Multi-AGV Path Planning for Indoor Factory by Using Prioritized Planning and Improved Ant Algorithm

Yi Zhang, Fuli Wang*, Fukang Fu & Zuqiang Su

School of Advanced Manufacturing Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.
*E-mail: fuli.wang27@foxmail.com.

Abstract. Multiple automated guided vehicle (multi-AGV) path planning in manufacturing workshops has always been technically difficult for industrial applications. This paper presents a multi-AGV path planning method based on prioritized planning and improved ant colony algorithms. Firstly, in dealing with the problem of path coordination between AGVs, an improved priority algorithm is introduced, where priority is assigned based on the remaining battery charge of the AGVs, which improves the power usage efficiency of the AGVs. Secondly, an improved ant colony algorithm (IAC) is proposed to calculate the optimal path for the AGVs. In the algorithm, a random amount of pheromone is distributed in the map and the amount of pheromone is updated according to a fitness value. As a result, the computational efficiency of the ant colony algorithm is improved. Moreover, a mutation operation is introduced to mutate the amount of pheromone in randomly selected locations of the map, by which the problem of local optimum is well overcome. Simulation results and a comparative analysis showed the validity of the proposed method.

Keywords: *ant algorithm; collision avoidance; decentralized algorithm; path planning.*

1 Introduction

In today's manufacturing industry, managers have shifted their strategy from manufacturing large numbers of individual products to a range of products and to improving quality and delivery time. Accordingly, many of today's logistic and manufacturing processes rely on the use of multi-AGV systems [1], especially in flexible manufacturing systems (FMS). FMS performance increases by better coordination of its components, such as AGVs [2]. In order to improve a multi-AGV system to operate efficiently and safely, the primary problem to solve is properly planning the path of multiple AGVs. This technology originates from robot path planning and aims to find a collision-free path from a start location to a target location while optimizing one or more objectives, such as path length, smoothness, and safety at the same time [3].

First of all, when dealing with multi-AGV path planning, it should be ensured that there will be no collisions between AGVs. At present, for this problem two

Received April 4th, 2018, 1st Revision July 4th, 2018, 2nd Revision September 7th, 2018, Accepted for publication September 26th, 2018.
 ISSN: 2337-5779, DOI: 10.5614/j.eng.technol.sci.2018.50.4.6

main two solutions are adopted: the coupled approach and the decoupled approach [4]. In the coupled approach, AGVs act as a single unit or run in a composite manner. For this situation, some researchers use graphs with specific topologies to search the minimum spanning tree of the roadmap [5,6] or restrict the problem domain to grid-worlds [7]. Notwithstanding some desirable properties such as completeness and the possibility of calculating optimal motion plans, coupled methods are highly demanding in terms of computational resources [1]. Decoupled methods, on the other hand, are able to be fast enough for real-time application. For example, the method used in [8] is a decoupled multi-agent motion planning method. It uses an algorithm for finding the path for the respective AGVs, but this method does not deal with conflict scenarios. A widely used decoupled scheme for multi-robot motion planning that has been shown to be effective in practice is prioritized planning [9]. Although prioritized planning is practical, it has a disadvantage in that it is incomplete, i.e. in some situations, it cannot find a path even if there exists one [10]. Recently, Čáp and co-workers [11] have proposed a revised prioritized planning algorithm to make up for this disadvantage. Although this improved method is suitable for the real-time application, it has a safety risk. Because it takes each AGV as a particle, it ignores its body size. To cope with the above concerns, this study proposes a revised prioritized planning algorithm for real-time application in a factory and defines a redundant time period to improve safety for AGVs.

