Comparison Of Standard PSO and Modified PSO Algorithm Using Benchmark Functions

1st Aniket Pradip Dere
Software Engineering
University of Europe for Applied Sciences
Potsdam, Germany
aniket.dere@ue-germany.de

2nd Aiswarya Sajeev
Software Engineering
University of Europe for Applied Sciences
Potsdam, Germany
aiswarya.sajeev@ue-germany.de

3rd Suraj Deshwal
Software Engineering
University of Europe for Applied Sciences
Potsdam, Germany
suraj.deshwal@ue-germany.de

4th Ajil Aleyas
Software Engineering
University of Europe for Applied Sciences
Potsdam, Germany
ajil.aleyas@ue-germany.de

5th Marcelina Owono
Software Engineering
University of Europe for Applied Sciences
Potsdam, Germany
marcelina.owono@ue-germany.de

***Abstract*—The paper presents a new variant of Particle Swarm Optimization (PSO) called Multi-Swarm PSO (MSPSO). MSPSO is designed to improve the performance of PSO by using multiple swarms of particles. The paper presents the results of a simulation study that compares MSPSO to standard PSO on a set of benchmark functions. The results show that MSPSO is able to achieve better results than standard PSO in terms of both convergence speed and solution quality.Particle Swarm Optimization (PSO) has gained significant attention in solving optimization problems due to its simplicity and effectiveness. This study presents a comprehensive comparison between the performance of the standard PSO algorithm and a modified version of PSO on a set of benchmark functions commonly used to evaluate optimization algorithms. The objective is to analyze and identify the strengths and weaknesses of both approaches in terms of convergence speed, solution accuracy, and robustness. The modified PSO algorithm incorporates novel enhancements or variations to the standard PSO to explore potential improvements in optimization performance. The benchmark functions chosen cover a range of complexities and characteristics to provide a diverse evaluation environment. Experimental results and statistical analyses are presented to illustrate the comparative outcomes, shedding light on the relative advantages and limitations of each algorithm variant. The findings of this study contribute valuable insights for researchers and practitioners seeking to choose or enhance PSO algorithms for solving diverse optimization challenges.**

Keywords—Modified PSO, Standard PSO , Multi Swarm PSO

I. Introduction

Particle Swarm Optimization (PSO) is a meta-heuristic optimization algorithm inspired by the social behaviour of bird flocking or fish schooling. It was first introduced by Eberhart and Kennedy in 1995. PSO is a population-based algorithm that maintains a population of particles, where each particle represents a potential solution to the optimization problem. Modified versions of the Particle Swarm Optimization (PSO) algorithm often incorporate enhancements or changes to the original algorithm to address specific challenges or improve its performance in certain scenarios. These modifications can target issues like premature convergence, exploration-exploitation balance, and handling constraints.

II. Standard PSO

Basic PSO Algorithm:
1. Initialization:
• Initialize a population of particles with random positions and velocities in the search space.
• Evaluate the fitness of each particle based on the objective function.
2. Update Particle's Velocity and Position:
• For each particle, update its velocity and position using the following equations:

$$\begin{aligned} v_i(t+1) = & w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_i(t) - x_i(t)) \\ & + c_2 \cdot r_2 \cdot (g(t) - x_i(t)) \end{aligned} \tag{1}$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{2}$$

Where:

- vi(t) is the velocity of particle i at time t
- xi(t) is the position of particle i at time t
- p i(t) is the personal best position of particle i found so far
- g i(t) is the global best position among all particles up to time
- w is the inertia weight
- c1 and c2 are the cognitive and social parameters
- r1 and r2 are random values between 0 and 1.

3.Update Personal and Global Best:
• Update each particle's personal best position (pbesti) if the current position has a better fitness value.
• Update the global best position (gbest) if any particle has a better fitness value than the current gbest.
4. Termination:
• Repeat steps 2 and 3 for a predefined number of iterations or until a termination criterion is met (e.g., reaching a satisfactory fitness level).

