A hybrid approach for composing groups in collaborative learning contexts

Julián Moreno*, Jesús D. Sánchez, Andrés F. Pineda

Universidad Nacional de Colombia-Sede Medellín, Medellín, Colombia

ABSTRACT

Considering that there are many alternatives in the literature for composing groups in collaborative learning contexts, we present a proposal that exhibits several features. First, and from the operational point of view, our proposal is highly flexible because i) it allows for several group sizes and an arbitrary array of grouping attributes, and ii) it may be easily adapted to consider several homogeneity/heterogeneity criteria. Second, and from the algorithmic point of view, it combines the best of two apparently opposite worlds: it uses a local brute-force search within an iterative process guided by a randomized heuristic criterion. Thus, this approach is still Non-Polynomial (NP) but in terms of the size of the groups, whereas is Polynomial (P) in terms of the number of students. Third, the experiments with several datasets, with student numbers varying from 20 to 3500, demonstrate reasonable performance and running times for this approach. We contrasted these times with those reported in 19 related works and, first taking into account certain considerations, we found that ours were lower in most cases. Nevertheless, and as the fourth feature, we make available both the datasets and the source code to allow for more objective comparisons of approaches, including our own.

1. Introduction

There is considerable evidence for the positive effects of collaborative learning not only in terms of academic performance but also in many other learning aspects (Barkley et al., 2005; Johnson and Johnson, 1989; Kreijns et al., 2007; So and Brush, 2008; Strijbos et al., 2004, 2007). Despite such evidence, several issues remain permanent objects of study. What are the most suitable collaboration techniques/activities for specific contexts? How should the grouping process be addressed? These are questions that remain objects of discussion. Regarding the latter question, the way in which learners are grouped has a critical impact on the successful development of the collaborative activities as well as on the quantity and quality of peer interactions (Cruz and Isotani, 2014; Isotani and Mizoguchi, 2008; Sadeghi and Kardan, 2015). Nevertheless, grouping is not an easy task, and numerous factors must be considered, including group size, group composition, grouping attributes, and grouping method. In fact, when we analyzed 33 research works about grouping approaches from the last 15 years, they varied greatly in how they deal with these four factors.

Regarding the first factor, the group size is considered by some researchers to be a significant variable that influences group productivity (Kreijns et al., 2003; Shaw, 2013; Strijbos et al., 2004). Interaction patterns and learning benefits differ between pairs, small groups (three to six members), and large groups (seven or more members), especially if participation equality or shared products are required (Lim and Yingqin, 2006). Whereas smaller groups may facilitate individual participation but lack the ability to evaluate potential solutions, larger groups may make communication difficult but provide an enhanced knowledge base (Sadeghi and Kardan, 2015). From the 33 analyzed works, some define a fixed value: 3 members (Zheng and Pinkwart, 2014), 4 members (Graf and Bekele, 2006; Zheng et al., 2016), 5 members (Ani et al., 2010), and 6 members (Long & Qing-hong, 2014; Yannibelli and Amandi, 2012), whereas the rest consider the group size as a variable.

Regarding the second factor, there are several alternatives for group composition. One is homogeneity or more precisely intra-homogeneity, i.e., when all members of the same group display similar attributes as possible, versus intra-heterogeneity which is the exactly opposite: the members of the same group display attributes as diverse as possible. This last case may be considered as a particularization of the maximally diverse grouping problem - MDGP (Feo and Mallek, 1990). Several authors have discussed the different benefits of the two alternatives. Sadeghi and Kardan (2015) state that whereas homogeneous groups may be better for obtaining particular goals, heterogeneous ones may tend to be more innovative and creative. Another alternative consists of mixed groups to satisfy homogeneity for some attributes and heterogeneity for another, whereas another one consists in obtaining inter-homogeneous
groups, i.e., each one is as diverse as possible as the population as a whole, versus inter-heterogeneity. These last two alternatives usually lead to intra-heterogeneity and intra-homogeneity respectively but not necessarily. Garshasbi et al. (2019) states that intra-heterogeneity enhances the success of a group by empowering the differences and complementary roles of its members according to their prequalified characteristics whereas inter-homogeneity ensures that all groups are equally built and therefore have the same chance of achieving such success. From the 33 analyzed works, most choose either intra-heterogeneity or inter-homogeneity, whereas a few choose the mixed alternative (Henry, 2013a; 2013b; Hubscher, 2010; Lambić et al., 2018) and even fewer adapt themselves to several alternatives (Abnar et al., 2012; Sadeghi and Kardan, 2015).

