STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 202x, pp 0–11.
Published online in International Academic Press (www.IAPress.org)

A Computational Time Analysis of Dhouib-Matrix-SPP versus Particle Swarm Optimization Metaheuristics for Grid-based Path Planning

Souhail Dhouib [1,*], Noura Béji [1], Dorra Kallel [1], Saima Dhouib [2]

[1] *University of Sfax, Higher Institute of Industrial Management of Sfax, Sfax, Tunisia*
[2] *University of Sousse, EPI-Polytechnique, Tunisia*

Abstract Actually, path planning is one of the most fundamental aspects of mobile robots study. The objective is to determine the shortest feasible trajectory from a starting point to a goal location while avoiding obstacles. Particle Swarm Optimization (PSO) has been widely applied to this problem. However, it is often complex, requiring careful parameter tuning and extensive computational resources, in spite of that it suffers from high computational complexity, sensitivity to parameter tuning, and local optima stagnation. To overcome these limitations, the new Dhouib-Matrix-SPP (DM-SPP) method is proposed, which is rapid, straightforward, and does not require parameter adjustment. Simulation experiments on four case studies (I-shaped, U-shaped, T-shaped and Randomly shaped) demonstrate that DM-SPP consistently outperforms the ranking Particle Swarm Optimization (rPSO) metaheuristic and the artificial potential field-based Particle Swarm Optimization (apfrPSO) metaheuristic in terms of computational time: DM-SPP is 66 time rapider than the rPSO metaheuristic and 31 time rapider than the apfrPSO metaheuristic. These findings indicate that DM-SPP is a powerful and scalable approach for mobile robot path planning.

Keywords Mobile robot path planning, Particle Swarm Optimization, Dhouib-Matrix-SPP, Metaheuristics, Optimization, Operations Research.

DOI: 10.19139/soic-2310-5070-3259

1. Introduction

With the rapid advancement of robotics in recent years, autonomous mobile robots have become increasingly common and path planning is a critical component of mobile robot autonomy [1]. Path planning has been a significant area of study in the field of mobile robotics for a long time. The goal of path planning is to devise a route that is both safe and free of collisions, guiding from the initial point to the destination in an environment filled with obstacles [2]. The primary objective is to allow robots to execute operations independently while minimizing the necessity for human involvement and determine the optimal route from the starting point to the endpoint [3]. Nowadays, mobile robots are widely applied in a diversity of applications, such as, the space exploration [4], medical applications [5], navigation system for transportation [6], road cracks [7, 8, 9] , the industrial electric vehicles [10, 11], Neural Network [12], and other disciplines.

Our motivation for investigating autonomous robotic navigation is to close the gap between existing technology and the various needs of practical applications [13]. Considerable obstacles remain in the path of making meaningful advancements in robotics, necessitating further innovations to ensure dependable and effective functioning in intricate settings. Despite the rapidity of industrial robots in executing repetitive tasks within

*Correspondence to: Souhail Dhouib (Email: souhail.dhouib@gmail.com). University of Sfax, Higher Institute of Industrial Management of Sfax, Sfax, Tunisia.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

controlled settings, they encounter constraints when operating in unfamiliar environments or under unforeseen conditions [3, 14]. Their fixed programming restricts their flexibility to manage with dynamic changes within their environment; they are limited to operating flexibility and autonomy and, in most cases, work within predetermined sequences of actions in highly tuned environments [15]. Moreover, these robots are anticipated to demonstrate the adaptability required to operate in complex environments, make rapid decisions, and execute missions without human control [16]. This growing demand highlights the acknowledged advantages of autonomous systems in enhancing safety, productivity, and efficiency across a wide range of industries. Therefore, there is a clear shift toward creating robots equipped with enhanced sensing and decision-making capabilities to address the evolving requirements of contemporary societies [17].
Because of their capacity to generate near-optimal results within a short time, heuristic algorithms have become increasingly prevalent in mobile robot path planning [18]. Traditional Metaheuristic algorithms used for mobile robot path planning include methods such as Genetic Algorithm (GA) Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) several others [19]. Each approach is suited to different applications, and therefore has its own advantages and limitations. An improved version of the A method, designed to minimize the number of turning points in a nuclear radiation environment, has been developed [20]. In this regard, several enhanced PSO variants have been proposed by researchers in recent years. For instance, an adaptive PSO approach is introduced in [21], and it is applied to both single and multiple humanoid robots for mobile robot path planning. In addition, other algorithms are used to solve this problem, such as the Firefly metaheuristic and Pelican Optimization Algorithm (POA). In addition, recent studies highlight a growing interest in hybrid metaheuristics that combine the strengths of several algorithms (GA-PSO or ACO-PSO) in order to achieve faster convergence, smoother trajectories, and better obstacle avoidance in complex environments [22]. Various studies have also adopted heuristic techniques and employed them to tackle different aspects of path-planning methods. For example, the initial PSO model was inspired by observing and graphically simulating the coordinated movement of a flock of birds [23]. Genetic Algorithms (GA), along with their modified versions, are frequently employed to determine the shortest path for mobile robot navigation in various environments [24]. A hybrid Genetic Algorithm, utilizing the Continuous Bezier Optimization technique, is presented for the robot path planning problem in [25]. A Whale Optimization Algorithm (WOA) has been implemented in a static environment to meet the requirements of finding the shortest and smoothest path [23]. An efficient Q-Learning method is designed to generate shortest-path planning with obstacle avoidance for a mobile robot in [26]. The A* method is enhanced to determine the most efficient route for a mobile robot in a fixed environment[27]. An artificial potential field-based Particle Swarm algorithm (apfrPSO) was developed to define more obstacle-free paths for a mobile robot in a grid map in [28]. Additionally, an optimal technique called Dhouib-Matrix-SPP (DM-SPP) has been developed to solve the shortest path problem for any type of graph [29]. Consequently, DM-SPP has been extended to generate the trajectory of a mobile robot using only eight possible movement directions [30, 31]. Furthermore, DM-SPP has been refined to operate in four movement directions (referred to as DM-SPP-4) [32]. All DM-SPP approaches are tested in multiple case studies of varying complexity and compared with various artificial intelligence techniques recently proposed in the literature.

