A Proposal for Applying Bio-inspired Optimization Algorithms: A Comparative Study of Genetic Algorithms and Particle Swarm Optimization for Sensor Selection

Daniel do Carmo Silva Ribeiro[1]*, Henrique Nunes Teixeira[1], Tiago Menezes de Oliveira[1], Fillipe Valente Braz[2], Tiago Silva e Silva[2], Samara Maria Assunção de Souza[2]

[1]Federal University of Bahia, DEEC, Salvador, Bahia; [2]Federal University of Recôncavo da Bahia, CETENS,Feira de Santana, Bahia, Brazil

This paper presents an application proposal and a comparative study between two bio-inspired optimization algorithms: Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). The main objective is to demonstrate the effectiveness of these algorithms in selecting a subset of sensors, aiming to minimize the variance of the collected data. Through the analysis of results from two distinct datasets, this work explores the convergence characteristics, final population distribution, and the profile of the sensors selected by each algorithm. The results indicate that while both algorithms are capable of finding satisfactory solutions, GA tends to achieve better optimization values (lower standard deviation), whereas PSO demonstrates faster convergence. This study contributes to the understanding of the capabilities and limitations of each approach in problems of feature selection and sensing systems optimization.

Keywords: Genetic Algorithms. Particle Swarm Optimization. Sensor Selection. Bio-inspired Optimization. Variance Analysis.

Abbreviations: GA, Genetic Algorithms. PSO, Particle Swarm Optimization.

Optimization is a fundamental field in various areas of science and engineering, seeking to find the best solutions for complex problems, often with a vast search space. In scenarios where traditional analytical methods prove to be unfeasible or inefficient, metaheuristics, inspired by natural phenomena, emerge as powerful alternatives. Among these, bio-inspired algorithms, which mimic biological and behavioral processes, have gained prominence due to their ability to handle non-linear, multimodal, and high-dimensional problems [1].

Among the most established metaheuristics for solving complex problems, Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) stand out. GA emulates the evolutionary process and principles of genetics, exploring the search space through operators such as selection, crossover, and mutation to converge towards the fittest solutions [1]. In turn, PSO models the collective intelligence of flocks or swarms, where particles adjust their trajectories based on individual experience and the global knowledge of the swarm [1,2].

The robustness of these approaches makes them ideal for solving combinatorial optimization challenges, such as sensor selection. This problem consists of identifying an optimal subset from a larger set of sensors, aiming not only for system efficiency—through the reduction of costs, complexity, and energy consumption—but also for improving the quality and reliability of the acquired data [1,2]. Minimizing data variance, for example, is a frequently adopted optimality criterion, as a reduced standard deviation suggests greater consistency and precision in measurements [3,4].

In this context, the present article details an application and a comparative analysis of GA and PSO in the sensor selection problem for variance minimization. The study evaluates the advantages and disadvantages of each metaheuristic in terms of convergence speed, final solution quality, and the characteristics of the selected sensor subset.

The paper was organized in topics to explores the theoretical foundations of Genetic Algorithms and Particle Swarm Optimization, respectively; describes the research methodology and the datasets

Received on 23 January 2026; revised 30 March 2026.
Address for correspondence: Daniel do Carmo Silva Ribeiro Rua Sucupió, Residencial Betaville. Camaçari – BA, Brazil Bahia, Brazil. Zipcode: 42825-016. E-mail: pedro.souza@aln.senaicimatec.edu.br.

J Bioeng. Tech. Health 2026;9(5):392-402

used; presents and discusses the experimental results; and consolidates the conclusions, highlighting the main contributions of the study and pointing out directions for future research.

Original Contribution and Novelty

This work fills a gap by offering a comparative analysis focused on variance minimization for sensor selection, a critical aspect for data reliability not extensively covered in prior studies [1,3].

A key novelty is the detailed justification for algorithm parameters, enhancing reproducibility. Furthermore, a dedicated discussion on limitations and practical implications provides a holistic view often over-looked. The qualitative analysis of selected sensor profiles reveals distinct search strategies, offering unique insights for designing sensor systems.

Genetic Algorithms (GA)

Principles

Genetic Algorithms (GAs) are a class of optimization algorithms inspired by the principles of natural selection and biological genetics, as proposed by Charles Darwin. They operate on a population of candidate solutions (individuals), which evolve over generations through genetic operators such as selection, crossover, and mutation [1].