In addition, the battery management in the AGV system is crucial, as it can reduce the costs and increase the efficiency of the system [12]. Some studies have focused on energy-harvesting methods to save battery energy [13,14]. Besides that, considering the AGV's remaining battery charge when planning for an AGV path is also important. In [15], the research was aimed at scheduling AGVs in an FMS environment by developing a multi-objective mathematical model that minimizes the makespan and the number of AGVs while considering the AGVs' battery charge. However, this model does not consider traffic problems, collisions, deadlock or conflicts. Therefore, for multi-AGV scheduling, this study considered both AGV battery charge and collision-free constraints.

Another issue of dealing with multi-AGV path planning is to find the optimal trajectory for each AGV. The current trend is to solve these problems by using robust heuristics algorithms [16]. For example, a genetic algorithm (GA) [15] or particle swarm optimization (PSO) [17]. Among these, the ant colony algorithm [18] is one of the most widely used approaches for AGV path planning because of its good robustness and positive feedback mechanism. However, the ant colony algorithm is faced with problems of low computational efficiency and easiness to obtain non-optimal solutions. Compared with other algorithms, the ant algorithm lacks a mutation operation. This operation is a crucial step in

intelligent algorithms, which can enhance population diversity to avoid the algorithm falling into a local optimum. For example, in [17] a mutation was added to the PSO algorithm to improve its performance. Likewise, the present study introduces a mutation operation to avoid the algorithm falling into a local optimum and improves computing efficiency by modifying the initialization and update the operations of the pheromone.

2 Coordination Between AGVs

2.1 Prioritized Planning Algorithm

Multi-AGV systems should have good coordination in order to ensure that no collisions occur between AGVs. In this paper, a revised priority planning [11] is proposed to deal with this problem. This algorithm is described as follows: each AGV is assigned a unique priority and the algorithm proceeds sequentially from the highest-priority AGV to the lowest-priority one. In each iteration, one of the AGVs plans its trajectory such that it avoids the higher-priority AGVs. Furthermore, the prioritized planning algorithm that seeks a trajectory for each AGV is revised in such a way that both (a) the start positions of all lower-priority AGVs are avoided, and (b) conflicts with higher-priority AGVs are avoided. The pseudocode of the revised prioritized planning algorithm is as follows:

Algorithm 1: Prioritized planning algorithm

Initialize: $\Delta \leftarrow \varnothing$

{for $i \leftarrow 1...n$

$S \leftarrow \bigcup_{j>i} S^j$

$\pi_j \leftarrow Best - traj(W \setminus S, \Delta)$;

If $\pi_i = \varnothing$, then

Report failure and terminate;

$\Delta \leftarrow \Delta \cup R_i^{\Delta}(\pi_i)$;

}

Function $Best - traj(W \setminus S, \Delta)$

Call algorithm 2 to obtain optimal satisfying trajectory for AGV *i* in the workshop (W) that avoids regions Δ if it exists. Otherwise, return $\varnothing$.

In the above pseudo-code, Δ acts as a dynamic obstacle. $S \leftarrow \bigcup_{j>i} S^j$ represents that AGV i can go through the start positions of higher-priority AGVs. The function

$$R_i^{\Delta}(\pi_i) := \{(x, y, t) : t \in [0, \infty) \wedge (x, y) \in R_i(\pi(t))\} \tag{1}$$

maps the trajectories of an AGV i to regions of spacetime that AGV i occupies when its center point follows given trajectory π.

The loop structure is for each AGV to plan paths, avoiding both start positions of all lower-priority AGVs and higher-priority AGVs. Furthermore, the function $Best - traj(W \setminus S, \Delta)$ calls Algorithm 2 (the improved ant algorithm that will be discussed in Section 3) to obtain the optimal path for each AGV.

As shown in Figure 1, AGV 1 (the blue one in Figure 1) has a higher priority than AGV 2 (the green one). Therefore, during route planning, the optimal path is first calculated for AGV 1 and then the path is calculated for AGV 2. If both AGVs reach the same point at the same time, a collision occurs. In this case, AGV 2 with a low priority should avoid the high-priority AGV 1. Actually, this is used to judge the intersection point on the AGV trajectory. If there is an intersection point between the paths of two AGVs and the two AGVs will reach the intersection point at the same time, in this case, to avoid a collision, low-priority AGVs have to avoid high-priority AGVs.