Regarding the third factor, grouping attribute options are very diverse, both in nature and in shape. They may include demographic variables such as gender or age, academic variables such as previous grades or enrolled subjects, cognitive variables such as learning styles or thinking styles, variables related to the learning activity itself such as previous knowledge or interest level, intra-personal variables such as personality type or interests, inter-personal variables as preferred team role or group work attitude, and many others. From the 33 analyzed works, most use several of those options, whereas some deal with an arbitrary set (Abnar et al., 2012; Henry, 2013a; Jozan and Taghiyareh, 2013; Lin et al., 2016; Moreno et al., 2012; Sadeghi and Kardan, 2015; Zheng et al., 2016; Zhou et al., 2016).

Regarding the fourth factor, several grouping methods have been used. From the 33 analyzed works, genetic algorithms (GA) is the most popular (Abnar et al., 2012; Ani et al., 2016; Garshasbi et al., 2019; Hwang et al., 2008; Krouska and Virvou, 2019; Lin et al., 2016; Long & Qing-hong, 2014; Moreno et al., 2012; Pinninghoff et al., 2017; Suk-striewong, 2017; Wang et al., 2007; Yannibelli and Amandi, 2012; Zheng et al., 2018), followed by particle swarm optimization (PSO) (Dascalu et al., 2014; Ghorbani and Montazer, 2012; Ho et al., 2010; Lin et al., 2010; Zervoudakis et al., 2019; Zheng and Pinkwart, 2014). From the rest, some use other heuristic approaches or combinations of them, some others use Binary Integer Programming (BIP) (Kardan and Sadeghi, 2014; Sadeghi & Kardan, 2015, 2016) and some others use a limited brute-force parallel algorithm (Henry, 2013a; 2013b).

Focusing on this fourth factor, determining which specific method is the best alternative, requires a thorough analysis. Considering the Non-Polynomial (NP) nature of the problem, described in detail in the next section, an algorithm that guarantees optimality like an exhaustive-search method is feasible only for small number of students. If this is not the case, some other method must be used, but choosing one in particular is difficult. From the algorithmic point of view, they could be compared with the corresponding time complexities, but from the 33 analyzed research works, this measure can only be inferred because no one presented it explicitly. A genuinely experimental comparison is not possible either, because datasets are very diverse and, except for two (Garshasbi et al., 2019; Pinninghoff et al., 2017), not available. Unavailability is also a problem for source codes. From the 33 analyzed works, only two claims that the corresponding software tool is available for use (Henry, 2013b; Pinninghoff et al., 2017), but in one case the reported link is broken. In a systematic mapping of the literature on grouping methods performed by Cruz and Isotani (2014), they reached a similar conclusion: only 2% or research works provided their source code, thereby preventing replication and reuse. Regarding numerical results, only one of the 33 works we analyzed present them. In some cases, the datasets correspond to actual data and in others to randomly generated data. Figure 1 summarizes those values, comparing the approaches used and the sizes of their largest dataset in terms of the number of students N. Note that most studies use a value of N up to 100, and only 4 use a value larger than 1000.

Considering this backdrop, this paper presents several contributions. First, we describe a hybrid algorithm that considers several grouping criteria and arbitrary parameters for group size and grouping attributes. We called it hybrid because it uses a local brute-force search within an iterative process guided by a randomized heuristic criterion. Although the idea of mixing algorithms that guarantee optimality, in this case locally rather than globally, with heuristics is not new in optimization problems, the way in which we presented had not been explored previously in this collaborative learning context. Second, we describe two datasets that can be easily replicated so further comparisons can be made. We then use these datasets to perform several experiments and, when possible, compare our results with some of those 33 related works.

The rest of the paper is organized as follows. In section 2, we formalize the grouping process as a combinatorial optimization problem and analyze its factorial solution space. Then, in section 3, we describe the proposed method as well as its corresponding time complexity. Later, in section 4, we describe the datasets generated and the corresponding experimental results. Finally, in section 5, we present some concluding remarks.