- **Initial Population:** The process begins with the creation of an initial population of individuals, usually generated randomly. Each individual represents a potential solution to the problem at hand [1].
- **Fitness Function:** A fitness function is used to evaluate the quality of each individual in the population. The higher the fitness, the better the solution [1].
- **Selection:** Individuals with higher fitness have a greater probability of being selected for reproduction, passing their characteristics on to the next generation. Common methods include roulette wheel selection and tournament selection, among others [1].
- **Crossover:** Two individuals (parents) are combined to generate new individuals (offspring) by exchanging genetic material. This allows for the recombination of features from good solutions [1].
- **Mutation:** Small, random changes are introduced into the genes of individuals. Mutation helps to maintain genetic diversity within the population and prevents the algorithm from getting stuck in local optima [1].
- **Replacement:** The new generation of individuals replaces the old population, and the process repeats for a predefined number of generations or until a stopping condition is met. Elitism is a strategy that ensures the best individual from the current generation survives to the next [1].

Applications

Genetic Algorithms are widely applied in a variety of fields due to their ability to explore complex search spaces and find optimal or near-optimal solutions for optimization, search, and learning problems [1]. Some notable applications include:

- **Optimization:** Solving combinatorial optimization problems (such as the Traveling Salesperson Problem), function optimization, engineering design, network and system optimization.
- **Machine Learning:** Feature selection, model parameter optimization (e.g., neural networks), rule learning.
- **Engineering:** Circuit design, industrial process optimization, route planning, resource allocation, and global localization of robots [1].
- **Finance:** Portfolio optimization, market forecasting.
- **Biology and Medicine:** DNA sequence analysis, drug discovery, modeling of biological systems.

Advantages and Disadvantages

Advantages

- **Robustness:** Ability to handle complex, non-linear problems with multiple local optima, as demonstrated in challenging localization scenarios [1].

- **Implicit Parallelism:** They explore multiple regions of the search space simultaneously through their population [1].
- **No Reliance on Gradient Information:** They do not require the objective function to be differentiable or continuous.
- **Flexibility:** They can be applied to a wide variety of problems with few modifications.
- **Find Global Solutions:** They exhibit a good trade-off between computational cost and exploration/exploitation capabilities, tending to find global solutions [1].

Disadvantages

- **Computational Cost:** They can be computationally expensive, especially for large populations and many generations [1].
- **Parameter Definition:** The choice of parameters (population size, crossover and mutation rates) can be challenging and significantly impact performance [1].
- **Slow Convergence:** Although robust, they may take many generations to converge to the optimal solution.
- **Fine-Tuning:** They can be slow at fine-tuning the solution in the later stages of optimization.
- **Encoding:** The problem representation (the individual's encoding) can be complex for some types of problems.

Principles

Particle Swarm Optimization (PSO) is an optimization metaheuristic inspired by the social behavior of bird flocks or fish schools. Proposed by Kennedy and Eberhart in 1995 [5], PSO is a population-based algorithm where each candidate solution (particle) moves through the search space, adjusting its trajectory based on its own experience (personal best position) and the experience of the swarm (global best position) [1,2].

- **Particles:** Each particle represents a potential solution in the search space and has a position and a velocity [2].
- **Personal Best Position (pBest):** Each particle keeps a record of the best position it has ever reached in the search space, along with its corresponding fitness value [1,2].
- **Global Best Position (gBest):** The swarm as a whole keeps a record of the best position found by any particle in the swarm, along with its corresponding fitness value [1,2].
- **Velocity and Position Update:** The velocity of each particle is updated based on three components: its inertia (previous velocity), the cognitive component (attraction to its personal best position), and the social component (attraction to the global best position). The new position of the particle is then calculated by adding the velocity to the current position [1,2].

Applications

PSO is known for its simplicity of implementation and its effectiveness on a variety of optimization problems, especially in continuous domains. Its applications include:

- **Function Optimization:** Finding the minima or maxima of non-linear mathematical functions [1].
- **Engineering:** Control system design, antenna optimization, route planning, resource allocation, and global localization of robots [1,2].
- **Signal Processing:** Adaptive filtering, pattern recognition.
- **Machine Learning:** Training neural network weights, feature selection, model parameter tuning, and in particle filters for robot localization [2].
- **Finance:** Portfolio optimization.

Advantages and Disadvantages

Advantages

- **Simplicity:** Easy to understand and implement, with few parameters to adjust.

- **Computational Efficiency:** Generally faster than other evolutionary algorithms for certain types of problems, reducing the number of particles needed compared to approaches like particle filters [2].
- **No Reliance on Gradient Information:** Like GAs, they do not require the objective function to be differentiable.
- **Good Exploration Capability:** Exhibits a good trade-off between computational cost and exploration/exploitation capabilities [1].

Disadvantages

- **Premature Convergence:** Tendency to converge prematurely to local optima in complex and multi-modal problems.
- **Parameter Sensitivity:** Performance can be sensitive to the choice of parameters (inertia weight, cognitive and social coefficients).
- **Difficulty with Discrete Problems:** Originally designed for continuous problems, it may underperform on discrete problems without specific adaptations.
- **Fine-Tuning:** It can have difficulty fine-tuning the solution in the later stages of optimization, and its precision may be limited without additional techniques [2].