The paper primarily aims to prove the performance of the novel DM-SPP method. The primary advantage of DM-SPP is its computational speed for finding a feasible path in simple grid worlds. For that, DM-SPP is compared to two PSO methods in four case studies. These comparative experiments are designed to illustrate not only the computational efficiency of DM-SPP, but also its capability to generate shorter and more reliable trajectories. As producing the shortest possible path is an important consideration in mobile robotics, as it directly influences execution time, energy consumption, and the overall sustainability of the system, the evaluation focuses on both the quality of the solution and the speed of resolution. By examining its performance in various representative situations, the study aims to highlight the robustness, scalability, and practical relevance of the DM-SPP method for real-world route planning problems.

For the remainder of this paper, the content is organized as follows. Section 2 describes the principles of the Particle Swam Optimization metaheuristic (PSO). Section 3 presents the proposed Dhouib-Matrix-SPP method.

Section 4 describes experimentation, simulation results, and discussion. And finally a conclusion and perspectives are given.

2. The Particle Swam Optimization metaheuristic

PSO was introduced by Kennedy and Eberhart in 1995 as an evolutionary computing technique. It is inspired by social behaviors such as bird flocking and fish schooling, and operates with a population of candidate solutions (particles) [33, 34]. Its core mechanism was largely inspired by simulations of animal social behavior as presented in Figure 1. Before foraging, individuals either disperse or gather while searching for food, similar to birds. Before foraging, birds either disperse or gather while searching for food and select areas where they can access it. Nevertheless, they migrate from one location to another in pursuit of food. A bird with a strong sense of smell is always present, guiding the flock to the food source [35]. The velocity of each particle is updated at every iteration based on both its social interactions and individual behavior. Furthermore, PSO has been extended in recent years through hybrid approaches and adaptive variants aimed at improving convergence speed, avoiding local optima, and addressing multi-objective optimization problems, making it applicable to a wide range of complex real-world scenarios.

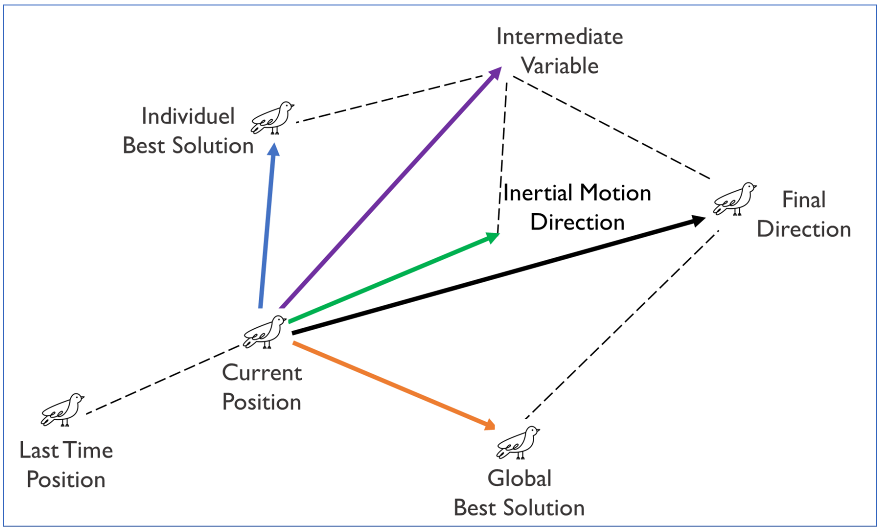

Figure 1. Particle Swarm Optimization (PSO) algorithm.

When exploring an n-dimensional hyperspace, the position of particle i denotes the location of the solution within the search space, as shown by $x_i = (xi1, xi2, xi3, ..., xin)$. The positional movement of particle i utilizes its velocity history, as in $v_i = (vi1, vi2, vi3, ..., vin)$. Each particle i keeps track of its best position, as indicated by $pbest_i = (pi1, pi2, pi3, ..., pin)$. The best position among all x_{pbest_i} in the group is identified as the global optimal position, x_{gbest_i}. The position and velocity of each particle i are updated using information from the selected global optimum and its own personal best, as shown in 1 and 2.

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

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

Where:
t : denotes the iteration number.
ω : represents the inertial weight.
c_1 and c_2 are the learning factors of the personal and global optimal particles respectively.
r_1 and r_2 are random numbers within the range [0,1].
In general, x_{pbest_i} represents the particle's best position and stores its memory. If the current position aligns with the recorded information, the position remains unchanged; otherwise, it is updated with the current position. x_{gbest_i} denotes the optimal solution obtained from the particle's neighborhood.

