Scientific Journal of Informatics
Vol. 13, No. 1, Feb 2026

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Optimizing Fair and Efficient Group Formation in Community Service Program Using Particle Swarm Optimization

Windarto[1]*
[1]Informatics, Faculty of Information Technology, Universitas Budi Luhur, Jakarta, Indonesia

Abstract.
Purpose: The rapid expansion and administrative complexity of community service programs (Kuliah Kerja Nyata/KKN) have made manual group formation increasingly inefficient, inconsistent, and prone to imbalance. This creates an urgent need for an automated, fair, and reliable optimization method capable of handling large-scale grouping constraints. This study aims to evaluate the performance of the Particle Swarm Optimization (PSO) algorithm in generating optimal KKN group formations, focusing on computational efficiency, convergence behavior, and solution quality.
Methods: PSO was implemented to form 27 KKN groups using 10 independent runs. Performance metrics included execution time, optimal iteration counts, initial fitness scores, and best final scores. Each run was analyzed to observe convergence patterns and stagnation behaviors.
Result: The results indicate that PSO is highly efficient, with very fast execution times and rapid convergence, often reaching optimal solutions in the first iteration. However, performance varied: some groups achieved low optimal scores (95–97), while many stagnated at extremely high scores (100000.0) with no improvement. This shows that PSO's effectiveness depends heavily on problem characteristics and initialization.
Novelty: This study identifies and explains stagnation patterns in PSO when applied to discrete, constraint-heavy academic group formation problems, an area rarely examined in prior research. The analysis provides insight into PSO's strengths and limitations and highlights the need for improved parameter tuning and initialization strategies. The findings serve as a foundation for developing more robust optimization approaches for fair and efficient KKN group formation.

Keywords: PSO, Optimization, Group formation, KKN, Algorithm performance
Received November 2025 / **Revised** November 2025 / **Accepted** December 2025

INTRODUCTION

The rapid advancement of information technology has increased the demand for intelligent systems capable of simplifying complex human tasks, including the formation of student groups for community service program (Kuliah Kerja Nyata or KKN). This need becomes increasingly relevant when considering that KKN group formation across Indonesian universities is governed by guidelines emphasizing multidisciplinary composition and gender balance as essential requirements. For example, the KKN guideline of Universitas Samudra mandates that each group must consist of students from at least three study programs and a minimum of two faculties, while also explicitly considering gender composition in the grouping process [1]. Similarly, the KKN–PPM guideline of Universitas Gadjah Mada (UGM) requires groups to be multidisciplinary, involving students from two to four different scientific clusters, and incorporates gender balance as a key criterion in the plotting system [2]. These standards demonstrate that KKN implementation nationally expects heterogeneous groups in both academic discipline and gender representation. However, at Universitas Budi Luhur, the current KKN grouping process is still performed manually using Microsoft Excel, making it time-consuming, prone to human error, and unable to systematically accommodate faculty diversity and gender balance. Given that optimal group formation is an NP-Complete problem, these challenges underscore the need for intelligent computational approaches capable of producing effective, fair, and equitable KKN group configurations.

Although manual grouping methods provide flexibility, they fail to ensure fairness and efficiency in large-scale implementations. Intelligent optimization algorithms have shown potential in automating similar complex tasks; however, their applications to KKN grouping remain limited. Existing approaches, such as the Genetic Algorithm (GA), have demonstrated efficiency in generating groupings but have not fully resolved issues of imbalance across gender and faculty representation. Thus, there is a need for alternative optimization methods capable of producing more balanced and adaptive group formations.

*Corresponding author.
Email address: windarto@budiluhur.ac.id (Windarto)
DOI: 10.15294/sji.v13i1.36007

While GA is widely utilized as a domain-independent approach for complex group formation—capable of handling numerous variables and producing optimal solutions—its conventional form has several limitations [3]. For example, traditional GA is less suitable for discrete and fixed problems such as course scheduling, even though enhanced GA variants have demonstrated superiority in discrete combinatorial optimization [4]. In addition, GA has been successfully applied to automate group formation for academic-advisor assignments, ensuring balanced distribution of demographic attributes including nationality, race, and gender across groups [5]. However, as grouping problems grow more complex, such as in NP-hard group-scheduling scenarios (e.g., in amusement parks), both GA and Particle Swarm Optimization (PSO) have been recognized as the most prominent evolutionary algorithms for such tasks [6]. In the context of highly complex, high-dimensional problems, standard PSO may struggle with issues such as sensitivity to initial parameters and susceptibility to local optima [7]. This has motivated the development of advanced PSO variants, including parallel PSO, which improves execution time in distributed environments [8], and multi-strategy group-based PSO (GPSOM), which significantly enhances optimization performance, stability, and efficiency for complex engineering applications [7]. These advancements further reinforce enhanced PSO as a strong methodological foundation for adaptive group formation.

Previous studies have explored the use of Genetic Algorithms (GA) to address group formation challenges. One study reported that GA produced efficient group distributions with an average of 3.74 faculties per group and relatively balanced gender composition, though inconsistencies remained in some cases [9]. Similarly, Rohmad and Akbar (2020) implemented GA for KKN student grouping at Mercu Buana University Yogyakarta, achieving solutions with low error rates and high fitness values but requiring a relatively long computation time—approximately 5 minutes and 26 seconds for 807 participants [10].

