Sánchez, J. (2024) Editor. Nuevas Ideas en Informática Educativa, Volumen 18, p. 69 - 79. Santiago de Chile.

# Group Formation in Programming Teaching: A Systematic Review of Literature

**Ilmara M. M. Ramos**
Instituto de Computação (ICOMP) - Universidade Federal do Amazonas (UFAM)
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM)
Parintins, Brasil
ilmaramonteverde@icomp.ufam.edu.br

**David Brito Ramos**
Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM)
Manaus, Brasil
david.brito@ifam.edu.br

**Bruno Gadelha**
Instituto de Computação (ICOMP) - Universidade Federal do Amazonas (UFAM)
Manaus, Brasil
bruno@icomp.ufam.edu.br

**Elaine H. T. de Oliveira**
Instituto de Computação (ICOMP) - Universidade Federal do Amazonas (UFAM)
Manaus, Brasil
elaine@icomp.ufam.edu.br

## ABSTRACT
This article presents the results of a Systematic Review of Group Formation in Programming Teaching. We collected publications from SCOPUS, IEEE, ACM, and Web of Science databases. The objective was to obtain information about how groupings that require collaboration in programming teaching are constructed, including group types, data collection methods, techniques/methods for forming groups, validation or evaluation of research experiments, and the courses and subjects in which they were applied. We found 8 papers published between 2011 and 2021. Some conclusions were: that approaches to forming groups include automatic selection; genetic algorithms are predominant in suggesting group formation; and works are mostly in higher education in Computing.

### Author Keywords
Group Formation; Systematic Review; Programming Teaching.

### ACM Classification Keywords
Applied computing-> Education-> Collaborative learning

## INTRODUCTION
Computer-supported collaborative learning - CSCL significantly contributes to the teaching-learning process by promoting communication and interaction among students [15]. For collaborative learning to be successful in any teaching modality, it is essential to form groups that can achieve their goals satisfactorily [19]. Forming groups is a complex process, whether in face-to-face or distance courses [2], especially when it comes to programming teaching.

Learning programming can be a challenging and non-intuitive process. Therefore, many educational institutions frequently review their curricula in Computer Science courses, seeking alternatives to reduce dropout rates [26].

In this context, forming groups in programming education becomes even more challenging in any modality, whether face-to-face, blended, or distance learning. This article presents the results of a Systematic Review of Literature (SRL) on group formation in programming education. The objective is to obtain information on how groupings that require collaboration in programming teaching are constructed, what types of groups are formed, the data collection methods, the techniques, and methods used to form groups, whether the experiments were validated or evaluated, and in which courses and disciplines the experiments were applied.

The results show the main data obtained by this research regarding group formation in programming teaching. This article is structured as follows: Section 2 describes the theoretical foundation, Section 3 describes related works, Section 4 the methodology used in this research, Section 5 the review conduction, Section 6 results and discussions, and Section 7 final considerations.

## THEORETICAL FOUNDATION
This section will address the theoretical foundations that will support the research, aiming to assist in the understanding of the technologies used to support its development. The theoretical principles related to Collaborative Learning,

Paste the appropriate copyright/license statement here. ACM now supports three different publication options:

- ACM copyright: ACM holds the copyright on the work. This is the historical approach.
- License: The author(s) retain copyright, but ACM receives an exclusive publication license.
- Open Access: The author(s) wish to pay for the work to be open access. The additional fee must be paid to ACM.

This text field is large enough to hold the appropriate release statement assuming it is single-spaced in Times New Roman 8-point font. Please do not change or modify the size of this text box.
Each submission will be assigned a DOI string to be included here.

Group Formation, and Collaborative Learning in Programming Teaching will be presented.

### Collaborative Learning

Collaborative learning is a pedagogical approach that emphasizes interaction between students to achieve common goals, promoting the exchange of knowledge and skills. The work of [4] reports that collaborative learning is an educational approach that places the student at the center of the teaching process, promoting group interaction and cooperation.

According to [5], collaborative learning occurs when a group of people is motivated to learn together. This learning brings significant benefits to the teaching-learning process of students, enabling the formation of groups. Collaborative learning is an essential approach in the formation of groups in programming teaching. It allows students to work in pairs or teams to solve problems, discuss ideas, and build knowledge together. In this way, students face the complex challenges of coding together, improving technical and social skills.

### Group Formation

To carry out collaborative activities, it is necessary to form groups of students so that they can carry out the activities. To do this, it is essential to understand the context of the group that will be approached to organize the learners. According to [25], a group is a small set of individuals who come together through a specific peculiarity. To facilitate this meeting, several particularities must be considered, such as approaches, criteria, characteristics, and student data. In this grouping situation, it is important to determine how to generate this motivation for involvement. Groups are basic social structures and, both inside and outside the academic world, they form and change in various ways for multiple purposes.

According to [4], groups can be formed manually or automatically within the system. Manual formation occurs when the grouping is created outside the system, that is, the teacher forms the group according to established criteria and then informs the system. This approach is common in face-to-face courses. On the other hand, automatic formation occurs when the system recommends the composition of the group based on criteria defined by the teacher, resulting in the automated selection of group members.