2. Problem formalization

Considering a number N of students which should be assigned to exactly g groups of size k, we can express the number of possible grouping schemes in combinatorial terms as:

\[
\binom{N}{k} \times \binom{N-k}{k} \times \cdots \times \frac{N-gk}{gk} \times \frac{gk}{k}
\]

Expressing each combinatorial as a factorial product, we obtain:

\[
\frac{N!}{k!^{g}(N-gk)!} \times \frac{(N-k)!}{k!^{g}(N-2k)!} \times \cdots \times \frac{2!}{k!^{g}k!} \times 1
\]

By removing terms that cancel each other, and considering that the group order is irrelevant, we can express the number of different solutions as:

\[
\frac{N!}{g!^{(g)}k!^{(g)}}
\]

This equation matches the one presented in (Sadeghi and Kardan, 2015), which confirms that applying a brute-force search method is infeasible in computational terms, even for N as small as 20.

Now, regarding the group composition criteria, there are several fitness functions that may be applied for each. Let \( S_i \) be a student with C attributes. If all those attributes are expressed as non-negative number, \( S_i \) may be represented by the array \( S_{i1}, S_{i2}, \ldots S_{iC} \) and the distance \( d_{ij} \) between element \( i \) and \( j \) may be simply defined by the Euclidean calculation:

\[
d_{ij} = \sqrt{(S_{i1} - S_{j1})^2 + (S_{i2} - S_{j2})^2 + \cdots + (S_{iC} - S_{jC})^2}
\]
\[ d_{ij} = \sum_{c=1}^{C} (S_{ic} - S_{jc})^2 \] (4)

Then intra-homogeneity may be obtained by:

\[ \text{Min} \left( F_1 = \frac{1}{g} \sum_{r=1}^{g} \sum_{c=1}^{C} d_{ij} \right) \] (5)

And consequently intra-heterogeneity may be obtained by Max \( F_1 \).

Now, if the mean of a group \( r \) is \( G_r = \{G_{r,1}, G_{r,2}, \ldots, G_{r,C}\} \) and the mean of all \( N \) students is \( S = \{S_1, S_2, \ldots, S_C\} \), inter-homogeneity may be obtained by:

\[ \text{Min} \left( F_2 = \frac{1}{g} \sum_{r=1}^{g} \sum_{c=1}^{C} |G_{r,c} - S_c| \right) \] (6)

And consequently intra-heterogeneity may be obtained by Max \( F_2 \).

3. Proposed method

As mentioned at the end of section 1, we propose a method that considers arbitrary parameters for group size and grouping attributes and, from the algorithmically point of view, it combines a local brute-force search algorithm with an iterative procedure that uses a straightforward heuristic criterion.

Consider for a moment that we have just two groups of \( k \) students each. We can then use a brute-force search algorithm to find all the ‘\( 2k \) choose \( k \)’ combinations for one of the two groups and then obtain the second as the direct complement. There are several ways to perform such an algorithm (Chase, 1970; Knuth, 2005; Payne and Ives, 1979; Ruskey, 1988); one is as follows.

Algorithm 1.

```plaintext
// arr is an array of m elements
// r is the size of the combination
// D is where combinations are stored
combination(arr, m, r, index, D, i) {
    if (index = r)
        return D
    if (i >= n)
        return
    D[index] = arr[i]
    combination(arr, m, r, index+1, D, i+1)
    combination(arr, m, r, index, D, i+1)
}
```

In this case, \( m = 2k \) and \( r = k \), so the algorithm has a time complexity \( O\left(\frac{2^k}{k!}\right)\). If we use it to “recombine” two given groups with \( k \) students each, we can obtain the optimal grouping scheme for that local problem. Next, the heuristic part of our proposal is applied. The idea is to repeat that procedure for the \( g/2 \) pairs of groups for the entire \( N \) students. Of course, the sum of the optimal solutions of the local problems does not guarantee by any means the optimal solution to the original problem. However, given a former grouping scheme, performing such a procedure
can only lead to a better, or at least equal, later solution. The demonstration is simple: there is just a $\frac{k!}{2k^2} k!$ chance that a particular grouping scheme for each couple is the best.

The next issue is to determine a procedure to select the $g/2$ pairs of groups. Here, the idea is also simple: try to match, most of the times but not all, “bad” groups with bad groups and “good” groups with good groups. In both cases, as explained earlier, the recombination would always lead to better solutions; but for the bad groups, we would expect a significant improvement each time.