Recent comparative studies evaluating the performance of the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in complex optimization problems yield varied yet instructive results depending on the application context. In estimating the parameter values of the epidemiological Susceptible-Exposed-Infected-Recovered (SEIR) model, one study demonstrated that PSO achieved a minimum Root Mean Square Error (RMSE) of 3,576.71 with an average computation time of 19.98 seconds. This outcome was superior in terms of accuracy and overall efficiency compared to the GA, which yielded a minimum RMSE of 3,923.36 in 19.86 seconds, thus establishing PSO as the more effective method for parameter estimation in this domain [11]. Conversely, in the domain of optimal cost management for on-grid microgrid energy systems (with PV-battery integration), while both algorithms successfully reduced operational costs, PSO consistently demonstrated superiority in finding the minimum cost solution with faster convergence and more stable exploration of the solution space. The GA, in this context, tended to become trapped in *local optima* and required longer to reach the global optimum [12]. Other study, comparing PSO and GA for designing optimal dietary recommendations for pregnant women, indicates that PSO achieves higher fitness values and adapts more effectively to multivariable and non-linear problems, whereas GA tends to stagnate and requires more complex parameter tuning [13]. Overall, these studies reinforce the superiority of PSO in terms of efficiency, stability, and solution quality, making it a strong methodological foundation for its application in KKN group formation, which similarly involves multiple attributes and constraints.

PSO, inspired by the collective movement patterns of bird flocks and fish schools, is a population-based optimization algorithm in which candidate solutions are iteratively refined through personal experience (personal best) and social learning driven by the global best solution [14], [15]. In the PSO method, a set of simple agents—referred to as particles—is distributed across the problem's search space, where each particle evaluates the objective function based on its current position [8]. The movement of each particle is then determined by combining information from its own past experiences—specifically its current and personal best positions (*pbest*)—with information obtained from other particles in the swarm, such as the global best position (*gbest*), along with stochastic (random) influences [16]. After all particles update their positions, the algorithm proceeds to the next iteration. In modern PSO formulations, particles traverse the search space by evaluating the fitness function at their current positions and updating their velocities based on cognitive, social, and stochastic components [14]. The mechanism integrates personal historical experience (*pbest*), collective swarm experience (*gbest*), and controlled randomness, enabling the swarm to move collectively toward near-optimal regions [7]. Updated studies in swarm intelligence reaffirm that this behaviour closely resembles cooperative foraging patterns found in natural species such as bird flocks and fish schools [17].

PSO has been widely applied in production systems, scheduling, and resource management due to its advantages in speed, flexibility, and its ability to handle nonlinear and multi-objective optimization problems [18]. For instance, Febiani [19] successfully applied PSO to optimize school timetables, demonstrating its capability to manage multiple complex parameters simultaneously. Likewise, in SMPIT Mufidatul Ilmi, PSO generated feasible course schedules that satisfied all hard constraints, although several soft constraints remained unmet, indicating the need for improved adaptability [20]. PSO was also effectively implemented in scheduling Baca Tulis Al-Qur'an (BTQ) mentoring activities at Ma'had Al-Jami'ah UIN Alauddin Makassar, where it successfully balanced lecturer availability and student schedules [21]. More recent studies further reinforce PSO's robustness: Magdalena et al. [22] validated PSO's capability to minimize course schedule conflicts in higher education through stable convergence and efficient computational performance. At the international level, PSO continues to advance through specialized variants, as seen in Zhang et al. [23], who applied a PSO-based scheduling mechanism to optimize queue management in large-scale satellite communication networks, achieving significant reductions in delay and packet loss. Furthermore, Han et al. [24] proposed an adaptive Group Learning PSO (GLPSO) that improves search diversity and avoids premature convergence, demonstrating superior performance in complex optimization scenarios.

Although PSO has been extensively applied in various scheduling and optimization problems—such as school timetabling, academic course scheduling, mentoring activity coordination, and large-scale communication systems—existing studies primarily focus on time-based scheduling tasks rather than multi-criteria group formation. Prior research demonstrates that PSO is capable of efficiently resolving scheduling conflicts and producing constraint-compliant solutions, but none of these studies address the complexity of forming heterogeneous student groups that must balance gender composition, faculty diversity, academic disciplines, and group size simultaneously. Furthermore, previous attempts to form KKN groups have predominantly relied on Genetic Algorithms (GA), which, despite producing feasible solutions, still result in uneven gender distribution, suboptimal faculty balance, and longer computation times. No existing research has explored the application of PSO for KKN group formation, nor examined its potential to outperform GA in producing equitable, diverse, and computationally efficient group assignments. This gap highlights the need for a PSO-based approach specifically designed to optimize multi-dimensional attributes required in KKN grouping.

Building on these prior studies, this research aims to apply the Particle Swarm Optimization (PSO) algorithm for automatic KKN group formation at Universitas Budi Luhur. The proposed approach seeks to optimize multiple grouping parameters—gender balance, faculty diversity, group size, and academic discipline—in a systematic and automated manner. By doing so, PSO is expected to produce more equitable, heterogeneous, and representative group compositions, while addressing the limitations of previous methods based on genetic algorithms.

METHODS

This study develops a group formation system for the Community Service Program (Kuliah Kerja Nyata/KKN) at Universitas Budi Luhur using the Particle Swarm Optimization (PSO) algorithm. The conceptual framework was designed based on a systems approach that connects the challenges of manual group formation, the principles of population-based optimization, and relevant supporting factors such as fairness and diversity. The main parameters considered in group formation include faculty diversity, gender balance, and group size, which are aligned with university regulations.

As illustrated in Figure 1, the conceptual framework shows the relationship between these parameters and the optimization objectives in achieving equitable and heterogeneous KKN groups.