Materials and Methods

We employed a methodology for the application and comparison of Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) to the sensor selection problem. The main objective is to identify a subset of sensors that minimizes the variance of the collected data, ensuring the consistency and quality of the measurements.

Problem Definition

The problem addressed consists of selecting a fixed number of sensors from a larger set in order to optimize a specific metric. In this study, the optimization metric is the minimization of the variance of the data collected by the selected sensors. Variance is a measure of the dispersion of data around the mean; therefore, a lower variance value indicates greater homogeneity and reliability in the sensor readings.

Formally, given a set of N sensors and a desired number of K sensors to be selected ($K < N$), the problem is to find the subset of K sensors such that the variance of the data collected by these K sensors is the smallest possible. The objective function (or fitness function, in the context of bio-inspired algorithms) is defined as the negative of the variance, since optimization algorithms generally seek to maximize fitness. Thus, maximizing the negative of the variance is equivalent to minimizing the variance. This type of optimization problem is common in various areas of robotics and autonomous systems, where localization and sensing are crucial.

Datasets

For the evaluation of the algorithms, two real-world datasets from sensor readings were used. The files, named `scan.csv` and `scan(1).csv`, contain multiple columns, of which only those prefixed with `ranges_` were considered. Each column that follows the `ranges_*` pattern represents the readings of a specific sensor over time. The use of two distinct datasets allowed for the evaluation of the robustness and generalization of each algorithm's performance in different scenarios.

Algorithm Configuration

Genetic Algorithm (GA)

The GA was configured based on literature recommendations and preliminary tests. A population of 50 individuals evolved over 100 generations, a setup that balanced diversity and computational cost, ensuring convergence without excessive processing. A mutation rate of 0.1 was chosen to maintain genetic variability and prevent premature convergence to local optima. The number of sensors to select was fixed at 10, as per the problem definition.

Particle Swarm Optimization (PSO)

The PSO was configured with 30 particles and 50 generations, leveraging its known rapid convergence. This setup proved sufficient for reaching stable solutions efficiently. The inertia weight (w=0.5) balanced exploration and exploitation, while cognitive (c1=1.5) and social (c2=1.5) coefficients were set to equally weigh individual and collective knowledge, fostering a robust search. The number of sensors to select was also 10.

Evaluation Metrics

To compare the performance of the algorithms, the following metrics and analyses were used:

- **Convergence of the Best Standard Deviation:** Convergence plots were generated to visualize how the best standard deviation (lowest variance) evolves over the generations for each algorithm and dataset. This allows for the analysis of the optimization's speed and stability.
- **Distribution of Standard Deviation in the Final Population/Swarm:** Box plots were used to analyze the distribution of standard deviation values of the solutions in the final population (GA) or final swarm (PSO). This provides insights into the diversity and quality of the solutions found.
- **Sensor Profile of the Best Solution:** Sensor profile plots were generated to visualize the average readings of the sensors selected by the best solution found by each algorithm. This helps to understand which sensors were considered most relevant for minimizing variance and how their readings behave.

Through this methodology, the aim is to provide a comprehensive and comparative analysis of the performance of GA and PSO in optimizing sensor selection, contributing to the choice of the most suitable approach in different application contexts.

Results and Discussion

This section presents and discusses the results obtained from the application of the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). The analysis focuses on the convergence of the algorithms, the fitness distribution of the final solutions, and the profile of the selected sensors. Two distinct datasets were used, hereafter referred to as Dataset A (`scan_2019-05-07-17-40-30`) and Dataset B (`scan_2019-05-07-15-54-45`), providing a basis for comparing their performances.

Convergence Analysis

The convergence graphs (Figures 1 to 4) illustrate the evolution of the best standard deviation value found by each algorithm over the generations. For Dataset A, the GA demonstrated a reduction in standard deviation from approximately 2.27 to 2.03 over 100 generations, showing a gradual and stable convergence. Similarly, for Dataset B, the GA converged from 2.22 to 1.97. In both cases, the GA's convergence curve exhibits discrete steps, indicating significant improvements in certain generations, followed by periods of stabilization.

In contrast, the PSO, executed for 50 generations, demonstrated faster convergence. For Dataset A, the standard deviation was reduced from 2.30 to 2.14. In the case of Dataset B, the convergence was from 2.23 to 2.15. The convergence speed of PSO is a notable characteristic, attributed to its nature of a guided search by the swarm's global best experience. However, it is observed that, despite the accelerated convergence, the GA achieved slightly lower standard deviation values, suggesting a superior ability to find higher-quality solutions in terms of variance minimization, given a larger number of iterations.

Figure 1. Convergence history of the best standard deviation for GA (Dataset A).

