Intelligent Path Planning for Cognitive Mobile Robot Based on Dhouib-Matrix-SPP Method

Article *in* Cognitive Robotics · February 2024

DOI: 10.1016/j.cogr.2024.02.001

1 author:

Souhail Dhouib
Higher Institute of Industrial Management, University of Sfax, Tunisia

Cognitive Robotics 4 (2024) 62–73

Contents lists available at ScienceDirect

Cognitive Robotics

journal homepage: http://www.keaipublishing.com/en/journals/cognitive-robotics/

Research Paper

Intelligent path planning for cognitive mobile robot based on Dhouib-Matrix-SPP method

Souhail Dhouib

OLID Laboratory, Higher Institute of Industrial Management, University of Sfax, Tunisia

ARTICLE INFO

Keywords:
Path planning
Mobile robot
Shortest path problem
Artificial intelligence
Operations research
Dhouib-Matrix-SPP

ABSTRACT

The Mobile Robot Path Problem looks to find the optimal shortest path from the starting point to the target point with collision-free for a mobile robot. This is a popular issue in robotics and in this paper the environment is considered as static and represented as a bidirectional grid map. Besides, the novel optimal method Dhouib-Matrix-SPP (DM-SPP) is applied to create the optimal shortest path for a mobile robot in a static environment. DM-SPP is a greedy method based on a column row navigation in the distance matrix and characterized by its rapidity to solve sparse graphs. The comparative analysis is conducted by applying DM-SPP on thirteen test cases and comparing its results to the results given by four metaheuristics the Max-Min Ant System, the Ant System with punitive measures, the A* and the Improved Hybrid A*. The outcomes acquired from different scenarios indicate that the proposed DM-SPP method can rapidly outperform the four predefined artificial intelligence methods.

1. Introduction

A cognitive mobile robot is a vehicle able to navigate, to explore autonomously complex environments and to move by itself from a starting position to a target position with avoiding obstacles. The mobile robot problem has been successfully applied in several areas such as industry, military and security environments.

An improved Ant Colony Optimization method is introduced in [1] in order to generate the optimal path planning for an autonomous robot. A survey of thirty-three papers with an application of three types of path planning techniques (the Generalized Voronoi Diagrams (GVD), the Rapidly exploring Random Tree (RRT) and the Gradient Descent Algorithm (GDA)) is presented in [2]. The Q learning based technique for robotic path planning and the comparison of its performance to the standard path finding methods (A* and Dijkstra) are proposed in [3]. A Hybrid method through integrating the Genetic Algorithm with the Dijkstra method is designed in [4]. An improved version of the Slap Swarm Algorithm and proving its performance by comparing it to five other evolution methods are developed in [5]. A Particle Swarm Algorithm for optimal path planning in the radiation environment (in nuclear facilities) is advanced in [6]. The Simulating Annealing method is used in [7] to generate the shortest path planning for an unmanned robot. A literature review for the path planning generation for an autonomous robot is illustrated in [8,9]. An Ant Colony metaheuristic is developed in [10] to unravel the path planning for an unmanned vacuum cleaner robot. Four techniques (Probabilistic Roadmaps, Rapidly exploring Random Tree, RRT* and A*) are exploded in [11] in order to experiment several scenarios of path planning for the Omni-wheeled mobile robots in a static environment. The Dijkstra algorithm is combined with the Ant Colony System in order to minimize the trajectory of a mobile robot [12]. Also, two variants of the Ant Colony Optimization metaheuristic are developed in [13] and an improved hybrid A* algorithm is developed and tested on eight 20×20 grid models in [14]. Moreover, an Ant Colony Optimization method is developed in [15] to solve the path for a mobile robot in the Context of COVID-19 and a

E-mail address: souh.dhou@gmail.com

https://doi.org/10.1016/j.cogr.2024.02.001
Received 18 October 2023; Received in revised form 3 January 2024; Accepted 13 February 2024
Available online 18 February 2024

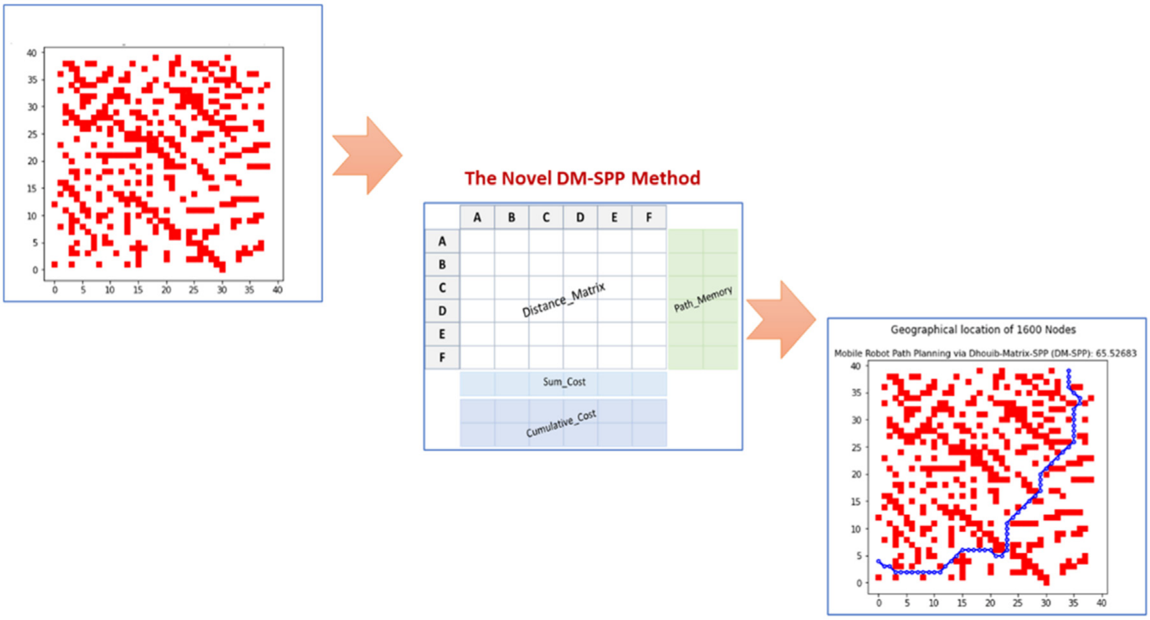

Fig. 1. Solving the path planning via DM-SPP.