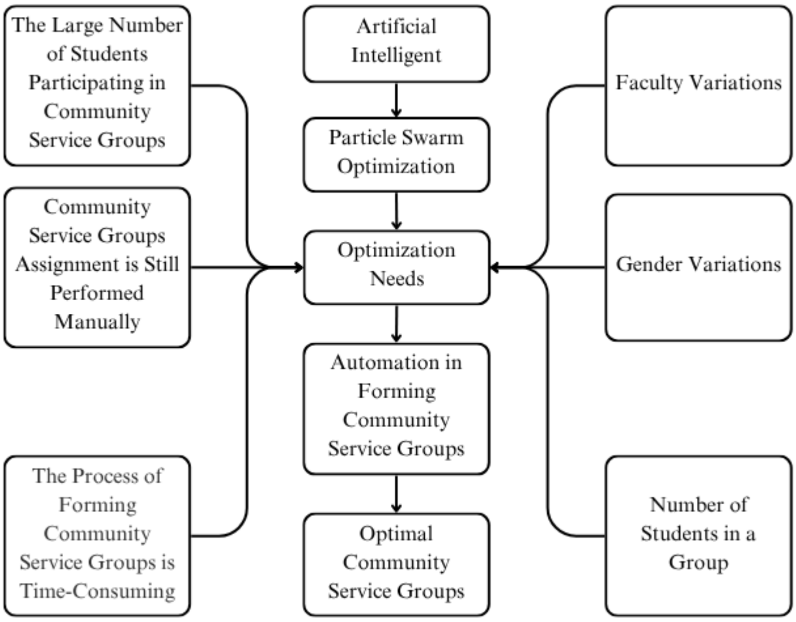

Figure 1. Research conceptual framework

To develop the system, this study employed the prototyping development method, which enables iterative interaction between system developers and stakeholders. A prototype is an early version of software developed to explore concepts, test design options, and better understand problems and solutions. It enables rapid, iterative development, allowing stakeholders to experiment early while controlling costs. Prototypes play a key role in requirements engineering—helping elicit and validate requirements—and in system design, where they assist in exploring solutions and developing user interfaces. By interacting with prototypes, users can identify strengths, weaknesses, and missing features, leading to refined requirements and improved system specifications based on real feedback and observations

To develop the system, this study employed the prototyping development method, which enables iterative interaction between system developers and stakeholders. A prototype is an early version of software created to explore concepts, evaluate alternative design options, and better understand the problem space as well as potential solution requirements. Prior literature emphasizes that prototyping supports rapid and iterative refinement, allowing stakeholders to experiment with system behaviour early in the development cycle while maintaining cost efficiency. This approach is strongly aligned with modern requirements engineering practices, where prototypes serve as an effective medium to elicit and validate user needs, uncover hidden requirements, and reduce ambiguity in system specifications, it showed that prototyping enables users to identify missing features, strengths, and weaknesses during early requirement exploration [25]. Furthermore, recent studies emphasize that prototyping is essential for reducing requirement-related errors and minimizing rework, as it supports continuous feedback and iterative improvement during system development [26]. Complementing this, established software engineering literature also notes that prototyping strengthens requirement validation and enhances system design by allowing early user interaction and clarifying unclear specifications [27]. In the context of system design, prototypes also help developers explore solution alternatives and design user interfaces more effectively, ensuring that the final system specification reflects real user expectations derived from direct interaction and evaluation.

This approach consists of several key stages: identifying user requirements, developing an initial prototype, evaluating it with stakeholders, revising based on feedback, and implementing the final version. This iterative process ensures that the developed system aligns closely with user needs and functional requirements. The PSO-based prototype was tested using actual student data to assess its ability to produce group formations that are faster, balanced, and compliant with the specified criteria. The stages of this development process are illustrated in Figure 2, which presents the prototyping methodology used in this study [28].

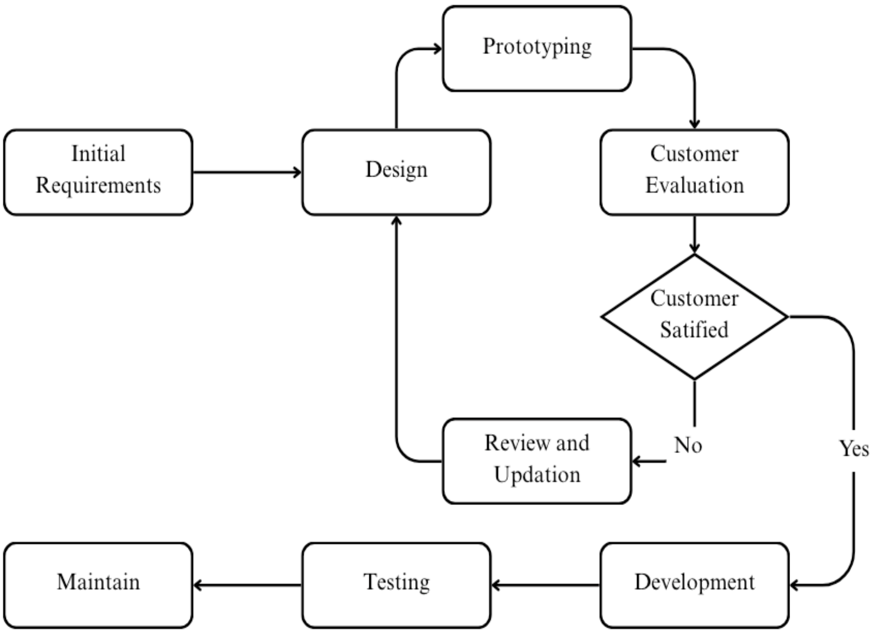

Figure 2. Prototyping development method

Data were collected through literature review, direct observation, and interviews with KKN administrators to determine user requirements and system expectations. The research was conducted at the Directorate of Research and Community Service, Universitas Budi Luhur. The detailed research stages are depicted in Figure 3, which outlines the workflow from problem identification to final system testing.

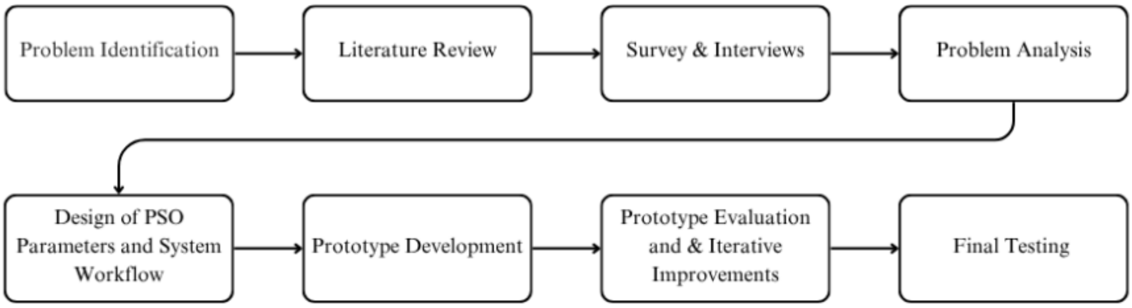

Figure 3. Research workflow

The dataset used in this study consists of 275 students enrolled in the Community Service Program (KKN) during the odd semester of the 2024/2025 academic year at Universitas Budi Luhur. These students come from five different faculties, each displaying varied gender compositions. As shown in Table 1, FISSIG contributed the highest number of participants (109 students), followed by FEB (85 students). Meanwhile, the Faculty of Engineering had the smallest representation with only two male students. Overall, the distribution reflects a diverse and heterogeneous student population, which is essential for analysing fairness and balance during the group formation process.