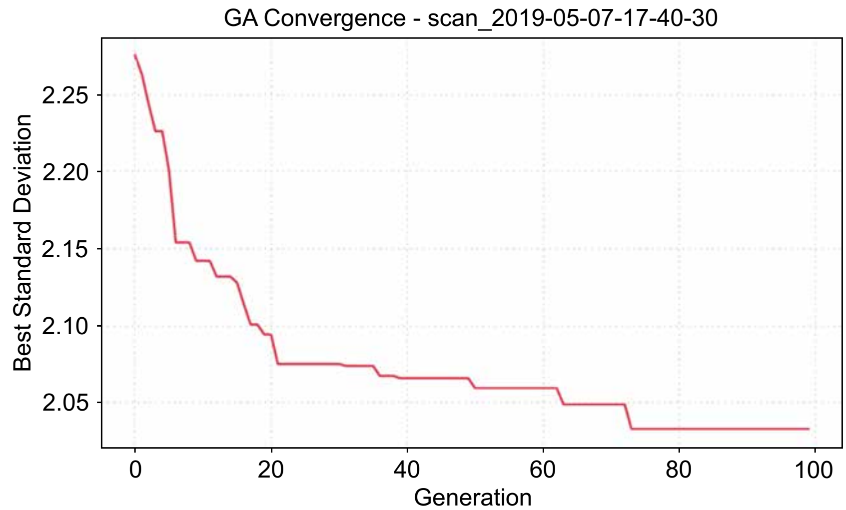


Figure 2. Convergence history of the best standard deviation for GA (Dataset B).

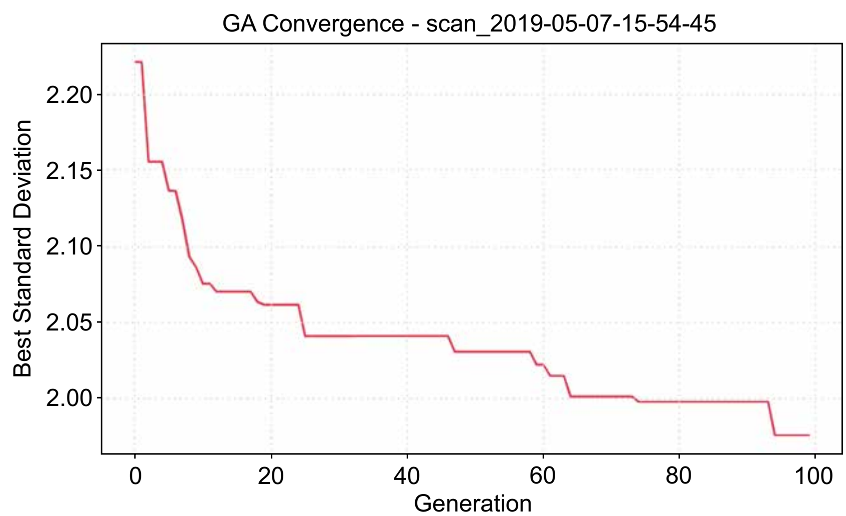


Figure 3. Convergence history of the best standard deviation for for PSO (Dataset A).

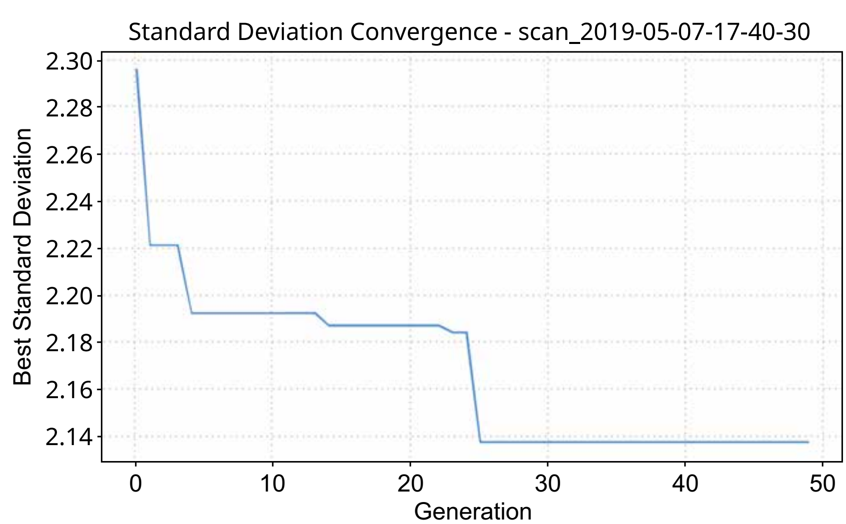
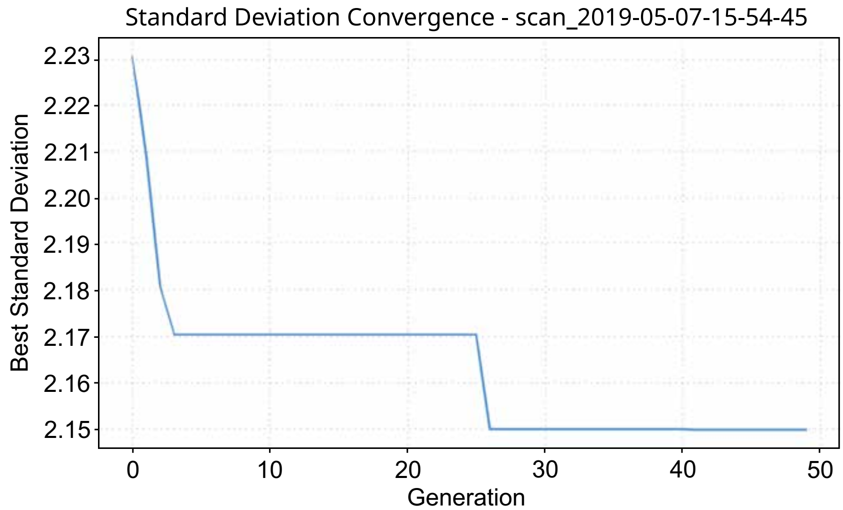


