VOL. 23, NO. 3, 2024, 30-37
elektrika.utm.my
ISSN 0128-4428

ELEKTRIKA
Journal of Electrical Engineering

Innovative Synergistic Genetic Algorithm and Particle Swarm Optimization for Scheduling Optimization in Mobile Robots

Mingyu Wu[1,2], **Chun Leong Lim**[1], **Eileen Lee Ming Su**[1], **Che Fai Yeong**[1], **Bowen Dong**[1] and **William Holderbaum**[3]

[1]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Johor Bahru, Johor, Malaysia
[2]Jiaxing Key Laboratory of Industrial intelligence and Digital Twin, Jiaxing Vocational and Technical College, Jiaxing, Zhejiang, China
[3]Department of Engineering, Manchester Metropolitan University, Manchester, UK

*Corresponding author: mingyu@graduate.utm.my

Abstract: Autonomous Mobile Robots (AMRs) are crucial in modern manufacturing for automating material handling and transportation. However, optimizing their task scheduling is challenging due to conflicting objectives like minimizing makespan and reducing energy consumption. Traditional algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) often yield suboptimal results. This study proposes an innovative Synergistic GA-PSO algorithm that combines the exploratory capabilities of GA with the fast convergence of PSO. Experiments conducted in MATLAB demonstrate that the Synergistic GA-PSO algorithm consistently outperforms GA, PSO, and ACO, especially in complex environments, by enhancing scheduling accuracy, reducing idle intervals, and lowering energy consumption.

Keywords: Autonomous Mobile Robots, Task Scheduling, Genetic Algorithm, Particle Swarm Optimization, Hybrid Algorithm

Article History: received 21 May 2024; accepted 22 September 2024; published 30 December 2024.

1. INTRODUCTION

Autonomous Mobile Robots (AMRs) have become increasingly indispensable in modern manufacturing processes and are critical in automating material handling and transportation functions. Despite their significant utility, optimizing their task scheduling poses a complex challenge, fraught with conflicting objectives such as minimizing makespan and reducing energy consumption. Traditional scheduling algorithms often yield suboptimal solutions, resulting in operational inefficiencies like increased downtime [1], [2].

Given this backdrop, this study aims to conduct a rigorous comparative evaluation of three AI-based optimization algorithms—Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO)—to enhance AMR scheduling in manufacturing environments. These algorithms are selected due to their prominence and proven effectiveness in various optimization problems. The evaluation focuses on their performance in terms of scheduling accuracy, idle interval reduction, and energy consumption.

The comparative analysis revealed that while each algorithm has its strengths, they also have notable limitations. GA is robust and adaptable but often struggles with scalability in larger systems [3], [4], [5]. PSO offers fast convergence and simplicity but may get trapped in local optima [6], [7]. ACO is effective for discrete path planning but can be computationally intensive and slow to converge in real-world scenarios [8], [9].

To address these limitations, this study proposes an Innovative Synergistic Genetic Algorithm and Particle Swarm Optimization (GA-PSO) approach. This synergistic method aims to leverage the exploratory capabilities of GA and the fast convergence properties of PSO, creating a more robust and efficient scheduling solution. The synergistic GA-PSO algorithm is designed to enhance scheduling accuracy, reduce idle intervals, and lower energy consumption by combining the strengths of both algorithms.

The primary objectives of this study are twofold: first, to rigorously compare the performance of GA, PSO, and ACO in AMR scheduling, and second, to develop and validate the proposed Synergistic GA-PSO algorithm. All computational modeling and algorithm deployments will be executed within the MATLAB environment.

The organization of this paper is as follows: Following this introduction, Section 2 reviews the existing literature and theoretical foundations of GA, PSO, and ACO. Section 3 details the methodology and experimental setup used for the comparative evaluation. Section 4 presents the innovative Synergistic GA-PSO algorithm and discusses its development. Section 5 compares the performance of the Synergistic algorithm with the traditional methods. Section 6 concludes the paper with key insights and recommendations for future research.