Table 1. Distribution of KKN Students by Faculty and Gender

Faculty	Male	Female	Total
Faculty of Economics and Business (FEB)	23	62	85
Faculty of Communication and Creative Design (FKDK)	22	12	34
Faculty of Social Sciences and Global Studies (FISSIG)	40	69	109
Faculty of Engineering (FT)	2	0	2
Faculty of Information Technology (FTI)	35	10	45
Total	**122**	**153**	**275**

The research procedure consisted of nine stages:
1. Identification of problems in the manual grouping process
2. Literature review on KKN management and PSO applications
3. Surveys and interviews to gather user requirements
4. Problem analysis and determination of success indicators
5. Design of PSO parameters and system workflow
6. Prototype development
7. Prototype evaluation and iterative improvements
8. Final testing and comparison with the manual method
9. Preparation of the final research report

This process was carried out over six months, with a primary focus on validating the performance of PSO in optimizing the KKN group formation process. To obtain the required data, several data-collection methods were employed. A literature study was conducted to gather theoretical and empirical evidence relevant to the research problem. Literature review methods published in recent years emphasise their role in establishing conceptual clarity, synthesizing existing findings, and providing methodological robustness [29]. In addition, a Rapid Literature Review (RLR) was conducted as a complementary approach to the traditional Systematic Literature Review (SLR), enabling faster analysis of recent research data [30]. This process included gathering references and scientific papers related to KKN information systems and academic studies on group formation, which were then critically reviewed and analyzed. Second, direct observation of the existing KKN information system was performed to understand the current manual group-formation workflow, an approach aligned with recent multilevel qualitative research emphasizing the importance of contextual system analysis in organizational environments [31]. Third, in-depth interviews were conducted with KKN administrators to capture procedural insights, operational challenges, and expectations for transitioning from manual to automated grouping. This approach is consistent with contemporary qualitative studies in higher education, which use semi-structured interviews to explore institutional practices and stakeholder needs [32].

The optimization technique applied in this study—Particle Swarm Optimization (PSO)—is a population-based algorithm inspired by the coordinated movement of social organisms. Contemporary PSO research highlights that each particle represents a potential solution whose trajectory is determined by iterative updates of its position and velocity through *pbest–gbest* interactions, enabling efficient exploration of complex search spaces [33].

In general, PSO operates through the following steps [14]:
1. Initializing the particle population.
2. Evaluating the fitness of each particle.
3. Updating *pbest* and *gbest* values.
4. Computing new velocities and positions.
5. Iterating until the stopping criteria are met.

The algorithm's performance depends on several parameters, including swarm size, inertia weight (ω), cognitive (c_1) and social (c_2) acceleration coefficients, as well as random factors (r_1, r_2) that introduce stochasticity. These components allow PSO to balance global exploration and local exploitation, enabling it to efficiently identify optimal solutions in complex, multidimensional problems such as KKN group formation.

In initializing the PSO algorithm, the initial velocity and position of each particle are assigned randomly. The subsequent development process consists of several main stages as described in [34]:
(1) Assume that the swarm size (number of particles) is represented by *N*. The initial velocity and position of each particle in the *N*-dimensional search space are generated randomly.
(2) Compute the initial velocity for all particles. Each particle moves toward an optimal point with a certain velocity. At the beginning, all velocities are assumed to be zero, and the iteration counter is set to $i = 1$.
(3) Evaluate the fitness value of each particle using the predefined objective function. If the current fitness value of a particle is better than its previous best position (*pbest*), then *pbest* is updated to the current position.

(4) Compare each particle's fitness value with the global best value (*gbest*). If a particle achieves a better fitness value than the current *gbest*, then *gbest* is updated accordingly.
(5) PSO updates particle velocity using inertia weight, cognitive learning, and social learning components, as expressed in (1) [14], [15], while particle positions are updated according to (2) [24], [23]. These equations determine how each particle adjusts its movement based on its previous velocity, the distance from its *pbest*, and the attraction toward *gbest*.

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (pbest_i - x_i^t) + c_2 r_2 (gbest_i - x_i^t) \tag{1}$$

$$x_i^{k+1} = x_i^t + v_i^{t+1} \tag{2}$$

The four studies indicate that solution updates in PSO are determined by the following components:
inertia weight (ω) : Preserves the particle's momentum
cognitive component (c_1) : Reflects the particle's individual learning (*pbest*) in the range [0, 1]
social component (c_2) : Represents collective learning (*gbest*) in the range [0, 1]
random factors (r_1, r_2) : Uniformly distributed random numbers in the range [0, 1]
velocity(v_i^t) : Velocity of particle i at iteration t
position (x_i^t) : Position of particle i at iteration t

(6) After updating the velocity and position, the algorithm evaluates whether the current solution has converged. Convergence is indicated when all particles move toward a similar region or value in the search space. If convergence has not yet been achieved, the algorithm returns to Step 2 by increasing the iteration counter to $i = i + 1$, followed by recalculating the updated values of pbest and gbest. This iterative process continues until all particles converge toward a common solution. Typically, convergence is determined using a stopping criterion, such as when the difference between the current solution and the previous solution becomes sufficiently small.
(7) There are two important considerations in determining the stopping condition [35]:
    (a) The stopping condition must prevent the PSO from premature convergence, in which the swarm collapses toward a suboptimal region too early, resulting in a non-optimal solution.
    (b) The stopping condition should avoid excessive oversampling; if it requires continuous or unnecessary computation, it may increase the complexity and processing time of the search.