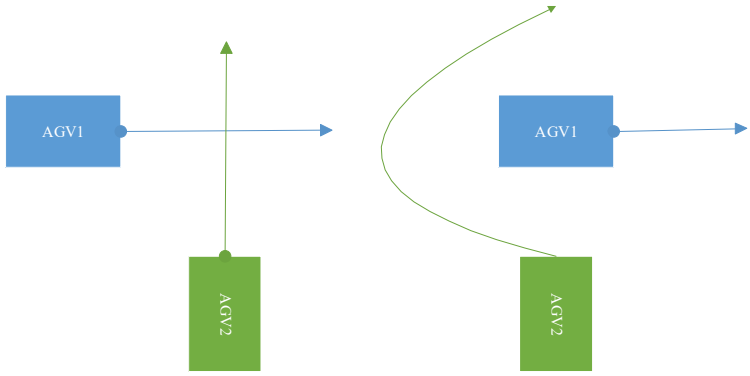


Figure 1 Avoidance between AGVs with different priorities.

2.2 Improvement of Prioritized Planning Algorithm for AGV

Most researches on prioritized path planning mainly focused on mobile robots [19-21]. In practical applications, AGVs tend to be designed with a long shape

for being able to load more materials. As one can see from Eq. (1), in the prioritized planning algorithm, it is determined if there will be a conflict by seeing whether two AGV appear in the same location (x, y) at the same time, *t*. In order to ensure safe avoidance between AGVs, a redundant time period is defined in this paper to improve the prioritized planning algorithm. That is, if two AGVs arrive at an intersection point within this time period, then a collision is determined. The definition is as follows:

If $(x, y, t \pm \triangle t) \in \Delta$, then we know that the intersection point (x, y) is occupied by the dynamic Δ in the time period $[t - \triangle t, t + \triangle t]$.

$$\triangle t = \frac{l + r}{v} \tag{2}$$

In Eq. (2), t is the time when the high-priority AGV reaches the intersection point (the center point of the AGV coincides with the intersection point); the number l indicates the width of the body of the AGV; r is the buffer distance, the purpose of which is to enhance the safety margin of AGV avoidance; v is the normal running speed of the AGV. When another low-priority AGV also reaches an intersection within the same time period, this point is considered an obstacle for the low-priority AGV.

On the other hand, in prioritized planning one of the neglected issues is how to assign priority to AGVs. This paper distributes the priority according to the AGVs' remaining battery charge. The purpose is to make AGVs with lower battery charge complete their task successfully. Therefore, the lower the AGV priority is, the higher its battery charge will be. For example, suppose there are four AGVs that need to perform a task. The current power (battery percentage) of each AGV is: 1-AGV (35%), 2-AGV (50%), 3-AGV (28%), 4-AGV (47%). The priority of these AGVs, from high to low, is:

3-AGV > 1-AGV > 4-AGV > 2-AGV

3 Improved Ant Colony Algorithm for Path Planning

Ant colony algorithms have been successfully applied in path planning of mobile robots due to their characteristics of strong robustness and parallel computing [22,23]. However, the performance of an ant colony algorithm depends on the adjustment of parameters, especially the initialization and update of pheromone. Therefore, most researches so far aimed to improve the pheromone operations. In this section, we first introduce the two most representative improved ant colony algorithms: Ant Colony System (ACS) [24]

MAX-MIN Ant System (MMAS) [25] and then compare them with the proposed algorithm (IAC).