2. ALGORITHMIC REVIEW

Optimizing the scheduling of Autonomous Mobile Robots (AMRs) is a critical research area at the intersection of manufacturing engineering and artificial intelligence. Traditional scheduling algorithms often fall short in the increasingly complex environment of modern manufacturing, necessitating the use of more advanced optimization techniques. While there is a broad array of algorithms being explored for AMR scheduling, this study focuses on a comparative analysis of three prominent bio-inspired algorithms: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO). Although Reinforcement Learning and Neural Networks also contribute to this field, the emphasis here is on the unique characteristics and applicability of GA, PSO, and ACO in optimizing AMR operations[10], [11], [12].

2.1 Genetic Algorithm (GA)

Genetic Algorithms (GAs) are inspired by the principles of natural selection and genetics [3]. They are used primarily in optimization and search problems, where they evolve a population of potential solutions over successive generations to find an optimal or near-optimal solution [4], [5], [6]. GAs are known for their robustness and adaptability, making them suitable for various optimization challenges, including AMR scheduling. However, their performance can be limited by scalability issues, particularly in large, complex systems, as they require substantial computational resources and longer convergence times.

2.2 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) was introduced by Kennedy and Eberhart in 1995 and is modeled after the social behaviors of birds flocking or fish schooling [7], [8]. Unlike gradient-based optimization methods, PSO relies on the collective behavior of a population of particles to explore the solution space, guided by the objective function [9]. PSO is valued for its simplicity and rapid convergence, which make it suitable for real-time applications such as AMR scheduling. However, PSO can sometimes become trapped in local optima, which limits its effectiveness in more complex scenarios.

2.3 Ant Colony Optimization (ACO)

Ant Colony Optimization (ACO) is inspired by the foraging behavior of ants and was developed in the early 1990s [10], [11]. It is particularly effective for solving combinatorial optimization problems by simulating how ants deposit pheromones to find the shortest paths to food sources [15], [16]. Despite its strengths, ACO can be computationally intensive and slower to converge compared to other algorithms, which can hinder its performance in dynamic and large-scale manufacturing environments [17].

3. DESIGN OF THE SYNERGISTIC GA-PSO ALGORITHM

The comparative analysis of GA, PSO, and ACO reveals that while each algorithm has distinct advantages, they also possess notable limitations. GA is robust and adaptable but struggle with scalability. PSO is efficient and fast but prone to getting trapped in local optima. ACO excels in combinatorial optimization but is computationally intensive and slow to converge. These limitations highlight the need for an approach that can combine the strengths of multiple algorithms while mitigating their weaknesses.

To address these challenges, this study proposes a Combined Genetic Algorithm and Particle Swarm Optimization (GA-PSO) Approach. By integrating the exploratory capabilities of GA with the fast convergence properties of PSO, the GA-PSO algorithm aims to provide a more robust and efficient solution for AMR scheduling. The combined method allows for an initial broad search of the solution space using GA, followed by fine-tuning through PSO, thus leveraging the individual strengths of each algorithm to overcome their respective limitations.

This synergistic approach is rooted in past research which shows that synergistic of algorithms can lead to improved performance. For instance, combining GA with local search techniques has been shown to enhance the efficiency of finding global optima in complex search spaces [18]. Similarly, integrating PSO with other optimization methods has demonstrated improvements in solution quality and convergence rates. Therefore, the proposed GA-PSO approach is designed to synthesize the evolutionary capabilities of GA with the social interaction dynamics of PSO, aiming for a more effective optimization strategy.

Recent advancements in hybrid optimization methods further support the potential of combining GA and PSO. In particular, hybrid algorithms have been shown to outperform their standalone counterparts in complex, multi-objective optimization problems, where multiple conflicting criteria must be balanced simultaneously. Studies have demonstrated that hybrid GA-PSO algorithms are especially effective in avoiding premature convergence by maintaining a balance between the global search capabilities of GA and the local refinement strengths of PSO. Moreover, hybrid methods provide a more adaptive search process, adjusting their exploration and exploitation mechanisms based on real-time feedback from the solution space. This adaptability is particularly beneficial in dynamic environments like AMR scheduling, where task priorities and constraints can change frequently. By utilizing the complementary strengths of both GA and PSO, the proposed approach can achieve superior performance in terms of convergence speed, solution accuracy, and robustness to varying problem scales and complexities.