Q-learning-based method is developed in [16] to unravel the path trajectory for an unmanned aerial vehicle in a dynamic movement. An improved Theta* method is designed in [17] to generate the 3D path planning for a mobile robot. An Ant Colony Optimization algorithm combined with Clustering Algorithm is used in [18] to plan the trajectory of an unmanned surface vehicle in the maritime environment. An improved A* algorithm is designed to plan the path for an automated guided vehicle in a Port Environment [19]. The Monte Carlo-based method is hybridized with the Ant Colony Optimization algorithm in order to create the 3D optimal path for a welding robot [20]. Three novel techniques (TAD, TRAD and TAD-TRAD) are designed to plan the shortest path for a mobile robot [21] and a hybrid Dijkstra algorithm is introduced in [22]. In addition, three artificial intelligence methods are tested to create the path for a mobile robot in a three-dimensional environment [23]. Several other methods are enhanced for the mobile robot path planning problem such as the artificial fish swarm algorithm [24], the RRT algorithm combined with Bezier Curves method [25], the Modified Ant Colony Optimization [26], the hybrid PSO [27], the Honey Badger algorithm [28] and the modified grey Wolf Optimization algorithm [29].

In this paper, the novel optimal method Dhouib-Matrix-SPP (DM-SPP) is applied to solve the mobile robot planning problem in a grid map (see Fig. 1) and compared to the Ant System metaheuristics. In point of fact, DM-SPP has been recently designed [30] in order to create the shortest path between any two nodes in a graph. Therefore, the grid map is transformed to a graph thanks to the use of the eight neighbors in the grid where the distance is the Euclidean distance between the centers of two cells. The simulation of DM-SPP on five 40×40 grid environment models with eight 20×20 grid maps are studied. Moreover, comparative results are carried out to four metaheuristics (Max-Min Ant System (MMAS), Ant System with punitive measures (AS-N), A* and Improved Hybrid A* (IH-A*)) originally developed by [13,14]. The generated results prove the performance of the proposed DM-SPP in rapidly creating the shortest path which can save time and energy for a mobile robot.

The paper is organized as follows. Section 2, presents the suggested model of the environment space. Section 3 describes in details the proposed DM-SPP for the mobile robot path planning problem. Section 4 illustrates the application of DM-SPP on the robot path planning for thirteen case studies and the comparison of its results to these given by four other metaheuristics (MMAS, AS-N, A* and IH-A*) taken from the literature. Section 5 presents the conclusion and the further works.

2. Modelization of the environment

Commonly, the physical environment of a mobile robot is converted to a virtual one using cameras or sensors. This virtual environment will be summarized as an image and in the literature, there are three map modeling environments (Grid, Geometric and Topological methods) where the Grid method is the simplest and the standard. It consists of subdividing the virtual environment (as an image) in small binary square units defined as obstacles and free to move spaces (in this paper, the solution generated by DM-SPP is generated using the Python programming language where the obstacles are represented by red squares and the free to move spaces are denoted by white squares). The mobile robot is used with a step size of one square with eight directions of the mobile robot (see Fig. 2): Right, Left, Up, Down, Right-Up, Right-Down, Left-Up and Left-Down.

Thanks to the motion direction (node expansion), the mobile robot path planning problem with a grid map can be transformed on a searching for the optimal shortest path problem from the starting to the ending points in a graph. Besides, the generated graph

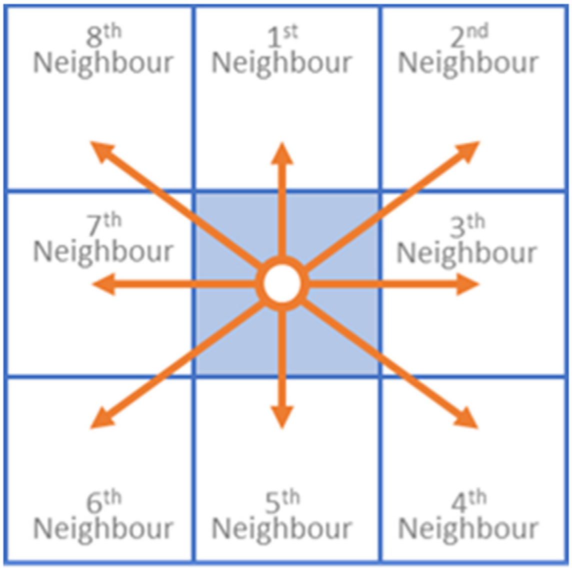

Fig. 2. Node expansion with DM-SPP.

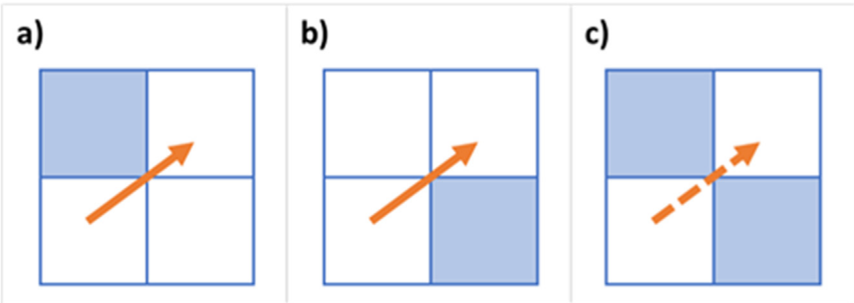

Fig. 3. The three diagonal path selections.

will be a sparse graph (where the number of edges will be at least equal to the number of vertices multiplied by eight: instead of a complete graph). In this graph, each node will be connected at least to eight nodes and DM-SPP has the advantage to rapidly generate the optimal shortest path. DM-SPP is the fastest method for this kind of graphs (for more details see [30]).

Moreover, there are three types of diagonal path selections and in this paper only the two first movements (see Fig. 3.a and b) are allowed. Whereas, the third movement (Fig. 3.c) is forbidden. In addition, the Euclidian distance is used: for the horizontal and vertical movements the distance is computed by adding the value of (1) and for the four diagonal movements (Right, Left, Up, Down, Right-Up, Right-Down, Left-Up and Left-Down) the distance is calculated by the value of (1.41421).

3. The novel Dhouib-Matrix-SPP (DM-SPP)

Very recently (in 2021), we have invented a new optimization concept entitled Dhouib-Matrix (DM) in order to rapidly solve combinatorial problems based on row data (contingency matrix) navigations. DM gathers several new heuristics, novel metaheuristics and original optimal methods (see Fig. 4). Concerning the DM heuristics, five new techniques are designed: DM-TSP1 and DM-TSP2 are developed to solve the Travelling Salesmen Problem, DM-AP1 and DM-AP2 to unravel the Assignment Problem and DM-TP1 to optimize the Transportation Problem. Regarding the DM metaheuristics, three novel approximation methods are considered: DM3 [31], DM4 and FtN. Relating to the DM optimal approaches, three greedy methods are developed: DM-SPP for the Shortest Path Problem [30], DM-ALL-SPP for the All-Pair Shortest Path Problem and DM-MSTP for the Minimum Spanning Tree Problem [32].