3.1 Ant Colony System

The pheromone update of ACS contains global update rules and local update rules. The global update is only used for the best ant in each loop:

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \rho\triangle\tau_{ij} \tag{3}$$

$$\triangle\tau_{ij} = \begin{cases} 1/L_{gb} \\ 0 \end{cases} \tag{4}$$

In Eq. (3), ρ is the pheromone heuristic factor, $\triangle\tau_{ij}$ is the pheromone increment. In Eq. (4), L_{gb} is the globally optimal path length. There will be partial updates after all ants have finished each action. ACS only adjusts the amount of pheromone on the path taken by the best individual in each generation to speed up convergence, but the algorithm will prematurely converge to a non-optimal solution.

3.2 MAX-MIN Ant System

In order to avoid premature convergence of the algorithm to non-optimal solutions, MMAS has three improvements:

1. Firstly, initialize the amount of pheromone as a constant, C.
2. After one loop, only the ants who find the shortest path can release pheromone on their path im Eq. (5):

$$\tau_{ij} = (1-\rho)\cdot\tau_{ij} + \triangle\tau_{ij}^{\min} \tag{5}$$

$$\tau_{ij}^{\min} = \frac{Q}{L} \tag{6}$$

   In Eq. (6), L is the shortest path of this iteration and Q is the pheromone constant.
3. The last is to limit τ_{ij} to a given range: $\tau_{ij} \in [\tau_{\min}, \tau_{\max}]$.

MMAS can improve the convergence speed of the algorithm, but there are still some defects. The MMAS initializes the amount of pheromone as a fixed value C, leading to a slower search speed in the early stage of the algorithm. Moreover, MMAS can still fall into local optimum solutions.

3.3 Improvement of Ant Colony Algorithm

Similar to MMAS, IAC limits the pheromone to $\tau_{ij} \in [\tau_{\min}, \tau_{\max}]$, but the amount of pheromone in each location of the map is randomly distributed between 0 and C:

$$\tau_{ij}(0) = \mathrm{c} \cdot rand() \tag{7}$$

In Eq. (7), $rand()$ is a random number of uniform distribution from 0 to 1, and $\tau_{ij}(0)$ represents the initial amount of pheromone at position *ij*.

Furthermore, the ant colony algorithm is complex in parameter tuning. The scope of pheromone Q has a large range of values according to the scale of the problem solved. Therefore, in the renewal stage of IAC, the pheromone constant Q is not used. With the iteration of the calculation, the amount of pheromone is updated by comparing the fitness values in Eq. (8):

$$\tau_{ij}(t+n) = (1-\rho) \cdot \tau_{ij}(t) + \triangle\tau_{ij} \tag{8}$$

$$\triangle\tau_{ij} = m\frac{L_{\max} - l}{l} \tag{9}$$

In Eq. (9), $L_{\max}$ is the longest path distance in the current iteration, i.e. the fitness value of inferior ants; l is the fitness value of the current ant; m is a constant and controls the update of the amount of pheromone. In the update operation of this algorithm, the pheromone is updated by comparing the current fitness value with the worst fitness value.

In addition, the results of the ant colony algorithm will always fall into a local optimum. Compared with other algorithms it is not difficult to find that the ant colony algorithm lacks a mutation operation to increase the diversity of the population. Therefore, a pheromone mutation operation is added to this algorithm.

$$\tau_{mn}(t') = \left|\tau_{mn}(t) + (\tau_{mn}(t) - \tau_{\min}) \cdot g\right| \tag{10}$$

where g is a Gaussian random number with a mean of 0 and a variance of 1. This algorithm mutates the amount of pheromone by randomly selecting part of the location. In Eq. (10), when the position (m,n) is selected, the amount of pheromone at that position is changed from $\tau_{mn}(t)$ to $\tau_{mn}(t')$. In order to ensure that the amount of pheromone is still positive, the absolute value symbol is added in Eq. (10). In summary, the entire improved ant colony algorithm can be obtained as follows:

Algorithm 2: Improved ant colony algorithm

Initialize: $ncycle = 1$;
$bestcycle = 1$;
η_{ij}; $\tau_{ij}(0)$=c$\cdot rand()$; $\tau_{\max}$; $\tau_{\min}$; $\Delta\tau_{ij} = 0$;
$tabu_k = null$;
While (not termination condition)
{for (k=1; k<m; k++)
{M ants are randomly placed on the initial map;}
for ($index$=0; $index$<n; $index$++) (Index is the location of the current loop)
{for(k=1; k<m; k++)
{Select next place with probability $p_{ij}^k(t)$;
The selected place is placed in $tabu_k$;
}
}
$ncycle = ncycle + 1$;
$L_{\min} = \min(l_k), k = 1,2,...,m$;
$L_{\max} = \max(l_k), k = 1,2,...,m$;
Calculate $\tau_{ij}(t+n)$, $\tau_{ij}(t+n)$ from Eq. (8) and (9);
If $\tau_{ij}(t+n) < \tau_{\min}$, then $\tau_{ij}(t+n) = \tau_{\min}$;
If $\tau_{ij}(t+n) > \tau_{\max}$, then $\tau_{ij}(t+n) = \tau_{\max}$;
Perform the mutation operation by using Eq. (10)
in some parts of the map;
}
Output the best path and results;
end

4 Simulation Results and Comparative Analysis

4.1 Analysis of Proposed Method Validity

This section validates the effectiveness of the path planning method proposed in this paper through two sets of simulations. All simulations were implemented in Matlab and run on a Core i3-2120 CPU 3.30GHz PC. The test environment maps used 30×30 grid maps [26], as shown in Figure 2.

In Figure 2, 'S1', 'S2', etc. denote different workstations, 'Charge' denotes a charging area, 'Depot' denotes a warehouse, and each warehouse location is assigned an AGV. Each AGV has its own driving route. After the simulation begins, each AGV is assigned a priority first and then the IAC is used for path planning.

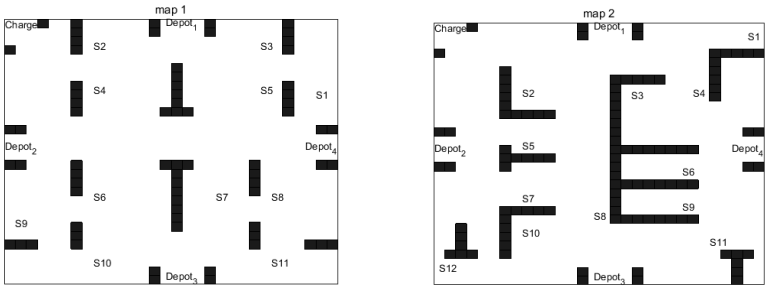


Figure 2 2D industrial environment maps.

The number and battery charge of each AGV are as follows: 1-AGV (41%), 2-AGV (45%), 3-AGV (25%), 4-AGV (30%). The priority of the AGVs can be obtained as follows:

$$3\text{-AGV} > 4\text{-AGV} > 1\text{-AGV} > 2\text{-AGV}$$

Because the path to be solved in the grid map is too tortuous, a B-spline curve [27] is used to smoothen the route before generating the result. The simulation results are shown in Figure 3, where the red route is 1-AGV's route, blue is 2-AGV's route, purple is 3-AGV's route, and green is 4-AGV's route.

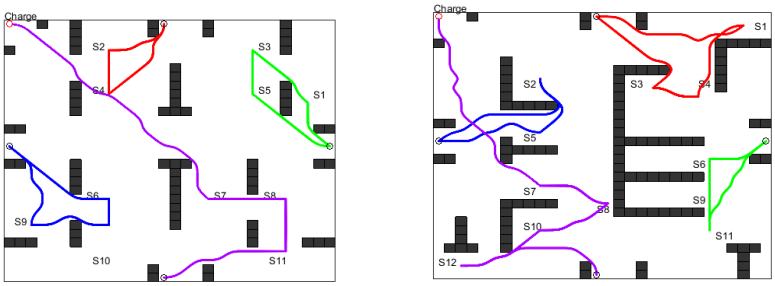


Figure 3 Path planning result diagrams of AGVs.