The flowchart in Figure 1 illustrates the detailed steps of the synergistic GA-PSO algorithm, from parameter definition and population initialization to fitness evaluation, selection, crossover, mutation, velocity update, and position update. This iterative process ensures continuous improvement and convergence towards the optimal solution.

Table 1. Algorithms Comparison [18], [19], [19]

Criteria	Genetic Algorithm (GA)	Particle Swarm Optimization (PSO)	Ant Colony Optimization (ACO)	Combined GA-PSO
Computational Complexity (Big O Notation)	*O*(*N*2) suitable for medium-scale AMR systems	*O*(*N*) suitable for real-time AMR systems	O(N2logN) suited for discrete path planning	O(N) suitable for scalable AMR systems
Parameter Sensitivity	High (fine-tuning needed for time-windows)	Moderate (few parameters ease AMR-specific adjustments)	High (requires tuning for AMR constraints)	Moderate (leverages strengths of GA and PSO)
Memory Requirement	Moderate (due to population, may slow down real-time tasks)	Low (better for real-time AMR systems)	Moderate (pheromone matrix may consume memory)	Moderate (optimized memory usage)
Convergence Rate (Iterations)	500-2000 (longer for complex AMR scenarios)	100-1000 (fast, suitable for real-time scheduling)	200-5000 (slower for real-world AMR tasks)	200-1000 (balanced convergence)
Optimality Gap for AMR Scheduling (%)	3-7%	1-4%	2-6%	1-3% (improved optimality)
Parallelizability	High (individual routes can be computed in parallel)	Moderate (parallel but interdependent updates)	Low to moderate (bottleneck at global pheromone update)	High (combines parallel capabilities)
Robustness in Dynamic Environments (%)	75-85%	80-95%	70-85%	85-95% (enhanced robustness)
Flexibility to AMR-specific Constraints	Moderate (requires encoding for time-windows, load capacity etc.)	High (easier to implement AMR-specific rules)	Moderate (problem-specific rules needed for tasks like deadlock avoidance)	High (adaptable to various constraints)
Implementation Complexity for AMR	Moderate-High (due to encoding/decoding tasks)	Low-Moderate (simpler models often suffice)	Moderate-High (due to pheromone updating and complex decision-making)	Moderate (balances complexity and efficienc

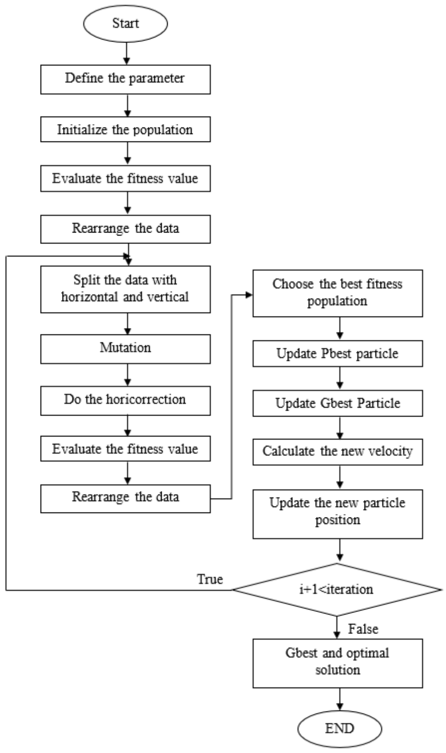

Figure 1. Synergistic GA-PSO algorithm flow chart

The following table 1 summarizes the characteristics and performance metrics of GA, PSO, ACO, and the proposed GA-PSO algorithm in the context of AMR scheduling:

4. EXPERIMENTAL SETUP AND RESULTS

4.1 Experimental Setup

The experimental setup aimed to evaluate the performance of the proposed Synergistic GA-PSO algorithm compared to traditional GA, PSO, and ACO algorithms. The experiments used three types of logical maps—simple, moderate, and complex—to represent different levels of task scheduling difficulty. Figures 2 illustrate these logical maps.