The primary steps of the PSO can be summarized as follows [36].

- Step 1: Initialization. Randomly assign the positions and velocities of particles within the n-dimensional problem space.
- Step 2: Particle Evaluation. Assess the fitness value of each particle within the n-dimensional optimization function.
- Step 3: Update Optimal Values. The fitness of each particle is compared with its personal best value, $pbest$. If it exceeds $pbest$, the particle's current position becomes its new $pbest$. Then, the particle's fitness is compared with the global best value, $gbest$. If the current value is better than $gbest$, the global best position is updated to the particle's current position.
- Step 4: Update Particle. Adjust the position and velocity of each particle according to the updates in 1 and 2.
- Step 5: Termination Condition. Repeat from Step 2 until the stopping criterion is met, typically based on achieving the desired fitness value or reaching the maximum number of iterations.

The advantages of the basic PSO are its easily adjustable parameters and straightforward implementation. However, its drawbacks include a tendency to get trapped in local minimum value and premature convergence. Moreover, the total number of iterations required to reach the global optimal solution is typically large, and the time complexity of the overall evaluation process is very high. To overcome these limitations, researchers have proposed various improvements such as hybridization with other metaheuristics, adaptive parameter tuning, and parallel implementations, which improve convergence speed, robustness, and applicability to large-scale and dynamic optimization problems.

3. The Dhouib-Matrix-SPP method

The novel DM-SPP is a new optimal method to generate the shortest path characterized by a time complexity of $O(n + m)$. The shortest path between a node and another specific node (Single Pair SPP) and between the source (or destination) and all other nodes can be calculated by DM-SPP. In addition, the All Pair Sorthest Path can be calculated by implementing DM-SPP in an iterative structure. DM-SPP is composed of four steps as illustrated in Figure 2.

The new DM-SPP method has a polyvalent and deterministic structure that requires no parameters, which is a significant revolution compared to other existing methods that require parameters. In mobile robotics, it is essential to generate the shortest possible path, as this has a direct impact on the robot's operational efficiency. A shorter path allows the robot to reach its destination faster while minimizing unnecessary movements and deviations from the trajectory. This not only helps improve task execution time, but also reduces mechanical wear and battery consumption. In addition, the performance of DM-SPP is confirmed by its speed, which allows the mobile robot to be faster, and its shorter resolution time with a more efficient solution. As a result, it reduces energy consumption, which improves sustainability performance. By producing optimized and shorter trajectories, this method helps reduce the robot's total energy consumption. This reduction in energy demand increases the robot's autonomy and improves its sustainability performance, an increasingly important criterion in modern robotic systems.

The Dhouib-Matrix Shortest Path Problem (DM-SPP) method introduced by [31] is an innovative polynomial-time algorithm designed to solve shortest path problems with great efficiency. Unlike traditional exact methods,

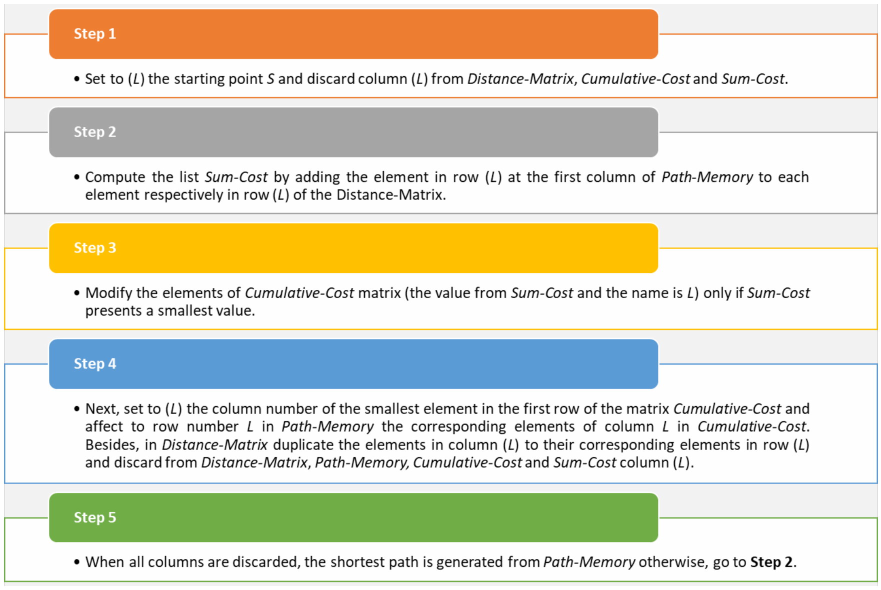

Figure 2. The four steps of the novel DM-SPP method.

which can require significant computational effort for large graphs, DM-SPP provides a simplified and structured procedure that significantly reduces computation time and energy consumption. This method is based on a column-row elimination mechanism that iteratively eliminates suboptimal paths while preserving the essential connectivity structure of the graph. Thanks to its lightweight operations and reduced memory requirements, DM-SPP can be easily implemented and adapted to real-time applications.

In order to present the efficiency of DM-SPP versus PSO, the 20x20 grid map (see Figure 3) is used to compare DM-SPP with two advanced variants of PSO (introduced in [28]): The ranking Particle Swarm Optimization (rPSO) and the artificial potential field-based Particle Swarm Optimization (apfrPSO).

