Improved Information Sharing Mechanism (I2SM) for Metaheuristic Efficiency: A PSO Case Study

Maria Zemzami[1], Chakib Benmhamed[2], Hakima Reddad[2], Farouk Yalaoui[3] and Nhan-Quy Nguyen[3]

Abstract— **This paper introduces a novel information sharing mechanism, the Improved Information Sharing Mechanism (I2SM), an adaptive real-time framework designed to enhance the performance of metaheuristic algorithms. I2SM dynamically collects and evaluates critical metrics, such as improvement rates and stagnation levels, through parallel processing, enabling real-time actions such as hybridization and parameter tuning. The mechanism's adaptive nature ensures efficient handling of diverse optimization challenges by dynamically balancing exploration and exploitation, with a reasonable trade-off in execution time.**

To assess the performance of the proposed I2SM mechanism, we selected the Particle Swarm Optimization (PSO) algorithm as a representative test framework. Empirical results from various benchmark functions demonstrate that PSO integrated with I2SM achieves superior performance, outperforming standard PSO in 90% of the cases. Although I2SM-PSO incurs a slightly higher execution time compared to standard PSO, significant improvements in solution quality validate its efficiency. However, this increase in execution time highlights a limitation that should be addressed in future research to optimize computational efficiency while maintaining performance gains.

I. INTRODUCTION

Metaheuristic algorithms, although widely utilized for solving complex optimization problems [1]–[4], have several inherent limitations. These issues include premature convergence [5]–[8], stagnation in local optima [9], [10], and inefficient exploration and exploitation of search space [6], [11]. A significant cause of these problems is the lack of effective communication within the algorithm [12]–[15]. Many metaheuristic algorithms do not fully utilize the valuable information from individuals in previous iterations to guide their current and future searches [16], [17].

For example, algorithms like Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) discard previous instances directly, while others, such as Cuckoo Search (CS) and Particle Swarm Optimization (PSO), only make use of the best previous individuals. To address these challenges, researchers have explored various strategies, such as hybridizing metaheuristics with other optimization techniques [7], [18], [19] or dynamically adjusting parameters [20] in response to the algorithm's performance. However, these adjustments are often reactive rather than proactive, relying on predefined rules or external interventions.

The literature is abundant with studies proposing various information sharing mechanisms (ISM) to address the limitations of optimization algorithms [12], [21], [22], [22]–[32]. These mechanisms aim to improve communication within the algorithm, allowing better coordination among particles or agents, and thereby improving overall performance [21], [22], [22]–[33].

The proposed **Improved Information Sharing Mechanism (I2SM)** builds on the foundational concept of the ISM mechanism [33] by introducing an adaptive framework that fully utilizes metaheuristic performance data throughout the optimization process. Unlike conventional ISM approaches, which can employ static or limited strategies [33], I2SM dynamically evaluates key performance indicators (KPI) — including the improvement rate (A), the convergence rate, and the stagnation (S) — in each iteration. Based on these metrics, the mechanism activates predefined triggers to implement targeted adjustments or hybrid strategies, ensuring a balance between exploration and exploitation.

To demonstrate its effectiveness, I2SM is integrated with the PSO algorithm as a case study, incorporating specific modifications to address challenges such as premature convergence [5] and unbalanced exploration [6].

The integration of adaptive triggers, hybridization, and parallelism enhances PSO's efficiency and robustness while offering a scalable framework for improving other metaheuristics. The PSO case study demonstrates the ability of I2SM to overcome key limitations of traditional metaheuristics, such as poor information sharing and limited adaptability.

The article begins with Section 2, which offers a detailed analysis of existing mechanisms through three key perspectives: information structure, information support, and information state: information structure, information support, and information state. Section 3 introduces the proposed I2SM mechanism, detailing its anatomy and functional features. Section 4 demonstrates a case study using the PSO algorithm, particularly sensitive to insufficient information sharing.

II. LITERATURE REVIEW

Metaheuristic algorithms are widely recognized as powerful optimization tools, each defined by unique mechanisms that guide selection, exploration, and exploitation strategies [8], [34]. Among these mechanisms, the Information-Sharing

*This work is ostensibly supported by the Chaire Connected Innovation and ZAY digital world Company.

[1]Maria Zemzami is with ENSAM, University Mohammed V, Rabat, Morocco mariazemzami@gmail.com

[2] Benmhamed C. and Reddad H. Researchers are with the Laboratory of Advanced Systems Engineering, National School of Applied Sciences ENSA, Ibn Tofail University, Kenitra, Morocco benmhamedchakib@gmail.com, reddadhakima@gmail.com

[2] Yalaoui F. and Nguyen N-Q. Researchers are with the Chaire Connected Innovation, University of Technology of Troyes, 12 rue Marie Curie, Troyes, France. farouk.yalaoui@utt.fr, nhan_quy.nguyen@utt.fr

Mechanism (ISM) plays a pivotal role by coordinating and facilitating communication between agents or particles during the optimization process. ISM manages the exchange of crucial information, including the best solutions discovered, fitness evaluations, and directional updates, which are essential to drive research progress.