Because the initial battery charge of 3-AGV was low, it needed to go to the charging area after the task was completed. Therefore, according to the requirements of the priority algorithm, 3-AGV's path was planned first and then the path planning was performed for other the AGVs in the priority order.

As can be seen from the above examples, the path planning method presented in this paper is quite efficient for multiple AGVs in an indoor factory.

4.2 Analysis of Prioritized Planning Algorithm

In order to more clearly reflect the function of the prioritized planning algorithm, we chose to use a more special map for simulation testing. As shown in Figure 4, when a shortage of materials occurs at workstations 1, 3, and 5, 1-AGV is required for delivering materials and thus we give 1-AGV the highest priority.

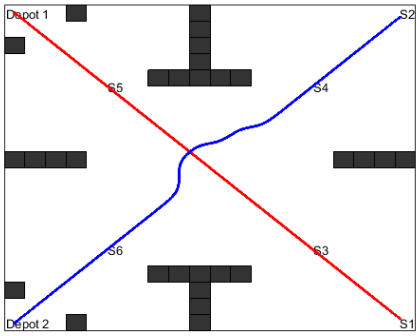


Figure 4 Diagram of avoidance between AGVs based on prioritized planning algorithm.

In Figure 4 it can be clearly seen that in the center of the map, 2-AGV (low priority) actively avoids 1-AGV (red curve in Figure 4). Therefore, the prioritized planning algorithm solves the avoidance problem between AGVs well.

4.3 Comparison Analysis of Ant Colony Algorithm

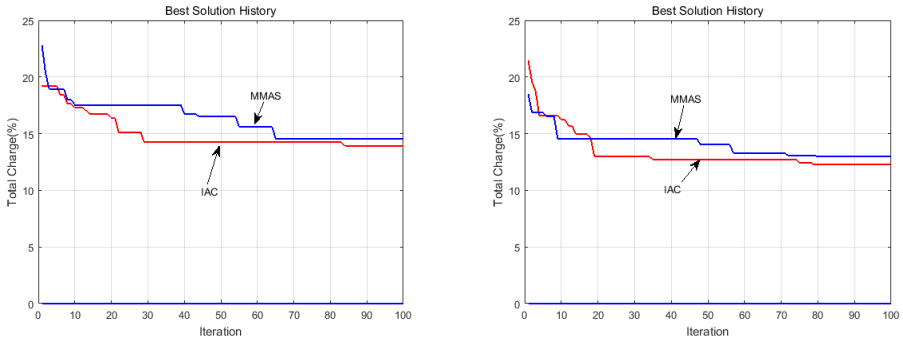
In order to compare the proposed ant colony algorithm with the existing ant colony algorithm, we used ACS, MMAS and IAC to calculate the optimal route of 3-AGV in Map 1 (as shown in Figure 2). Each algorithm was iterated 20 times. The average battery charge consumption and the calculation cost are shown in Table 1. The pheromone constant Q was set to 15, the ant colony size was set to 40, the pheromone factor $\alpha = 1$, the heuristic factor $\beta = 6$, the pheromone volatilization coefficient $\rho = 0.1$, and the maximum number of iterations was set to $N = 200$.

Table 1 Comparison of Algorithms' Performance

Algorithm	Average power consumption	Average calculation time (s)
ACS	14.59%	184.933
MMAS	14.48%	192.484
IAC	14.24%	171.311

In Table 1, it can be seen that IAC had the highest computational efficiency and better results compared to other well-known algorithms. Therefore, the power usage efficiency of the AGVs was improved.

Furthermore, the convergence speed of the MMAS and IAC was also compared. Both algorithms were applied to solve the results of 3-AGV in the two maps in Figure 2. As shown in Figure 5, the blue and red curves are the convergence curves of MMAS and IAC, respectively, where $N = 100$.