Figure 4. Convergence history of the best standard deviation for for PSO (Dataset B).



Limitations and Practical Implications

This study has limitations, including parameter sensitivity and the use of a single optimization metric (variance). The scalability to larger sensor networks was not exhaustively tested. The sensor selection was static, whereas dynamic environments might require adaptive strategies.

Practically, the results guide algorithm choice: GA is preferable for applications prioritizing data quality, while PSO excels when rapid solutions are critical. This work demonstrates that sensor subset selection can optimize resources (cost, energy) and improve data reliability, offering a foundation for advanced, adaptive sensing systems.

Fitness Distribution in the Final Popula-tion/Swarm

The box plots (Figures 5 to 8) provide a view of the distribution of standard deviation values of the solutions in the final population (GA) and the final swarm (PSO). For the GA, in Dataset A, most solutions had a standard deviation concentrated between 2.15 and 2.20, with an outlier at 2.30. In Dataset B, the distribution was slightly wider, ranging between 2.15 and 2.25.

For PSO, the standard deviation distribution in the final swarm for Dataset A was concentrated between 2.35 and 2.42. In Dataset B, the distribution varied between 2.32 and 2.40, with an outlier at 2.52. The analysis of the plots reve-als that both algorithms generated outliers, but the GA, in general, produced solutions with a lower average standard deviation and a more compact distribution, indicating gre-ater consistency in solution quality. The PSO, on the other hand, showed a distribution at slightly higher values and, in some cases, which might indicate greater diversity in the swarm, but also less convergence to high-quality solutions across all particles.

Selected Sensor Profile

The sensor profile graphs (Figures 9 to12) illustrate the indices of the sensors selected by the best solution of each algorithm, as well as their respective average readings.

For the GA, in Dataset A, the selected sensors were: `110, 177, 180, 181, 202, 209, 222, 223, 224, 225`. In Dataset B, the chosen sensors were: `149, 175, 183, 188, 190, 191, 194, 196, 208, 210`.

We observed that the GA tended to select sensors with numerically close indices, suggesting that the optimization sought a cohesive set to minimize the variance of the readings.

For the PSO, in Dataset A, the selected sensors were: `64, 80, 101, 116, 118, 163, 165, 168, 182, 194`. In Dataset B, the sensors were: `156, 159, 182, 195, 197, 216, 284, 323, 388, 431`.

Figure 5. Box-plot of the standard deviation distribution in the final population for GA (Dataset A).

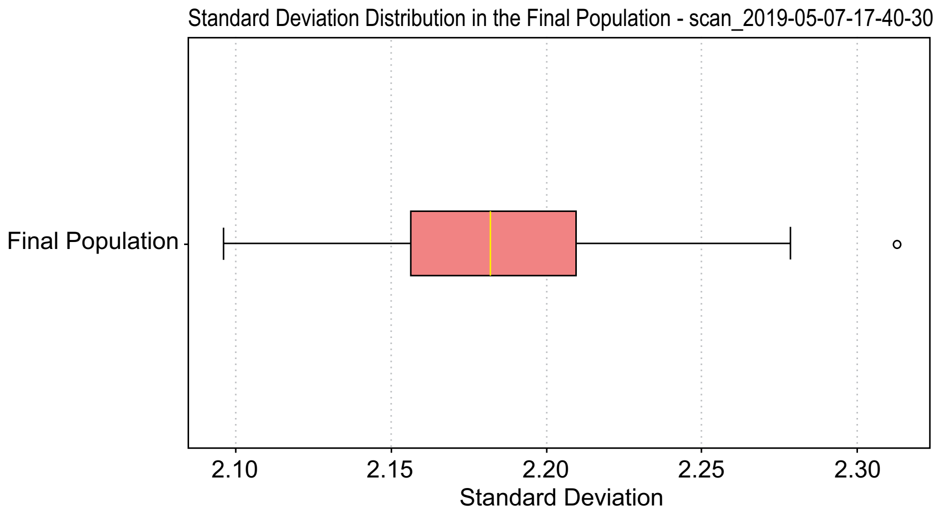


Figure 6. Box-plot of the standard deviation distribution in the final population for GA (Dataset B).