III. Modified Particle Swarm Optimization

MPSO utilizes the same parameters and design criteria as Standard PSO. The key factor that influences the performance of the smart PSO is the addition of the new term in the position and velocity equations. The idea behind this new term is to reduce the distance between each particle over time which can result in an increase in the velocity of the particle. The new velocity equation will be

$$\begin{aligned} v_{i,d} = & w \cdot v_{i,d} + c_1 \cdot R_1 (p_{i,d} - x_{i,d}) \\ & + c_2 \cdot R_2 (p_{g,d} - x_{i,d}) + w \left(\frac{c_1}{c_2} \right) (p_{i,d} - p_{g,d}) \end{aligned} \tag{3}$$

where:

- $v_{i,d}$: Velocity of particle i in dimension d.
- w: Inertia weight, controlling the impact of the previous velocity.
- v_{id}: Previous velocity of particle i in dimension d.
- c_1, c_2: Acceleration coefficients controlling the influence of the particle's personal best ($p_{i,d}$) and the global best ($p_{g,d}$) positions.
- R_1, R_2: Random values between 0 and 1.
- $p_{i,d}$: Personal best position of particle i in dimension d.
- $x_{i,d}$: Current position of particle i in dimension d.
- $p_{g,d}$: Global best position among all particles in dimension d.

The term $(p_{i,d} - x_{i,d})$ represents the difference between the current position of the particle and its personal best position, and $(p_{g,d} - x_{i,d})$ represents the difference between the current position of the particle and the global best position.

The additional term $\left(\frac{c_1}{c_2}\right) \cdot (p_{i,d} - p_{g,d})$ is multiplied by the inertia weight w. This term is often used to further adjust the velocity based on the difference between the personal best and global best positions. It can be considered as an additional influence on the particle's movement, combining the individual and global information.

Dividing the two acceleration constants in (1) helps to make the resulting value neither too big nor very small as both C1 and C2 have an enormous influence on the agent's mobility. The position update equation is given by

$$\mathrm{x}_{id}(k+1) = w * x_{id}(k) + v_{id}$$

IV. Impact of Modified PSO

Modified versions of Particle Swarm Optimization (PSO) aim to address the limitations and improve the performance of the standard PSO algorithm. Here are some potential benefits and impacts of using modified PSO:

Improved Convergence Speed:

Many modifications are designed to accelerate the convergence speed of PSO, allowing the algorithm to reach optimal or near-optimal solutions more quickly. Enhanced Exploitation and Exploration:

Adjustments to parameters, such as inertia weight and constriction coefficients, can strike a better balance between exploitation (fine-tuning around current solutions) and exploration (searching for new solutions). This can lead to improved global and local search capabilities.

Better Handling of Constraints:

Modifications that address constraint handling make the algorithm more robust in dealing with optimization problems that have constraints. This is particularly important for real-world problems with practical limitations.

Adaptability to Dynamic Environments:

Dynamic PSO variants can adapt to changes in the optimization landscape over time, making them suitable for problems where the optimum may shift or evolve during the optimization process.

Increased Robustness:

Incorporating mechanisms for local search or hybridizing PSO with other optimization techniques can enhance the algorithm's robustness, making it more effective across a broader range of problem types.

Parameter Adaptation:

Adaptive PSO variants dynamically adjust algorithm parameters during runtime, allowing the algorithm to adapt to the specific characteristics of the optimization problem. This adaptability can result in improved performance across different problem domains.

Addressing Premature Convergence:

Some modifications aim to mitigate premature convergence issues by introducing mechanisms that encourage exploration, preventing the algorithm from getting stuck in suboptimal solutions too early.

Increased Applicability:

Modified PSO algorithms can be tailored to specific problem domains, making them more versatile and applicable to a wider range of optimization challenges.

Hybridization with Other Algorithms:

Hybrid PSO algorithms, combining PSO with other optimization or machine learning techniques, can leverage the strengths of multiple methods, potentially achieving better performance than standalone algorithms.

Facilitation of Multi-Objective Optimization:

Some modifications extend PSO to handle multi-objective optimization problems, enabling the algorithm to find trade-off solutions among conflicting objectives. It's important to note that the impact of modifications depends on the specific characteristics of the optimization problem. The selection and effectiveness of modifications should be carefully considered based on the nature of the problem at hand. Additionally, empirical testing and benchmarking are often necessary to validate the benefits of modified PSO variants for specific applications Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—LaTeX will do that for you.