Formally, we propose a roulette’s procedure similar as the one used in other heuristic approaches like GA. For each group, we measure the corresponding value of $F$ and, after dividing by their sum, we use the result as the selection probability. This way, the larger the individual $F$, the larger the chance of being selected, and vice versa. This is for intra-heterogeneity or inter-homogeneity, otherwise the ones with smaller $F$ would have the larger chance to be selected. If we perform this procedure without replacement $g$ times and make pairs with consecutive groups, we obtain the $g/2$ pairs according to our heuristic criterion. We used this pseudo-random approach, rather than simply ordering groups by their $F$, to prevent the algorithm from getting stuck in local optima from one iteration to the next.

In summary, the proposed method is described as follows.

Algorithm 2.
Step 1: Preprocess student data: convert to non-negative numbers.
Step 2: Generate an initial grouping scheme
Step 3: Select $g/2$ couples of groups according to the roulette mechanism
Step 4: Perform the brute-force search algorithm for each couple and select the optimal local grouping
Step 5: Repeat steps 3 to 4 until an iterations limit is reached. We propose to use a factor of the number of groups

The time complexity of the entire method, considering that g may be expressed as $N/k$, is:

$$O\left(\frac{N^2(2k!)^g}{k!}\right)$$

(7)

Compared with the total number of solutions described in (3), this approach is factorial regarding $k$, but polynomial in terms of $N$. The latter is highly useful from the algorithmic point of view, considering the combinatorial nature of the problem discussed in the previous section for large values of $N$. In addition, the former is not so bad either, considering that for practical implications, the value of $k$ is not larger than a single digit. In fact, weighing all the pros and cons of small and large groups, some authors state that a medium value close to six is optimal (Yannibelli and Amandi, 2012).

As an example for intra-homogeneity, Figure 2 presents the values of $F_1$ for a simulation with $N = 500$, $g = 100$, $k = 5$, and $C = 1$ randomly generated data with a uniform distribution in $[0,1]$. The X-axis represents the iterations and the Y-axis is presented on a logarithmic scale. In addition, the former is not so bad either, considering that for practical implications, the value of $k$ is not larger than a single digit. In fact, weighing all the pros and cons of small and large groups, some authors state that a medium value close to six is optimal (Yannibelli and Amandi, 2012).

For each dataset, we scaled all values to $[0,1]$ and then used values of $g = (2, 3, 4, 5, 6, 7)$ and $k = (10, 20, 50, 100, 200, 500)$, with values of $N$ between 20 and 3500. Using inter-homogeneity as grouping criterion, we ran each scenario ten times using $2g$ as the iterations limit, coding the algorithm in Java and using a MacBook Air with a 1.3 GHz Intel Core i5 processor and 4 GB 1600 MHz DDR3 RAM. The corresponding results are presented in Tables 1, 2, 3, 4. For the running times, we present the average values for the 10 runs, whereas for the values of $F_2$ we present the minimum values. In Table 2, we highlighted the cases where the theoretical optimum of zero for $F_2$ was reached considering at least 10 decimal places.

The average running time values are summarized in Figure 4. We used the main Y-axis for the values of $k = 2,3,4$ and the secondary Y-axis for the values of $k = 5,6,7$. The results in both datasets coincide with the

4. Experimental results

To validate our method and to provide an objective benchmark for further comparisons, we propose the following procedures for the generation of two datasets.

Dataset 1 (the “simplest of all” dataset, with one attribute): Given $N$ students, and considering $C = 1$, $S_h = h$ for $h = 1:N$. In other words, it forces a discrete uniform distribution for just one grouping attribute. This distribution guarantees at least one optimal grouping scheme, i.e., if intra and inter homogeneity are the grouping criteria then $F_1$ and $F_2$ may be zero when $k$ is even and $N$ is a multiple of $k$. Dataset 2 (the “realistic” dataset, with three attributes): Given $N$ students, generate random values for $C = 3$ attributes considering different probability distributions for each. The first attribute resembles a normal distribution in $[0,5]$ and may simulate, for example, a grade on a previous knowledge test. The second attribute resembles a Weibull distribution in $[15, 30]$ and may simulate, for example, the student’s age. The third attribute resembles a geometric distribution in $[1, 5]$ and may simulate, for example, the student’s motivation level. Figure 3 presents the histograms for the three attributes with $N = 3500$. The exact datasets used, as well as the proposed algorithm source code, are available at https://github.com/Blesfiz/Grouping.