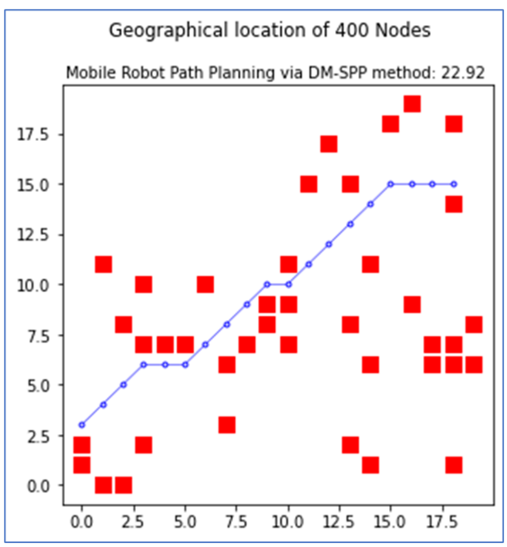

Figure 3. The 20x20 Grid map.

Figure 4 illustrates the effect of varying the population size on the computational time (CPU) in seconds. The x-axis represents the population size of PSO methods and Y-axis the required computational time. For each population, an average of 30 iterations is represented. From this figure it can be concluded that DM-SPO is instable method versus the clear stability of DM-SPP. However, the most important remark is the rapidity of DM-SPP: For the case study of 20x20 grid map, DM-SPP generates a solution after just 0.017 second (where as rPSO and apfrPSO require respectively an average computational time of 251.89 seconds and 120.07 second).).

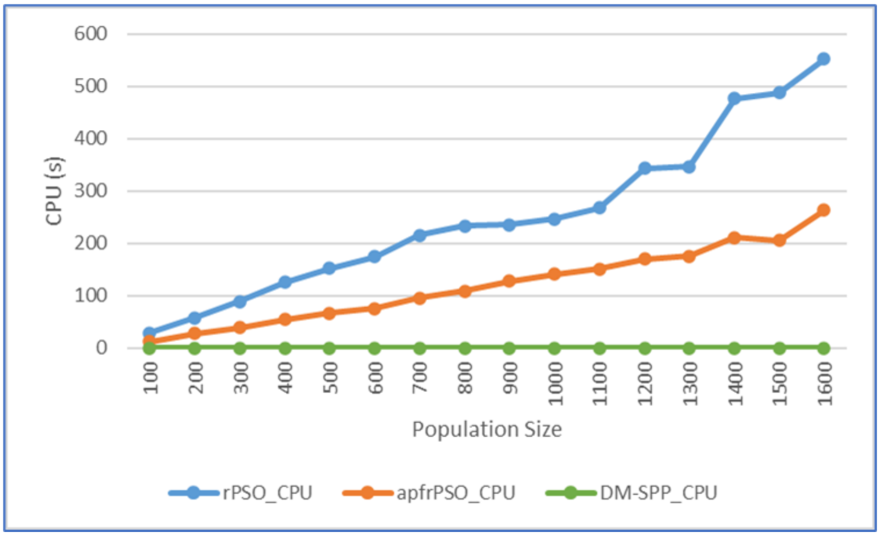

Figure 4. Stability of DM-SPP versus instability of PSO method on average running time.

Dm-SPP is a component of the general concept of Dhouib-Matrix (DM) where several other methods are developed such as: The DM-TSP1 method to unravel the Travelling Salesman Problem [37], the DM-AP1 and the DM-AP2 methods to optimize the Assignment Problem [38, 39] and the DM-TP1 to solve the Transportation Problem [40]. Moreover, three novel metaheuristics are developed: The iterated stochastic DM3 [41], the multi-start DM4 [42] and the local search FtN [43]. Also, to solve the Minimum Spanning Tree Problem the DM-MSTP method is designed in [44] and to unravel the all-pairs shortest path problem the DM-ALL-SPP technique is developed in [45].

4. Simulation results

This section presents a comparative analysis of the proposed DM-SPP method against two recent PSO-based metaheuristics from the literature, namely the ranking Particle Swarm Optimization (rPSO) and the artificial potential field-based Particle Swarm Optimization (apfrPSO) introduced in [28]. DM-SPP is implemented in Python and executed on a Dell laptop equipped with an Intel Core i7-1255U processor and 16 GB RAM.
In accordance with the evaluation protocol adopted in Sections 2 and 3, the assessment focuses on computational time, path length, and algorithmic effort (iteration count). The four benchmark environments— I-shaped, U-shaped, T-shaped, and randomly shaped—were selected due to their use in prior studies and their differing structural complexity. Across all cases, DM-SPP demonstrated highly favorable performance, reinforcing its deterministic and parameter-free advantages compared to population-based metaheuristics.

4.1. I-shaped example

This first example is known as I-Shaped taken from [28]. DM-SPP generates the solution illustrated in Figure 5 with just 0.25 seconds.