However, for collaborative learning to be successful, it is important to form groups that can be effective and efficient in achieving the task objectives. Collaborative groups generally range from two to six students. According to [2], groups of two students tend to work better in most cases because they allow for quick exchanges and have fewer interruptions. Additionally, smaller groups are more effective because they promote greater involvement and facilitate coordination among members.

To form student groups, it is important to establish which group approach will be used. In this regard, the research conducted by [16] reports three types of approaches for group formation: Random, Self-Selected, and Selected. Random: Groups are proposed by the teacher, who groups the students without any defined criteria. Self-Selected: Students are allowed to choose the group they want to belong to. Selected: The teacher defines the criteria for the system to form the groups.

Additionally, it is necessary to determine the type of data collection that will be used to identify user preferences. According to [14], information can be collected in two ways: Explicit and Implicit. Explicit: Information is directly requested from users through requirement elicitation techniques, such as questionnaires. Implicit: Information is extracted from user navigation without them having to provide it directly, allowing their needs and preferences to be inferred. Another requirement to consider is the type of group classification that will be used for group formation. In the research by [4], group classifications were adapted to the following criteria: Homogeneous: Group members have similar characteristics. Heterogeneous: Group members have different characteristics. Both: Allows the formation of both homogeneous and heterogeneous groups.

All these approaches described above will serve as a basis for understanding and addressing the questions raised in the Systematic Review of Literature (SLR), which is the focus of this article.

### Collaborative Learning in Programming Teaching

In programming teaching, collaborative learning can be implemented in various ways, including group projects, pair programming, and communities of practice. These approaches allow students to share their experiences, discuss solutions, and learn from each other, which can lead to a deeper understanding of programming concepts [28][8].

In programming teaching, collaborative learning can be implemented in several ways, including:

- Group Projects: Students work together on programming projects, sharing responsibilities and collaborating to achieve a common goal [6].
- Pair Programming: Two students work together on a single computer, alternating between the roles of “driver” and “navigator”. This practice promotes knowledge exchange and joint problem-solving [28][21].
- Communities of Practice: Groups of students who share a common interest in programming meet regularly to discuss problems, share solutions, and learn from each other[11].

## RELATED WORKS

The following are the works related to this research. The proposal by [20] describes a quasi-systematic literature review on group formation in virtual learning environments, aiming to gather information on how groups are constructed for collaborative activities in Virtual Environments. As a result of the review, 19 publications related to the proposed theme were identified, with the research being published between the years 2002 and 2015.

The research by [15] presents a literature review on group formation in computer-supported collaborative learning, to update information on group formation and compare the results with previously conducted research within the related theme. As a result of the review, 13 publications were identified for data extraction, with the researchs spanning the period from 2009 to 2019.

The work by [9] reports on a systematic review of literature on clustering solutions in collaborative learning, to construct a model for collaborative learning group formation to meet the needs of Computer Supported Collaborative Learning (CSCL) and Mobile-Supported Collaborative Learning (MSCL), by examining the existing literature. The result of the systematic literature review (SLR) was 8 publications, with the research spanning the years 2006 to 2017.

On the other hand, the research by [17] describes a systematic review of group formation in intelligent learning environments, to conduct a SRL on the need for intelligent technology for group formation in an intelligent learning environment. As a result of the review, 22 publications were presented, covering the periods from 2001 to 2018.

This research distinguishes itself from others by conducting a Systematic Review of Literature (SRL) on group formation in programming teaching, investigating how groupings that require collaboration are structured in this context, with no limitation on the publication period of the articles. The following section describes the methodology used to conduct this research.

## METHODOLOGY

This SRL is fundamentally based on the guidelines of [10] and, according to [27], the review can be classified as a quasi-systematic review, as it is an exploratory research that preserves formalism, does not make comparisons between publications, and follows the same process as a systematic review.

The main objective of this research was defined based on the need to understand how group formation processes have been applied in programming teaching. The description of the objective according to the Goal-Question-Metric Paradigm of [3] can be observed below:

To analyze scientific publications through a study based on a systematic review. To identify how the process and creation of group formation used in the literature for programming teaching occurs, in relation to student grouping. From the researchers' point of view, through the academic or industrial context. The SRL sought to answer the following main question (MQ): How are groupings constructed to carry out collaborative activities in programming education?

The following sub-research questions were defined to answer the main question (MQ):

SQ1. What techniques/algorithms are used in the formation of student groups in programming teaching?

SQ2. What type of data collection is used?

SQ3. What data sources are extracted for group formation?

SQ4. What types of methods are used for group formation?

SQ5. What criteria are used for group formation?

SQ6. What is the target audience of the accepted research?

SQ7. What research data are used for group formation?

### Publication Search Method

To perform the automated search for the SRL, the following Academic Search Engines (ASE) were used: Computer Science Digital Library (IEEE), Scopus, Association for Computing Machinery (ACM), and Web of Science, with no publication date limits. Additionally, [10] states that Scopus is the largest abstract and citation indexing database, providing an overview of the world's scientific production in the fields of science, technology, medicine, social sciences, arts, and humanities.