We present the average values for the 10 runs, whereas for the values of $F_2$ we present the minimum values. In Table 2, we highlighted the cases where the theoretical optimum of zero for $F_2$ was reached considering at least 10 decimal places.

The average running time values are summarized in Figure 4. We used the main Y-axis for the values of $k = 2,3,4$ and the secondary Y-axis for the values of $k = 5,6,7$. The results in both datasets coincide with the

### Table 5. Related works with reported running times.

| Reference                  | Approach  | Largest dataset                                      | PC features                                                                 | Running time (in seconds) |
|----------------------------|-----------|------------------------------------------------------|------------------------------------------------------------------------------|---------------------------|
| Zervoudakis et al. (2019)  | PSO       | $N = 24, k = 6$, simulated and actual data (not available) | Intel Core i5 3.3 GHz processor and 8 GB RAM memory                          | 14.13                     |
| Dascu et al. (2014)        | PSO       | $N = 25, k = 5$, actual data (not available)         | Not mentioned                                                              | 0.36                      |
| Sukstreniong (2017)        | GA        | $N = 25, k = 3$, actual data (not available)         | Intel Core i5-450M 2.4GHz and 2GB DDR3 RAM                                   | >1                        |
| Sadeghi and Kardan (2016)  | BIP       | $N = 40, k = 8$, actual data (not available)         | Intel Core 2 Duo 2.53 GHz processor and 4 GB RAM memory                     | 373.28                    |
| Balmaceda et al. (2014)    | Other heuristic | $N = 44, k = 10$, actual data (not available) | Intel Dual Core 2.3 GHz processor and 4 GB RAM memory                     | >5                        |
| Krouska and Virvou (2019)  | GA        | $N = 48, k = 17$, simulated data (available)         | Not mentioned                                                               | >1.2                      |
| Pinninghoff et al. (2017)  | GO        | $N = 50, k = 5$, simulated data (available)          | Not mentioned                                                               | >180                      |
| Ho et al. (2010)           | PSO       | $N = 61, k = 4$, actual data (not available)         | Intel Core 2 Duo 2.66 GHz processor and 2 GB RAM memory                     | 0.83                      |
| Lin et al. (2016)          | GA        | $N = 100, k = 5$, random data (not available)        | Intel Core i7 2.1 GHz processor and 4 GB RAM memory                        | 1.35                      |
| Sadeghi and Kardan (2015)  | BIP       | $N = 100, k = 4$, random data (not available)        | Intel Core i7 2.20 GHz processor and 8 GB RAM memory                       | 14591.4                   |
| Lambic et al. (2018)       | Other heuristic | $N = 108, k = 4$, actual data (not available) | AMD FD-7500 2.10 GHz processor and 8GB RAM memory                     | 11.33                     |
| Zheng et al. (2016)        | Other heuristic | $N = 160, k = 4$, random data (not available) | Intel Core i5 3.3 GHz processor and 16 GB RAM memory                       | 2284.93                   |
| Takači et al. (2017)       | Other heuristic | $N = 172, k = 4$, actual data (not available) | Intel Core 2 Duo 2.4 GHz processor and 2 GB RAM memory                     | 17                        |
| Moreno et al. (2012)       | GA        | $N = 180, k = 3$, random data (not available)        | Intel Core i5 4590 3.3 GHz processor and 16 GB RAM memory                   | 0.66                      |
| Zheng et al. (2018)        | GA        | $N = 180, k = 3$, simulated data (not available)     | Intel Core i7 2.1 GHz processor and 2 GB RAM memory                        | 1005.93                   |
| Hwang et al. (2008)        | GA        | $N = 2000, k = 5$, random data (not available)       | Not mentioned                                                               | 87.53                     |
| Lin et al. (2010)          | PSO       | $N = 2000, k = 5$, random data (not available)       | Intel Pentium IV 1.5 GHz processor and 1 GB RAM memory                     | 111.6                     |
| Yannibelli and Amandi (2012)| GA       | $N = 3000, k = 6$, deterministic data (not available) | Not mentioned                                                               | 771.65                     |
| Zheng and Pinkwart (2014)   | PSO       | $N = 3000, k = 3$, random data (not available)       | Intel Core i7 2.0 GHz processor and 8GB RAM memory                        | 809.69                    |


time complexity discussed in section 3, the algorithm exhibits a polynomial running time about $N$ and factorial about $k$.

