Intelligent Service Robotics (2021) 14:313–325
https://doi.org/10.1007/s11370-021-00363-w

ORIGINAL RESEARCH PAPER

Coordination of multi-robot path planning for warehouse application using smart approach for identifying destinations

Kaushlendra Sharma[1] · Rajesh Doriya[1]

Received: 1 September 2020 / Accepted: 3 March 2021 / Published online: 27 March 2021

Abstract
Path planning and coordination in a multi-robot system are important and complex tasks in any environment. In a multi-robot system, there can be multiple objectives to be achieved by multiple robots simultaneously. Nowadays, many mobile service robots are being used in warehouses to reduce running costs and overheads. In a large warehouse, there can be multiple robots to handle the number of operations. Planning a path means to find out the optimal route, and coordinating them means a collision-free route. To get both the parameters to reach their optimal level becomes a tedious task to achieve. The efficiency of overall warehouse operation can be improved by adequately addressing the coordination and path planning issues among the robots. In warehouses, each robot has to navigate to its destination by finding a collision-free optimal route in coordination with other robots. In this paper, a comparative study with the acclaimed path planning and coordination has been presented. The proposed smart approach has been presented for a multi-robot system to find a collision-free optimal path in a warehouse to handle storage pods. This paper proposes a smart distance metric-based approach for a multi-robot system to identify their goals smartly and traverse only a minimal path to reach their goal without getting being collided. It uses a smart distance metric-based approach to find the intended path. The proposed work performs better when compared with other works like A* and ILP. It is strictly monitored that there is no collision occurred during execution. Three different instances of a warehouse have been considered to carry out the experiments with parameters such as path length, average path and elapsed time. The experiments with 800 pods and 16 robots report the improvement in performance up to 2.5% and 13% in average path length and elapsed time.

Keywords Multi-robot · Path planning · Coordination · Warehouse

1 Introduction and related work

Multi-robot system is required to coordinate well with each other to carry out a challenging and complex task like managing the goods and services in the warehouse [1–3]. Achieving more than one objective at the same instance and in the shared workspace makes it an uphill task to achieve. Multiple robots require two crucial parameters to get optimized up to their maximum in such cases. The first one is to derive an optimal path for each robot, and the second is to maintain coordination among them. A considerable number of work has been done in the area of path planning for robots, though most of them focus on a single robot, and some of them inherited those concepts into multi-robot systems as well. However, the moment when path planning is applied to the multi-robot system, there exists one more issue which needs to addresses thoroughly, which is known as coordination. There are also some work present which has been proposed by the researchers regarding the coordination scheme [4]. Combining these two issues parallel requires a smart approach to achieve synchronization between these two parameters. Here, in this section, we present a detailed definition of all the essential aspects related to multi-robot path planning and coordination. Also, this section covers some notable contribution made and a literature survey done in this regard.

1.1 Path planning

Due to the working of the path planning algorithm and its behavioural complexity, it is considered as the NP-hard

✉ Kaushlendra Sharma
kaushlendra84@gmail.com

[1] National Institute of Technology Raipur, Raipur, India

problem [5]. Initially, the path planning was done for the robotics arm only for some manufacturing purposes in industries. Later on, with the wide acceptability of robotics in multiple sectors, different types of robots were brought into use for serving the task, which leads to an increase in the work related to path planning algorithms. There exist a long history of path planning algorithms. Starting from the Heuristic approach to solve the path planning problem to the evolutionary and hybrid algorithm to find an optimal path from source to destination, while the robot tends to traverse in the given configuration space. The definition of path planning is to plan a collision-free route from source to destination, while the goal of different path planning algorithm is to make that path optimum and safe. Several new technologies were popping up with each passing days. A good number of optimization techniques and algorithms has been on the top for a few years. Several researchers have consumed those concepts for solving the path planning problem for robots. Some of the latest entrants in the queue are, memetic algorithm [6], genetic and adaptive fuzzy control [7], delayed PSO algorithm [8], beetle antennae search algorithm [9], Smoothed A* [10]. There are also many advanced versions of the A*, RRT, PRM algorithm proposed by the researchers to solve the path planning problem in multiple domains [11] [12]. These algorithms can also be considered as the core algorithms for solving robot path planning. Path planning again can be studied in terms of its environmental set-up like static, dynamic, two-dimensional, three-dimensional and many other ways. To work in a dynamic environment, path planning is done by considering virtual static obstacle to avoid collisions. Each time the path planning is done considering the time parameter, the updated virtual position of an obstacle is considered. Though that fall separately into other zones, that's why we do not include the discussion of that part.