In this paper, the novel DM-SPP method is simulated to generate the shortest path for a mobile robot where the environment is represented as a grid map with obstacles and compared to several methods developed in literature. In fact, DM-SPP is a deterministic optimal method and it is very rapid for the case of sparse graphs (DM-SPP is faster than Dijkstra: for more details see [30]). DM-SPP is composed of five steps (see Fig. 5) with a computational time complexity of $O(n + m)$ where n is the number of vertices and m is the number of edges. A detailed step-by-step application of DM-SPP on a simple example is presented in [30].

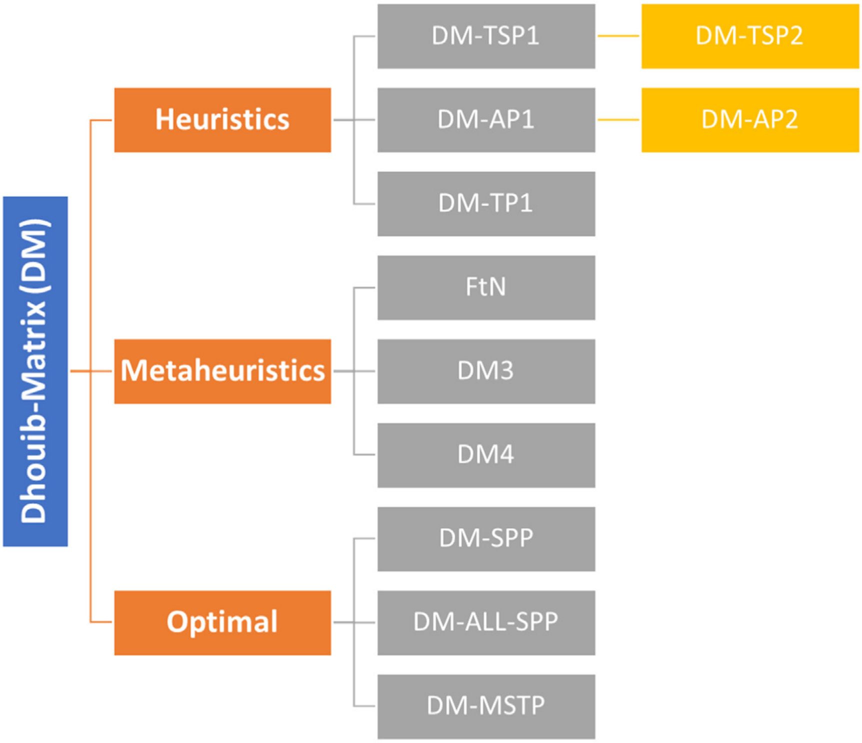

Fig. 4. The DM methods.

Step 1	Set to (*L*) the starting point *S* and discard column (*L*) from *Distance_Matrix*, *Cumulative_Cost* and *Sum_Cost*.
Step 2	Compute the list *Sum_Cost* by adding each element in the first column of *Path_Memory* to each element respectively in row (*L*) of the distance matrix.
Step 3	Modify the elements of *Cumulatice_Cost* matrix (the value from *Sum_Cost* and the name is *L*) only if *Sum_Cost* presents a smallest value.
Step 4	Next, set to (*L*) the column number of the smallest element in the first row of the matrix *Cumulatice_Cost* and affect to row number *L* in *Path_Memory* the corresponding elements of column *L* in *Cumulatice_Cost*. Besides, in *Distance_Matrix* duplicate the elements in column (*L*) to their corresponding elements in row (*L*) and discard from *Distance_Matrix*, *Path_Memory*, *Cumulatice_Cost* and *Sum_Cost* column (*L*).
Step 5	When all columns are discarded, the shortest path is generated from *Path_Memory* otherwise, go to **Step 2**.

Fig. 5. The five steps of DM-SPP.

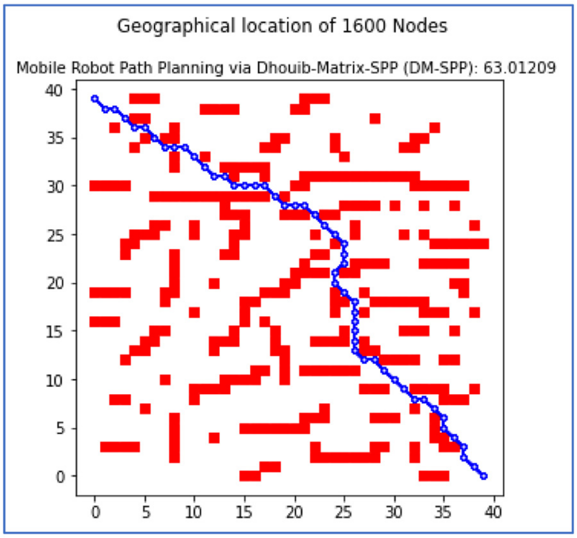

Fig. 6. The path diagram obtained by the proposed DM-SPP method for Environment 1.

DM-SPP starts by presenting data as a contingency matrix with the creation of a new list named *Sum-Cost* and two other novel matrices entitled *Path-Memory* and *Cumulative-Cost*. The *Sum-Cost* list is used as an intermediate memory to archive the added path value. Whereas, the *Path-Memory* matrix is handled to regenerate the optimal path and the *Cumulative-Cost* matrix is applied to drive the selection process through the contingency matrix (a detailed application of the novel DM-SPP method is illustrated in [30]).

4. Computational results

In order to prove the performance of the proposed DM-SPP method, it is simulated on five 40×40 grid environment models with eight other 20×20 grid models and the generated results are compared to the Max-Min Ant System (MMAS), the Ant System with punitive measures (AS-N), the A* and the Improved Hybrid A* (IH-A*) algorithms originally designed in [13,14]. The simulation environment adopts Windows 10 operating system on a Dell Laptop with 1.70 GHz Intel Core i7 CPU and 16GB RAM. Moreover, DM-SPP is developed on Python programming language and using *matplotlib* library.

DM-SPP is a deterministic optimal method, so it will be executed only once (the same result will be generated if it is executed again: So, there is no need to present the best, the worst, the average and the Standard Deviation length path for the DM-SPP method). However, the stochastic MMAS and the AS-N methods are iterated 30 times in order to compute their performance. Accordingly, DM-SPP is reiterated 30 times in order to compute its average CPU time.

To evaluate the performance of DM-SPP, two evaluation metrics (*DM-SPP Improvement* and *DM-SPP Speed*) are developed: The percentage improvement of DM-SPP in accuracy entitled *DM-SPP improvement* is computed as a percentage based on Eq. (1) and the rapidity of DM-SPP compared to the other methods called *DM-SPP Speed* is computed based on Eq. (2).

$$DM-SPP\ IMPROVEMENT = \frac{\left(P_{other} - P_{DM-SPP}\right)}{P_{DM-SPP}} \tag{1}$$

$$DM-SPP\ SPEED = \frac{CPU_{others}}{CPU_{DM-SPP}} \tag{2}$$

Where P_{other} is the distance path generated by the other methods (MMAS, AS-N and IH-A*) and P_{DM-SPP} is the distance path produced by DM-SPP. Moreover, CPU_{others} and CPU_{DM-SPP} designate respectively the planning computational time (CPU) for the other methods (MMAS, AS-N and IH-A*) and the DM-SPP method. Accordingly, the performance of DM-SPP is evaluated based on two criteria: On the one hand, the distance path using the *DM-SPP Improvement* criterion (as a percentage); and on the other hand, the computational time via the *DM-SPP Speed* criterion (as a ratio).