    Common stopping criteria that can be applied in Particle Swarm Optimization include [35]:
    (a) Terminating the process when the maximum allowed number of iterations has been reached.
    (b) Terminating the process when an acceptable solution has been found.
    (c) Terminating the process when no significant improvement is observed after a certain number of iterations.

RESULT AND DISCUSSION

The study involved 275 students from five faculties at Universitas Budi Luhur—the Faculty of Economics and Business (FEB), Faculty of Communication and Design (FKDK), Faculty of Social Sciences and Global Studies (FISSIG), Faculty of Engineering (FT), and Faculty of Information Technology (FTI)—with a nearly balanced gender distribution. The group formation process was guided by three primary criteria: group size (7–10 members), faculty diversity (3–5 faculties per group), and gender balance.

Using the Particle Swarm Optimization (PSO) algorithm, groups were formed efficiently and equitably. Most groups consisted of ten members, representing multiple faculties and maintaining balanced gender compositions. The detailed performance results, as summarized in Table 2, demonstrate PSO's high efficiency in computation and convergence speed, achieving optimal group formations with minimal processing time. However, the table also highlights several score anomalies in certain groups, indicating the need for further parameter tuning and data normalization to enhance accuracy and stability.

Table 2. PSO Performance Metrics

Aspect	PSO Results	Notes
Computation Time	Very fast (average < 0.5 seconds per group)	Significantly faster than the genetic algorithm; suitable for large-scale datasets
Optimal Iterations	Majority: 1 iteration	Indicates high efficiency in reaching optimal solutions
Score Stability	Consistent, minor variations (95–97 range)	Score improvement minimal, can be enhanced with further tuning

Aspect	PSO Results	Notes
Convergence Speed	High and stable	Validation required for groups with fitness scores of 100,000.0
Score Anomalies	Groups 16–27 produced extreme scores (100,000.0)	Requires normalization or parameter adjustment
Group Composition Fairness	275 students distributed into 28 groups; each group consisted of 7–10 members	All groups met diversity and size criteria
Faculty Representation	Each group includes 2–5 faculties (FEB, FISSIG, FKDK, FT, FTI)	Ensures interdisciplinary and equitable distribution
Gender Balance	Average ratio per group: 45% male – 55% female	Gender composition balanced across all groups; no single-gender domination observed
Overall System Reliability	High, with minor data-related anomalies	Performs effectively on validated datasets; robust for automated group formation

3.1. Summary of Findings

Following the performance metrics presented in Table 2, this section elaborates on the fairness and efficiency aspects achieved through the application of the Particle Swarm Optimization (PSO) algorithm in forming Community Service (KKN) student groups.

The fairness aspect was demonstrated through balanced group compositions across gender and faculty representation. As shown in Table 3, PSO successfully distributed 275 students—122 males and 153 females—from five faculties (FEB, FISSIG, FKDK, FT, and FTI) into 28 groups, most containing 10 members. Each group represented between 2 and 5 faculties, indicating a high degree of heterogeneity. The gender ratio was maintained proportionally, preventing bias or concentration of participants from a single faculty, as illustrated in Figure 4. This reflects PSO's capability to generate fair and well-distributed group formations.

Table 3. Gender and Faculty Composition Summary

Parameter	Value / Observation
Total Students	275 (122 male, 153 female)
Total Groups	28 (mostly 10 members, one with 5)
Faculties Represented	5 (FEB, FISSIG, FKDK, FT, FTI)
Faculty Representation per Group	2–5 faculties
Gender Distribution	Balanced across groups
Notable Observation	FT had only 2 male members, no female — still distributed to maintain diversity

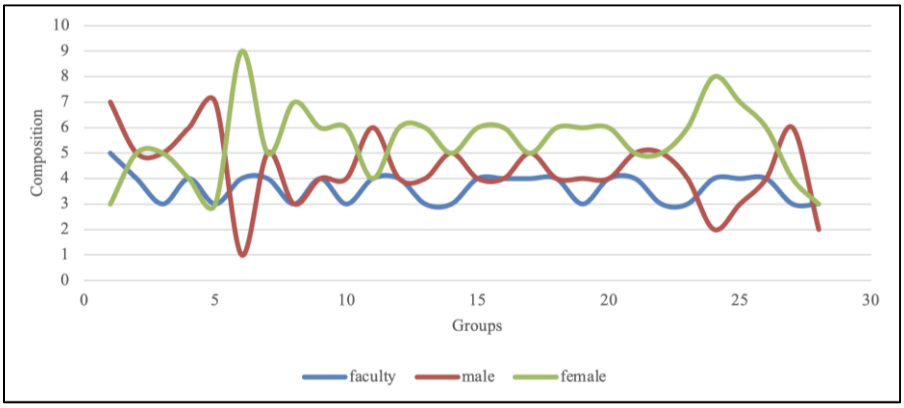

Figure 4. Gender and Faculty Composition

In terms of efficiency, PSO demonstrated very fast computation times and rapid convergence during the optimization process. As summarized in Table 4, the average computation time per group ranged between 0.17 and 0.26 seconds across ten trials, indicating sub-second performance. A few groups, specifically in the first trial of the ten test runs (e.g., Groups 11–13) required slightly longer computation times (>0.7 seconds), likely due to initial particle variation during random initialization. Overall, the algorithm maintained computational stability and efficiency suitable for real-time applications, as illustrated in Figure

5.