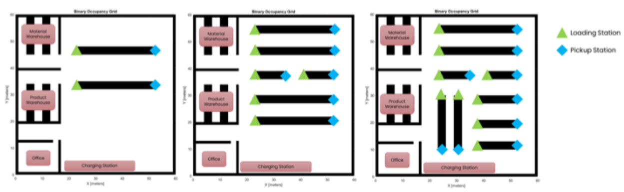

Figure 2. Logical maps

The simulation process was implemented using MATLAB. It begins with inputting data and setting up the environment, followed by executing the optimization scheduling algorithms. This algorithms updates the best solution based on fitness values over 100 iterations, resulting in path planning visualized through robot operation simulations and Gantt charts.

Three types of simulation maps were used: simple, moderate, and complex. The number of robots (2, 5, and

10) and tasks varied, and three types of fitness selection (time, distance, and time x distance) were considered. Two types of path generation were employed: sequential task allocation and simultaneous task allocation. Path planning utilized the A* algorithm due to its simplicity and efficiency [20]. Simulations were conducted on a computer with an i7-10875H CPU, RTX2070 GPU, and 32 GB RAM.

In the experimental setup, the Genetic Algorithm (GA) was configured with a population size of 50, a crossover rate of 0.8, and a mutation rate of 0.05, using tournament selection. For Particle Swarm Optimization (PSO), we used a population of 50 particles, an inertia weight starting at 0.9 and decreasing linearly to 0.4, and cognitive and social coefficients both set at 1.5. The Ant Colony Optimization (ACO) algorithm was configured with 50 ants, an evaporation rate of 0.5, pheromone influence (α) set at 1.0, and heuristic influence (β) set at 2.0. Finally, the Synergistic GA-PSO algorithm combined GA and PSO, using GA's population size of 50, a crossover rate of 0.8, and a mutation rate of 0.05, followed by a PSO phase with 50 particles, using the same inertia, cognitive, and social parameters as in standalone PSO. All algorithms were run for a maximum of 1000 iterations or until convergence.

4.2 Results and Analysis

The results are presented in terms of average distance, travel time, processing time, and maximum travel distance for each robot. The performance of the Synergistic GA-PSO algorithm was compared with traditional GA, PSO, and ACO algorithms across different fitness functions and path generation types. Lower fitness values indicate a better performance of the algorithm.

Path Generation Type 1 (Tasks Follow Sequence)

- **Fitness Using Time:** The results for different environments using time as the fitness function are shown in Figure 3 to 5. The Synergistic GA-PSO algorithm performed similarly to GA with fewer robots but improved significantly as the number of robots increased.

Figure 3. Results by Fitness Using Time in simple map

Figure 4. Results by Fitness Using Time in moderate map

Figure 5. Results by Fitness Using Time in complex map

- **Fitness Using Distance:** Figures 6 to 8 present the results using distance as the fitness function. The Synergistic GA-PSO algorithm consistently outperformed GA with more robots, demonstrating its effectiveness in minimizing travel distance.