4.1. Comparing DM-SPP to MMAS and AS-N

DM-SPP is tested on the first environment (40×40 two-dimensional grid model) originally designed in [13]. Fig. 6 illustrates the generated solution by DM-SPP where the red squares represent the obstacles to be avoided and the blue line denotes the shortest path for the mobile robot (generated by DM-SPP). DM-SPP creates very rapidly the shortest path (63.0121) in (0.0421) second (for the 40×40 grid so 1600 nodes). Whereas, MMAS needs (1.1963) seconds to generate the solution (63.5451) and AS-N requires (1.2305) seconds to generate a solution of (63.4512). Thus, we can conclude that DM-SPP is (28.42) times rapider than MMAS and (29.23)

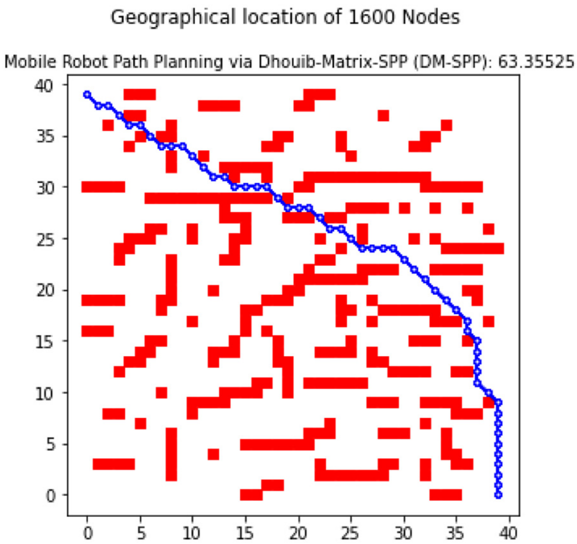

Fig. 7. The path diagram obtained by the proposed DM-SPP method for Environment 2.

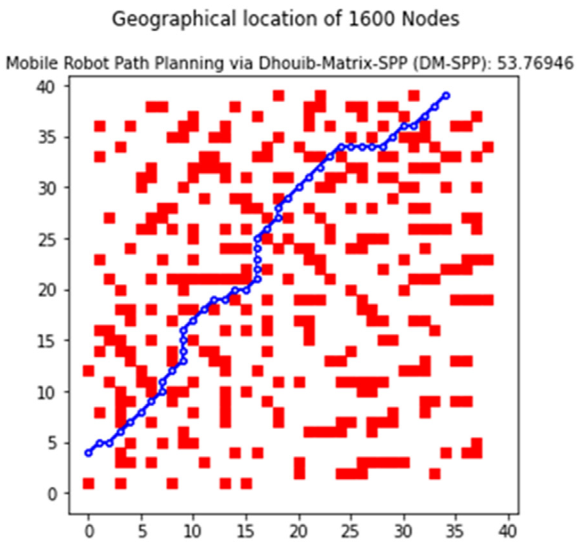

Fig. 8. The path diagram obtained by the proposed DM-SPP method for Environment 3.

times faster than AS-N. Also concerning the accuracy of the distance improvement, DM-SPP improves the shortest path by (0.85 %) and (0.70 %) respectively to MMAS and AS-N. Furthermore, DM-SPP is an optimal method (it always generates the optimal solution) whereas MMAS and AS-N are approximative methods where the optimality is not guaranteed.

The second environment is also a 40×40 two-dimensional grid model. DM-SPP solves optimally (63.3553) this problem in (0.0416) second (see Fig. 7, where the red squares represent the obstacles to be avoided and the blue line denotes the shortest path for the mobile robot). However, MMAS generates a solution of (64.7329) after (1.1963) seconds and AS-N finds a solution of (64.6763) in (1.2305) seconds. Again, DM-SPP offers a massive CPU improvement of (28.76) and (29.58) respectively to MMAS and AS-N with a distance improvement of (2.17 %) and (2.09 %).

Fig. 8 illustrates the generated solution by DM-SPP for the third environment (40×40 two-dimensional grid model). The shortest path (53.7695) is rapidly found after just (0.0429) second. However, MMAS needs (3.3951) seconds to generate the solution (61.3387) and AS-N requires (3.1737) seconds to generate a solution of (61.0082). Once more, DM-SPP offers a huge CPU improvement of (79.14) and (73.98) respectively to MMAS and AS-N methods and also with an important distance improvement of (14.08 %) and (13.46 %).

Fig. 9 shows the generated solution by DM-SPP for the fourth environment (40×40 two-dimensional grid model) where the red squares represent the obstacles to be avoided and the blue line denotes the shortest path for the mobile robot. DM-SPP produces very rapidly the shortest path (57.7695) in (0.0411) second. Despite the fact, MMAS needs (3.3951) seconds to generate the solution

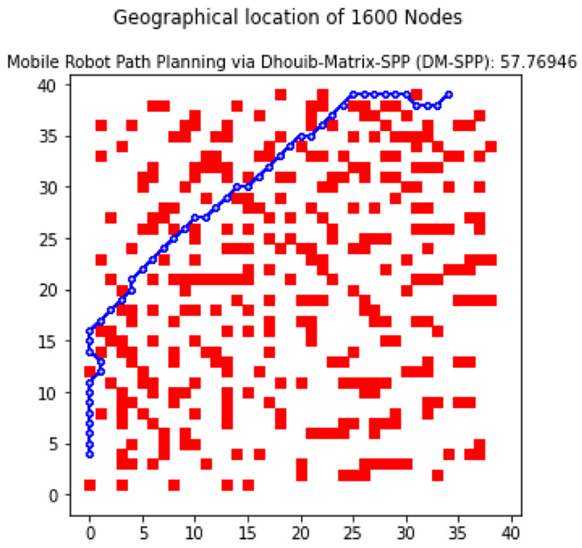

Fig. 9. The path diagram obtained by the proposed DM-SPP method for Environment 4.