Table 4. PSO Computation Time Summary

Metric	Average Value	Observation
Average Computation Time	0.17–0.26 seconds	Indicates high computational efficiency
Outlier Groups	11–13	Initial run >0.7 seconds due to initialization variance
Stability	High	Consistent across 10 trials
Suitability	Excellent	Ideal for large-scale or real time applications

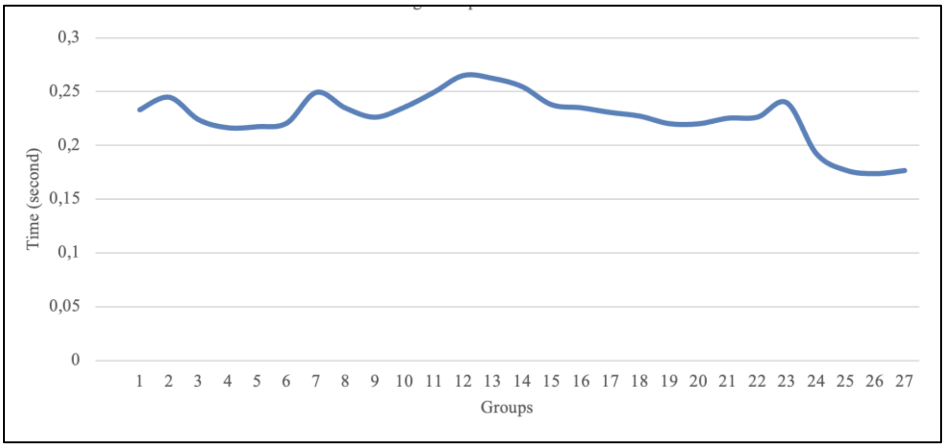

Figure 5. Average Computation Time

Furthermore, the algorithm demonstrated rapid convergence performance, as presented in Table 5. Out of 50 possible iterations, most groups (particularly Groups 12–26) reached optimal solutions within a single iteration. A few early groups required slightly more iterations (2–3 on average), and only one group (Group 27) showed an anomaly requiring up to 19 iterations in one test. The overall average iteration count across all groups ranged between 1 and 2, indicating exceptional stability of the optimization process, as illustrated in Figure 6.

Table 5. Optimal Iteration Summary

Metric	Result	Observation
Total Iterations Allowed	50	—
Average Iterations to Converge	1–2	Extremely fast convergence
Groups Achieving Optimal in 1 Iteration	12–26	Majority of all groups
Outlier Group	Group 27 (up to 19 iterations)	Due to data variation
Overall Stability	Very High	Consistent convergence behaviour

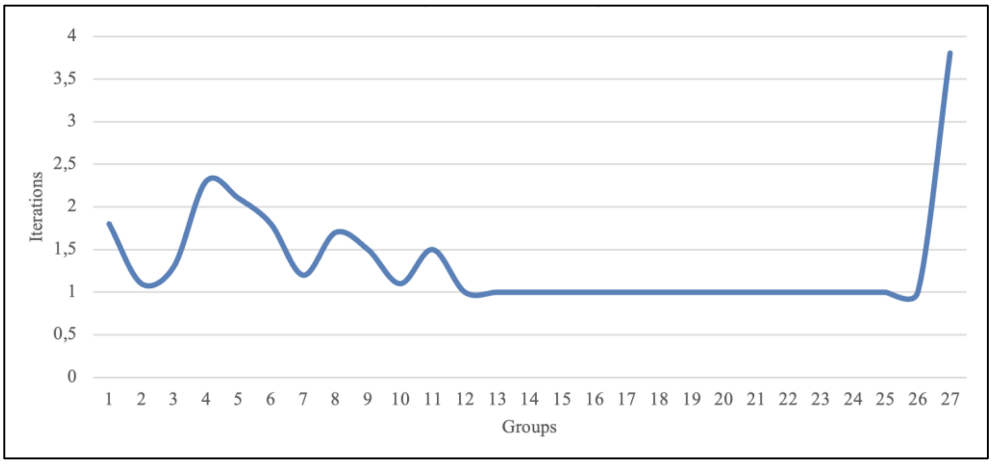

Figure 6. Average Iteration to Converge

In summary, these findings demonstrate that PSO not only achieves computational efficiency through rapid processing and convergence but also ensures fairness by maintaining balanced and diverse group compositions. The algorithm's ability to produce equitable and efficient outcomes reinforces its potential as a practical tool for automated student group formation in community service programs.

Regarding fitness score behaviour, two distinct performance categories were observed. Category A (Groups 1–15) exhibited realistic and consistent fitness improvements, with final scores between 95–97, indicating successful optimization and balanced group formation. In contrast, Category B (Groups 16–27) produced abnormally high fitness scores (~100,000), suggesting data anomalies or parameter misconfiguration. Notably, Group 27 displayed extreme outlier behaviour, with fitness values reaching up to 1,000,000 and irregular convergence patterns, likely caused by errors in input or configuration settings. These anomalies highlight the importance of robust data validation and careful parameter control to ensure consistent and reliable outcomes.

Overall, the PSO algorithm performed effectively on well-structured datasets (Category A), demonstrating fast, stable, and reliable optimization results. However, its performance declined when faced with data irregularities (Category B), emphasizing the need for proper input validation and parameter tuning. To enhance the system's reliability, several improvements are recommended:
(1) implementing strict data validation to eliminate invalid inputs;
(2) fine-tuning particle size, inertia weight, and learning coefficients for smoother convergence;
(3) refining the fitness function to consider academic discipline and student preference; and
(4) performing thorough debugging to ensure logical consistency and reproducibility across diverse datasets.

In conclusion, PSO proves to be a viable and efficient approach for automating Community Service (KKN) group formation, particularly when operating on validated datasets. Despite minor anomalies, its capability to maintain fairness, stability, and scalability demonstrates strong potential for wider implementation in educational institutions.