1.2 Multi-robot system

A multi-robot system is used to accomplish a complex task. A single robot is capable of handling most of the simplified jobs. Multi-robot is required when multiple objectives need to be fulfilled simultaneously, which makes it complicated. Application like dynamic mission planning [13], collective construction [14], multiple task allocation [15], mapping environmental variables [16] and many other applications have used multiple robots in the past. It requires an entirely different architecture to work upon. Each robot is assigned a unique goal to satisfy; every single robot has to traverse an optimal path for reaching their respective destination from the start point. Managing multiple robots simultaneously in a shared workspace makes it more dynamic and complex. Dynamic in the sense that each other robot is an obstacle for every other robot, apart from the static obstacles already present in the workspace. Thus any conventional path planning algorithm has to behave differently for finding an optimal path for each robot due to the continuous change in the environment. The second thing which makes it complicated is that it is required in the working architecture to imbibe a coordination mechanism to manage all the robots working in that workspace [17]. Thus, while dealing with the problem of the multi-robot system, it becomes mandatory to deal with the two issues simultaneously, the first one being the path planning algorithm to find an optimal path for each robot as well as ensuring the optimal average path of all the robots working in the shared workspace. The second issue to deal with is having an effective coordination approach such that it makes all the robots to move on their optimal path and that too, without any collision.

1.3 Coordination

Coordination is required when there is more than one task to instate. The multi-robot scenario is one among it [18]. The basic principle of coordination is to avoid collision among the robots while traversing, and after doing so, the second most crucial thing to manage is whether they are centralized or decentralized coordinated [19]. In a centralized scheme [20], there is a supervisor robot which synthesis all the robots and performs all the required calculation for safe execution [21]. While in a decentralized scheme [22, 23] all the robots perform individually and distribute their information to other robots [24]. Comparing it with the centralized scheme, it is much more flexible and reliable in adapting to the environment but is more expensive in terms of the cost it incurs. There are also some applications that uses the hybrid approach [25, 26] by combining both the centralized and decentralized methods of coordination. There are multiple techniques [27, 28] proposed by the researchers to coordinate the multiple robots in the given workspace. There are some more additional issues which are required to handle and manage simultaneously. The issues like how the robots are made to traverse by keeping a minimal distance with other robots as increasing the distance between them will affect their optimal paths. In the same way, it is also required to take care of the speed of each robot with which they are moving. It should also be maintained at its optimal level, as increasing or decreasing from their optimal speed will lead to safety concerns and may make the whole workspace an unsafe environment.

1.4 Related work

Here, in this subsection, we have analyzed some of the prominent work in the area of path planning and coordination. A short discussion is presented below, which helps in indulging into the core working of a multi-robot system.

Xinye et al. in the [29] paper presents the problem of path planning for a group of mobile robots with multiple targets as multi-travelling salesman problem (MTSP) with one or more depots. A bi-objective ant colony optimization (ACO) algorithmic approach, which is based on the memetic algorithm, is presented to solve the problem. So the essential requirement is to ensure that every robot must visit at least one target, and each target is visited once by one of the robots. In addition to this, the simultaneous optimization of the total path length and maximum path length of the robots are to be achieved. The approach employees ACO based on a memetic algorithm that uses local search and integrates it with sequential neighborhood descent for optimization [30]. The experimental results of this approach tested in a static environment are compared with other classical algorithms, which show that the proposed approach gives better results.

Rami et al. paper [31] describes an approach based on probabilistic neuro-fuzzy logic. This approach involves two layers of fuzzy architectures and based on the leader-follower scenario. First is the probabilistic fuzzy control that takes care of uncertainties and errors to avoid disturbance in path traversal and approximates the position of the robot relatively. The neuro-fuzzy Inference system layer is designed to establish a leader-follower movement among the robots. The problem formulation is depicted using fuzzy architecture diagrams. The neuro-fuzzy inference system acts as a navigation controller for each robot. Its input coordinates and the orientation of the robot, as well as the output, are the linear speed and angular speed of the robot. The experimental results of this approach are obtained by simulation on different environments; the leader and follower movements are observed in the experiment. The future scope involves testing the current approach in a real complex environment and to develop a technique to improve the performance and to reduce the complexity.