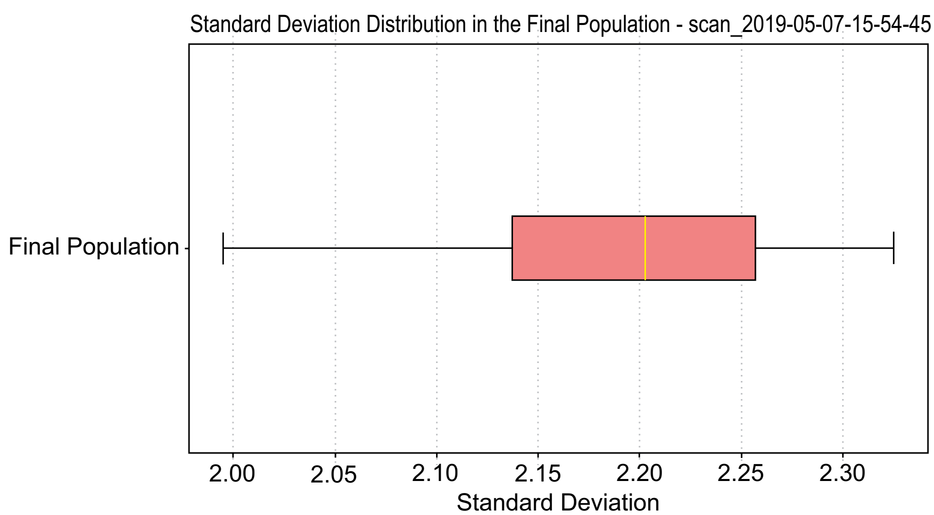


Figure 7. Box-plot of the standard deviation distribution in the final population for PSO (Dataset A).

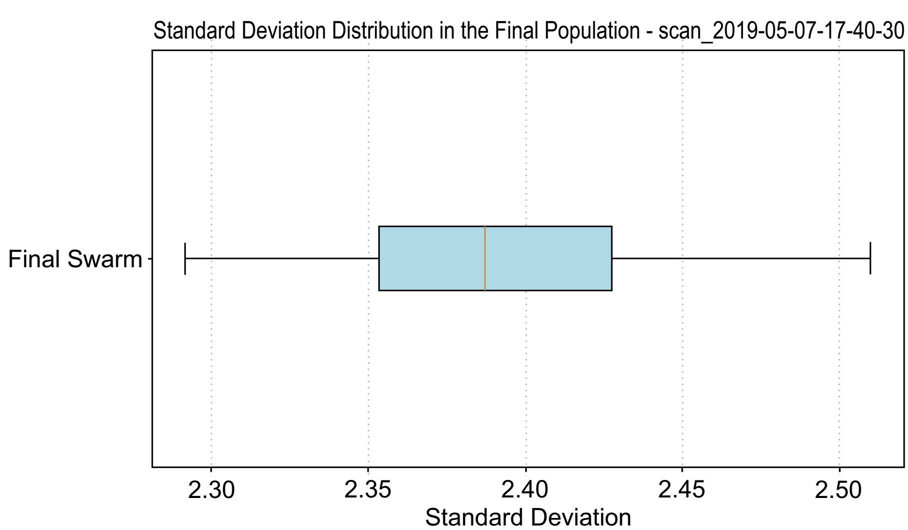


Figure 8. Box-plot of the standard deviation distribution in the final population for PSO (Dataset B).

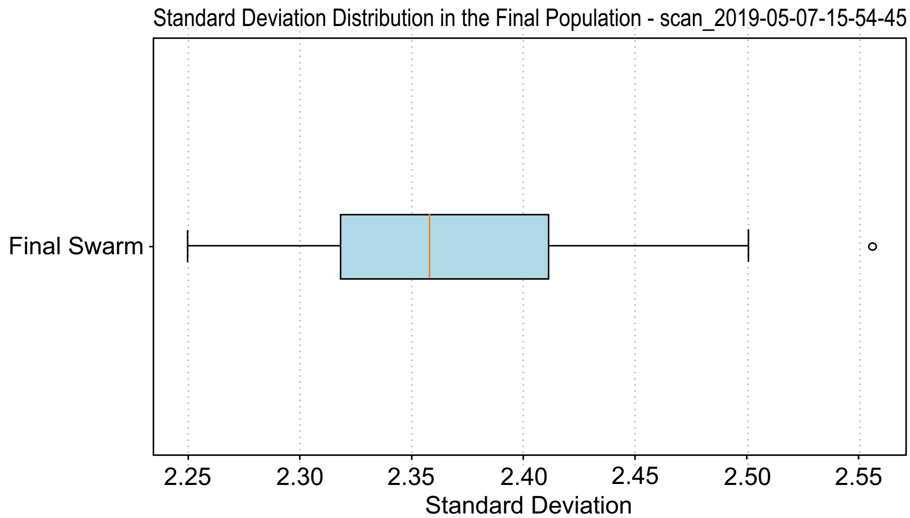


Figure 9. Sensor profile of the best solution for GA (Dataset A).