When comparing these results with those of the 33 research works we analyzed (discussed in section 1), we found that only in 19 of those works the corresponding running times for their largest dataset were reported, as presented in Table 5. When examining each case and comparing them with similar values in terms of $N$ and $k$ in Tables 1 and 3, our proposal demonstrates lower running times in all cases. Of course, this statement is completely arguable due to the diversity of parameters, datasets and PC features of each approach. Nevertheless, and considering that practically none of the implementations are available, in Figure 5 we summarized those comparisons, including the values of Table 3 for $k = 6$ to match the largest dataset found in those 19 works ($N = 3000$).

Finally, the average values of $F_2$ are summarized in Figure 6. For dataset 1, when $k$ is even, the algorithm reaches the optimal value of zero in some scenarios at least once in one of the runs (see Table 2). When $k$ is odd, and a theoretical zero does not exist, the larger the $k$ and the $N$, the better the solutions in terms of the fitness function. For dataset 2, where there are three attributes and each one comes from a different distribution, the fitness function, in general, does not seem to be affected by the size of $N$. Instead, as $k$ increases, the value of $F_2$ decreases.

Another way to look at the results for dataset 2, considering that a theoretical zero is not guaranteed, is analyzing the improvement rate between the value of $F_2$ for the initial grouping scheme obtained at step 2 of Algorithm 1, and the value obtained when step 5 finishes (see section...
3). Figure 7 presents the maximum values of these rates for ten runs of each scenario.

Except for the scenarios with \( k = 2 \), where the grouping attribute distributions, mainly the second and third, make it difficult to achieve inter-homogeneity, the method achieves improvement rates higher than 75%. For values of \( k = 4,5 \), and 6 the rates are close to 90% on average. For dataset 1, the rates are 99.7% on average for all values of \( k \).

5. Conclusions

Many studies have examined group formation in collaborative learning contexts with significant diversity in four main factors: group size, group composition, grouping attributes, and grouping method. Regarding the latter, we proposed a method that uses a local brute-force search approach but immersed into an iterative process guided by a heuristic criterion. And regarding the former three, our proposal i) considers an arbitrary value for the group size; ii) considers an arbitrary array of grouping attributes; and iii) can adapt for various homogeneity/heterogeneity grouping criteria.

After several experiments, we found that our proposal exhibits a good performance. Considering an ideal dataset, it obtains optimal values in some cases and gets close for the rest. It also displays reasonable running times even for a large number of students for that dataset but also for other more “realistic”. We run several scenarios with \( N \) as large as 3500 students and the average running time was less than 30 min. In the 33 research works we analyzed, we did not find any experiment with a sample size that large. In the works that were close, \( N = 3000 \) (Yannibelli and Amandi, 2012; Zheng and Pinkwart, 2014), and even if a direct comparison is not necessarily fair, our running times considering similar values for the group size were at least twice as fast. The datasets are available in a repository along with the source code of our proposal. Future research works can use those resources to make objective comparisons.

These findings do not mean that we recommend our approach in all scenarios. In fact, for small values of \( N \), we think that other approaches like Binary Integer Programming are better options. In (Sadeghi and Kardan, 2015) for example, they report a running time of 243 min with their largest dataset \( N = 100 \), which is not unreasonable. Where we think we are a strong competitor is in scenarios with hundreds or even thousands of students, as long as the group size remains single-digit. Our hypothesis in those cases is that the brute-force search component of our approach allows for better solutions, and when not, for similar solutions with faster rates than strictly heuristic approaches.

As future work, it would be interesting to define another stop criterion besides the number of iterations as a factor of the number of groups. Considering the asymptotic shape of the value of the fitness function across iterations, a minimum threshold for its change rate could be an interesting option.

Declarations

Author contribution statement

J. Moreno: Conceived and designed the experiments; Analyzed and interpreted the data; Wrote the paper.

J. D. Sanchez: Performed the experiments; Contributed reagents, materials, analysis tools or data.

A. F. Pineda: Contributed reagents, materials, analysis tools or data.

Funding statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data availability statement

Data associated with this study has been deposited at https://github.com/Blesfia/Grouping.

Declaration of interests statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

References

Ahmar, S., Orooji, F., Taghiyareh, F., 2012. An evolutionary algorithm for forming mixed groups of learners in web based collaborative learning environments. Proceedings - 2012 IEEE International Conference on Technology Enhanced Education. ICTEE 2012.