### Search Expression

To arrive at the search string used in the digital libraries of Scopus, IEEE, ACM, and Web of Science, tests were conducted based on a control article. Control Articles are those that serve as a reference to compare the results of the studies included in the systematic review. These well-known and established works should appear in the results during the execution of the search string. The search term used in this SLR is illustrated below in Table 1. In constructing the search string, the keywords that should be contained in the publications, as well as their synonyms or alternative terms, were identified. The keywords of the search expression are presented below. For searches in ACM, quotation marks ("") were replaced by brackets ([]).

| Terms | Alternative Terms | Search String |
|---|---|---|
| Group Formation | Group Creation | ( ( "group formation" **OR** "group creation" ) **AND** "collaborative learning" **AND** ("computer programming" **OR** "programming" ) ) |
| Collaborative Learning | Programming | |

**Table 1. Search String.**

| IC | EC |
|---|---|
| **IC1**.Articles that present representations of models, techniques, processes, and clustering algorithms for the formation of student groups in programming education. | **EC1.** Publications in which the search keywords do not appear in the title, abstract, and/or keywords will not be selected<br>**EC2**. Does not meet the inclusion criteria.<br>EC3. Publications that do not have content available for reading (for example, where the works are paid or not provided by search engines or authors) will be excluded.<br>**EC4**. Articles that only present experience reports of a model, technique, or tool.<br>**EC5.** Articles that only mention group formation but do not address it as the main focus.<br>EC6. Publications that do not relate to group formation and programming education will not be selected. Publications that do not relate group formation and programming education will not be selected. |

**Table 2. Inclusion Criteria (IC) and Exclusion Criteria (EC)**

### Study Selection and Data Extraction

Table 2 presents the Inclusion Criteria (IC) and Exclusion Criteria (EC) used during the execution of the SRL. The selection of articles was based on the IC and EC listed below, which were created from the control articles.

The entire process of data extraction obtained during the execution of the SLR is described below.

### Data Extraction

The data extracted during the reading of the articles approved in the first and second filters followed a list of predefined items, as presented in Table 4.

The data extracted during the reading of the articles approved in the first and second filters followed a list of predefined items, as presented in Table 4. The items considered were:

- Research: Field refers to the numbering of the article resulting from the SRL.
- Form: Types of group formation, which can be automatic, manual, or a combination of both.
- Criterion: Whether the group formation is homogeneous, heterogeneous, or hybrid (homogeneous and heterogeneous).
- Type of Data Collection: Whether the collection of information used during the research was conducted explicitly or implicitly.
- Target Audience: Whether the research participants were technical education students, undergraduate, or postgraduate, among others.
- Techniques or Algorithms: Which techniques or algorithms were used during the group formation?

Other relevant data extracted during the execution of the SLR included the types of methods used to form the groups, whether by automatic selection, selected by the teacher, or selected by the students, among others.

The sources of data used to collect the information, whether by questionnaires, forms, access logs, or others. Another piece of information was in which subjects the experiments were conducted, in which courses they were applied, whether the experiments were validated or evaluated, and finally, which research data or attributes were used for group formation, such as personality traits, and learning paths, among others.

The following section how describes the SLR was conducted and which databases were used in this SLR.

### CONDUCTING THE REVIEW

Once all the planning part of the review was completed, the RSL conducting phase began in May 2022, resulting in the execution of the search string in the digital libraries, 362 searches, of which 265 in the Scopus database, 9 in the IEEE, 67 in the ACM and 21 in the Web of Science database, the result per database is shown in Table 3.

According to Table 3, the Scopus database presented the highest concentration of research, returning 265 publications representing 73.2% of the total research returned, unlike IEEE, which returned only 9 works, representing the lowest number of research with only 2.5% of the total publications. From 362 publications, we found 45 duplicate publications, leaving 317 articles for analysis by the first filter.

| Database | Number of Papers | % |
|---|---|---|
| SCOPUS | 265 | 73,2% |
| IEEE | 9 | 2,5% |
| ACM | 67 | 18,5% |
| Web of Science | 21 | 5,8% |
| **Total** | **362** | **100%** |

**Table 3. Number of research returned from automated search.**

In the first filter, we analyzed 317 articles according to the inclusion (IC) and exclusion (EC) criteria described in Section 3.3. Of these 317, 3 are from IEEE, 63 from ACM, 8 from Web of Science, and 243 from SCOPUS. The title, abstract, and keywords of each publication were analyzed to verify which articles are useful for the context of this research. The analysis of each article was according to each criterion, as illustrated in Figure 1.

To execute the second filter, 27 articles resulting from the analysis of the first filter were accepted. In the second filter process, all articles were read in full and analyzed according to the selection criteria described in Section 3.3. After the execution of the second filter, 8 publications were accepted and 19 rejected, as illustrated in Figure 2.