(a) results of graph 1 (b) results of graph 2

Figure 5 Convergence graphs of ant algorithm.

Through comparison of the convergence curves in Figure 5 we can see that IAC got better solutions compared with the MMAS algorithm. This is because IAC has an added mutation operation to avoid falling into a local optimum. Therefore, using the route solved by IAC can calculate the AGV route in a shorter time and the power usage efficiency of the AGVs is also improved.

5 Conclusions

This paper presented a path planning method for a fleet of AGVs in indoor factory environments. Firstly, a priority algorithm is used to assign priorities based on the remaining power of each AGV and a redundant time period is defined to improve the safety of avoidance. Then, an improved ant colony algorithm is applied to solve the optimal paths for the AGVs. The improved method was mainly developed to increase the computational efficiency of the algorithm and keep the algorithm from obtaining a local optimum. Simulation experiments showed the effectiveness of the proposed method. Compared with other ant colony algorithms, the improved ant colony algorithm can save computing time and find optimal paths.

Acknowledgement

This work was supported by the Science and Technology Research Project of Chongqing Municipal Education Commission (KJ1600443), Chongqing Science and Technology Commission (cstc2015jcyjBX0066).

References

[1] Draganjac, I., Miklić, D., Kovačić, Z., Vasiljević, G. & Bogdan, S., *Decentralized Control of Multi-AGV Systems in Autonomous Warehousing Applications*, IEEE Transactions on Automation Science & Engineering, **13**(4), pp. 1433-1447, 2016.

[2] Pan, X.Y., Wu, J., Zhang, Q.W., Lai, D., Xie, H.L. & Zhang C, *A Case Study of AGV Scheduling for Production Material Handling,* Applied Mechanics and Materials, **411**, pp. 2351-2354, 2013.

[3] Wang, F. & Zhu, Z., *Global Path Planning of Wheeled Robots Using a Multi-Objective Memetic Algorithm*, Integrated Computer Aided Engineering, **22**(4), pp. 387-404, 2015.

[4] Li, B., Liu, H., Xiao, D. & Zhang, Y., *Centralized and Optimal Motion Planning for Large-scale AGV Systems: A Generic Approach*, Advances in Engineering Software, **106**(C), pp. 33-46, 2017.

[5] Peasgood, M., Clark, C. & McPhee, J., *A Complete and Scalable Strategy for Coordinating Multiple Robots within Roadmaps*, IEEE Transactions on Robotics, **24**(2), pp. 282-292, 2008.

[6] Clark, C., *Probabilistic Road Map Sampling Strategies for Multi-robot Motion Planning*, Robotics and Autonomous Systems, **53**(3), pp. 244-264, 2005.

[7] Wang, K.H.C. & Botea, A., *Tractable Multi-Agent Path Planning on Grid Maps*, International Joint Conference on Artificial Intelligence IJCAI-09, Pasadena, CA, USA, 2009, pp. 1870-1875.

[8] Ma, X., Jiao, Z., Wang, Z. & Panagou, D., *Decentralized Prioritized Motion Planning for Multiple Autonomous UAVs in 3D Polygonal Obstacle Environments*, International Conference on Unmanned Aircraft Systems (ICUAS), Arlington, VA USA, 2016, pp. 292-300.

[9] Erdmann, M. & Lozano-Pèrez, T., *On Multiple Moving Objects*, Algorithmica, **2**, pp. 1419-1424, 1987.

[10] Dewangan, R.K., Shukla, A. & Godfrey, W.W., *Survey on Prioritized Multi Robot Path Planning,* 2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Chennai, pp. 423-428, 2017.

[11] Čáp, M., Novák, P., Kleiner, A. & Selecký, M., *Prioritized Planning Algorithms for Trajectory Coordination of Multiple Mobile Robots*, IEEE

Transactions on Automation Science & Engineering, **12**(3), pp. 835-849, 2015.