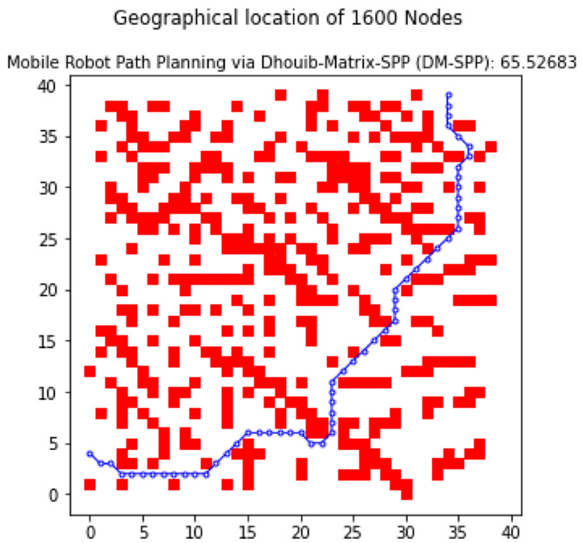

Fig. 10. The path diagram obtained by the proposed DM-SPP method for Environment 5.

(63.8864) and AS-N requires (3.1737) seconds to generate a solution of (63.0581). Another time, DM-SPP offers a vast CPU improvement of (82.61) and (77.22) respectively to MMAS and AS-N methods and also with a distance improvement of (10.59 %) and (9.15 %).

Fig. 10 shows the generated solution by DM-SPP for the fifth environment (40×40 two-dimensional grid model). DM-SPP produces very rapidly the shortest path (65.5268) in (0.0409) second. Despite the fact, MMAS needs (3.3951) seconds to generate the solution (64.9977) and AS-N requires (3.1737) seconds to generate a solution of (64.2471). For this example, DM-SPP represents the same number of points and degrees as AS-N. However, both of the two methods do not find the same result: DM-SPP finds a path length of (65.5268) and AS-N claims a distance of (64.2471) and for the next steps these results are kept. Once again, DM-SPP offers a vast CPU improvement of (83.01) and (77.60) respectively to MMAS and AS-N methods and also with a distance variation of (−0.81 %) and (−1.95 %).

Table 1 gathers all the results generated by MMAS, AS-N and DM-SPP for the five environments of 40×40 two-dimensional grid models.

From the comparison of the planning computational time on the five environments (see Fig. 11), it can be concluded that DM-SPP significatively needs a smaller time than MMAS and AS-N: DM-SPP is 60.39 rapider than MMAS and 57.52 faster than AS-N (see Table 2).

Table 1
The results generated by MMAS, AS-N and the proposed DM-SPP.

Problems	MMAS		AS-N		DM-SPP	
	Path	CPU	Path	CPU	Path	CPU
Environment 1	63.5451	1.1963	63.4512	1.2305	63.0121	0.0421
Environment 2	64.7329	1.1963	64.6763	1.2305	63.3553	0.0416
Environment 3	61.3387	3.3951	61.0082	3.1737	53.7695	0.0429
Environment 4	63.8864	3.3951	63.0581	3.1737	57.7695	0.0411
Environment 5	64.9977	3.3951	64.2471	3.1737	65.5268	0.0409

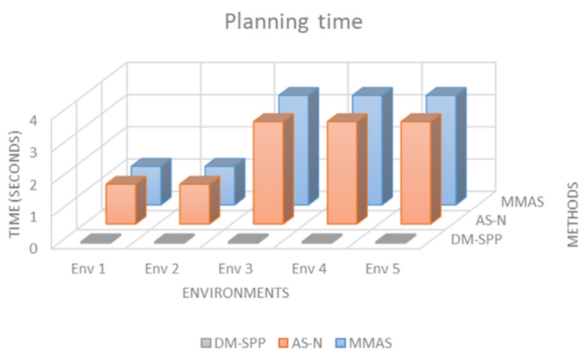

Fig. 11. Comparing the planning computational time.

Table 2
Comparing DM-SPP to MMAS and AS-N for five 40×40 two-dimensional grid models.

Problems	MMAS		AS-N	
	% DM-SPP Improvement	*DM-SPP Speed*	*% DM-SPP Improvement*	*DM-SPP Speed*
Environment 1	0.85	28.42	0.70	29.23
Environment 2	2.17	28.76	2.09	29.58
Environment 3	14.08	79.14	13.46	73.98
Environment 4	10.59	82.61	9.15	77.22
Environment 5	−0.81	83.01	−1.95	77.60
Average	**5.38**	**60.39**	**4.69**	**57.52**

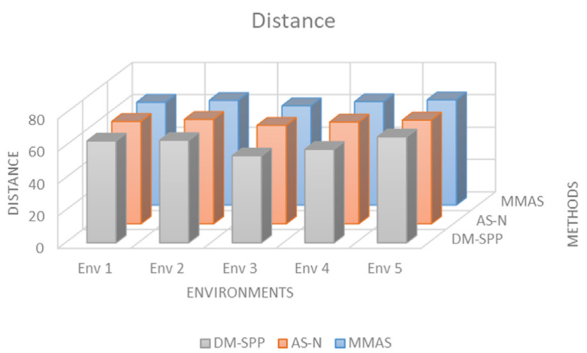

Fig. 12. Comparing the short planning path.

From the comparison of the shortest path on the five environments (see Fig. 12), it can be found that the generated results by DM-SPP are better than the results of the other methods with an average improvement of distance (path length) deviation of (5.38 %) to MMAS and (4.69 %) to AS-N (see Table 2).

Table 2 illustrates the deviation of the MMAS and the AS-N techniques to the proposed DM-SPP method for the five 40×40 two-dimensional grid maps. By comparing the data in Table 2, it is clear that DM-SPP is better than MMAS and AS-N (see the evaluation ratio *DM-SPP Speed*). For example and concerning environment 3, DM-SPP is (79.14) rapider than MMAS and (73.98) faster than AS-N with a corresponding improvement of (14.08 %) and (13.46) on the path distance.

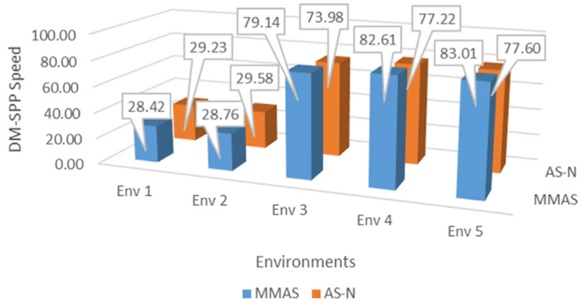

Fig. 13. Comparing the rapidity of DM-SPP to the other methods (MMAS and AS-N).