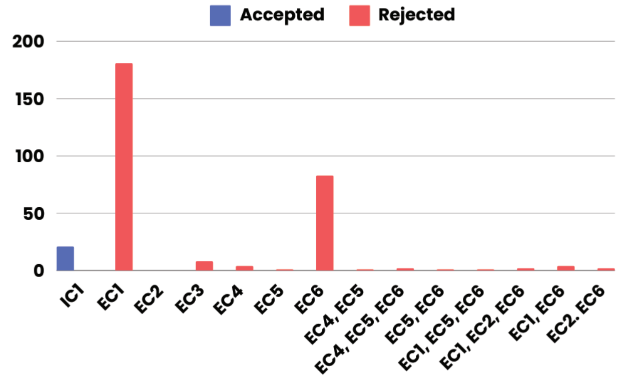


**Figure 1. Results of the first filter.**

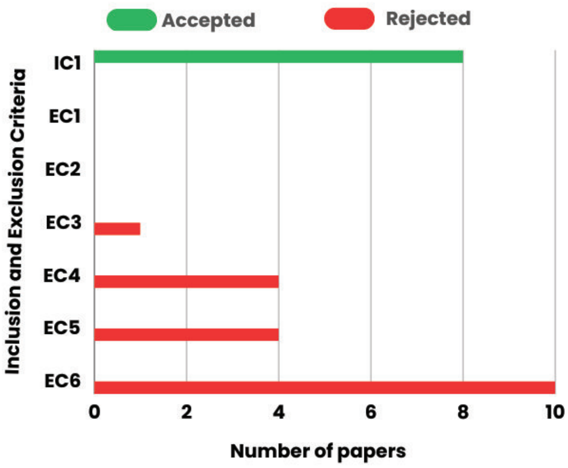


**Figure 2. Results of the second filter.**

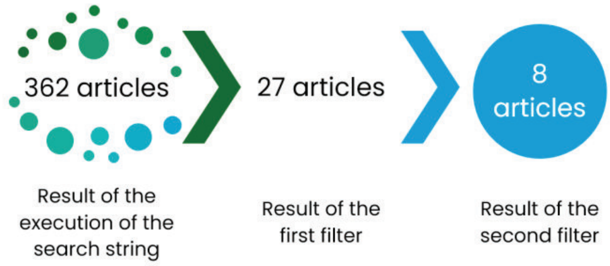


**Figure 3. SRL flow diagram.**

Figure 3 represents the entire RSL execution process and the result of each step.

Next, we carry out analyses and discussions regarding the information obtained.

## DATA ANALYSIS AND DISCUSSION

In RSL, the main question addressed was how to build groups to carry out collaborative activities in programming teaching. At the end of RSL, we identified 8 relevant publications. We present the information in Table 4.

This model was based on the taxonomy of [4]. All items are mandatory, except when not informed (-) or when they do not apply to the publication.

As a result, it was noticed that to form groups in the research, the data collection used to extract information about the students' preferences was carried out explicitly by 7 works and implicitly by 1 work, responding to SQ2.

Regarding the types of methods used to form groups (SQ4), 3 studies performed it automatically. In 1 study, the groups were selected by the teacher. The same occurred for the joint selection methods, such as those selected by the teacher and student selection. Regarding automatic selection and self-selection, two studies combined teacher selection and automatic selection. Finally, two studies used a joint selection of those selected by the teacher and those selected automatically. In other words, the studies performed the formation of groups in the sequence of first the teacher selecting the data, and then the tool automatically generating the groups based on the data defined by the teacher. As can be seen in Figure 4.

| Research | 1 [22] | 2 [23] | 3 [24] | 4 [7] | 5 [12] | 6 [13] | 7 [1] | 8 [18] |
|---|---|---|---|---|---|---|---|---|
| **Form** | Automatic | Automatic | Automatic | Automatic | Automatic | Manual and Automatic | Automatic | Automatic |
| **Criterion** | Homogeneous | Homogeneous | Homogeneous, Heterogeneous | Homogeneous and Heterogeneous | Homogeneous and Heterogeneous | - | Homogeneous and Heterogeneous | Heterogeneous |
| **Collection Type** | Explicit | Explicit | Explicit | Explicit | Explicit | Explicit | Explicit | Implicit |
| **Target audience** | Undergraduate Students | Undergraduate Students | Undergraduate Students | Postgraduate Students (PhD) | Postgraduate Students (PhD) | Undergraduate Students | Undergraduate Students | - |
| **Technique/Algorithm** | Genetic Algorithm | Genetic Algorithm | Genetic Algorithm | Fuzzy Logic | Fuzzy Logic | Network-flow-based Matching Algorithm in Bipartite Graphs | Group Organizer tool | Adaptive Genetic Algorithm |
| **Types of Methods** | Automatic Selection | Automatic Selection | Selected by professor | Selected by professor and Automatic Selection | Selected by professor and Automatic Selection | Selected by professor and selected by students | Selected by professor and Self-selection | Automatic Selection |
| **Data Source** | Questionnaire | Questionnaire | Questionnaire | Questionnaire | Questionnaire | Form | Questionnaire | - |
| **Discipline** | Programming I and Fundamentals of Programming | Programming I and Fundamentals of Programming | Programming | Computer Programming and Data Mining | Computer Programming and Data Mining | Algorithms | Algorithm and data structure | Basic Programming |
| **Validation or Evaluation** | Validation | Validation | Validation | Validation | Validation | Validation | Validation | Validation |
| **Course** | Systems Engineering and Electronics | Systems Engineering and Electronics | Systems Engineering | Computer Science | Computer Science | Computer Science | Bachelor of Science in Computer Science | - |
| **Research Data** | Personality Traits | Personality Traits | Personality Traits | Data Depth | Data Depth | Ranks students (Grade Point Average (GPA) or performance in discrete mathematics) | Social-cognitive conflict and learning styles | Exam score and participation score |