Over the past decade, substantial progress has been achieved in developing metaheuristic algorithms that integrate innovative strategies to boost optimization performance. For instance, the Improved Light Spectrum Optimizer (ILSO) integrates novel update systems to improve exploration and exploitation capabilities, achieving superior parameter estimation for photovoltaic models [35]. Similarly, Improved Binary Quadratic Interpolation Optimization (BIQIO) incorporates crossover and swap operators to improve search space exploration for 0-1 knapsack problems, demonstrating higher accuracy and computational efficiency compared to classical methods [**?**]. Furthermore, the Improved Spider Wasp Optimizer (ISWO) combines local search strategies with the Spider Wasp Optimizer, improving parameter estimation in double-diode models and excelling in both performance and convergence speed against established methods [37].

These advancements align closely with the objectives of the Improved Information-Sharing Mechanism (I2SM), which dynamically adapts metaheuristic processes in real time through metric-driven strategies, such as parameter tuning and hybridization. Using adaptive methodologies, I2SM improves both the efficiency and effectiveness of metaheuristic algorithms. To provide a structured understanding, the ISM is analyzed in more detail in this section through its three fundamental elements: **Information Structure**, **Information Carrier**, and **Information State**.

A. Information Structure

The Information Structure element of the ISM mechanism addresses the organization, classification, and sources of information in metaheuristic algorithms. In the classical PSO algorithm, the information structure defines how particle data - such as position and velocity - are organized into personal and global best information points, guiding the swarm towards optimal solutions.

Table I provides a detailed summary of the information structure strategies reviewed, highlighting their mechanisms and respective contributions to the improvement of metaheuristic algorithms.

TABLE I
SUMMARY OF INFORMATION STRUCTURES IN METAHEURISTIC ALGORITHMS

Article	Structure
[23], [41]	Neighborhood and Direction Information
[24], [42], [12]	Individual Information
[25], [21], [26], [43]	Mutual Information
[40]	Task-Based Information
[44]	Neighborhood Information
[27]	Population and Neighborhood Information
[45]	Surrogate Information
[30], [29], [28], [39]	Neighborhood Information
[31], [22]	Population Information
[20], [46]	Population/Swarm Information
[47]	Information From Dynamical Environments
[32]	Neighborhood/Individual Information

B. Information Support

Information support includes the tools and frameworks essential for storing, managing, sharing, and disseminating information within metaheuristic optimization algorithms. This includes databases, information management systems, communication protocols, and any other mechanism that facilitates the management and transmission of information within a system.

In the literature, two main categories have been identified: **clustering and neighborhood-based approaches**, and **memory-based ISMs**. Table II provides a detailed summary of the information support classes reviewed.

TABLE II
INFORMATION SUPPORT CLASSES

Information Support Class	Papers
Clustering and neighborhood-based	[44], [40], [43], [27], [22], [28], [39], [41], [32]
Memory and storage-based	[25], [12], [46]

Clustering and neighborhood-based approaches rely on structural techniques to efficiently partition and organize the search space. Clustering methods guide individuals towards optimal solutions within their designated groups, improving search efficiency and convergence speed. For example, some approaches dynamically form groups where leaders share information on global best solutions (gbest) while receiving updates on personal best solutions (pbest) from their members. The neighborhood-based approach also includes the use of ring topology, where the best solution in each neighborhood is stored at each iteration to impact the next search space solution [28]. Memory- and storage-based ISMs extend the optimization potential by using dedicated storage systems to improve adaptability and robustness.

C. State of Information

The state of the information represents the quality, completeness, and impact of the information shared within ISMs, which significantly influences the efficiency and adaptability of metaheuristic algorithms. The high quality of the information indicates that the data are accurate and complete, enabling optimal decisions to be made and parameters to be adjusted. Conversely, a low information state signals incomplete or noisy data, requiring refinement or additional processes to improve algorithm performance.

Several studies focus on methods for dynamically controlling and exploiting the information state. Evolutionary state estimation (ESE) [42], for example, evaluates the average distances between particles to adaptively adjust the PSO parameters, ensuring a balance between exploration and exploitation. Metrics such as mean absolute velocity (vAVG(t)) [20] are used to measure the energy of particle motion, allowing real-time optimization of algorithmic behavior. Stagnation detection measurements are particularly useful for identifying when particles become static, thus avoiding premature convergence and maintaining swarm search efficiency.

Despite significant advancements in ISM mechanisms, several limitations remain. Many methods struggle to balance exploration and exploitation, often relying on static information structures or pre-defined neighborhood and clustering frameworks, which can reduce adaptability in dynamic environments. In addition, memory and storage-based mechanisms face scalability issues, particularly for large-scale or high-dimensional problems. These challenges highlight the need for more flexible and adaptive solutions.

The proposed I2SM mechanism addresses these limitations by dynamically adjusting information sharing based on evolving conditions. By integrating adaptive triggers, hybridization techniques, and parallelism, I2SM improves the efficiency, robustness, and scalability of metaheuristic algorithms, offering a more versatile approach to overcome the constraints of traditional ISM methods.

III. The Improved Information Sharing Mechanism (I2SM)

As discussed earlier, various approaches have been developed to enhance metaheuristic algorithms by leveraging information sharing through various mechanisms. The proposed **Improved Information Sharing Mechanism (I2SM)** is notable for focusing specifically on the critical analysis of information that drives optimization. By evaluating these data and dynamically adjusting the performance of the metaheuristics, I2SM aims to significantly improve the optimization results. The mechanism is composed of three core components, as depicted in Figure 1.