V. Benchmark Functions

Benchmark functions are mathematical functions often used in optimization and machine learning to evaluate the performance of algorithms. These functions serve as test cases to assess how well an optimization algorithm can find the optimal solution. It commonly used in the context of Particle Swarm Optimization (PSO) and its modified versions like Multi-Objective Particle Swarm Optimization (MOPSO) to evaluate and compare the performance of these optimization algorithms. Here are a few reasons why benchmark functions are applied in the context of PSO and MOPSO:

Performance Evaluation: Benchmark functions provide a standardized set of problems with known solutions or global optima. By applying PSO or MOPSO to these functions, researchers and practitioners can evaluate how well the algorithms perform in terms of convergence speed, accuracy, and robustness.

Algorithm Tuning: Benchmark functions serve as a tool for fine-tuning the parameters and configurations of PSO and MOPSO. Experimenting with various benchmark functions helps researchers understand how the algorithms respond to different types of optimization challenges, leading to better-tailored algorithms.

Algorithm Comparison: Since benchmark functions are widely used and well-documented, they provide a common ground for comparing different optimization algorithms, including PSO and MOPSO. Researchers can assess which algorithm performs better across a variety of problems, leading to advancements and insights into the strengths and weaknesses of each approach.

Generalization of Results: Using benchmark functions ensures that the results obtained from the optimization algorithms are not specific to a particular problem domain. This allows researchers to make more general claims about the algorithm's performance and applicability.

Understanding Algorithm Behavior: Benchmark functions often have specific characteristics like multimodality, nonlinearity, and separability. Applying PSO and MOPSO to these functions helps in understanding how well the algorithms handle these challenges, shedding light on their behavior in different scenarios.

Publication and Reproducibility: When researchers publish their work on PSO or MOPSO, using benchmark functions enhances the reproducibility of results. Others in the field can replicate experiments easily since the benchmark functions are well-defined and widely accepted.

In summary, benchmark functions play a crucial role in assessing, comparing, and improving the performance of optimization algorithms, including PSO and MOPSO. They provide a standardized and fair way to evaluate how well these algorithms tackle various optimization challenges.

VI. Results and Discussions

This study presents a far reaching investigation of the presentation of the standard PSO calculation compared with a changed PSO variation. The emphasis is on improving a different arrangement of benchmark functions, each presenting exceptional difficulties for improvement calculations. We fastidiously examine the outcomes got by both the standard and modified PSO calculations, digging into the best positions and relating objective capability values accomplished for every benchmark. The overarching objective is to identify patterns and trends that indicate whether the modified PSO algorithm possesses superior convergence and optimization capabilities across this collection of benchmark functions.

The evaluation involves breaking down the results for each benchmark function, looking at both the quantitative improvements in terms of the best values achieved and the qualitative improvements, like changes in the best positions. This examination intends to give experiences into the flexibility and strength of the changed PSO calculation across a range of improvement challenges.

Through this comprehensive examination, we seek to contribute to the understanding of how modifications to the standard PSO algorithm impact its performance on a diverse set of benchmark functions. The results and comparisons presented here shed light on the effectiveness of the modified PSO algorithm in addressing the inherent complexities posed by various optimization landscapes and provide valuable insights into its strengths and weaknesses.

if the modified PSO yields a lower best value compared to the standard PSO, it is generally considered an improvement.

The compared results are explained as improve or not improve. Basically, If the modified PSO

TABLE I
PERFORMANCE COMPARISON: STANDARD PSO VS. MODIFIED PSO - BEST POSITIONS AND VALUES FOR VARIOUS BENCHMARK FUNCTIONS