Jianjun et al. in the paper [32] mainly focuses on improving the path planning for a multi-robot system using the genetic algorithm (GA), which is achieved by using a memetic approach. The proposed improved memetic algorithm-based (IMA) approach involves in implementing GA with variable length chromosome using the two-point crossover and bacterial mutation operations, which avoids the optimum local problem. A search strategy that combines local neighbors search with disorder strategy is used to improves the overall convergence rate. Furthermore, an approach to deal with multi-robot path planning in a dynamic environment is also proposed. The experimental results obtained by real-time testing and simulations are specified that shows the significant improvement in the results in comparison with the general genetic and memetic approach.

Ebtehal et al. paper [33] aims at developing a complete and optimal solution for multi-robot path planning problem. The extant push and spin (PASp) algorithm ensure providing a complete solution to the multi-robot path planning problem. An improved push and spin (PASp+) algorithm are devised to obtain an optimal path using the standard PASp algorithm two ways were adapted, which include the use of smooth operation that eliminates redundant moves in the path, and the other involves usage of heuristic search to explore other available paths. The PASp algorithm involves the usage of push and spin operations when two robots are detected simultaneously at a common vertex position along their paths. Instead of spin operation, a heuristic value comparison with other available paths is done, and the smooth operation is applied at the end on the list of solutions obtained. The proposed approach was tested and compared with the standard algorithms, which shows that the proposed approach is better in performance.

Reducing down the related work, this paper focuses on the warehouse applications implementing a multi-robot system. In context to that a small survey is presented in Table 1. Warehouse application is one of the areas which is best utilizing the multi-robot system to serve and fulfil the requirements. It is again found that most of the solution proposed is centralized or distributed in nature.

2 Problem definition

The problem is set up in a static warehouse environment with multiple robots whose initial configurations are defined, and a set of goal positions are given. So the problem can here be viewed as optimal target assignment to the robots followed by path planning to develop a collision-free path for each robot to reach the target. In a multi-robot system, path planning requires coordination to obtain a feasible collision-free path in minimum moves for the robot. The static warehouse environment here consists of static obstacles, which are a non-traversable region for the robots, and the remaining free space region is traversable are shown in Fig. 1 The necessary information required for path planning involves target position to be reached, so target assignment plays a key role here.

In general, the multi-robot path planning system has a defined mapping of a target for each robot to reach the target. So target assignment is an additional task that effects the final path obtained. The primary objective in target assignment is to assign targets such that future paths obtained have minimum path length and computation time. Using random permutations for target assignment result in considerable path length and computation time. So a proper approach with prior consideration of path lengths as a factor is desirable.

Consider a set of n robots in a multi-robot system where each robot is represented as R_i where i =1, 2,..., n indicates

Table 1 A brief survey on the multi-robot path planning algorithms and coordination techniques for different applications

Author	Path Planning	Coord. Tech. C	Coord. Tech. D	Application	Parameters	Year	R.I.	Ref
Warren et al.	Artificial PF	✓	No	Static Envi.	Path length	1990	281	[34]
Svestka et al.	PRM Algo	No	✓	Car Parking	Time Factor	1995	209	[35]
Petr et al.	PRM Algo	✓	No	Robot Car	Comp. Time	1998	303	[36]
Burgard et al.	Probabilistic	✓	No	Unkown Envi	Target Sele	2000	932	[37]
Bennewitz et al.	Hill Climbing	✓	No	Real Robots	Path Length	2001	194	[38]
Sanchez et al.	PRM Planner	✓	No	Welding Station	Completeness	2002	188	[39]
Kalra et al.	Hoplites Algo	No	✓	Mult Agent Plan	Coordination	2005	179	[40]
Wurm et al.	Segmentation	No	✓	Unknown Envi.	Exp.Time	2008	248	[41]
Hollinger et al.	Approximation	No	✓	Indoor Envi	Capture Time	2009	159	[42]
Van et al.	Decoupling	✓	No	Unknown	Running Time	2009	145	[43]
Hollinger et al.	Online Algo.	✓	No	Simulation	Comp. Cost	2010	86	[44]
Ding et al.	Optimaility	✓	No	UAV Convoy	Path Time	2010	127	[45]
Wagner et al.	M* StarAlgo	No	✓	Warehouse	Min Cost Path	2011	149	[46]
Michael et al.	Any-Com Algo	No	✓	Dynamic Teams	Completeness	2014	17	[47]
Contreras et al.	ABC-EP Algo	✓	No	2D-Map	Path Length	2015	140	[48]
Huijiao et al.	Auction Algo	No	✓	Dynamic Envi.	Task Alloc.	2015	08	[49]
Glenn et al.	M* Star Algo	✓	No	Subdimensional	Min Cost Path	2015	160	[50]
Claes et al.	Monte Carlo	No	✓	Static Envir.	Planning Time	2017	28	[51]
Araki et al.	Priority Plan	No	✓	Swarm Robots	Motion Control	2017	29	[52]
Bolu et al.	Modified A*	No	✓	Smart Warehouse	Completion Time	2019	02	[53]
Kumar et al.	GA and A*	✓	No	Warehouse	Order Picking	2018	15	[54]
Han et al.	Heuristic	No	✓	Warehouse Env.	Comp. Time	2019	05	[30]