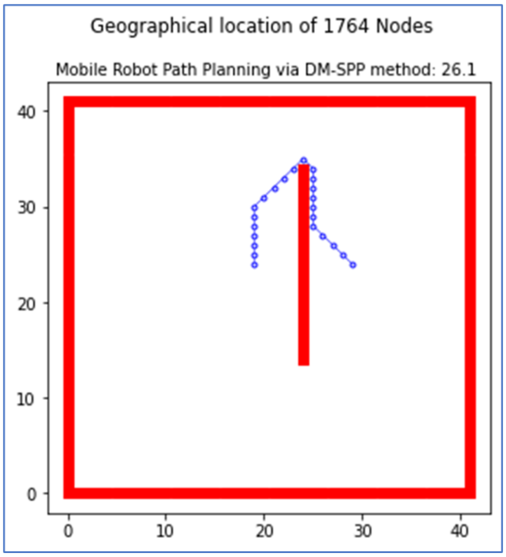

Figure 5. DM-SPP simulation result for I-shaped example.

Table 1 presents a comparative summary of performance indicators (represented by an average of thirty independent runs) for the three methods. DM-SPP achieves speed-up factors of 83× over rPSO and 42× over apfrPSO, confirming its ability to reach a feasible solution substantially faster than metaheuristic-based approaches.

Table 1. Comparing DM-SPP to PSO methods on I-shaped example.

Methods	Distance	Average CPU	Iterations	DM-SPP Improvement
rPSO	27.21	20.67	146	83
apfrPSO	24.38	10.44	36	42
DM-SPP	26.14	0.25	1	1

4.2. U-shaped example

The second experiment evaluates the U-shaped environment [28]. DM-SPP generates the solution in Figure 6 within 0.25 s.

As indicated in Table 2, DM-SPP is 58× faster than rPSO and 28× faster than apfrPSO. The results (represented by an average of thirty independent runs) highlight the stability of DM-SPP's computational performance regardless of obstacle geometry.

Table 2. Comparing DM-SPP to PSO methods on U-shaped example.

Methods	Distance	Average CPU	Iterations	DM-SPP Improvement
rPSO	24.97	14.39	24	58
apfrPSO	21.56	6.89	30	28
DM-SPP	37.51	0.25	1	1

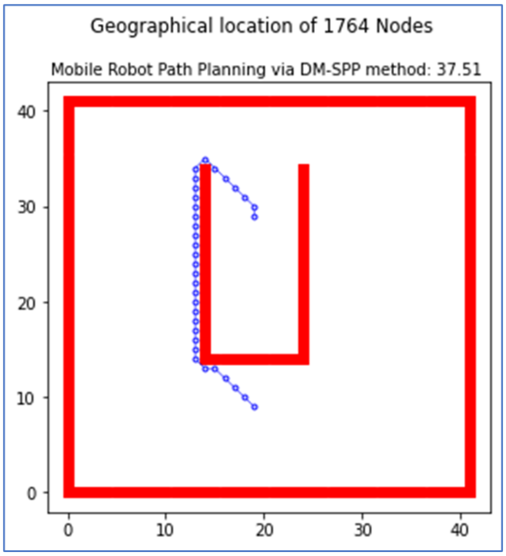

Figure 6. DM-SPP simulation result for U-shaped example.

4.3. T-shaped example

The third experiment examines the T-shaped environment taken from [28]. DM-SPP computes the trajectory depicted in Figure 7 in 0.29 s.

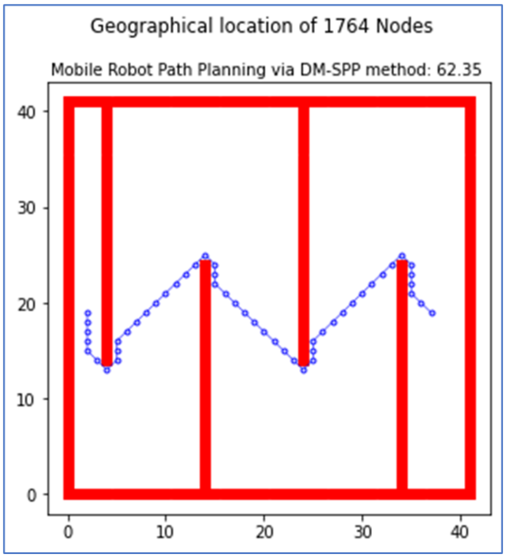

Figure 7. DM-SPP simulation result for T-shaped example.

Table 3 summarize the results generated by DM-SPP and the derivative versions of PSO. DM-SPP is 52 time rapider than rPSO and 26 time rapider than apfrPSO.

4.4. Randomly shaped example

The final experiment uses the randomly generated environment from [28]. DM-SPP determines the trajectory shown in Figure 8 in 0.24 s.

Table 3. Comparing DM-SPP to PSO methods on T-shaped example.

Methods	Distance	Average CPU	Iterations	DM-SPP Improvement
rPSO	22.14	15.14	71	52
apfrPSO	19.31	7.49	2	26
DM-SPP	62.35	0.29	1	1

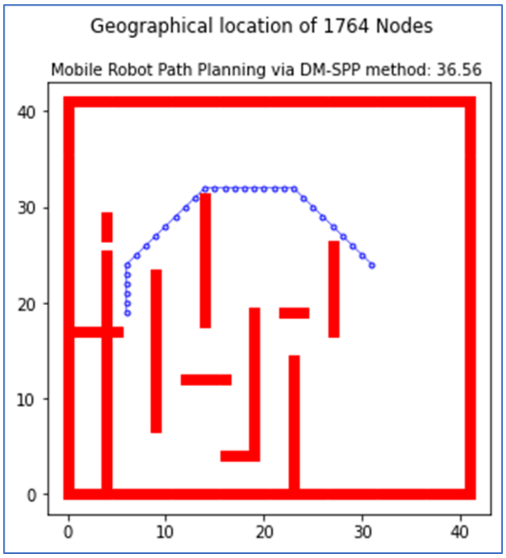

Figure 8. DM-SPP simulation result for Randomly shaped example.