Function	Standard PSO	Modified PSO
Sphere	Best Position [2.736, -1.526], Best Value: 9.813E-14	Best Position [4.842, -2.552], Best Value: 0.567
Rosenbrock	Best Position [1.000, 1.000], Best Value: 7.482E-13	Best Position [1.364, 9.385], Best Value: 7.257
Step 2	Best Position [-0.439, 0.121], Best Value: 0.000	Best Position [-10.000, -3.622], Best Value: 1.000
Quartic	Best Position [0.000, -0.000], Best Value: 1.179E-10	Best Position [2.004, 6.000], Best Value: 0.609
Schwefel 2.21	Best Position [-0.000, -0.000], Best Value: 7.304E-07	Best Position [5.740, -4.075], Best Value: 1.313
Schwefel 2.22	Best Position [2.467, 2.467], Best Value: -2.000	Best Position [0.689, 2.247], Best Value: -1.999
Foxholes	Best Position [-3.604, 5.941], Best Value: 0.950	Best Position [10.000, -10.000], Best Value: 0.916
Kowalik	Best Position [1.000, 1.000], Best Value: 1.882E-08	Best Position [6.299, 10.000], Best Value: 0.096
Six-hump Camel Back	Best Position [0.090, -0.714], Best Value: -1.032	Best Position [-3.828, -3.464], Best Value: -0.649
Levi N13	Best Position [-4.599, -4.512], Best Value: -2.000	Best Position [6.068, 5.305], Best Value: -1.608
Rastrigin	Best Position [0.000, -0.000], Best Value: 3.436E-09	Best Position [-1.007, 0.536], Best Value: 4.105
Griewank	Best Position [0.000, 0.000], Best Value: 1.577E-12	Best Position [6.046, -4.730], Best Value: 0.011
Ackley 1	Best Position [0.000, 0.000], Best Value: 2.690E-05	Best Position [0.982, -0.369], Best Value: 2.742
Schwefel 2 26	Best Position [0.000, 0.000], Best Value: 5.930E-06	Best Position [-9.012, 9.665], Best Value: 1.271
Branin Rcos	Best Position [9.425, 2.475], Best Value: 0.398	Best Position [9.879, 2.916], Best Value: 1.374
Goldstein Price	Best Position [0.000, -1.000], Best Value: 3.000	Best Position [-2.493, 6.706], Best Value: 115.205
Penalized 1	Best Position [-1.000, 1.000], Best Value: 1.655E-09	Best Position [-6.038, 2.132], Best Value: 0.004

achieves a lower best value than the standard PSO, it is considered an improvement. This implies that the modified algorithm is more effective at finding solutions with lower objective function values, which is the primary goal in optimization. However, If the modified PSO results in a higher best value than the standard PSO, it suggests that the modification may not have been successful in improving the algorithm's performance for that particular problem. In this case, the modified algorithm did not find a solution that is better (lower) than what the standard algorithm achieved.

Definitely, the changed PSO calculation didn't in every case outflank the standard PSO across all benchmark capabilities. While it showed improvement at times, in others, it either didn't improve or even brought about a less ideal arrangement. The viability of the changed PSO seems, by all accounts, to be capability explicit, and further examination might be expected to figure out its conduct across a more extensive scope of enhancement issues.

TABLE II
STATISTICAL ANALYSIS RESULTS: T STATISTICS AND P VALUES FOR MULTIPLE RUNS

Run Number	T Statistic	P Value
1	-3.985718982	7.55818E-05
2	3.705283609	0.000233149
3	1.060350368	0.28942394
4	-2.711930369	0.006866426
5	-0.418491619	0.675677445
6	0.333863812	0.738556262
7	-0.838626901	0.401904102
8	1.60782158	0.108218916
9	-0.666309834	0.505491121
10	0.701415135	0.483206898
11	0.952812646	0.341088497
13	-2.712489451	0.006887818
14	-1.596384516	0.110719654
15	-2.250297042	0.024649837

The statistical analysis conducted on fifteen experimental runs comparing the standard and modified PSO algorithms yielded notable findings regarding their performance differences. Table II encapsulates these findings, presenting the T statistics and P values corresponding to each run.

The analysis revealed significant performance differences between the algorithms in five out of fifteen runs, as indicated by P values less than 0.05. Specifically:

Run 1 exhibited a T statistic of -3.985718982 with a P value of 7.55818E-05, indicating a statistically significant difference where the standard PSO algorithm outperformed the modified version.

Run 2 showed a T statistic of 3.705283609 and a P value of 0.000233149, suggesting the modified PSO algorithm's superiority over the standard version in this instance.

Run 4 had a T statistic of -2.711930369 with a P value of 0.006866426, further supporting scenarios where the standard PSO algorithm performed better than the modified version.

Runs 13 and 15 reported T statistics of -2.712489451 and -2.250297042 with P values of 0.006887818 and 0.024649837, respectively, indicating the standard algorithm's better performance in these runs.

Conversely, the remaining ten runs did not show statistically significant differences between the two algorithms, as denoted by P values greater than 0.05. This observation suggests that under certain conditions, the performance of the standard and modified PSO algorithms is comparable. Specifically:

Runs 3, 5, 6, 7, 8, 9, 10, 11, and 14 reported P values ranging from 0.108218916 to 0.738556262, indicating no significant performance difference between the algorithms.