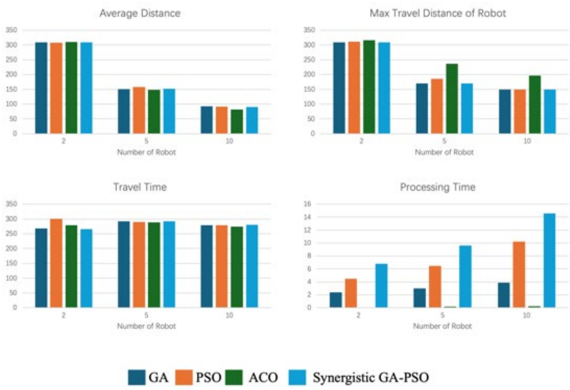

Figure 6. Results by Fitness Using Distance in simple map

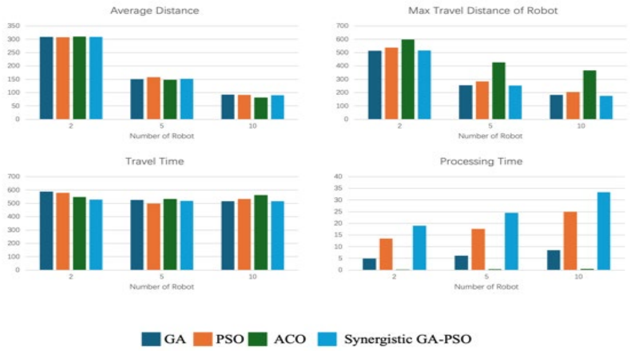

Figure 7. Results by Fitness Using Distance in moderate map

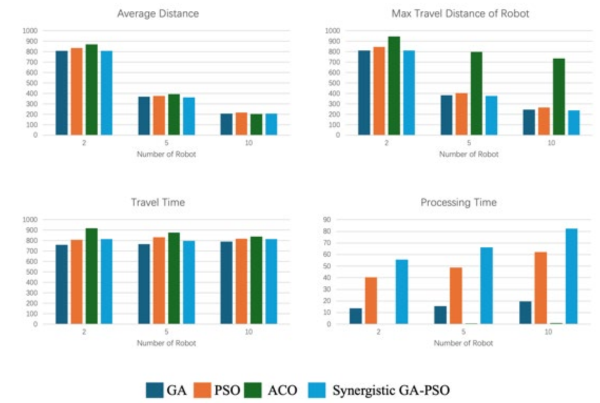

Figure 8. Results by Fitness Using Distance in complex map

- **Fitness Using Time x Distance:** Figures 9 to 11 illustrate the results using a combination of time and distance as the fitness function. The Synergistic GA-PSO algorithm showed superior performance, particularly in complex environments, effectively balancing both travel time and distance.

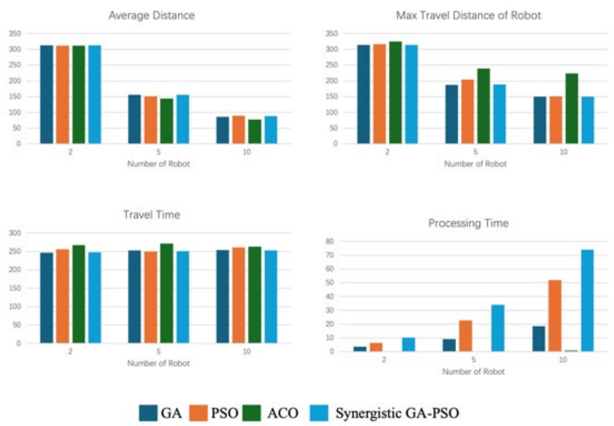

Figure 9. Results by Distance*Time in simple map

Figure 10. Results by Distance*Time in moderate map

Figure 11. Results by Distance*Time in complex map

Path Generation Type 2 (All Tasks Together)

- **Fitness Using Time x Distance:** For the scenario where all tasks are assigned simultaneously, the results in Figure 12 to 14 demonstrate that the Synergistic GA-PSO algorithm performs the best in complex environments, optimizing for both time and distance.