As reported in Table 4, DM-SPP is 71× faster than rPSO and 31× faster than apfrPSO. This consistently high improvement ratio across all case studies reflects the deterministic efficiency of the method.

Table 4. Comparing DM-SPP to PSO methods on Randomly shaped example.

Methods	Distance	Average CPU	Iterations	DM-SPP Improvement
rPSO	21.14	17.12	44	71
apfrPSO	19.73	7.38	48	31
DM-SPP	36.56	0.24	1	1

The aggregated comparison illustrated in Figure 9 confirms the substantial computational advantage of DM-SPP across all four benchmark environments. Consistent with the methodological characteristics, the deterministic structure and absence of parameter tuning enable DM-SPP to achieve overall speed-up factors of 66× relative to rPSO and 31× relative to apfrPSO. These outcomes reinforce the relevance of DM-SPP as a rapid and scalable approach for the mobile robot path planning problem.

In mobile robotics, computing speed is a key performance factor, particularly in scenarios that require immediate navigation decisions. Fast trajectory generation algorithms enable robots to react quickly to environmental changes, avoid collisions, and maintain operational continuity in real time. In industrial environments and warehouses, speed has a direct impact on productivity, task completion rates, and coordination between multiple robots. In rescue, surveillance, or time-critical missions, delays in trajectory computation can compromise safety or mission success. Therefore, although optimality in terms of distance is desirable, many practical applications favor fast and reliable trajectory generation, making fast deterministic algorithms such as DM-SPP particularly relevant when

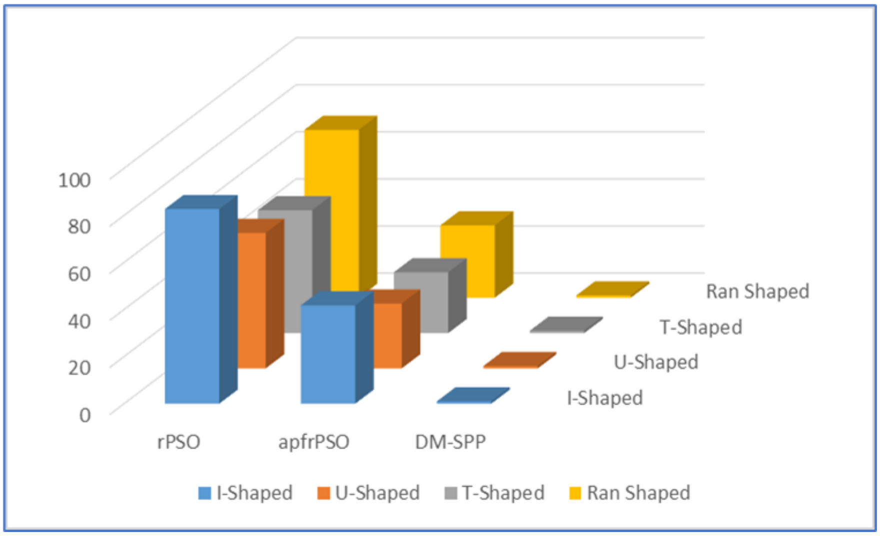

Figure 9. Comparing DM-SPP to PSO methods on four examples.

immediate feasibility is more important than absolute optimality.

It is done with gratitude. DM-SPP has one major limitation: its application is restricted to discrete environments represented by grids. Indeed, DM-SPP operates on a graph where the grid map is transformed into a connected graph. To overcome this limitation, continuous environments can be easily converted into discrete graphs, and then DM-SPP can be applied.

5. Conclusion

This paper has presented a comprehensive comparison between advanced versions of the Particle Swarm Optimization metaheuristic (the ranking Particle Swarm Optimization (rPSO) and the artificial potential field-based Particle Swarm Optimization (apfrPSO) and the novel DM-SPP method for mobile robot path planning. The results establish DM-SPP as a superior approach in terms of computational efficiency, and reliability. The method's parameter-free nature and deterministic behavior make it particularly suitable for practical applications. DM-SPP is tested on four examples (I-shaped, U-shaped, T-shaped and Randomly shaped) and compared with rPSO and apfrPSO metaheuristics. Simulation results prove that DM-SPP can generate rapidly the shortest trajectory, moreover, it is 66 time rapider than the rPSO metaheuristic and 31 time rapider than the apfrPSO metaheuristic. These results show that DM-SPP is particularly advantageous for rapid feasibility assessments in simple, static environments where computational speed is essential. Nevertheless, in complex or highly dynamic scenarios requiring continuous adaptation, more flexible metaheuristic approaches, such as methods based on particle swarm optimization (PSO), may offer advantages. Further research will investigate multi-objectives aspects, trajectory smoothing and 3d resolutions domains.

Interest Declaration

All authors declare that they have no conflicts of interest.