**Table 4. Data extracted during the reading of research.**


**Figure 4. Result of Method Types.**

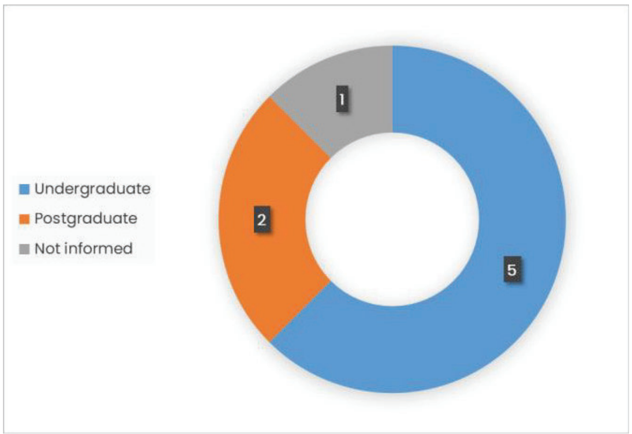


**Figure 5. Target Audience of Research.**

During the execution of the research experiments, the target audience (SQ6) was verified, as shown in Figure 5. Five studies were carried out with undergraduate students and two with Stricto Sensu Postgraduate (Doctorate) students; one study did not identify its target audience. An interesting fact is that none of the studies carried out their experiments with high school technical education students.

The algorithms used to form groups in programming teaching are presented in Figure 6 and respond to SQ5. In 4 works genetic algorithms were used, two used Fuzzy Logic, one work used a proprietary tool called Group-Organizer, and, finally, one work used a matching algorithm based on bipartite flow.

The data source used to extract user information was through questionnaires in 6 of the 8 studies, one study used a form, and one study did not inform its data source. Thus, we answered SQ3.

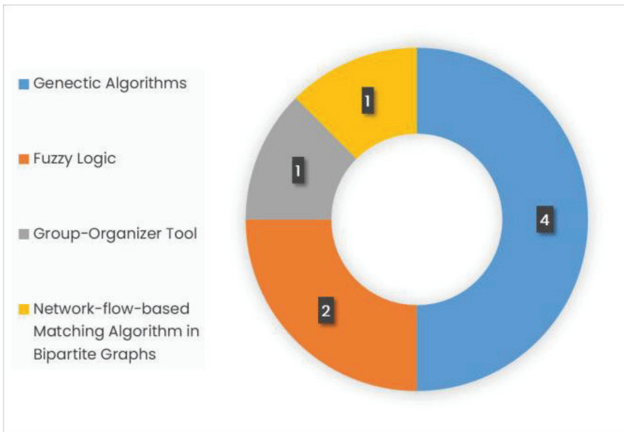


**Figure 6. Algorithms used in research.**

In all studies, the experiments were 100% validated. In response to SQ7, the research data or attributes that were used to form the groups in three studies were personality traits, in two studies they were in-depth data, in one study it classified students by grade, specifically by weighted average, in another study it used cognitive conflict and learning styles, and in the last study it used a combination of assessment grades and participation.

In response to question SQ5, 4 of the 8 publications gave the option of grouping students using either homogeneous or heterogeneous criteria, two works addressed only homogeneous groups, one work formed heterogeneous groups and one work did not present this information.

Figure 7 shows the evolution of the selected research that uses group formation in programming teaching over the years. The temporal distribution showed that despite not obtaining a significant number of selected publications per year, there was an increase in publications between 2011 and 2021, which demonstrates the interest of researchers in group formation work in programming teaching. Some general information about these 8 articles is listed in Table 5 below.