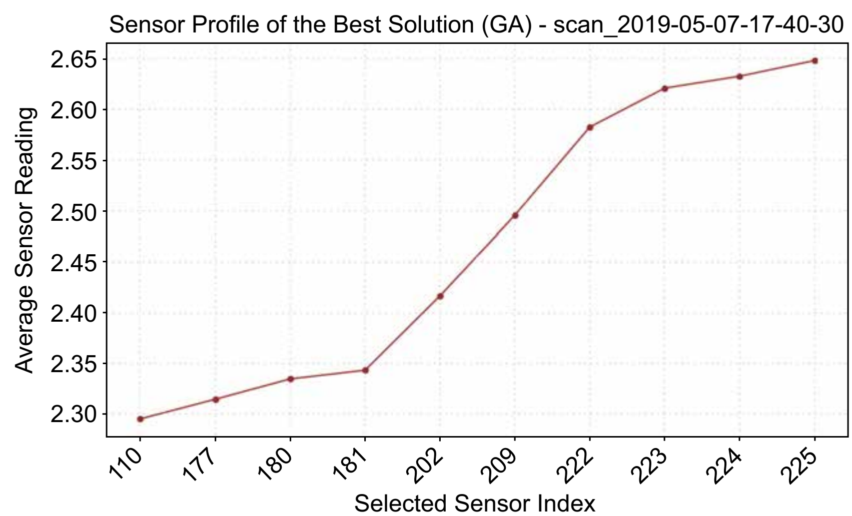


Figure 10. Sensor profile of the best solution for GA (Dataset B).

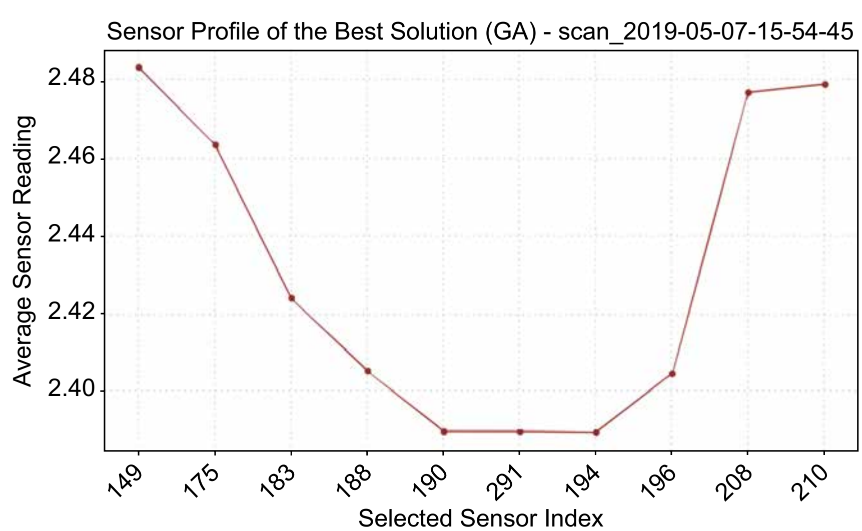


Figure 11. Sensor profile of the best solution for PSO (Dataset A).