Ahi, Z.C., Yasin, A., Husin, M.Z., Hamid, Z.A., 2010. A method for group formation using genetic algorithm. Int. J. Comput. Sci. Eng. 2 (9), 3060-3064.

Balsameda, J.M., Schiaffino, S.N., Diaz-Pace, J.A., 2014. Using constraint satisfaction to aid group formation in GSCIL. Ibero-Am. J. Artif. Intell. 17 (3), 35-45.

Barley, E.F., Cross, K.P., Major, C.H., 2005. Collaborative Learning Techniques: A Handbook for College Faculty, Wiley, San Francisco, CA.

Chase, P.J., 1970. Algorithm 382: combinations of M of N out of objects [G6]. Commun. ACM 13 (6), 366.

Cruz, W.M., Isotani, S., 2014. Group formation algorithms in collaborative learning context: a systematic mapping of the literature. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6658. LNCS, pp. 199–214.

Dascaliu, M.I., Bodex, C.N., Lynets, M., De Pablo, P.O., Burlacu, A., 2014. Improving e-learning communities through optimal composition of multidisciplinary learning groups. Comput. Hum. Behav. 30, 362–371.

Feo, T.A., Mallek, K., 1990. A class of bounded approximation algorithms for graph partitioning. Networks 20 (2), 181-195.

Garshasbi, S., Mohammadi, Y., Graf, S., Garshasbi, S., Shen, J., 2019. Optimal learning group formation: a multi-objective heuristic search strategy for enhancing inter-group homogeneity and intra-group heterogeneity. Expert Syst. Appl. 118, 506-521.

Ghorbani, F., Montazeri, G.A., 2012. Learners grouping improvement in e-learning environment using fuzzy inspired PSO method. In: 3rd International Conference on eLearning and eTeaching, IClELET 2012, pp. 65-70.

Graf, S., Bekele, R., 2006. Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization. Int. Tutor. Syst. 4053, 217–226.

Henry, T.R., 2013a. Forming productive student groups using a massively parallel brute-force algorithm. In: Proceedings of the World Congress on Engineering and Computer Science, I, pp. 23–25.

Henry, T.R., 2013b. Creating effective student groups: an introduction to GroupFormation.org. In: Proceeding of the 44th ACM Technical Symposium on Computer Science Education. ACM, New York, NY, USA, pp. 645-650.

Ho, T.P., Shyu, S.J., Li, C.T.J., 2010. Utilizing particle swarm optimizations on student grouping problem. In: International Conference on Machine Learning and Cybernetics, ICMLC 2010, 4, pp. 1899-1904.

Hubricher, R., 2010. Assigning students to groups using general and context-specific criteria. IEEE Trans. Learn. Technol. 3 (3), 178–189.

Hwang, C., Yin, P., Hwang, C., 2008. An enhanced genetic approach to composing cooperative learning groups for multiple grouping criteria. Educ. Technol. Soc. 11 (1), 148–167.

Isotani, S., Mrsiap, R., 2008. Theory-driven group formation through ontologies. In: Woof, R.P., Almeir, E., Nkambou, R., Lajoie, S. (Eds.), Intelligent Tutoring Systems: 9th International Conference, ITS 2008, Montreal, Canada, June 23-27, 2008 Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 646–655.

Johnson, D.W., Johnson, R.T., 1989. Cooperation and Competition: Theory and Research. Interaction Book, Edina, MN.

Jozan, M.M.B., Taghiyareh, F., 2013. A semi-pareto optimal set based algorithm for grouping of students. In: 2013 Fourth International Conference on E-Learning and E-Teaching (ICELEIT), pp. 10-13.

Kardan, A., Sadeghi, H., 2014. Modeling the learner group formation problem in computer-supported collaborative learning using mathematical programming. In: The 8th National and the 5th International Conference on E-Learning and E-Teaching (ICETEL 2014), pp. 1-5.

Knuth, D.E., 2005. The Art of Computer Programming, Volume 4, Fascicle 3: Generating All Combinations and Partitions. Addison-Wesley Professional, Kreijns, K., Kirschner, P.A., Jochems, W., 2003. Identifying the pitfalls for social interaction in computer-supported collaborative learning environments: a review of the research. Comput. Hum. Behav. 19 (3), 335-355.

