the performance of the evolutionary search, regarding both effectiveness and efficiency.

Unlike the evolutionary algorithms proposed in [5, 9, 8], the hybrid evolutionary algorithm proposed here uses adaptive crossover and mutation processes. These processes adapt their behavior regarding the evolutionary search state, to enhance the performance of the evolutionary search.

6 Conclusions and Future Work

In this paper, the collaborative learning team formation problem described in [5], was addressed. This problem involves a grouping criterion that has been both satisfactorily and widely evaluated in higher education contexts. Such criterion corresponds to that defined by Belbin's team role model [3].

For solving the addressed problem, a hybrid evolutionary algorithm was proposed. This proposed hybrid evolutionary algorithm integrates adaptive crossover, mutation and simulated annealing processes. These processes adapt their behavior regarding the state of the evolutionary search. The integration of such adaptive processes is meant to enhance the performance of the evolutionary search [6, 12, 13].

The performance of the hybrid evolutionary algorithm was evaluated on ten data sets with very different complexity levels.

Subsequently, the performance of this algorithm on these data sets was compared with those of the algorithms previously reported in the literature for solving the addressed problem. Based on the obtained results, it may be stated that the proposed hybrid evolutionary algorithm considerably outperforms the previous algorithms.

In future works, the incorporation of other adaptive processes into the framework of the evolutionary algorithm will be evaluated. In particular, other adaptive crossover and mutation processes, as well as adaptive selection processes, will be exhaustively evaluated. Furthermore, the incorporation of other search and optimization techniques into the framework of the evolutionary algorithm will be evaluated.

References

1. Barkley, E. F., Cross, K. P., Howell-Major, C.: Collaborative learning techniques. John Wiley & Sons (2005)
2. Michaelsen, L. K., Knight, A. B., Fink, L. D.: Team-based learning: A transformative use of small groups in college teaching. Stylus Publishing (2004)
3. Belbin, R. M.: Team Roles at Work. Taylor & Francis (2011)
4. Alberola, J., Del-Val, E., Sanchez-Anguix, V., Palomares, A., Teruel, M.: An artificial intelligence tool for heterogeneous team formation in the classroom. Knowledge-Based Systems, 101(1), pp. 1–14 (2016)
5. Yannibelli, V., Amandi, A.: A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context. Expert Systems with Applications, 39(10), pp. 8584–8592 (2012)
6. Eiben, A. E., Smith, J. E.: Introduction to Evolutionary Computing, Springer (2015)
7. Srinivas, M., Patnaik, L. M.: Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics, 24(4), pp. 656–667 (1994)
8. Yannibelli, V., Amandi, A.: A Hybrid Algorithm Combining an Evolutionary Algorithm and a Simulated Annealing Algorithm to Solve a Collaborative Learning Team Building Problem. J.-S. Pan et al. (eds.), (HAIS), LNCS, Springer, Heidelberg, 8073, pp. 376–389 (2013)
9. Yannibelli, V., Amandi, A.: A memetic algorithm for collaborative learning team formation in the context of software engineering courses. Cipolla-Ficarra, F., Veltman, K., Verber, D., Cipolla-Ficarra, M., Kammüller, F. (eds.), (ADNTIIC), LNCS, Springer, 7547, pp. 92–103 (2012)
10. Cruz, W. M., Isotani, S.: Group Formation Algorithms in Collaborative Learning Contexts: A Systematic Mapping of the Literature. Baloian, N., Burstein, F., Ogata, H., Santoro, F., Zurita, G. (eds), Collaboration and Technology, (CRIWG), LNCS, Springer, 8658, pp. 199–214 (2014).
11. Ounnas, A., Davis, H. C., Millard, D. E.: A Framework for Semantic Group Formation in Education. Educational Technology & Society, 12(4), pp. 43–55 (2009)
12. Rodriguez, F.J., García-Martínez, C., Lozano, M.: Hybrid metaheuristics based on evolutionary algorithms and simulated annealing: taxonomy, comparison, and synergy test, IEEE Trans. Evol. Comput., 16(6), pp. 787–800 (2012)
13. Talbi, E.: Hybrid Metaheuristics. (SCI), Springer, 434 (2013)