REFERENCES

1. L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, “Path planning techniques for mobile robots: Review and prospect,” *Expert Systems with Applications*, vol. 227, p. 120254, 2023.
2. J. Ahmad, M. N. Ab Wahab, A. Ramli, M. Y. Misro, W. Z. Ezza, and W. Z. W. Hasan, “Enhancing performance of global path planning for mobile robot through alpha-beta guided particle swarm optimization (abgpso) algorithm,” *Measurement*, p. 118633, 2025.
3. A. Benmachiche, M. Derdour, M. S. Kahil, M. C. Ghanem, and M. Deriche, “Adaptive hybrid pso–apf algorithm for advanced path planning in next-generation autonomous robots,” *Sensors*, vol. 25, no. 18, p. 5742, 2025.
4. F. Luo, Q. Zhou, J. Fuentes, W. Ding, and C. Gu, “A soar-based space exploration algorithm for mobile robots,” *Entropy*, vol. 24, no. 3, p. 426, 2022.
5. M. A. Asghar, A. Aslam, S. Bakhet, M. U. Saleem, M. Ahmad, A. Gohar, and H. Khan, “An efficient integration of artificial intelligence-based mobile robots in critical frames for the internet of medical things (iomts) using (adp2s) and convolutional neural networks (cnns),” *Annual Methodological Archive Research Review*, vol. 3, no. 4, pp. 160–183, 2025.
6. W. Di and T. Ahamed, “Design of navigation system for transportation mobile robot for agricultural farms,” in *IoT and AI in Agriculture: Smart Automation Systems for increasing Agricultural Productivity to Achieve SDGs and Society 5.0*, pp. 57–84, Springer, 2024.
7. J. Zhang, X. Yang, W. Wang, J. Guan, W. Liu, H. Wang, L. Ding, and V. C. Lee, “Cross-entropy-based adaptive fuzzy control for visual tracking of road cracks with unmanned mobile robot,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 39, no. 6, pp. 891–910, 2024.
8. T. Feng, J. Li, H. Jiang, S. X. Yang, P. Wang, Y. Teng, S. Chen, Q. Fu, and B. Luo, “The optimal global path planning of mobile robot based on improved hybrid adaptive genetic algorithm in different tasks and complex road environments,” *IEEE Access*, vol. 12, pp. 18400–18415, 2024.
9. D. Kozjek, A. Malus, and R. Vrabič, “Reinforcement-learning-based route generation for heavy-traffic autonomous mobile robot systems,” *Sensors*, vol. 21, no. 14, p. 4809, 2021.
10. L. Manuguerra, F. Cappelletti, M. Rossi, and M. Germani, “Eco-design tool to support the design of industrial electric vehicles. the case studies of an electric shuttle and an autonomous mobile robot,” *Journal of Industrial Information Integration*, vol. 39, p. 100605, 2024.
11. L. Manuguerra, F. Cappelletti, M. Rossi, and M. Germani, “Design of electric vehicles for industry 4.0: the case of an autonomous mobile robot,” *Procedia CIRP*, vol. 120, pp. 980–985, 2023.
12. D. Gurin, V. Yevsieiev, S. Maksymova, and A. Abu-Jassar, “Effect of frame processing frequency on object identification using mobilenetv2 neural network for a mobile robot,” *Multidisciplinary Journal of Science and Technology*, vol. 4, no. 8, pp. 36–44, 2024.
13. K. H. P. Y. Tang, M. C. Ghanem, P. Gasiorowski, V. Vassilev, and K. Ouazzane, “Synchronisation, optimisation, and adaptation of machine learning techniques for computer vision in cyber-physical systems: A comprehensive analysis,” *IET Cyber-Physical Systems: Theory & Applications*, pp. 1–43, 2025.
14. C. Okonkwo and I. Awolusi, “Environmental sensing in autonomous construction robots: Applicable technologies and systems,” *Automation in Construction*, vol. 172, p. 106075, 2025.
15. K. Katona, H. A. Neamah, and P. Korondi, “Obstacle avoidance and path planning methods for autonomous navigation of mobile robot,” *Sensors*, vol. 24, no. 11, p. 3573, 2024.
16. J. Zhong, D. Kong, Y. Wei, X. Hu, and Y. Yang, “Efficiency-optimized path planning algorithm for car-like mobile robots in bilateral constraint corridor environments,” *Robotics and Autonomous Systems*, vol. 186, p. 104923, 2025.
17. P. G. Luan and N. T. Thinh, “Hybrid genetic algorithm based smooth global-path planning for a mobile robot,” *Mechanics Based Design of Structures and Machines*, vol. 51, no. 3, pp. 1758–1774, 2023.
18. S. K. Sahoo and B. B. Choudhury, “A review of methodologies for path planning and optimization of mobile robots,” *Journal of process management and new technologies*, vol. 11, no. 1-2, pp. 122–140, 2023.
19. S. K. Sahoo and B. B. Choudhury, “A fuzzy ahp approach to evaluate the strategic design criteria of a smart robotic powered wheelchair prototype,” in *Intelligent Systems: Proceedings of ICMIB 2020*, pp. 451–464, Springer, 2021.
20. B. Zhang, J. Cao, X. Li, W. Chen, X. Zheng, Y. Zhang, and Y. Song, “Minimum dose walking path planning in a nuclear radiation environment based on a modified a* algorithm,” *Annals of Nuclear Energy*, vol. 206, p. 110629, 2024.
21. C. Sahu, P. B. Kumar, and D. R. Parhi, “An intelligent path planning approach for humanoid robots using adaptive particle swarm optimization,” *International Journal on Artificial Intelligence Tools*, vol. 27, no. 05, p. 1850015, 2018.
22. R. Yao, J. Wang, Y. Lv, and D. Wang, “The time-impact optimal trajectory planning for multi-joint robot based on ga-pso,” in *2024 4th International Conference on Computer, Control and Robotics (ICCCR)*, pp. 284–288, IEEE, 2024.
23. T.-K. Dao, T.-S. Pan, and J.-S. Pan, “A multi-objective optimal mobile robot path planning based on whale optimization algorithm,” in *2016 IEEE 13th international conference on signal processing (ICSP)*, pp. 337–342, IEEE, 2016.
24. C. Lamini, S. Benhlima, and A. Elbekri, “Genetic algorithm based approach for autonomous mobile robot path planning,” *Procedia Computer Science*, vol. 127, pp. 180–189, 2018.
25. J. Ma, Y. Liu, S. Zang, and L. Wang, “Robot path planning based on genetic algorithm fused with continuous bezier optimization,” *Computational intelligence and neuroscience*, vol. 2020, no. 1, p. 9813040, 2020.
26. A. Maoudj and A. Hentout, “Optimal path planning approach based on q-learning algorithm for mobile robots,” *Applied Soft Computing*, vol. 97, p. 106796, 2020.
27. B. Fu, L. Chen, Y. Zhou, D. Zheng, Z. Wei, J. Dai, and H. Pan, “An improved a* algorithm for the industrial robot path planning with high success rate and short length,” *Robotics and Autonomous Systems*, vol. 106, pp. 26–37, 2018.
28. L. Zheng, W. Yu, G. Li, G. Qin, and Y. Luo, “Particle swarm algorithm path-planning method for mobile robots based on artificial potential fields,” *Sensors*, vol. 23, no. 13, p. 6082, 2023.
29. S. Dhouib, “Intelligent path planning for cognitive mobile robot based on dhouib-matrix-spp method,” *Cognitive Robotics*, vol. 4, pp. 62–73, 2024.