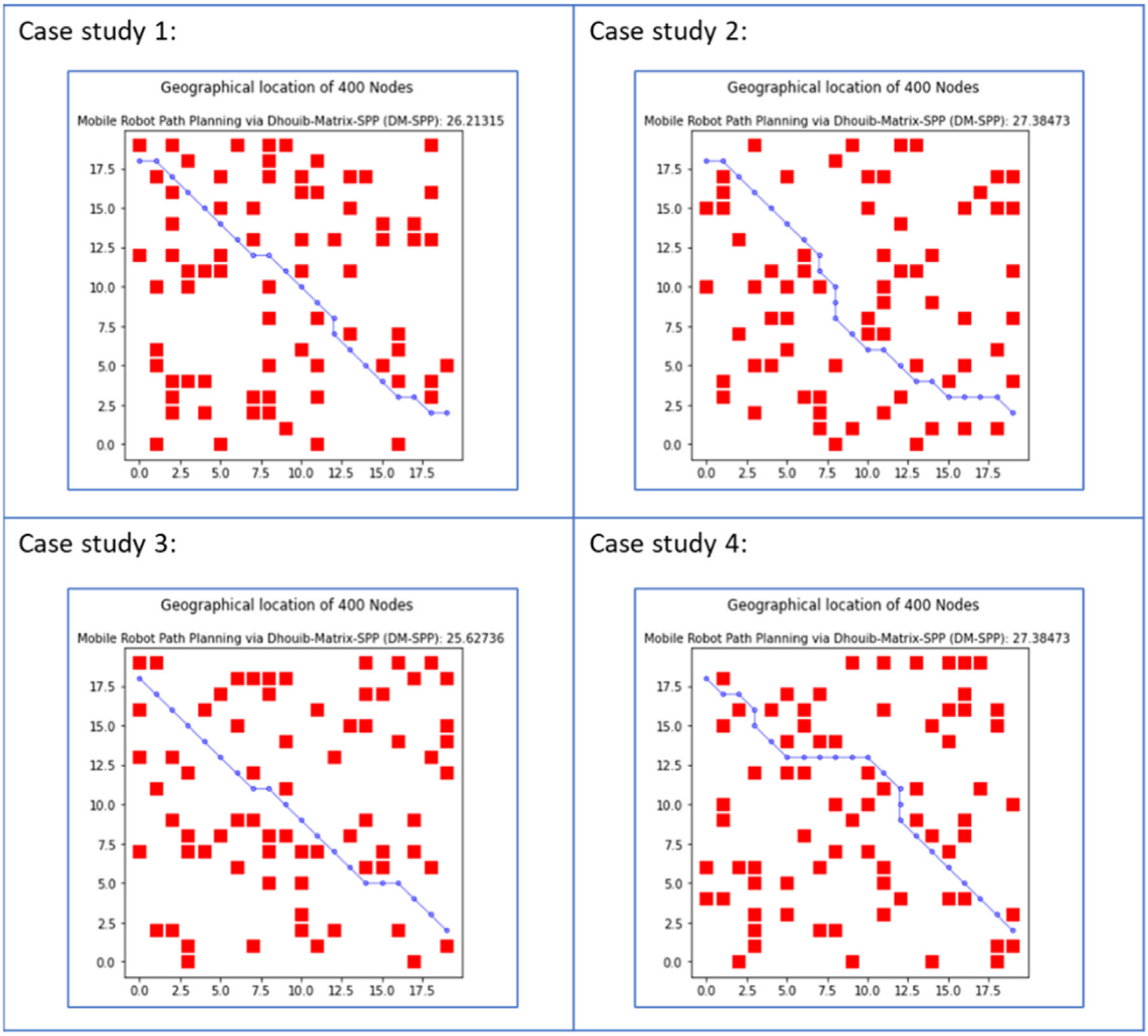

Fig. 14. The simulation experimental results of DM-SPP on 20×20 grid models with an obstacle ratio (0.2).

Fig. 13. depicts the *DM-SPP Speed* indicator for the five environments where the rapidity of DM-SPP is compared to the rapidity of the MMAS and the AS-N methods. It is clear that DM-SPP is the fasted technique for all the environments.

From all the analyses, it can be concluded that the DM-SPP method is better than the MMAS and the AS-N techniques with an average speed improvement of (58.95) in the planning computational time (DM-SPP is (58.95) times rapider than the MMAS and AS-N metaheuristics) and (5.03 %) in the path length. The rapidity of DM-SPP can be explained thanks to its deterministic optimal