In this study, I2SM is integrated with the selected metaheuristic algorithm, allowing the parallel execution of two interdependent processes during each iteration: the *metaheuristic optimization process* and the *data collection and analysis process*. This parallelization is achieved through multithreading.

During each iteration, the optimization process starts with the metaheuristic algorithm that generates candidate solutions. During the data collection and analysis process, key metrics such as the improvement rate, the iterations of stagnation, and the convergence rate were used. These metrics are continuously monitored and analyzed in an evaluation phase powered by a **multi-agent system (MAS)**.

The evaluation phase is critical for dynamically controlling the performance of the metaheuristic. It comprises three key activities:

- **Data Collection:** Gathering relevant performance metrics from the ongoing optimization process.
- **Data Analysis:** Assessing the metrics to determine the state of the algorithm.
- **Decision-Making:** Activating specific triggers based on the results of the evaluation to optimize the behavior of the algorithm.

These triggers are designed to dynamically adjust the search process, ensuring a balance between diversification (exploration of new solution spaces) and intensification (exploitation of promising regions). The operational logic of I2SM, including its triggers and tasks, is detailed in the following sections and visually summarized in Figure 2.

In the study, the I2SM mechanism was designed specifically to address the challenges of multidimensional optimization problems. Meanwhile, optimization problems in real-world applications are inherently complex and often involve a diverse spectrum of challenges that Zhanas [49] referred to as 5-M challenges: large-scale optimization, dynamic optimization, multimodal optimization, multiobjective optimization, and constrained and expensive optimization problems.

In fact, we recognize that dynamic environments, characterized by varying objective functions and varying constraints, introduce additional levels of complexity. Currently, the I2SM framework lacks adaptive mechanisms to detect and respond to these environmental changes. This limitation presents a compelling direction for future research, where enhancing the triggering mechanism could allow I2SM to seamlessly adapt to ever-changing research spaces.

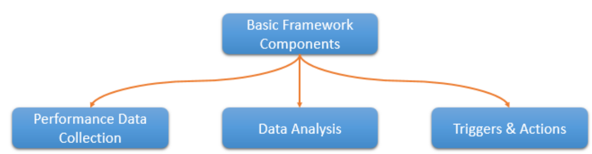

Fig. 1. I2SM Data Analysis Components.

A. Performance Data Collection

At each iteration, comprehensive information is gathered on various aspects of the metaheuristic's performance. This includes metrics such as:

- **Improvement Rate KPI:** This metric tracks how quickly the algorithm finds better solutions over time. A higher improvement rate means that the algorithm is efficiently discovering better solutions.
- **Convergence Rate KPI:** The convergence rate evaluates how steadily the algorithm approaches an optimal or near-optimal solution. A consistent convergence rate indicates stability and reliability.
- **Stagnation KPI:** This metric identifies iterations with little to no improvement in solution quality, indicating potential problems such as premature convergence or lack of exploration.

B. Data Analysis: Triggers & Actions

The collected KPIs provide quantitative insights into the metaheuristic's overall performance, allowing us to assess its effectiveness throughout the optimization process. By systematically collecting and analyzing these indicators against predefined thresholds, we can track the evolution of the