3.2. Discussion
This study significantly establishes the superiority of the Particle Swarm Optimization (PSO) algorithm over the Genetic Algorithm (GA) in solving the *NP-Complete* problem of KKN group formation, particularly concerning computational efficiency and solution quality (fairness). Regarding efficiency, PSO achieved exceptionally fast processing times, averaging 0.17–0.26 seconds per group, with rapid convergence typically occurring within 1–2 iterations. This performance markedly surpasses the computational time of the GA, which required approximately 5 minutes and 26 seconds in a previous study [10] for a similar context, thereby demonstrating PSO's superior scalability and real-time potential. In terms of solution quality, PSO successfully generated consistently balanced KKN groups, maintaining an average composition of 45% male to 55% female and representing 2–5 faculties per group. This addresses the gender inconsistency issues frequently observed in prior GA implementations [9]. Although some score anomalies were detected in certain groups, likely due to parameter sensitivity—a known challenge in standard PSO [7]—the algorithm's stable performance on validated datasets confirms that PSO provides a more robust, efficient, and effective methodological foundation for the automated creation of fair, multi-criteria groups compared to conventional GA approaches.

CONCLUSION
The results of this study confirm that the Particle Swarm Optimization (PSO) algorithm offers an effective, fair, and computationally efficient solution for automating Community Service (KKN) group formation at Universitas Budi Luhur. PSO consistently produced balanced groups across gender and faculty representation while maintaining rapid execution times—averaging below 0.5 seconds per group—and achieving convergence in as little as one iteration for most cases. Although several groups exhibited anomalous fitness values due to data inconsistencies or suboptimal parameter settings, these findings highlight the importance of robust preprocessing and careful parameter calibration to ensure stable outcomes. Overall, this research demonstrates that PSO not only enhances the speed and accuracy of group allocation but also supports institutional goals of fairness, diversity, and scalability. Its successful implementation indicates substantial practical impact, offering higher education institutions a reliable and

efficient mechanism to manage large-scale, equity-focused group formation processes for community service programs. This study concludes that the Particle Swarm Optimization (PSO) algorithm is effective and efficient for automating KKN group formation. PSO succeeded in generating fair group compositions—with balanced gender proportions and diverse faculty representation—while maintaining very fast computation times and rapid convergence, often within a single iteration. The research contributes empirical evidence that PSO can support large-scale and equity-oriented group allocation in higher education contexts. However, several anomalies in fitness values indicate that the system remains sensitive to data irregularities and parameter settings. These findings highlight the need for improved data validation, refined fitness formulations, and more adaptive parameter tuning. Future work should explore alternative initialization strategies, hybrid metaheuristics, and expanded constraints such as student preferences to enhance solution stability and practical applicability.

REFERENCES

[1] *Buku Panduan Kuliah Kerja Nyata Periode II Tahun 2024 Universitas Samudra*. Langsa: Lembaga Penelitian, Pengabdian Kepada Masyarakat dan Penjaminan Mutu Universitas Samudra, 2024.

[2] *Hibah Pengabdian Masyarakat, dan KKN-PPM UGM Tahun 2023*. Yogyakarta: Direktorat Pengabdian Kepada Masyarakat UGM, 2023.

[3] A. Krouska, C. Troussas, and C. Sgouropoulou, "A novel group recommender system for domain-independent decision support customizing a grouping genetic algorithm," *User Model. User-adapt. Interact.*, vol. 33, no. 5, pp. 1113–1140, 2023, doi: 10.1007/s11257-023-09360-3.

[4] R. Chen, L. Su, and J. Guan, "AI-Powered Genetic Algorithm for Optimal Scheduling in Medical Research and Educational Management," *Comput. Des. Appl.*, vol. 21, no. S24, pp. 17–34, 2024.

[5] T. X. Ying, A. M. Kassim, and N. A. Abdullah, "A Genetic Algorithm-based Group Formation to Assign Student with Academic Advisor : A Study on User Acceptance Using UTAUT," *JOIV*, vol. 7, no. 3, pp. 967–976, 2023.

[6] Y.-C. Hsieh and P.-S. You, "A comparison of three evolutionary algorithms for group scheduling in theme parks with multitype facilities," *Sci. Prog.*, vol. 107, no. 4, pp. 1–24, 2024, doi: 10.1177/00368504241278424.

[7] J. Yan, G. Hu, H. Jia, A. G. Hussien, and L. Abualigah, "GPSOM : group - based particle swarm optimization with multiple strategies for engineering applications," *J. Big Data*, vol. 12, no. 114, 2025, doi: 10.1186/s40537-025-01140-7.

[8] M. I. Capel, A. S. Hidalgo, and J. A. H. Terriza, "Parallel PSO for Efficient Neural Network Training Using GPGPU and Apache Spark in Edge Computing Sets," *Algorithms*, vol. 17, no. 378, pp. 1–26, 2024.

[9] Windarto, "Pembuatan Kelompok Kuliah Kerja Nyata di Universitas Budi Luhur Dengan Pendekatan Algoritma Genetika," Jakarta, 2025.

[10] A. N. Rohmad and M. Akbar, "Penerapan Algoritma Genetika Dalam Pengelompokan Mahasiswa KKN (Studi Kasus: KKN Angkatan XLII Universitas Mercu Buana Yogyakarta)," *JIKO (Jurnal Inform. dan Komputer)*, vol. 8, no. 1, p. 50, 2024, doi: 10.26798/jiko.v8i1.1073.

[11] A. Sa'adah, A. Sasmito, and A. A. Pasaribu, "Comparison of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) for Estimating the Susceptible-Exposed- Infected-Recovered (SEIR) Model Parameter Values," *J. Inf. Syst. Eng. Bus. Intell.*, vol. 10, no. 2, pp. 290–301, 2024.

[12] M. EL-Qasery, A. Abbou, M. Laamim, L. Id-Khajine, and A. Rochd, "Comparative analysis of GA and PSO algorithms for optimal cost management in on-grid microgrid energy systems with PV-battery integration," *Glob. Energy Interconnect.*, vol. 8, no. 4, pp. 572–580, 2025, doi: 10.1016/j.gloei.2025.05.003.

[13] D. Kurnianingtyas, N. Daud, I. Indriati, and L. Muflikhah, "Comparison Genetics Algorithm and Particle Swarm Optimization in Dietary Recommendations for Maternal Nutritional Fulfillment," *J. Sains dan Teknol. Ind.*, vol. 21, no. 2, p. 216, 2024, doi: 10.24014/sitekin.v21i2.28937.