Kreijns, K., Kirschner, P.A., Jochems, W., van Buuren, H., 2007. Measuring perceived sociability of computer-supported collaborative learning environments. Comput. Educ. 49 (2), 176-192.
Krouska, A., Virvou, M., 2019. An enhanced genetic algorithm for heterogeneous group formation based on multi-characteristics in social networking-based learning. IEEE Trans. Learn. Technol. 1–12 in Press.

Lambić, D., Lazović, B., Djenić, A., Marić, M., 2018. A novel metaheuristic approach for collaborative learning group formation. J. Comput. Assist. Learn. 34 (6), 907–916.

Lim, J., Yingjin, Z., 2006. The interaction and effects of perceived cultural diversity, group size, leadership, and collaborative learning systems: an experimental study. Inf. Ressour. Manag. J. 19 (4), 56–71.

Lin, Y.S., Chang, Y.C., Chu, C.P., 2016. Novel approach to facilitating tradeoff multi-objective grouping optimization. IEEE Trans. Learn. Technol. 9 (2), 107–119.

Lin, Y.-T., Huang, Y.-M., Cheng, S.-C., 2010. An automatic group composition system for composing collaborative learning groups using enhanced particle swarm optimization. Comput. Educ. 55 (4), 1483–1493.

Long, C., Qing-hong, Y., 2014. A group division method based on collaborative learning elements. In: 26th Chinese Control and Decision Conference (2014 CCDC), pp. 1701–1705.

Moreno, J., Ovalle, D.A., Vicari, R.M., 2012. A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics. Comput. Educ. 58 (1), 560–569.

Payne, W.H., Ives, F.M., 1979. Combination generators. ACM Trans. Math Software 5 (2), 163–172.

Pinninghoff, M.A., Contreras, R., Salcedo, P., 2017. Genetic algorithms as a tool for structuring collaborative groups. Nat. Comput. 16, 231–239.

Ruskey, F., 1988. Adjacent interchange generation of combinations. J. Algorithm 9 (2), 162–180.

Sadeghi, H., Kardan, A.A., 2016. Toward effective group formation in computer-supported collaborative learning. Interact. Learn. Environ. 24 (3), 382–395.

Sadeghi, H., Kardan, A.A., 2015. A novel justice-based linear model for optimal learner group formation in computer-supported collaborative learning environments. Comput. Hum. Behav. 48, 436–447.

Shaw, R.-S., 2013. The relationships among group size, participation, and performance of programming language learning supported with online forums. Comput. Educ. 62, 196–207.

So, H.J., Brush, T.A., 2008. Student perceptions of collaborative learning, social presence and satisfaction in a blended learning environment: relationships and critical factors. Comput. Educ. 51 (3), 318–336.

Strijbos, J.W., Martens, R.L., Jochems, W.M.G., 2004. Designing for interaction: six steps to designing computer-supported group-based learning. Comput. Educ. 42 (4), 403–424.

Strijbos, J.W., Martens, R.L., Jochems, W.M.G., Broers, N.J., 2007. The effect of functional roles on perceived group efficiency during computer-supported collaborative learning: a matter of triangulation. Comput. Hum. Behav. 23 (1), 353–360.

Suksrienswong, A., 2017. A genetic-algorithm approach for balancing learning styles and academic attributes in heterogeneous grouping of students. Int. J. Emerg. Technol. Learn. 12 (3), 4–25.

Strijbos, J.-W., Martens, R.L., Jochems, W.M.G., Broers, N.J., 2007. The effect of functional roles on perceived group efficiency during computer-supported collaborative learning: a matter of triangulation. Comput. Hum. Behav. 23 (1), 353–360.

Zervoudakis, K., Mastrophanasis, K., Tsafarakis, S., 2019. Forming automatic groups of learners using particle swarm optimization for applications of differentiated instruction. Comput. Appl. Eng. Educ. 1–11.

Zheng, Y., Li, C., Liu, S., Li, W., 2018. An improved genetic approach for composing optimal collaborative learning groups. Knowl. Base Syst. 139, 214–225.

Zheng, Z., Pinkwart, N., 2014. A discrete particle swarm optimization approach to compose heterogeneous learning groups. In: 14th IEEE International Conference on Advanced Learning Technologies, pp. 14–16.

Zhou, M., Ye, Y., Ren, Y., Xu, Y., 2016. Collaborative learning group formation with density clustering. Int. J. Grid Distrib. Comput. 9 (6), 117–126.