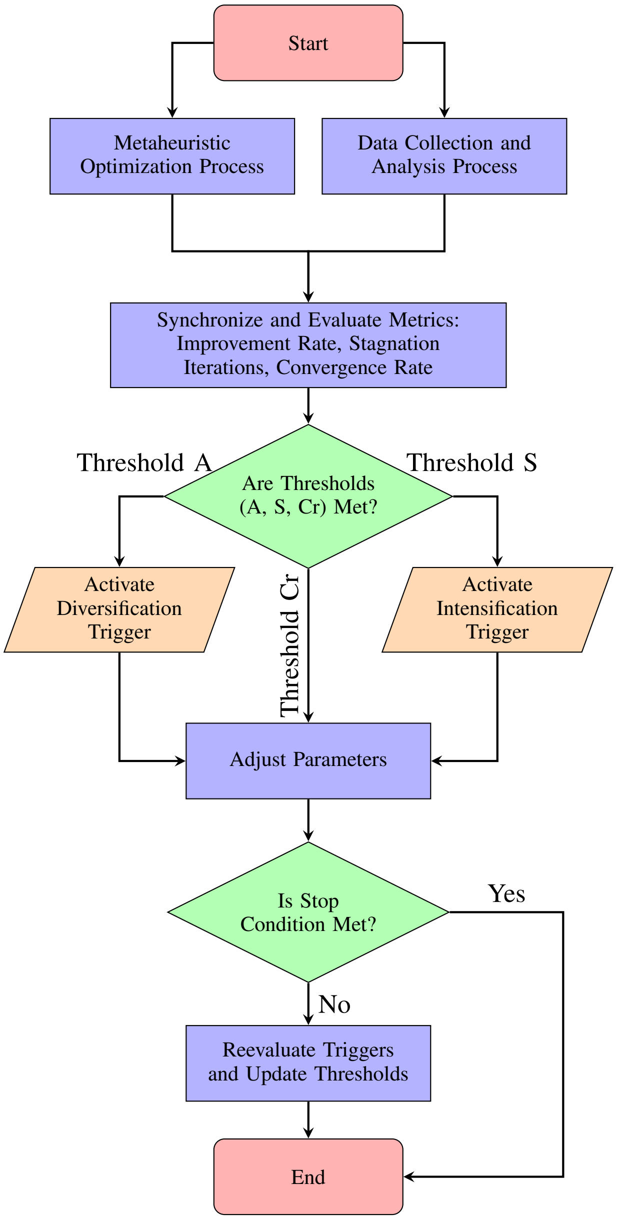

Fig. 2. I2SM Flowchart

metaheuristic across iterations. This approach enables us to make appropriate adjustments and activate specific triggers for necessary actions. A multi-agent system [50] is considered Information Support, with each action managed by a dedicated agent responsible for a specific KPI and its associated trigger, ensuring effective adjustments based on KPI evaluations.

- **Fine-tuning Parameters:** Adjustments to algorithm parameters are triggered when the convergence rate falls below a specific threshold value, which is set as a control parameter. If the convergence rate is slow, it indicates that the algorithm's parameters (mutation rate, crossover rate, population size) need adjustment. Fine-tuning these parameters can improve the results and allow the algorithm to adapt to changing conditions, improving search effectiveness.
- **Incorporating Hybrid Approaches:** Adjustments to the algorithm are triggered when the performance analysis indicates slow convergence or stagnation, based on predefined threshold values for the improvement rate and the stagnation KPIs. Subsequently, hybridization is employed by combining the metaheuristic with local or global search techniques, or by applying problem-specific heuristics.

Threshold-based triggers ensure that the metaheuristic algorithm remains effective and adaptable, leading to improved performance and more robust solutions.

C. Pseudo-code

1) Initialize Metaheuristic Parameters (e.g., population size, maximum number of iterations max_iter, threads, c1, c2, thresholds.)
2) **For each iteration** $i = 1$ to max_iter
   a) Run the Metaheuristic (PSO)
   b) Check for stopping condition
   c) If the stopping condition is met, exit the loop
   d) Collect performance metrics:
      - Stagnation iteration number ($stagnation\_iter$)
      - Improvement rate ($improvement\_rate$)
      - Convergence rate (cv_rate)
   e) If $stagnation\_iter$ exceeds the threshold 'S', activate the Diversification Trigger.
   f) If $improvement\_rate$ is less than the threshold 'A', activate the Intensification Trigger.
   g) If cv_rate is low, adjust the Metaheuristic parameters
   h) Continue to the next iteration

IV. Case Study: I2SM-PSO

To demonstrate the effectiveness of I2SM, PSO is selected as a case study. This evaluation focuses on the Information State, quantified through metrics such as Stagnation (S) and Improvement Rate (A). These metrics enable the dynamic activation of triggers by agents, initiating tailored enhancement actions.

For example, when stagnation is detected, the system employs hybridization with a random walk to boost diversification and explore new regions of the search space. Conversely, during cases of weak improvement, hybridization with Hill Climbing is applied to intensify the search and refine solutions.

A. Particle Swarm Optimization

PSO is a population-based metaheuristic. Unlike evolutionary algorithms that use genetic operators, such as crossover and mutation, standard PSO [51] emulates the behavior of swarms in nature, such as insects, flocking animals, birds in group flight, and schooling fish, which collaboratively search for food. Each member of the swarm adapts its search patterns by learning from its own experience and that of other members. These phenomena have been studied, and mathematical models have been constructed.

The classical PSO algorithm operates on the following two equations:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 \cdot r_1 \cdot (pbest_{ij} - x_{ij}(t)) + c_2 \cdot r_2 \cdot (gbest_j - x_{ij}(t)) \quad (1)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2)$$

where $v_{ij}(t+1)$ is the updated velocity of the particle i in dimension j in iteration $t+1$; c_1 is the cognitive learning factor, which controls the influence of the particle's personal best position ($pbest$) on its velocity update; r_1 is a random number in the range [0, 1], used to scale the influence of $pbest$;$pbest_{ij}$ is the best position achieved by the particle i in dimension j during its search history; $x_{ij}(t)$is the current position of the particle i in dimension j in iteration t; c_2 is the social learning factor, which determines the influence of the best global solution of the swarm ($gbest$) on the update of the velocity of the particle; r_2 is a random number in the range [0, 1], used to scale the influence of $gbest$; and $gbest_j$:] is the best global position found by the entire swarm in dimension j.

B. I2SM-PSO

As discussed in the previous section, the I2SM mechanism employs a multi-agent system where each agent monitors a specific metric. Based on collected KPIs and predefined thresholds, triggers are executed on dedicated threads to accelerate execution. For example, if the PSO encounters stagnation, a trigger is activated to hybridize the process with a random walk, diversifying the solutions, and enabling a global search for better options. This process loops until stagnation is resolved, and the output of the agent is fed back into the PSO to continue the optimization.

Similarly, in cases involving improvement rates or a combination of stagnation and improvement, specific triggers are executed. For improvement rates, a hill climbing mechanism is used to intensify the local search and exploit promising regions. In scenarios combining stagnation and low improvement rates, a dual trigger approach balances diversification and exploitation. These mechanisms collectively improve the state of the search, both locally and globally, ensuring adaptive and efficient optimization across dynamic and complex landscapes.