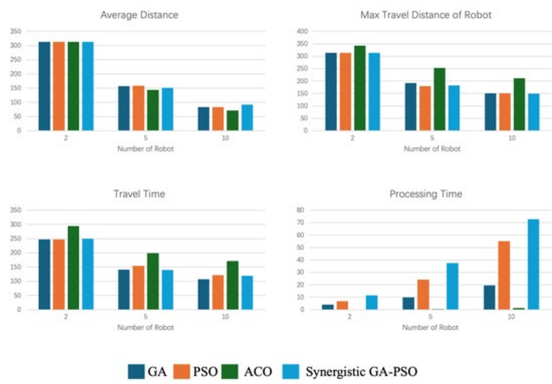

Figure 12. Results by Distance*Time in simple map All task

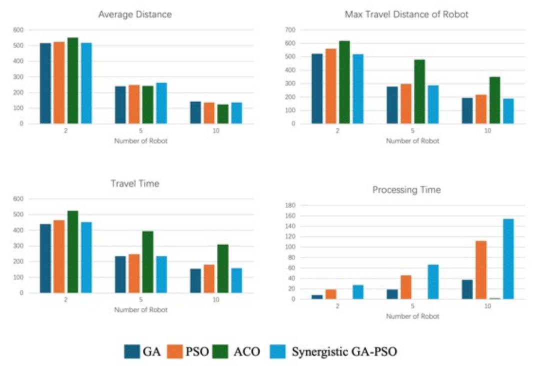

Figure 13. Results by Distance*Time in moderate map All task

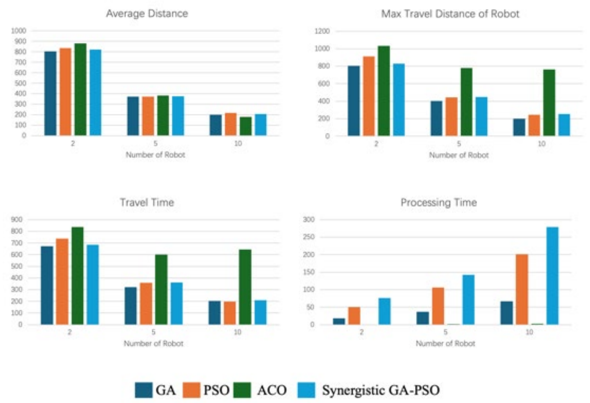

Figure 14. Results by Distance*Time in complex map All task.

4.3 Summary of Results

Based on the experiments, the Synergistic GA-PSO algorithm consistently outperformed traditional GA, PSO, and ACO algorithms, especially in complex environments and with an increasing number of robots. It effectively balances trade-offs between time and distance, making it a robust solution for optimizing AMR scheduling. While the ACO algorithm has the fastest processing time, its performance in terms of travel distance and time is inferior.

The analysis concludes that the Synergistic GA-PSO algorithm provides enhanced scheduling accuracy, reduced idle intervals, and lower energy consumption, addressing the limitations of using GA or PSO independently and resulting in improved overall performance.

5. DISCUSSION AND CONCLUSION

5.1 Discussion

The experimental results provide valuable insights into the performance of the Synergistic Genetic Algorithm and Particle Swarm Optimization (GA-PSO) for scheduling optimization in mobile robots, compared to traditional GA, PSO, and ACO algorithms.

Ant Colony Optimization (ACO): Despite its fast processing time, ACO showed the weakest performance in terms of fitness value and task completion time. ACO's limitations are due to its need for a higher number of iterations to achieve optimal performance, as it does not utilize a population-based approach like GA or PSO. Consequently, ACO's effectiveness diminishes in scenarios with limited iterations.

Genetic Algorithm (GA): GA demonstrated strong performance, particularly in simpler environments and with fewer robots. Its robustness and adaptability allowed it to achieve low fitness values. However, GA's scalability issues became apparent in more complex scenarios with a larger number of robots and tasks, where it required more computational resources and longer convergence times.

Particle Swarm Optimization (PSO): PSO offered rapid convergence and was effective in real-time applications. However, it was prone to getting trapped in local optima, which limited its performance in more complex environments. The algorithm's simplicity and fast convergence were advantageous, but its overall effectiveness was compromised when compared to the synergistic approach.

Synergistic GA-PSO: The Synergistic GA-PSO algorithm excelled in balancing the trade-offs between exploration and exploitation. It consistently achieved the lowest fitness values and effectively minimized both travel time and travel distance. The Gantt charts for the complex environment showed that while GA-PSO had slightly longer travel times in some cases, it significantly reduced the maximum travel distance, leading to more efficient and balanced task scheduling. The primary drawback of GA-PSO was its higher processing time, which highlights the need for further optimization.