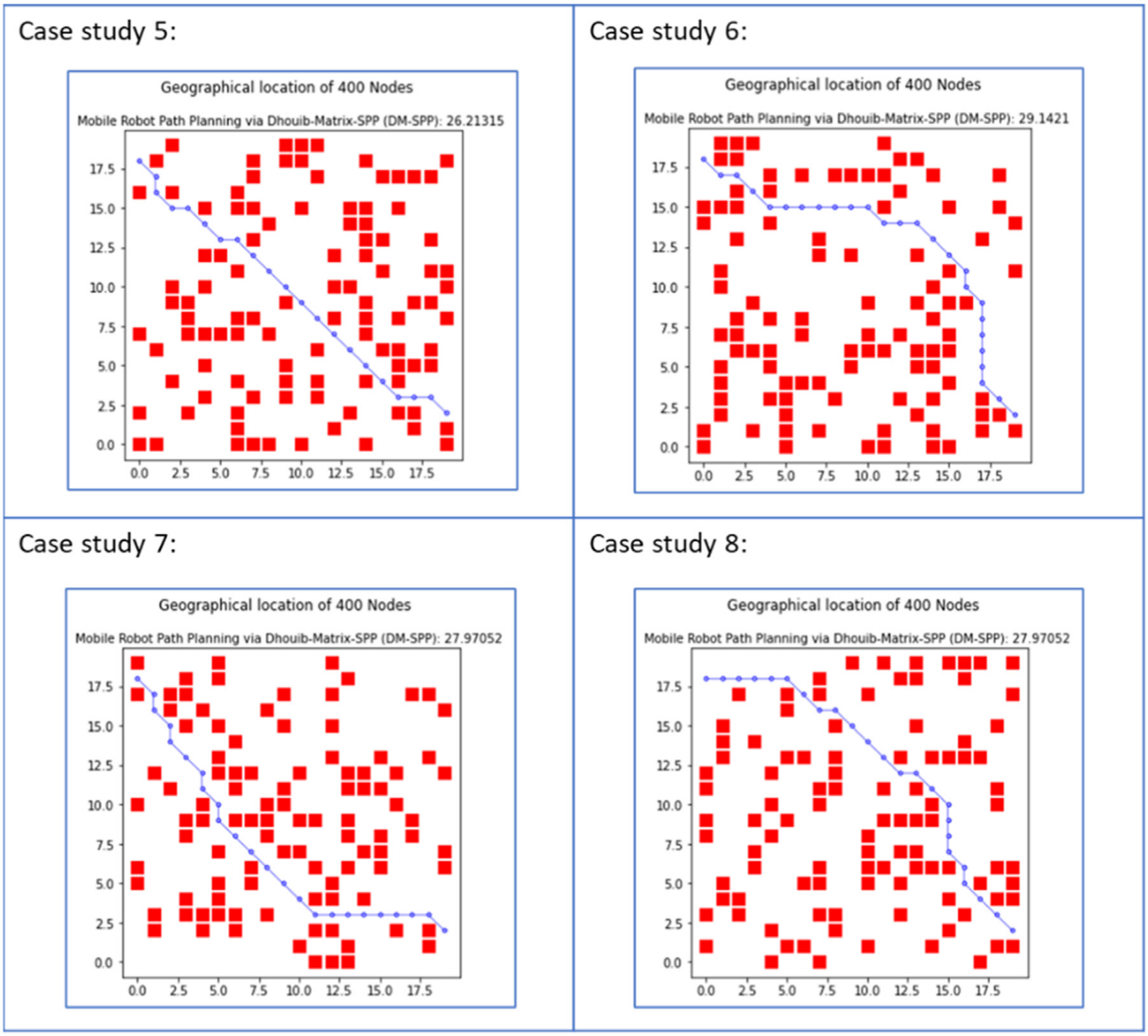

Fig. 15. The simulation experimental results of DM-SPP on 20×20 grid models with an obstacle ratio (0.3).

structure: only one iteration is required to generate the optimal solution differently to the stochastic and iterated structure of MMAS and AS-N: Creating a population and improving each solution by an iterative structure.

4.2. Comparing DM-SPP to A and IH-A**

In this section, the performance of DM-SPP is tested on eight 20×20 grid models and compared to the A* and the Improved Hybrid A* (IH-A*) methods developed in [14]. In fact, DM-SPP is simulated on four 20×20 grid models with an obstacle ratio (0.2 %) and the other four 20×20 grid models with an obstacle ratio (0.3 %). Besides, for all the experiments the starting point is at position (2,0) and the ending point is at position (20,18). Fig. 14 illustrates the DM-SPP simulation experiments with an obstacle ratio (0.2 %) where the average computation time (CPU) ratio is (0.006 s). Whereas, A* and IH-A* require respectively an average computation time of (1.722 s) and (1.300 s).

Moreover, the DM-SPP experimental results with an obstacle ratio (0.3 %) are represented in Fig 15 with an average computation ratio of (0.006 s). While, A* and IH-A* require respectively an average computation time of (2.060 s) and (1.800 s).

The Average searching time for all the methods are represented in Fig 16 where DM-SPP requires a reduced total average computation time of (0.006) second to solve these problems. However, A* and IH-A* require respectively a total average computation time of (1.891 s) and (1.550 s).

Obviously, DM-SPP is faster than A* and IH-A* for the eight case studies (see Fig. 17). Indeed, for the first four case studies with (0.2 %) obstacles DM-SPP is (287.00) faster than A* and (216.67) rapider than IH-A*. Also, concerning the second four case studies with (0.3 %) obstacles DM-SPP is (343.33) faster than A* and (300.00) rapider than IH-A*. Accordingly, the total average percentage improvement (time reduction) of DM-SPP is (315.17) compared to A* and (258.33) compared to IH-A*.

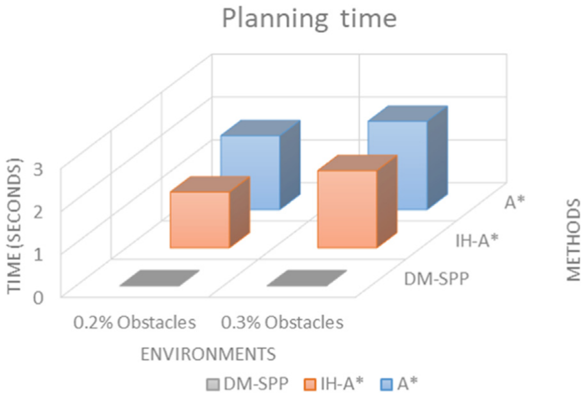

Fig. 16. Comparing the average planning computational time.

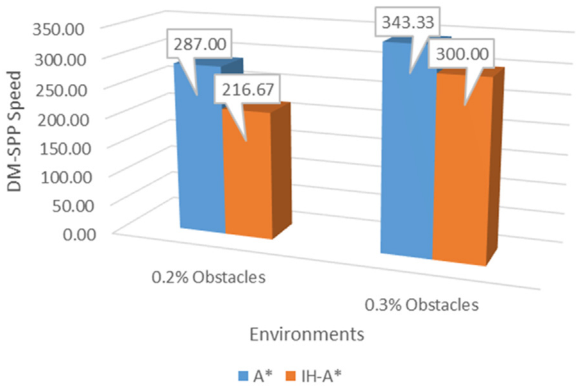

Fig. 17. Comparing the rapidity of DM-SPP to the other methods (A* and IH-A*).

5. Conclusion

This paper makes a valuable contribution in the mobile robot path problem by ensuring an important improving in terms of convergence speed and shortest path acquirement through the enhancement of the novel optimal method Dhouib-Matrix-SPP (DM-SPP) to optimally solve this problem. DM-SPP is tested against the MMAS, the AS-N, the A* and the IH-A* methods for five 40×40 and eight 20×20 two-dimensional grid maps in a static space. The optimization results show that DM-SPP is the fastest and the most accurate method, it rapidly outperforms the MMAS, the AS-N, the A* and the IH-A* methods. In summary, DM-SPP presents a giant improvement in the CPU performance for the mobile robot path planning problem. In this paper, DM-SPP is stimulated using a simple eight nodes movement and a new direction for further research will be to test twelve, sixteen or even twenty-four node direction movements. Also, a further research work will focus on the application of the novel DM-SPP to solve the unmanned area vehicle in a dynamic environment or unknown environment (where the robot does not have a full knowledge about its environment).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] H. Huang, G. Tan, L. Jiang, Robot path planning using improved ant colony algorithm in the environment of internet of things, J. Robot. 2022 (2022) 1739884, doi:10.1155/2022/1739884.
[2] B.K. Jogeshwar, K. Lochan, Algorithms for path planning on mobile robots, IFAC-PapersOnLine 55 (1) (2022) 94–100, doi:10.1016/j.ifacol.2022.04.016.
[3] P. Chintala, R. Dornberger, T. Hanne, Robotic path planning by q learning and a performance comparison with classical path finding algorithms, Int. J. Mech. Eng. Robot. Res. 11 (6) (2022), doi:10.18178/ijmerr.11.6.373-378.
[4] E. Boğar, S. Beyhan, A hybrid genetic algorithm for mobile robot shortest path problem, Int. J. Intell. Syst. Appl. Eng. 1 (4) (2016) 264–267.
[5] X. Cheng, L. Zhu, H. Lu, J. Wei, N. Wu, Robot path planning based on an improved salp swarm algorithm, J. Sens. 2022 (2022) 2559955, doi:10.1155/2022/2559955.
[6] Y. Liu, M. Li, C. Xie, M. Peng, F. Xie, Path-planning research in radioactive environment based on particle swarm algorithm, Prog. Nucl. Energy 74 (2014) 184–192, doi:10.1016/j.pnucene.2014.03.002.