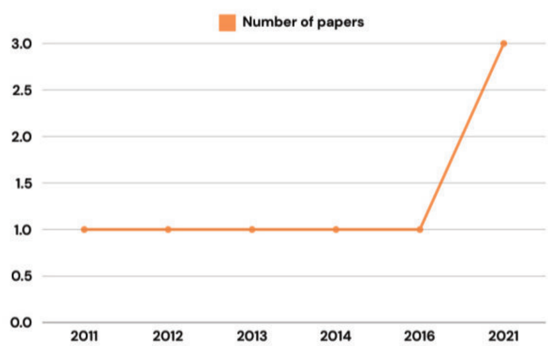


**Figure 7. Year of Publication.**

| Number of Research | Year of Publication | Source | Paper | Journal or Conference |
|---|---|---|---|---|
| 1 | 2021 | IEEE | Homogeneous Group Formation in Collaborative Learning Scenarios: An Approach Based on Personality Traits and Genetic Algorithms[22] | IEEE Transactions on Learning Technologies |
| 2 | 2021 | WEB OF SCIENCE | Collaborative Learning Group Formation Based on Personality Traits: An Empirical Study in Initial Programming Courses[23] | Interaction Design and Architecture(s) |
| 3 | 2021 | SCOPUS | A strategy based on genetic algorithms for forming optimal collaborative learning groups: An empirical study[24] | Electronics |
| 4 | 2014 | WEB OF SCIENCE | Analyzing collaboration and interaction in learning environments to form learner groups[7] | Computers in Human Behavior |
| 5 | 2013 | WEB OF SCIENCE | A method to form learners groups in computer-supported collaborative learning systems[12] | First International Conference on Technological Ecosystems for Enhancing Multiculturality |
| 6 | 2012 | ACM | Forming project groups while learning about matching and network flows in algorithms[13] | Forming Project Groups while Learning about Matching and Network Flows in Algorithms |
| 7 | 2011 | SCOPUS | Forming groups for collaborative learning of introductory computer programming based on students' programming skills and learning styles [1] | International Journal of Information and Communication Technology Education |
| 8 | 2016 | SCOPUS | Student grouping using adaptive genetic algorithm[18] | International Electronics Symposium |

**Table 5. General information about selected articles.**

SRL identified 8 relevant publications that address this issue. The data extracted from these studies are presented in Table 4, which details the forms of grouping, criteria, types of data collection, target audience, techniques/algorithms used, selection methods, data sources, disciplines, validation or evaluation, courses, and specific research data.

The results presented in this section show the data currently used in the literature to build clusters and carry out collaborative activities in programming teaching. Data collection to extract information about students' preferences was carried out explicitly in seven studies and implicitly in one study.

Regarding group formation, the automatic method was highlighted in seven articles, while a combination of automatic and manual methods was used in one article. Regarding the methods used for group formation, three studies used automatic methods, one study had groups selected by the professor, and others combined professor selection with automatic selection or self-selection.

Regarding the classification criteria of the groups, two studies used homogeneous criteria, one study used heterogeneous criteria, and the others combined homogeneous and heterogeneous criteria. The predominant target audience of the articles was undergraduate students in five studies, postgraduate students in two studies, and one study did not identify the target audience.

The most used data source to collect information was the questionnaire, present in six studies, one work used form, and one study did not identify the data source. All studies performed validation.

The courses in which the experiments were performed include three in Computer Science, one in Computer Science Degree, one in Systems Engineering, two in Systems and Electronics Engineering, and one unidentified. Regarding the techniques/algorithms used in the publications, four works used genetic algorithms, two used Fuzzy logic, one used a matching algorithm based on network flow in bipartite graphs, and one used the Group Organizer tool.

Finally, the predominant data in the research were personality traits, present in three studies. Two studies used in-depth data, two used grades, and one used student classification (weighted average - GPA or performance in discrete mathematics). These approaches help to understand how clusters are built to carry out collaborative activities in programming teaching, answering the main question (MQ).

## LIMITATIONS OF THE REVIEW

The SRL is conducted through a robust and objective protocol that governs the entire review process; however, this does not completely eliminate the researchers' inherent subjectivity. Concurrently, the exclusion of any relevant work, which might be eliminated for not meeting some of the selection criteria, is always an imminent possibility. In this context, it is possible that the execution of the first filter may have excluded some relevant works. This is because publications that did not contain the search terms in the keywords, titles, or abstracts are discarded at this stage, and no full reading is carried out. Another limitation arises from subjective decisions made during the reading and comprehension process of the works, as some studies did not present a clear description, making it difficult to directly apply the inclusion and exclusion criteria.

## CONCLUSION

This article presented the data produced during the SRL on the state of the art of works related to group formation in programming teaching. The analysis focused on key aspects to obtain detailed information, aiming to identify how the works have directed their solutions. The results showed that the publications frequently use automatic group formation, explicit data collection through questionnaires, homogeneous and heterogeneous criteria, and genetic algorithms. The main target audience is undergraduate students in Computer Science courses, and the experiments are validated.

The importance of these results lies in understanding how group formation can be optimized to improve programming teaching. The use of genetic algorithms and explicit data collection indicates a trend towards seeking precise and personalized solutions for group formation, which can lead to more effective and collaborative learning. Additionally, the validation of the experiments reinforces the reliability of the approaches used.

This work significantly contributes to the field of AI and Informatics in Education by gathering and analyzing the aspects that the current literature presents on group formation in programming teaching. By identifying the most effective practices and methodologies, this study provides a solid foundation for future research and developments in the area.

## FUTURE WORK

Based on the results of this systematic literature review, several directions for future research can be explored to enhance group formation and collaboration in programming education. Below are suggestions for future directions:

- **Exploration of New Group Formation Methodologies:** The analysis revealed a predominance of automatic methods, such as genetic algorithms and Fuzzy Logic. Future research could investigate the effectiveness of other methodologies, such as neural networks or machine learning algorithms, which could potentially offer more personalized and adaptive groupings.
- **Application in Different Educational Levels:** It was noted that most studies focused on undergraduate and postgraduate students, leaving a gap in research on group formation in high school contexts, especially in technical education. Investigating how these methodologies can be applied to these groups could contribute to more effective programming education from early levels.
- **Evaluation of Selection Methods Effectiveness:** The combination of automatic and manual methods in group formation was used in some studies. Future research could compare the effectiveness of these methods, analyzing how students' perceptions of their inclusion and group dynamics affect learning and collaboration.
- **Diversification of Data Sources:** Although questionnaires were the main data source, there is an opportunity to explore more dynamic data collection methods, such as analyzing interactions on online learning platforms, real-time feedback, and qualitative observations. This could enrich the understanding of students' preferences and group formation.
- **Validation of Results in Diverse Contexts:** The validation of approaches used in the studies is a strong point; however, it is essential to replicate these validations in varied contexts, including different disciplines and educational environments. This would help better understand the generalization of group formation methodologies.
- **Investigation of Alternative Classification Criteria:** While the research focused on characteristics such as personality traits and academic performance, it would be interesting to explore additional criteria, such as learning styles, motivations, and previous experiences, which could offer a more holistic view of group formation.
- **Longitudinal Analysis of Group Dynamics Impact:** Studies that follow the evolution of groups over time can provide valuable insights into the effectiveness of formation methodologies. This longitudinal analysis would help identify performance and collaboration patterns that develop in response to different grouping strategies.

These future works will not only expand the understanding of group formation in programming education but also contribute to the evolution of pedagogical practices, promoting more collaborative and effective learning environments.

## ACKNOWLEDGMENTS

To the Federal Institute of Education, Science and Technology of Amazonas -IFAM Manaus East Zone campus and Parintins Campus. This study was funded in part by the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES-PROEX) - Finance Code 001 and was partially supported by the Amazonas State Research Support Foundation - FAPEAM - through the POSGRAD 2024/2025 project. In addition, this research, carried out within the scope of the Samsung-UFAM Project for Education and Research (SUPER), according to Article 39 of Decree n°10.521/2020, was funded by Samsung Electronics of Amazonia Ltda., under the terms of Federal Law n°8.387/1991 through agreement 001/2020, signed with UFAM and FAEPI, Brazil.

**REFERENCES**


1. Adán-Coello, J. M.; Tobar, C. M.; José de Faria, E. S.; Serafim de Menezes, W.; Luís de Freitas, R. Forming Groups for Collaborative Learning of Introductory Computer Programming Based on Students' Programming Skills and Learning Styles. International Journal of Information and Communication Technology Education (IJICTE). 2011, 7(4), 34-46. https://doi.org/10.4018/jicte.2011100104
2. Barkley, E. F., Major, C. H. e Cross, K. P. (2014) "Collaborative Learning Techniques", 2 ed, San Francisco, Jossey-Bass. Knuth, D. E. (1984), The TeXbook, Addison Wesley, 15th edition.
3. Basili, V., Caldiera, G. e Rombach, H. (1994) "The Experience Factory". Encyclopedia of Software Engineering. New York: John Wiley & Sons.
4. Citadin, J., Kemczinski, A. e Matos, A. (2014) "Formação de Grupos para Aprendizagem Colaborativa: Um Mapeamento Sistemático da Literatura", In: XIX Conferência Internacional sobre Informática na Educação, 19,v. 10,p. 46-54.
5. Dillenbourg, P., Baker, M. Blaye, A. e Malley, C. (1996) "The Evolution of Research on Collaborative Learning", In: Spada, E., Reiman, P.(eds) Learning in Humans andMachine: Towards an Interdisciplinary Learning Science, p.189-211. Elsevier, Oxford
6. De Jesus, Â. M., & Silveira, I. F. (2022). A Collaborative Learning Framework for Computational Thinking Development through Game Programming. Informatics in Education, 21(2), 253-281. https://doi.org/10.15388/infedu.2022.141
7. Duque, R; Gómez-Pérez, D.; Nieto-Reyes, A.; Bravo, C. Analyzing collaboration and interaction in learning environments to form learner groups. Computers in Human Behavior. 2015, 47, 42-49.https://doi.org/10.1016/j.chb.2014.07.012
8. Garcia, M.B. Cooperative learning in computer programming: A quasi-experimental evaluation of Jigsaw teaching strategy with novice programmers. Educ Inf Technol 26, 4839–4856 (2021). https://doi.org/10.1007/s10639-021-10502-6
9. Ge, Z.; Qi, Z e Yang, L. (2018) "A Literature Review of Grouping Solutions in Collaborative Learning", 2018 IEEE International Conference on Progress in Informatics and Computing (PIC), 2018, pp. 393-397, doi: 10.1109/PIC.2018.8706326
10. Kitchenham, B. e Chartes, S. (2007) "Guidelines for performing Systematic Literature Reviews in Software Engineering" EBSE Technical Report EBSE-2007-01, Software Engineering Group Department of Computer Science Keele University
11. Li, W., Liu, CY. & Tseng, J.C.R. Effects of the interaction between metacognition teaching and students' learning achievement on students' computational thinking, critical thinking, and metacognition in collaborative programming learning. Educ Inf Technol 28, 12919–12943 (2023). https://doi.org/10.1007/s10639-023-11671-2
12. Medina, R. D.; Gómez-Pérez, D.; Nieto-Reyes, A.; Santos, C. B. A method to form learners groups in computer-supported collaborative learning systems. In Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality (TEEM '13), Salamanca, Spain, 14 – 15 November 2013; pp. 261–266. https://doi.org/10.1145/2536536.2536576
13. Mehta, Dinesh; Kouri, Tina; Polycarpou, Irene. Forming project groups while learning about matching and network flows in algorithms. In Proceedings of the 17th ACM annual conference on Innovation and technology in computer science education (ITiCSE '12), Haifa, Israel, 3 – 5 July 2012; pp. 40–45, doi: 10.1145/2325296.2325310
14. Neto, M. (2011) "Estratégia Híbrida para Recomendação Personalizada utilizando o Guia de Programação Eletrônico".106p. Dissertação (Mestrado Acadêmico em Ciência da Computação), Universidade Estadual do Ceará, Fortaleza.
15. Oliveira, L. C., Rosa, S. S. e Pimentel, A. R. (2019). Revisão Sistemática da Literatura: Formação de Grupos na Aprendizagem Colaborativa com Suporte Computacional. In: XXX Simpósio Brasileiro de Informática na Educação, 30, p. 1955-1964.
16. Ounnas, A.; Davis, H. C.; Millard, D. E. Towards semantic group formation. In: Proceedings of the 7th IEEE International Conference on Advanced Learning Technologies, ICALT 2007, p. 825–827, 2007.
17. Putro, B. L., Rosmansyah, Y and Suhardi. (2018) "Group Formation in Smart Learning Environment: A Literature Review," International Conference on Information Technology Systems and Innovation (ICITSI), 2018, pp. 381-385, doi: 10.1109/ICITSI.2018.8695917.
18. Putu I. Ciptayani,Kadek C. Dewi, Wayan B. Santana.(2016). Student grouping using adaptive genetic algorithm. In *2016 International Electronics Symposium (IES)* (pp. 1-6). IEEE. https://doi.org/10.1109/ELECSYM.2016.7861034
19. Ramos, I. M. M., Ramos, D. B., Gadelha, B. F. and Oliveira, E. H. T. (2021). An Approach to Group Formation in Collaborative Learning Using Learning Paths in Learning Management Systems. In: IEEE Transactions on Learning Technologies, vol. 14, no. 5, pp. 555-567, 1 Oct. 2021, doi: 10.1109/TLT.2021.3117916.