Fitness Selection Analysis: The experiments revealed that distance-based fitness minimized the maximum travel distance but did not control travel time effectively, leading to inefficiencies. Time-based fitness reduced travel time but failed to control the maximum travel distance, potentially shortening the robots' operational lifespan. The combined time x distance fitness function provided the most balanced and effective optimization, reducing both travel time and distance and thereby enhancing overall efficiency and robot longevity.

Overall, the Synergistic GA-PSO algorithm emerged as the most robust and efficient solution, effectively addressing the limitations of GA and PSO when used independently. It demonstrated superior performance in diverse and complex scheduling scenarios, making it a highly effective approach for optimizing mobile robot scheduling in manufacturing environments.

5.2 Conclusion

This study aimed to develop and validate an innovative Synergistic Genetic Algorithm and Particle Swarm Optimization (GA-PSO) for scheduling optimization in mobile robots. The primary objectives were to design an AI-based scheduling simulation system, compare the performance of GA, PSO, and ACO algorithms, and develop a synergistic GA-PSO algorithm.

The AI-based scheduling simulation system was successfully developed using MATLAB, incorporating GA, PSO, and ACO algorithms. The comparative analysis revealed that while each algorithm has distinct strengths, the Synergistic GA-PSO algorithm provided the most robust and efficient scheduling solution. It achieved the lowest fitness values, minimized travel time and distance, and handled a large number of robots and tasks effectively.

The study highlighted the significant benefits of the Synergistic GA-PSO algorithm, including improved task allocation, reduced idle times, and extended robot life. These advantages directly impact the productivity and profitability of manufacturing operations, making the Synergistic GA-PSO algorithm a valuable tool for scheduling optimization in mobile robots.

5.3 Future Work

Future research should focus on several enhancements to further improve the Synergistic GA-PSO algorithm:

1. Time-based Simulation System: Developing a dynamic time-based simulation system to visualize real-time optimization of robot scheduling and task management.

2. Speed Limit Integration: Incorporating speed limits in the simulation to better represent realistic travel times and improve scheduling accuracy.

3. Total Distance Inclusion in Fitness Function: Including total travel distance in the fitness function to enhance energy efficiency and reduce overall travel distance.

4. Processing Time Optimization: Streamlining the Synergistic GA-PSO algorithm to reduce processing time, making it more practical for real-time applications and larger-scale deployments.

These enhancements would refine the scheduling optimization process, making the Synergistic GA-PSO algorithm even more efficient and effective for practical applications in diverse manufacturing environments.

REFERENCES

[1] M. Wu, C. F. Yeong, E. L. M. Su, W. Holderbaum, and C. Yang, "A review on energy efficiency in autonomous mobile robots," *Robot. Intell. Autom.*, vol. 43, no. 6, pp. 648–668, Jan. 2023, doi: 10.1108/RIA-05-2023-0060.

[2] M. Wu, S. E. Chua, E. L. M. Su, and C. F. Yeong, "Investigation of Effects of Path Planning Algorithms on Mobile Robot's Performance," in *2024 IEEE International Conference on Industrial Technology (ICIT)*, Mar. 2024, pp. 1–6. doi: 10.1109/ICIT58233.2024.10540892.

[3] X. Ma and X. Zhou, "Research on the Scheduling of Mobile Robots in Mixed-Model Assembly Lines Considering Workstation Satisfaction and Energy Consumption," *IEEE Access*, vol. 10, pp. 84738–84753, 2022, doi: 10.1109/ACCESS.2022.3197791.

[4] O. V. Darintsev and A. B. Migranov, "The Use of Genetic Algorithms for Distribution of Tasks in Groups of Mobile Robots with Minimization of Energy Consumption," in *2019 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon)*, Oct. 2019, pp. 1–6. doi: 10.1109/FarEastCon.2019.8934927.

[5] V. Sathiya, M. Chinnadurai, S. Ramabalan, and A. Appolloni, "Mobile robots and evolutionary optimization algorithms for green supply chain management in a used-car resale company," *Environ. Dev. Sustain.*, vol. 23, no. 6, pp. 9110–9138, Jun. 2021, doi: 10.1007/s10668-020-01015-2.