C. Test Functions

To evaluate the performance of I2SM when combined with PSO, five benchmark functions [52], [53] are selected for testing and comparison with the results obtained by the classical PSO. These functions represent a variety of optimization challenges, including unimodal functions, complex multimodal functions with numerous local optima, and multimodal functions with fewer local optima. The details of these benchmark functions are presented in Table 3. For further details, refer to [52], [53].

- Ackley Function:

$$f(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 \quad (3)$$

- Styblinski-Tang Function:

$$f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n}\left(x_i^4 - 16x_i^2 + 5x_i\right) \quad (4)$$

- Rastrigin Function:

$$f(\mathbf{x}) = \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right) \quad (5)$$

- Rosenbrock Function:

$$f(\mathbf{x}) = \sum_{i=1}^{n-1}\left[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right] \quad (6)$$

- Sphere Function:

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 \quad (7)$$

D. Parameter Settings for PSO algorithm and I2SM

For both I2SM-PSO and PSO, the social and cognitive coefficients are set as $c_1 = 1.25$ and $c_2 = 2.25$, respectively. The stagnation threshold is $S = 5$ iterations and the improvement rate threshold is $A = 5\%$. Moreover, the population size of 50, and 30 is the maximum iterations number. To reduce the influence of stochastic error, 50 independent trials are run on each test function, and the mean results are recorded for comparison. The parameter settings for PSO were selected based on the work of [54].

E. Experimental Results

This section presents a comprehensive comparative analysis of the I2SM-PSO and classical PSO algorithms, evaluated in a range of benchmark test functions in varying dimensional settings Table III.

The results, summarized in Table IV, highlight significant improvements achieved by I2SM-PSO in terms of the accuracy of the solution and robustness. These differences underscore the enhanced ability of I2SM-PSO to effectively balance exploration and exploitation, even in complex or high-dimensional multimodal optimization problems.

Performance on Unimodal Functions:

- **Sphere Function (f_5):** I2SM-PSO consistently achieved significantly lower fitness values compared to PSO in all dimensions, demonstrating superior convergence properties. For instance, at 30 dimensions, I2SM-PSO achieved a fitness value of 5.97×10^{-3} in an execution time of 0.93 seconds, compared to PSO's 1.21×10^{3} in 0.10 seconds. Although PSO was executed faster, the marked fitness improvement by I2SM-PSO validates its robustness in handling

TABLE III

BENCHMARK FUNCTIONS AND THEIR DIMENSIONS, BOUNDS, FUNCTION VALUES, AND MINIMUM FITNESS VALUES.

Function	Dimension	Bounds	f_i	Min Fitness Value
Ackley	n	$[-5, 5]$	f_1	0
Styblinski-Tang	n	$[-5, 5]$	f_2	$-39.16599 \times n$
Rastrigin	n	$[-5.12, 5.12]$	f_3	0
Rosenbrock	n	$[-30, 30]$	f_4	0
Sphere	n	$[-100, 100]$	f_5	0

TABLE IV

COMPARISON OF I2SM-PSO AND PSO ON VARIOUS TEST FUNCTIONS.

Function	Dimensions	Best Fitness I2SM-PSO	Best Fitness PSO	I2SM-PSO Execution Time (s)	PSO Execution Time (s)
f_1	5	7.35×10^{-6}	7.35×10^{-6}	0.13	0.04
f_1	10	8.53×10^{-2}	7.28×10^{-2}	0.24	0.12
f_1	15	6.00×10^{-2}	1.34×10^{-1}	0.42	0.14
f_1	20	1.17×10^{0}	1.84×10^{0}	0.61	0.33
f_1	25	3.44×10^{0}	3.76×10^{0}	0.83	0.25
f_1	30	2.32×10^{0}	2.94×10^{0}	0.93	0.19
f_2	5	-1.82×10^{2}	-1.82×10^{2}	0.12	0.02
f_2	10	-3.63×10^{2}	-3.63×10^{2}	0.25	0.02
f_2	15	-5.17×10^{2}	-5.58×10^{2}	0.35	0.05
f_2	20	-6.70×10^{2}	-6.65×10^{2}	0.74	0.12
f_2	25	-8.38×10^{2}	-8.38×10^{2}	0.93	0.20
f_2	30	-1.01×10^{3}	-9.67×10^{2}	0.93	0.12
f_3	5	2.00×10^{0}	1.99×10^{0}	0.12	0.02
f_3	10	1.11×10^{1}	2.28×10^{1}	0.23	0.04
f_3	15	3.67×10^{1}	4.34×10^{1}	0.33	0.06
f_3	20	5.75×10^{1}	6.07×10^{1}	0.58	0.08
f_3	25	6.03×10^{1}	1.09×10^{2}	0.76	0.11
f_3	30	7.89×10^{1}	1.10×10^{2}	0.93	0.12
f_4	5	1.34×10^{0}	1.04×10^{1}	0.11	0.03
f_4	10	5.11×10^{0}	8.48×10^{1}	0.31	0.05
f_4	15	3.05×10^{2}	8.18×10^{2}	0.54	0.07
f_4	20	1.55×10^{1}	2.41×10^{3}	1.84	0.10
f_4	25	2.15×10^{1}	1.69×10^{5}	2.54	0.11
f_4	30	2.44×10^{1}	2.77×10^{5}	2.54	0.13
f_5	5	1.21×10^{-3}	1.75×10^{-8}	0.13	0.02
f_5	10	1.66×10^{-3}	9.66×10^{-3}	0.24	0.04
f_5	15	3.21×10^{-3}	1.32×10^{1}	0.42	0.05
f_5	20	3.89×10^{-3}	2.33×10^{1}	0.61	0.07
f_5	25	4.53×10^{-3}	4.76×10^{2}	0.83	0.08
f_5	30	5.97×10^{-3}	1.21×10^{3}	0.93	0.10

simpler unimodal landscapes, justifying the additional computation time.