[12] Oliveira, M.M., Galdames, J.P.M., Vivaldini, K.T., Magalhães, D.V. & Becker, M., *Battery State Estimation for Applications in Intelligent warehouses*, IEEE Conference on Robotics and Automation, pp. 5511-5516, 2011.

[13] Anton, S.R. & Inman, D.J., *Vibration Energy Harvesting for Unmanned Aerial Vehicles*, Proc Spie, p. 6982, 2008.

[14] Aminzahed, I., Zhang, Y. & Jabbari, M., *Energy Harvesting from a Five-story Building and Investigation of Frequency Effect on Output Power*, International Journal on Interactive Design & Manufacturing, **10**(3), pp. 1-8, 2016.

[15] Mousavi, M. Yap, H.J Musa, S.N., Tahriri, F. & Md Dawal, S.Z., *Multi-Objective AGV Scheduling in an FMS Using a Hybrid of Genetic Algorithm and Particle Swarm Optimization*, Plos One, **12**(3), e0169817, 2017.

[16] Pandey, A., *Mobile Robot Navigation and Obstacle Avoidance Techniques: A Review*, International Journal of Robotics & Automation, **2**(3), pp. 1-12, 2017.

[17] Faraji, H., Hajimirzaalian, H., Farzadpour, F. & Legha, M.M., *A New Hybrid Particle Swarm Optimization Approach for Sizing and Placement Enhancement of Distributed Generation*. Energy and Electrical Drives (POWERENG), 2013 Fourth International Conference on (pp. 1277-1281). IEEE.

[18] Dorigo, M., Di, C.G. & Gambardella, L.M., *Ant Algorithms for Discrete Optimization*, Artificial Life, **5**(2), pp. 137, 1999.

[19] Čáp, M., *Algorithms for Multi-Robot Trajectory Planning in Well-formed Infrastructures*, Association for the Advancement of Artificial Intelligence, pp. 1-5, 2015.

[20] Van Den Berg, J.P. & Overmars, M.H., *Prioritized Motion Planning for Multiple Robots*, IEEE/RSJ International Conference on Intelligent Robots and Systems IEEE, pp. 374-382, 2005.

[21] Cáp, M., Novák, P., Selecký, M., Faigl, J. & Vokffnek, J., *Asynchronous Decentralized Prioritized Planning for Coordination in Multi-robot System*, IEEE/RSJ International Conference on Intelligent Robots and Systems IEEE, pp. 3822-3829, 2013.

[22] Liu, G., Li, T., Peng, Y. & Hou, X., *The Ant Algorithm for Solving Robot Path Planning Problem*, International Conference on Information Technology and Applications, IEEE Computer Society, pp. 25-27, 2005.

[23] Rashid, R., Perumal, N., Elamvazuthi, I., Tageldeen, M.K., Ahamed Khan, M.K.A. & Parasuraman, S., *Mobile Robot Path Planning Using Ant Colony Optimization*, IEEE International Symposium on Robotics and Manufacturing Automation, IEEE, pp. 1-6, 2017.

[24] Dorigo, M. & Gambardella, L.M., *Ant Colony System: a Cooperative Learning Approach to the Traveling Salesman Problem*, IEEE Transactions on Evolutionary Computation, **1**(1), pp. 53-66, 1997.
[25] Stutzle, T. & Hoos, H.H., *MAX-MIN Ant System*, *Future Generation Computer Systems*, The International Journal of Escience, **16**(8), pp. 889-914, 2000.
[26] Elfes, A., *Using Occupancy Grids for Mobile Robot Perception and Navigation*, Computer, **22**(6), pp. 46-57, 2002.
[27] Huang, C., Fei, J., Liu, Y. & Liu, X., *Smooth Path Planning Method Based on Dynamic Feedback A Ant Colony Algorithm*, Transactions of the Chinese Society for Agricultural Machinery, **48**(4), pp. 34-40, 2017.