[6] Y. Shi, T. Boudouh, and O. Grunder, "A hybrid genetic algorithm for a home health care routing problem with time window and fuzzy demand," *Expert Syst. Appl.*, vol. 72, pp. 160–176, Apr. 2017, doi: 10.1016/j.eswa.2016.12.013.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Nov. 1995, pp. 1942–1948 vol.4. doi: 10.1109/ICNN.1995.488968.

[8] M. S. Innocente and J. Sienz, "Particle Swarm Optimization: Development of a General-Purpose Optimizer," Jan. 24, 2021, *arXiv*: arXiv:2101.09835. doi: 10.48550/arXiv.2101.09835.

[9] X. Tao *et al.*, "Self-Adaptive two roles hybrid learning strategies-based particle swarm optimization," *Inf. Sci.*, vol. 578, pp. 457–481, Nov. 2021, doi: 10.1016/j.ins.2021.07.008.

[10] M. F. M. Sabri, K. A. Danapalasingam, and M. F. Rahmat, "A review on hybrid electric vehicles architecture and energy management strategies," *Renew. Sustain. Energy Rev.*, vol. 53, pp. 1433–1442, Jan. 2016, doi: 10.1016/j.rser.2015.09.036.

[11] F. H. Ajeil, I. K. Ibraheem, M. A. Sahib, and A. J. Humaidi, "Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm," *Appl. Soft Comput.*, vol. 89, p. 106076, Apr. 2020, doi: 10.1016/j.asoc.2020.106076.

[12] D. Zhang, Y. Yin, R. Luo, and S. Zou, "Hybrid IACO-A*-PSO optimization algorithm for solving multiobjective path planning problem of mobile robot in radioactive environment," *Prog. Nucl. Energy*, vol. 159, p. 104651, May 2023, doi: 10.1016/j.pnucene.2023.104651.

[13] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.*, vol. 1, no. 4, pp. 28–39, Nov. 2006, doi: 10.1109/MCI.2006.329691.

[14] C. Blum, "Ant colony optimization: Introduction and recent trends," *Phys. Life Rev.*, vol. 2, no. 4, pp. 353–373, Dec. 2005, doi: 10.1016/j.plrev.2005.10.001.

[15] H. Li and H. Zhang, "Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints," *Autom. Constr.*, vol. 35, pp. 431–438, Nov. 2013, doi: 10.1016/j.autcon.2013.05.030.

[16] H. R. Gaikwad, P. K. Mahind, and S. U. Mane, "Comparative Analysis of GPGPU based ACO and PSO Algorithm for Employee Scheduling Problems," Mar. 23, 2022, *arXiv*: arXiv:2203.12239. doi: 10.48550/arXiv.2203.12239.

[17] S. U. Mane, P. S. Lokare, and H. R. Gaikwad, "Overview and Applications of GPGPU Based Parallel Ant Colony Optimization," Mar. 22, 2022, *arXiv*: arXiv:2203.11487. doi: 10.48550/arXiv.2203.11487.

[18] Z. Wu, "A Comparative Study of solving Traveling Salesman Problem with Genetic Algorithm, Ant Colony Algorithm, and Particle Swarm Optimization," in *Proceedings of the 2020 2nd International Conference on Robotics Systems and*

Vehicle Technology, in RSVT ’20. New York, NY, USA: Association for Computing Machinery, Jun. 2021, pp. 95–99. doi: 10.1145/3450292.3450308.

[19] “Optimization of Fairhurst-Cook Model for 2-D Wing Cracks Using Ant Colony Optimization (ACO), Particle Swarm Intelligence (PSO), and Genetic Algorithm (GA).” Accessed: Nov. 20, 2023. [Online].

[20] Cong Liu, X. Xu, X. Li, Z. Pan, K. Hu, and Y. Shu, “Path Planning for an Omnidirectional Mobile Robot Based on Modified A* Algorithm with Energy Model,” in *2021 IEEE International Conference on Progress in Informatics and Computing (PIC)*, Shanghai, China: IEEE, Dec. 2021, pp. 462–468. doi: 10.1109/PIC53636.2021.9687067.