- **Rosenbrock Function (f_4):** I2SM-PSO outperformed PSO in all dimensions, especially in higher-dimensional spaces. At 30 dimensions, I2SM-PSO obtained a fitness value of 2.44×10^1 in 2.54 seconds, compared to PSO's 2.77×10^5 in 0.13 seconds. Although the PSO was executed faster, the ability of I2SM-PSO to navigate the narrow and curved valleys of the Rosenbrock function more efficiently highlights its superiority in optimizing unimodal functions with challenging landscapes.

Performance on Multimodal Functions:

- **Ackley Function (f_1):** I2SM-PSO demonstrated significant improvements over PSO, particularly in higher dimensions. For example, at 30 dimensions, the I2SM-PSO achieved a fitness value of 2.32×10^0 in 0.93 seconds, outperforming PSO's 2.94×10^0 in 0.19 seconds. Although the PSO was executed faster, the ability of I2SM-PSO to explore complex landscapes and avoid local optima justifies its slightly longer execution time.
- **Styblinski-Tang Function (f_2):** the results were mixed, with PSO outperforming I2SM-PSO in certain dimensions. For example, at 15 dimensions, PSO achieved -5.58×10^2 in 0.05 seconds compared to I2SM-PSO's -5.17×10^2 in 0.35 seconds. However, in 30 dimensions, I2SM-PSO demonstrated superior performance (-1.01×10^3 in 0.93 seconds compared to PSO -9.67×10^2 in 0.12 seconds), indicating its effectiveness in higher-dimensional optimization tasks.
- **Rastrigin Function (f_3):** I2SM-PSO showed notable advantages over PSO, particularly in dimensions greater than 10. For instance, at 30 dimensions, I2SM-PSO achieved a fitness value of 7.89×10^1 in 0.93 seconds compared to PSO's 1.10×10^2 in 0.12 seconds. Although PSO was executed faster, the strength of I2SM-PSO in navigating the highly oscillatory landscape of the function and avoiding local optima makes it more effective for complex multimodal problems.

The execution time increases as the dimensionality of the problem scales, for both I2SM-PSO and PSO. I2SM-PSO generally takes longer than PSO due to its enhanced mechanisms, such as multithreading and resource coordination, which are designed to improve performance.

For smaller dimensions, execution times remain low and manageable. However, as the problem size exceeds 20 dimensions, we observe a noticeable increase in execution time, particularly for I2SM-PSO. This suggests that resource demand (e.g. memory, CPU usage) scales with dimensionality, highlighting the need for careful resource management and optimization when addressing larger problems.

This analysis emphasizes the trade-off between execution time and solution quality. Although PSO was generally executed faster, I2SM-PSO consistently demonstrated better fitness values, particularly in higher dimensions, validating its enhanced optimization capabilities.

V. Conclusion

The proposed I2SM mechanism introduces a robust framework to enhance metaheuristic optimization by integrating advanced information management and evaluation strategies. Using a Multi-Agent System (MAS), I2SM collects critical metrics such as improvement rates and stagnation levels, activating adaptive triggers to implement targeted actions. The parallelism of the model facilitates simultaneous optimization and performance analysis, enabling significant performance improvements while effectively balancing execution time. Empirical results indicate that PSO combined with I2SM achieves up to 90% better fitness outcomes than conventional PSO, particularly in unimodal functions such as Sphere and Rosenbrock, as well as complex multimodal functions such as Rastrigin and Ackley, especially in higher dimensions. Although PSO demonstrates competitiveness on Styblinski-Tang, the ability of I2SM to dynamically balance exploration and exploitation enhances both solution accuracy and convergence speed. Importantly, although I2SM-PSO incurs slightly higher execution times compared to standard PSO, this trade-off is justified by the substantial gains in solution quality and robustness.

References

[1] Y. Alaouchiche, Y. Ouazene, and F. Yalaoui, "Multi-objective optimization of energy-efficient buffer allocation problem for non-homogeneous unreliable production lines," IEEE Access, vol. PP, pp. 1-1, 2021.

[2] Y. Alaouchiche, Y. Ouazene, and F. Yalaoui, "Energetic and Economic Performance Evaluation of Production Systems: Perspective Analysis," IFAC-PapersOnLine, vol. 53, pp. 11150-11155, 2020.

[3] A. Berrichi, F. Yalaoui, L. Amodeo, and M. Mezghiche, "Bi-Objective Ant Colony Optimization approach to optimize production and maintenance scheduling," Computers Operations Research, vol. 37, no. 9, pp. 1584-1596, 2010.