These results underscore the context-specific efficacy of the PSO algorithm modifications. While some experimental conditions highlighted the modified algorithm's advantages, others favored the standard version, and yet in many instances, both algorithms performed similarly.

The detailed examination of T statistics and P values serves as a quantitative basis for assessing the relative performance of standard and modified PSO algorithms. The significant variations observed in specific runs underscore the importance of algorithm selection based on the specific optimization problem and conditions at hand.

VII. Conclusion

This comprehensive comparative analysis of the standard Particle Swarm Optimization (PSO) algorithm and its modified counterpart across various benchmark functions provides a difference understanding of the strengths and limitations of these optimization techniques. The evaluation considered the best position and best value achieved by each algorithm for each benchmark function, offering valuable insights into the efficacy of the modified PSO.

The results across benchmark functions revealed a diverse range of outcomes, showing the variable effect of the changed PSO. While certain capabilities exhibited negligible enhancements, others showed no massive changes or, at times, a deterioration in execution. This variability underscores the importance of tailoring optimization algorithms to specific problem characteristics, as there is no one-size-fits-all solution.

The function-specific behavior observed in this study suggests that the modified PSO's effectiveness is intricately tied to the underlying mathematical structure of the optimization problem. Functions such as Step 2 witnessed notable improvements, emphasizing the potential benefits of customization. Be that as it may, functions like Circle and Quartic showed almost no improvement, featuring the calculation's particular effect.

The modified PSO's performance trade-offs became evident as the algorithm excelled in certain functions while faltering in others. This complexity raises questions about the adaptability of the modified approach across a broader spectrum of optimization challenges. Researchers and experts must carefully weigh the benefits against potential drawbacks, considering the specific characteristics of the optimization problem at hand.

The review's results highlight the requirement for more in-depth investigations into the modified PSO's behavior. Understanding the factors influencing its performance could lead to refinements or hybridizations that enhance its versatility. Furthermore, extending the investigation to incorporate a more extensive array of benchmark functions and real-world optimisation issues could give a more far reaching evaluation of the calculation's utility.

The modified PSO's performance distinction imply that algorithmic modifications should not be universally applied. Instead, a careful assessment of the optimization landscape is essential before implementing customized approaches. This finding aligns with the broader trend in optimization research, emphasizing the significance of algorithmic specificity for improved problem-solving capabilities.

Future examination should delve deeper into understanding the modified PSO's behavior, potentially exploring adaptive strategies or dynamic parameter changes. Comparative analyses across various optimization algorithms and hybrid approaches may unveil synergies that enhance overall optimization efficiency. Additionally, real-world applications and large-scale optimization problems should be prioritized to validate the algorithm's practical utility.

In conclusion, this examination gives an extensive evaluation of the changed PSO calculation's performance across a different arrangement of benchmark functions. While the algorithm exhibited promising results in specific cases, its function-specific impact and occasional suboptimal outcomes highlight the necessity for further refinement and exploration. A difference comprehension of problem-specific characteristics and careful consideration of the trade-offs associated with algorithmic modifications are necessary on the path to improving optimization algorithms.

REFERENCES

[1] Davis, L., Ed. (1991). Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York,NY.

[2] Eberhart, R. C. and R. W Dobbins (1990). Neural Network PC Tools: A Practical Guide. Academic Press, San Diego, CA

[3] Fisher, R.A. (1936). The use of multiple measurements in taxonomic problems. Annals of Eugenics, 7: 179-188.

[4] Heppner, F. and U. Grenander (1990). A stochastic nonlinear model for coordinated bird flocks. In S . Krasner, Ed., The Ubiquity of Chaos. AAAS Publications, Washington, DC

[5] Holland, J. H. (1992). Adaptation in Natural and Artijlcial Systems. MIT Press, Cambridge, MA.

[6] Cao Y, Ye Y, Zhao H, Jiang Y, Wang H, Shang Y, Wang J (2018) Remote sensing of water quality based on hj-1a hsi imagery with modified discrete binary particle swarm optimization-partial least squares (mdbpso-pls) in inland waters: a case in weishan lake. Eco Inform 44:21–32

[7] Cao Y, Zhang H, Li W, Zhou M, Zhang Y, Chaovalitwongse WA (2018) Comprehensive learning particle swarm optimiza- tion algorithm with local search for multimodal functions. IEEE Trans Evol Comput