[7] M. Chiu, Path-planning for an autonomous robot using a simulating annealing, J. Inf. Optim. Sci. 32 (2) (2011) 297–314, doi:10.1080/02522667.2011.10700058.
[8] H. Zhang, W. Lin, A. Chen, Path planning for the mobile robot: a review, Symmetry. (Basel) 10 (450) (2018), doi:10.3390/sym10100450.
[9] D. Agarwal, B.S. Bharti, Nature inspired evolutionary approaches for robot navigation: survey, J. Inf. Optim. Sci. 41 (2) (2020) 421–436, doi:10.1080/02522667.2020.1723938.
[10] M. Chiu, The numerical assessment of path planning for an autonomous robotic vacuum cleaner using an ant colony optimization method, J. Inf. Optim. Sci. 32 (5) (2011) 1039–1058, doi:10.1080/02522667.2011.10700103.
[11] M.M. Massoud, A. Abdellatif, M.R.A. Atia, Different path planning techniques for an indoor omni-wheeled mobile robot: experimental implementation, comparison and optimization, Appl. Sci. 12 (24) (2022), doi:10.3390/app122412951.
[12] C. Gu, A. Feng, G. Wang, X. Liu, Robot path planning of improved adaptive ant colony system algorithm based on Dijkstra, J. Robot. 2022 (2022) 9229155, doi:10.1155/2022/9229155.
[13] L. Yue, H. Chen, Unmanned vehicle path planning using a novel ant colony algorithm, J. Wirel. Commun. Netw. 2019 (136) (2019), doi:10.1186/s13638-019-1474-5.
[14] Z. Chi, Z. Yu, Q. Wei, Q. He, G. Li, S. Ding, High-efficiency navigation of nonholonomic mobile robots based on improved hybrid A* algorithm, Appl. Sci. 13 (2023) 6141, doi:10.3390/app13106141.
[15] Y. Shao, X. Deng, L. Feng, Path planning of ant colony algorithm based on decision tree in the context of COVID-19″, Wirel. Commun. Mob. Comput. 2023 (2023) 8984451, doi:10.1155/2023/8984451.
[16] A. Sonny, S.R. Yeduri, L.R. Cenkeramaddi, Q-learning-based unmanned aerial vehicle path planning with dynamic obstacle avoidance, Appl. Soft. Comput. 147 (2024) 110773, doi:10.1016/j.asoc.2023.110773.
[17] L. Han, L. He, X. Sun, Z. Li, Y. Zhang, An enhanced adaptive 3D path planning algorithm for mobile robots with obstacle buffering and improved Theta* using minimum snap trajectory smoothing, J. King Saud Univ. - Comput. Inf. Sci. 35 (10) (2023) 101844, doi:10.1016/j.jksuci.2023.101844.
[18] Liu, X., Li, Y., Zhang, J., Zheng, J., and Yang, C., "Self-adaptive dynamic obstacle avoidance and path planning for USV under complex maritime environment", Vol. 7, (2019), 114945–114954, DOI: 10.1109/ACCESS.2019.2935964.
[19] G. Tang, C. Tang, C. Claramunt, X. Hu, P. Zhou, Geometric A-star algorithm: an improved A-star algorithm for AGV path planning in a port environment, IEEe Access. 9 (2021) 59196–59210, doi:10.1109/ACCESS.2021.3070054.
[20] T. Wang, L. Wang, D. Li, J. Cai, Y. Wang, Monte Carlo-based improved ant colony optimization for path planning of welding robot, J. King Saud Univ. - Comput. Inf. Sci. 35 (7) (2023) 101603, doi:10.1016/j.jksuci.2023.101603.
[21] N.A. Eapen, Path planning of a mobile robot among curved obstacles through tangent drawing and trapezoidal decomposition, Eng. Sci. Technol. Int. J. 24 (6) (2021) 1415–1427, doi:10.1016/j.jestch.2021.03.002.
[22] G. Klančar, A. Zdešar, M. Krishnan, Robot navigation based on potential field and gradient obtained by bilinear interpolation and a grid-based search, Sensors 22 (9) (2022) 3295, doi:10.3390/s22093295.
[23] F. Aslan, A. Durdu, K. Sabanci, "Goal distance-based UAV path planning approach, path optimization and learning-based path estimation: GDRRT*, PSO-GDRRT* and BiLSTM-PSO-GDRRT*", Appl. Soft. Comput. 137 (2023) 110156, doi:10.1016/j.asoc.2023.110156.
[24] T. Zhang, L. Yu, S. Li, F. Wu, Q. Song, X. Zhang, Unmanned aerial vehicle 3D path planning based on an improved artificial fish swarm algorithm, Drones 7 (10) (2023) 636, doi:10.3390/drones7100636.
[25] Z. Durakli, V. Nabiyev, A new approach based on Bezier curves to solve path planning problems for mobile robots, J. Comput. Sci. 58 (2022) 101540, doi:10.1016/j.jocs.2021.101540.
[26] L. Wu, X. Xiaodong Huang, J. Cui, C. Liu, W. Xiao, Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot, Expert. Syst. Appl. 215 (2023) 119410, doi:10.1016/j.eswa.2022.119410.
[27] F.H. Ajeil, I.K. Ibraheem, M.A. Sahib, A.J. Humaidi, Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm, Appl. Soft. Comput. 89 (2020) 106076, doi:10.1016/j.asoc.2020.106076.
[28] G. Hu, J. Zhong, G. Wei, SaCHBA_PDN: modified honey badger algorithm with multi-strategy for UAV path planning, Expert. Syst. Appl. 223 (2023) 119941, doi:10.1016/j.eswa.2023.119941.
[29] X. Liu, G. Li, H. Yang, N. Zhang, L. Wang, P. Shao, Agricultural UAV trajectory planning by incorporating multi-mechanism improved grey wolf optimization algorithm, Expert. Syst. Appl. 233 (2023) 120946, doi:10.1016/j.eswa.2023.120946.
[30] S. Dhouib, An optimal method for the shortest path problem: the Dhouib-Matrix-SPP (DM-SPP), Results Control Optim. 12 (2023) 100269, doi:10.1016/j.rico.2023.100269.
[31] S. Dhouib, Novel metaheuristic based on iterated constructive stochastic heuristic: Dhouib-Matrix-3 (DM3), Appl. Comput. Intell. Soft Comput. 2021 (2021) 7761993, doi:10.1155/2021/7761993.
[32] S. Dhouib, Innovative method to solve the minimum spanning tree problem: the Dhouib-Matrix-MSTP (DM-MSTP), Results Control Optim. 14 (2024) 100359, doi:10.1016/j.rico.2023.100359.