[4] M. Zemzami, A. Makhloufi, N. El hami, A. ELHami, M. Itmi, and N. Hmina, "Applying a New Parallelized Version of PSO Algorithm for Electrical Power Transmission," IOP Conference Series: Materials Science and Engineering, vol. 205, pp. 012032, 2017.

[5] G. Xu, Z.-H. Wu, and M.-Z. Jiang, "Premature convergence of standard particle swarm optimisation algorithm based on Markov chain analysis," Int. J. Wire. Mob. Comput., vol. 9, no. 4, pp. 377–382, Jan. 2015.

[6] M. Zemzami, N. Elhami, M. Itmi, and N. Hmina, "A new parallel approach for the exploitation of the search space based on PSO algorithm," in 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), 2016, pp. 104-110.

[7] L. Velasco, H. Guerrero, and A. Hospitaler, "A Literature Review and Critical Analysis of Metaheuristics Recently Developed," Archives of Computational Methods in Engineering, vol. 31, July 2023.

[8] R. Hakima, Z. Maria, H. Nabil, et al., "A comparative study of several metaheuristic algorithms for optimization problems," in 2022 8th International Conference on Optimization and Applications (ICOA), 2022, pp. 1-9.

[9] R. Atha, A. Rajan, and S. Mallick, "An enhanced Equilibrium Optimizer for solving complex optimization problems," Information Sciences, vol. 660, pp. 120077, 2024.

[10] Q. Fan, S. Zhao, M. Shang, Z. Wei, and X. Huang, "An improved genetic salp swarm algorithm with population partitioning for numerical optimization," Information Sciences, vol. 679, pp. 120895, 2024.

[11] X. Yang, H. Li, and Y. Huang, "An adaptive dynamic multi-swarm particle swarm optimization with stagnation detection and spatial exclusion for solving continuous optimization problems," Engineering Applications of Artificial Intelligence, vol. 123, pp. 106215, 2023.

[12] G.-G. Wang and Y. Tan, "Improving metaheuristic algorithms with information feedback models," IEEE transactions on cybernetics, vol. 49, no. 2, pp. 542-555, 2017.

[13] J.-S. Pan, Z. Zhang, S. C. Chu, Z.-J. Lee, and W. Li, "Application of Diversity-Maintaining Adaptive Rafflesia Optimization Algorithm to Engineering Optimisation Problems," Symmetry, vol. 15, pp. 2077, 2023.

[14] Y. Zhang, X. Liu, F. Bao, J. Chi, C. Zhang, and P. Liu, "Particle swarm optimization with adaptive learning strategy," Knowledge-Based Systems, vol. 196, pp. 105789, 2020.

[15] Y. Ren, Q. Chen, Y. Y. Lau, M. A. Dulebenets, B. Li, and M. Li, "A multi-objective fuzzy programming model for port tugboat scheduling based on the Stackelberg game," Scientific Reports, vol. 14, no. 1, pp. 25057, Oct. 2024.

[16] C. Gong, Y. Nan, M. Pang, H. Ishibuchi, and Q. Zhang, "Performance of NSGA-III on Multi-objective Combinatorial Optimization Problems Heavily Depends on Its Implementations," in Proceedings of the Genetic and Evolutionary Computation Conference, 2024, pp. 511–519.

[17] N. Van Thieu, N. H. Nguyen, and A. A. Heidari, "Feature selection using metaheuristics made easy: Open source MAFESE library in Python," Future Generation Computer Systems, vol. 160, pp. 340-358, 2024.

[18] C. Blum, J. Puchinger, G. R. Raidl, A. Roli, et al., "A brief survey on hybrid metaheuristics," Proceedings of BIOMA, pp. 3-18, 2010.

[19] V. Tomar, M. Bansal, and P. Singh, "Metaheuristic Algorithms for Optimization: A Brief Review," RAiSE-2023, 2024.

[20] G. Xu, "An adaptive parameter tuning of particle swarm optimization algorithm," Applied Mathematics and Computation, vol. 219, no. 9, pp. 4560-4569, 2013.

[21] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," IEEE transactions on evolutionary computation, vol. 10, no. 3, pp. 281-295, 2006.

[22] H. Handa, N. Baba, O. Katai, T. Sawaragi, and T. Horiuchi, "Genetic algorithm involving coevolution mechanism to search for effective genetic information," in Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC'97), 1997, pp. 709-714.

[23] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02, vol. 2, 2002, pp. 1671-1676.

[24] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," IEEE transactions on evolutionary computation, vol. 8, no. 3, pp. 204-210, 2004.

[25] Y. Li, Z.-H. Zhan, S. Lin, J. Zhang, and X. Luo, "Competitive and cooperative particle swarm optimization with information sharing mechanism for global optimization problems," Information Sciences, vol. 293, pp. 370-382, 2015.

[26] J. C. Bansal, K. Deep, K. Veeramachaneni, and L. Osadciw, "Information sharing strategy among particles in particle swarm optimization using laplacian operator," in 2009 IEEE swarm intelligence symposium, 2009, pp. 30-36.