[14] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications : A Systematic Review," *Arch. Comput. Methods Eng.*, vol. 29, no. 5, pp. 2531–2561, 2022, doi: 10.1007/s11831-021-09694-4.

[15] M. Ghazali, M. Hamami, and Z. H. Ismail, "A Systematic Review on Particle Swarm Optimization Towards Target Search in The Swarm Robotics Domain," *Arch. Comput. Methods Eng.*, no. 0123456789, 2022, doi: 10.1007/s11831-022-09819-3.

[16] H. Xu, Q. Deng, Z. Zhang, and S. Lin, "A hybrid differential evolution particle swarm optimization algorithm based on dynamic strategies," *Sci. Rep.*, vol. 15, no. 4518, pp. 1–49, 2025.

[17] D.-D. Ramírez-Ochoa, L. A. Pérez-Domínguez, E.-A. Martínez-Gómez, and D. Luviano-Cruz,

"PSO, a Swarm Intelligence-Based Evolutionary Algorithm as a Decision-Making Strategy: A Review," *Symmetry (Basel).*, vol. 14, no. 455, pp. 1–23, 2022.
[18] M. H. P. Swari, C. A. Putra, and I. P. S. Handika, "Analisis Perbandingan Algoritma Genetika dan Modified Improved Particle Swarm Optimization dalam Penjadwalan Mata Kuliah," *J. Nas. Pendidik. Tek. Inform.*, vol. 11, no. 2, pp. 92–101, 2022, doi: 10.23887/janapati.v11i2.49061.
[19] Alfina Febiani, Agung Mulyo Widodo, Nizirwan Anwar, Binastya Anggara Sekti, and Alivia Yulfitri, "Implementasi Algoritma 'Particle Swarm Optimization' (PSO) Penjadwalan Belajar Mengajar," *IKRA-ITH Inform. J. Komput. dan Inform.*, vol. 8, no. 1, pp. 152–161, 2024, doi: 10.37817/ikraith-informatika.v8i1.3210.
[20] M. Muhardeny, M. H. Irfani, and J. Alie, "Penjadwalan Mata Pelajaran Menggunakan Algoritma Particle Swarm Optimization (PSO) Pada SMPIT Mufidatul Ilmi," *J. Softw. Eng. Comput. Intell.*, vol. 1, no. 1, pp. 51–63, 2023, doi: 10.36982/jseci.v1i1.3047.
[21] Serliana, Rahman, Hastuti, F. Yusuf, and A. M. Abidin, "Penerapan Algoritma Particle Swarm Optimization (PSO) pada Sistem Penjadwalan Kegiatan Mentoring Baca Tulis Al-Qur'an," *Switch J. Sains dan Teknol. Inf.*, vol. 3, no. 1, pp. 206–222, 2025.
[22] M. Simanjuntak, M. P. U. Sitompul, J. N. Sitompul, K. Lanang, and R. Faizah, "Optimasi Penjadwalan Mata Kuliah Dengan Menggunakan PSO," *J. Mahajana Inf.*, vol. 10, no. 1, pp. 9–18, 2025.
[23] Z. Zhang *et al.*, "A Particle Swarm Optimization-Based Queue Scheduling and Optimization Mechanism for Large-Scale Low-Earth-Orbit Satellite Communication Networks," *Sensors*, vol. 25, no. 4, pp. 1–16, 2025, doi: 10.3390/s25041069.
[24] J. Han, Y. Chen, and X. Huang, "An Advanced Adaptive Group Learning Particle Swarm Optimization Algorithm," *Symmetry (Basel).*, vol. 17, no. 5, pp. 1–27, 2025, doi: 10.3390/sym17050667.
[25] B. Kang, N. Crilly, W. Ning, and P. O. Kristensson, "Prototyping to elicit user requirements for product development: Using head-mounted augmented reality when designing interactive devices," *Des. Stud.*, vol. 84, p. 101147, 2023, doi: 10.1016/j.destud.2022.101147.
[26] A. Alzayed, "Evaluating the Role of Requirements Engineering Practices in the Sustainability of Electronic Government Solutions," *Sustainability*, vol. 16, p. 433, 2024.
[27] I. Sommerville, *Software Engineering*, 10th ed. London: Pearson Education, 2016. [Online]. Available: https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software Engineering - Ian Sommerville.pdf
[28] Iqbal, "Tahapan Metode Prototype, Pengertian Metode Prototype dan Contohnya," nextgen.
[29] U. E. Chigbu, S. O. Atiku, and C. C. Du Plessis, "The Science of Literature Reviews : Searching, Identifying , Selecting , and Synthesising," *Publications*, vol. 11, no. 2, pp. 1–16, 2023.
[30] B. Smela *et al.*, "Rapid literature review: definition and methodology," *J. Mark. Access Heal. Policy*, vol. 11, no. 1, 2023, doi: 10.1080/20016689.2023.2241234.
[31] T. Köhler, "Multilevel qualitative research : Insights from practice," *Eur. Manag. J.*, vol. 42, no. March, pp. 503–514, 2024, doi: 10.1016/j.emj.2024.03.011.
[32] R. Kleimola, L. Hirsto, and H. Ruokamo, "Promoting higher education students ' self-regulated learning through learning analytics : A qualitative study," *Educ. Inf. Technol.*, vol. 30, pp. 4959–4986, 2025.
[33] F. Zhang and Z. Chen, "A Novel Reinforcement Learning-Based Particle Swarm Optimization Algorithm for Better Symmetry between Convergence Speed and Diversity," *Symmetry (Basel).*, vol. 16, no. 1290, pp. 1–27, 2024.
[34] R. M. Chen and H. F. Shih, "Solving University Course Timetabling Problems Using Constriction Particle Swarm Optimization with Local Search," *Algorithms*, vol. 6, no. 2, pp. 227–244, 2013, doi: 10.3390/a6020227.
[35] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2006.