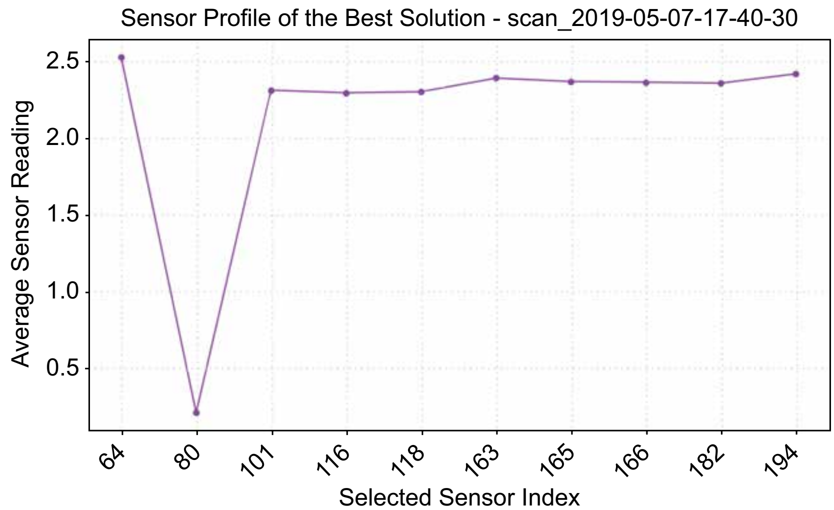
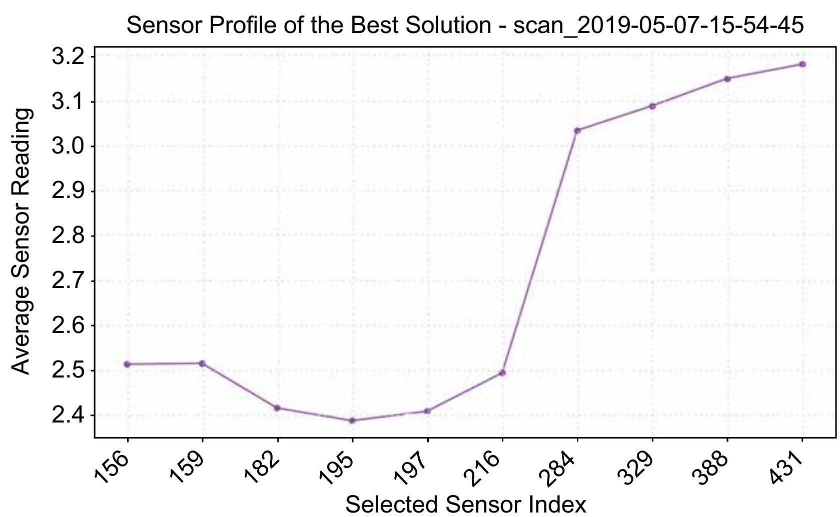


Figure 12. Sensor profile of the best solution for PSO (Dataset B).



Unlike the GA, the PSO selected sensors with more scatte-red indices in the search space. This may indicate a broader exploration, but it does not necessarily result in a cluster of sensors with homogeneous characteristics, which is the objective of variance minimization.

Overall Comparison

In terms of performance, the GA demonstrated a superior ability to find solutions with a lower standard deviation, resulting in a more effective optimization for the proposed problem. Its convergence, although slower, was more sta-ble and consistent. On the other hand, the PSO stood out for its convergence speed, reaching satisfactory solutions in a smaller number of generations. However, the quality of the PSO solutions, on average, was slightly inferior to that of the GA, and the fitness distribution of the final swarm was more dispersed.

The choice of the most suitable algorithm will depend on the specific requirements of the application. If the pri-ority is the quality of the solution and robustness in opti-mization, the GA may be more indicated. If convergence speed is a critical factor, PSO may be preferable, even if it implies a slightly less optimized solution. The difference in the selected sensor profiles also suggests that each algo-rithm explores the search space differently, which can be an advantage in problems where solution diversity is desirable.

Conclusion

This work presented a comparative study between GA and PSO for sensor selection, focusing on data variance minimization. The results demonstrated that GA, despite slower convergence, achieved superior solutions with lower standard deviation, indicating greater effectiveness in opti-mizing data quality. PSO, on the other hand, excelled in convergence speed, quickly reaching satisfactory solutions, though of slightly lower quality.

The analysis of sensor profiles revealed distinct selec-tion patterns: GA favored cohesive sensor clusters, while PSO selected more dispersed sensors, reflecting different search strategies. This trade-off between solution quality (GA) and convergence speed (PSO) is a key takeaway. The choice of algorithm should depend on application-specific needs: prioritize optimality with GA or rapid deployment with PSO.

Future work could explore hybrid approaches combi-ning the strengths of both algorithms and investigate their application in dynamic environments with multi-objective optimization criteria.

References

1. Carvalho JLC, Farias PCMA, Simas Filho EF. Global localization of unmanned ground vehicles using swarm intelligence and evolutionary algorithms. J Intell Robot Syst. 2023;107:45. doi:10.1007/s10846-023-01813-6.
2. Zhang Q, Wang P, Bao P, Chen Z. Mobile robot global localization using particle swarm optimization with a 2D range scan. In: Proceedings of the 2017 International Conference on Robotics and Automation; 2017. doi:10.1145/3175603.3175618.
3. Sobreira H, Costa CM, Sousa I, Rocha L, Lima J, Farias PCMA, et al. Map-matching algorithms for robot self-localization: a comparison between perfect match, iterative closest point and normal distributions transform. J Intell Robot Syst. 2020;93:533-546. doi:10.1007/s10846-017-0765-5.
4. Menines MS, Entringer AMM, Silva LA, Sá FB, Izabel VH, Vassallo RF. Detection and localization of mobile robots in intelligent spaces using artificial neural networks. [Publication details unavailable].
5. Kennedy J, Eberhart R. Particle swarm optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.