[27] T. Ray and K.-M. Liew, "A swarm with an effective information sharing mechanism for unconstrained and constrained single objective

optimisation problems," in Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, 2001, pp. 75-80.
[28] L. Â. da Silveira, J. L. Soncco-Álvarez, and M. Ayala-Rincón, "Parallel genetic algorithms with sharing of individuals for sorting unsigned genomes by reversals," in 2017 IEEE Congress on Evolutionary Computation (CEC), 2017, pp. 741-748.
[29] D. E. Goldberg, J. Richardson, et al., "Genetic algorithms with sharing for multimodal function optimization," in Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms, vol. 4149, 1987, pp. 414-425.
[30] Y. Gonzalez-Fernandez and S. Chen, "Leaders and followers—a new metaheuristic to avoid the bias of accumulated information," in 2015 IEEE congress on evolutionary computation (CEC), 2015, pp. 776-783.
[31] M. Shi and H. Wu, "Pareto cooperative coevolutionary genetic algorithm using reference sharing collaboration," in Proceedings of the 11th Annual conference on Genetic and evolutionary computation, 2009, pp. 867-874.
[32] Y. Tamura, T. Sakiyama, and I. Arizono, "Ant Colony Optimization Using Common Social Information and Self-Memory," Complexity, vol. 2021, no. 1, pp. 6610670, 2021.
[33] L. Cheng, J. Cao, W. Wang, and L. Cheng, "Multiple Learning Strategies and a Modified Dynamic Multiswarm Particle Swarm Optimization Algorithm with a Master Slave Structure," Applied Sciences, vol. 14, pp. 7035, 2024.
[34] K. Rajwar, K. Deep, and S. Das, "An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges," Artificial Intelligence Review, April 2023.
[35] S. Saber and S. Salem, "Improved Light Spectrum Optimizer for Parameter Identification of Triple-Diode PV Model," Sustainable Machine Intelligence Journal, vol. 4, Sept. 2023.
[36] S. Salem, "Improved Binary Quadratic Interpolation Optimization for 0-1 Knapsack Problems," Sustainable Machine Intelligence Journal, vol. 4, Sept. 2023.
[37] S. Saber and S. Salem, "High-Performance Technique for Estimating Unknown Parameters of Photovoltaic Cells and Modules Using Improved Spider Wasp Optimizer," Sustainable Machine Intelligence Journal, vol. 5, Oct. 2023.
[38] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Proceedings of ICNN'95-international conference on neural networks, vol. 4, 1995, pp. 1942-1948.
[39] W. Deng, J. Xu, and H. Zhao, "An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem," IEEE access, vol. 7, pp. 20281-20292, 2019.
[40] M. Roshanzamir, M. A. Balafar, and S. N. Razavi, "Empowering particle swarm optimization algorithm using multi agents' capability: A holonic approach," Knowledge-Based Systems, vol. 136, pp. 58-74, 2017.
[41] B. Bullnheimer, R. F. Hartl, and C. Strauss, "A New Rank Based Version of the Ant System-A Computational Study," Central European Journal of Operations Research, vol. 7, no. 1, 1999.
[42] M. Isiet and M. Gadala, "Self-adapting control parameters in particle swarm optimization," Applied Soft Computing, vol. 83, pp. 105653, 2019.
[43] L. Fang and Y. Ge, "Enhanced PSO Based on Multi-Agent System," in 2008 International Symposium on Computational Intelligence and Design, vol. 1, 2008, pp. 290-293.
[44] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 39, no. 6, pp. 1362-1381, 2009.
[45] J. Kennedy, "Stereotyping: Improving particle swarm performance with cluster analysis," in Proceedings of the 2000 congress on evolutionary computation. CEC00, vol. 2, 2000, pp. 1507-1512.
[46] J. Kennedy, "Bare bones particle swarms," in Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03, 2003, pp. 80-87.
[47] W. Hu and G. G. Yen, "Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system," IEEE Transactions on Evolutionary Computation, vol. 19, no. 1, pp. 1-18, 2013.
[48] W. Gao, F. T. Chan, L. Huang, and S. Liu, "Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood," Information Sciences, vol. 316, pp. 180-200, 2015.
[49] Z.-H. Zhan, L. Shi, K. C. Tan, and J. Zhang, "A survey on evolutionary computation for complex continuous optimization," Artificial Intelligence Review, vol. 55, no. 1, pp. 59-110, 2022.
[50] M. Zemzami, N. Elhami, M. Itmi, and N. Hmina, "A Proposal Of A Multi-agent Modeling Based On A Parallel Model Of The PSO Algorithm," in 2020 IEEE 6th International Conference on Optimization and Applications (ICOA), 2020, pp. 1-6.
[51] M. Zemzami, N. El Hami, M. Itmi, and N. Hmina, "A comparative study of three new parallel models based on the PSO algorithm," International Journal for Simulation and Multidisciplinary Design Optimization, vol. 11, pp. 5, 2020.
[52] M. Jamil and X.-S. Yang, "A literature survey of benchmark functions for global optimisation problems," International Journal of Mathematical Modelling and Numerical Optimisation, vol. 4, no. 2, pp. 150-194, 2013.
[53] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," IEEE Transactions on Evolutionary computation, vol. 3, no. 2, pp. 82-102, 1999.
[54] M. Zemzami, "Variations sur PSO: approches parallèles, jeux de voisinages et applications," Ph.D. dissertation, Normandie Université; Ecole nationale des sciences appliquées (Kénitra, Maroc), 2019.