30. S. Dhouib, “Faster than dijkstra and a* methods for the mobile robot path planning problem using four movement directions: the dhouib-matrix-spp-4,” in *Mechatronics and Automation Technology*, pp. 284–290, IOS Press, 2024.
31. S. Dhouib, “An optimal method for the shortest path problem: the dhouib-matrix-spp (dm-spp),” *Results in Control and Optimization*, vol. 12, p. 100269, 2023.
32. S. Dhouib, “Shortest path planning via the rapid dhouib-matrix-spp (dm-spp) method for the autonomous mobile robot,” *Results in Control and Optimization*, vol. 13, p. 100299, 2023.
33. J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *Proceedings of ICNN’95-international conference on neural networks*, vol. 4, pp. 1942–1948, ieee, 1995.
34. Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*, pp. 69–73, Ieee, 1998.
35. A. A. Alabdalbari and I. A. Abed, “New robot path planning optimization using hybrid gwo-pso algorithm,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 1289–1296, 2022.
36. L. Zhang, Y. Zhang, and Y. Li, “Mobile robot path planning based on improved localized particle swarm optimization,” *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6962–6972, 2020.
37. S. Dhouib, “Novel heuristic for new pentagonal neutrosophic travelling salesman problem,” *Neutrosophic Sets and Systems*, vol. 51, no. 1, p. 22, 2022.
38. S. Dhouib, “An intelligent assignment problem using novel heuristic: The dhouib-matrix-ap1 (dm-ap1): Novel method for assignment problem,” *International Journal of Intelligent Systems and Applications in Engineering*, vol. 10, no. 1, pp. 135–141, 2022.
39. S. Dhouib, “Novel optimization method for unbalanced assignment problems with multiple jobs: The dhouib-matrix-ap2,” *Intelligent Systems with Applications*, vol. 17, p. 200179, 2023.
40. S. Dhouib, “Solving the single-valued trapezoidal neutrosophic transportation problems through the novel dhouib-matrix-tp1 heuristic,” *Mathematical Problems in Engineering*, vol. 2021, no. 1, p. 3945808, 2021.
41. S. Dhouib, “Novel metaheuristic based on iterated constructive stochastic heuristic: Dhouib-matrix-3 (dm3),” *Applied Computational Intelligence and Soft Computing*, vol. 2021, no. 1, p. 7761993, 2021.
42. S. Dhouib, “Minimizing the drilling robot arm movement by the advanced dhouib-matrix-4 metaheuristic,” *Concurrent Engineering*, p. 1063293X241311734, 2025.
43. S. Dhouib, “Hole drilling route optimization in printed circuit boards using far-to-near metaheuristics: Optimizing the hole drilling route via far-to-near metaheuristic,” *International Journal of Strategic Engineering (IJoSE)*, vol. 5, no. 1, pp. 1–12, 2022.
44. S. Dhouib, “Innovative method to solve the minimum spanning tree problem: The dhouib-matrix-mstp (dm-mstp),” *Results in Control and Optimization*, vol. 14, p. 100359, 2024.
45. S. Dhouib, “Original optimal method to solve the all-pairs shortest path problem: Dhouib-matrix-all-spp,” *Data Science and Management*, vol. 7, no. 3, pp. 206–217, 2024.