20. Ramos, I. M. M., Ramos, D. B., Nascimento, P. B., Gadelha, B. F., e Oliveira, E. H. T. (2016) Formação de Grupos em Ambientes Virtuais de Aprendizagem: Uma Revisão Sistemática da Lietratura. In: XXVII Simpósio Brasileiro de Informática na Educação, 26, p. 310-319.
21. Renate Andersen, Anders I. Mørch & Kristina Torine Litherland (2022) Collaborative learning with block-based programming: investigating human-centered artificial intelligence in education, Behaviour & Information Technology, 41:9, 1830-1847.https://doi.org/10.1080/0144929X.2022.2083981
22. Revelo, O. S.; Ordóñez, C. A. C.; Duque, M. Á. R.; Pinto, I. I. B. S. Homogeneous Group Formation in Collaborative Learning Scenarios: An Approach Based on Personality Traits and Genetic Algorithms. IEEE Transactions on Learning Technologies. 2021, 4(14), 486-499. doi: 10.1109/TLT.2021.3105008.
23. Revelo-Sánchez, O., Collazos, C.A., Redondo, M.A. (2020). Collaborative Learning Group Formation Based on Personality Traits: An Empirical Study in Initial Programming Courses. In: Agredo-Delgado, V., Ruiz, P.H., Villalba-Condori, K.O. (eds) Human-Computer Interaction. HCI-COLLAB 2020. Communications in Computer and Information Science, vol 1334. Springer, Cham. https://doi.org/10.1007/978-3-030-66919-5_8
24. Revelo Sánchez, O.; Collazos, C.A.; Redondo, M.A. A Strategy Based on Genetic Algorithms for Forming Optimal Collaborative Learning Groups: An Empirical Study. Electronics. 2021, 10, 463. https://doi.org/10.3390/electronics10040463
25. Santos, S., Castro, J. e Castro, T. (2007) "Alteração do Modelo de Grupos do Moodle para apoiar a Colaboração", In: Anais do XVIII Simpósio Brasileiro de Informática na Educação (SBIE 2007), 18, p.123-134. Porto Alegre
26. Souza, K.H.F. e Melo, L.B. (2021). Uma revisão Sistemática do uso de gamificação no ensino de programação. In: XXXII Simpósio Brasileiro de Informática na Educação, 32, p. 440-450.
27. Travassos, G.; Dos Santos, P.; Neto, P.; et al. An Environment to Support Large Scale Experimentation in Software Engineering. In: Proceedings of 13th IEEE International Conference on the Engineering of Complex Computer Systems (ICECCS), p. 193-202, Belfast, United Kingdom, 2008.
28. Xu, W., Wu, Y. & Ouyang, F. Análise de aprendizagem multimodal de padrões colaborativos durante a programação em pares no ensino superior. Int J Educ Technol High Educ 20, 8 (2023). https://doi.org/10.1186/s41239-022-00377-z