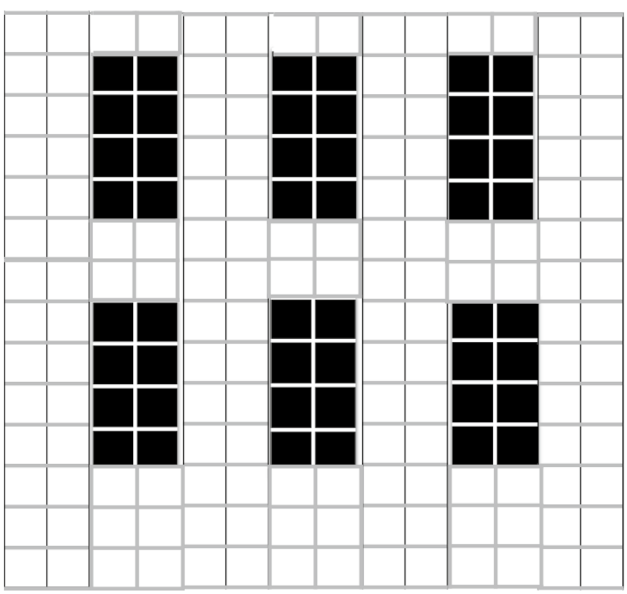

Fig. 1 A sample warehouse map where the black cell represents the storage place and are also the destination for robots, whereas the white cell represents the free space where traversal is possible

the robot id. A matrix denoting mapping between robots and goals D. Configuration of robot R_i at time t is given as per the Eq. (1).

$$C_i(t) = (x_i, y_i, t) \tag{1}$$

Now, the collision between two robots R_i and R_j is defined as per Eq. (2).

$$\begin{aligned} &\text{Collision}(i,j) \rightarrow \exists t : C_i(t) = C_j(t) \text{ where } i \neq j \\ &\quad \text{i.e,} \rightarrow \exists t : (x_i, y_i, t) = (x_i, y_i, t) \text{ where } i \neq j \end{aligned} \tag{2}$$

To obtain globally optimal paths, we define a parameter average path length(μ_{pl}) mathematically in 3.

$$\mu_{pl} = \sum_{i=1,j=1}^{n} p_{ij} \tag{3}$$

where n is the number of robots in the system and p_{ij} denotes the length of the path traced from Robot $R_{(i)}$ to the goal G_j . The objective is to determine a collision-free and path with minimal path length for each robot in the system to reach the goal state. These two conditions are to be satisfied to ensure the solution is optimal, which can be mathematically represented, as mentioned in the following equation. Firstly, collision-free implies 4.

$$\text{Collision }(i,k) = \phi \ \forall\, R_i, R_k \text{ where, } i \neq k \tag{4}$$

Which states that all the pairs of robots whose paths involved in a collision should be empty. Secondly, μ_{pl} should be minimum. Consider the set of optimal paths (S) to all the goals in the system is given by the following 5

$$S = [P_{ij,\dots}]\ \forall i,j \text{ where } D_{i,j} = 1 \tag{5}$$

And, min (μ_{pl}) and collision $(i,k)\ = \phi$ for each pair pair of R_i, R_k robots in the system

3 Proposed solution

This section presents the description of the proposed path planning method for the multi-robot environment and the process of their coordination. Here in this paper, we intend to propose an improved smart selection of pairs to find the nearest best possible destination for each Robot Ri. This improved smart selection has an advantage over primitive selection as primitive selection uses the traditional distance metrics, which does not consider obstacles while the former one will consider obstacles. Figures 2 and 3 represent the pictorial description of the robot's smart selection mechanism. Subfigures (i–iv) of Fig. 2 represents the incremental instances of the existing problem. Figure 2i is showing the positions of all the robots and destinations in the workspace. The block adjacent to it is showing that all the robots present in the workspace are connected to the destination. After that, their respective distance is calculated. Then in the next stage, the distance of all the robots from all the available destination is calculated. After that, the pairing is done, which means each robot is selected for a specific destination to traverse. This is done by selecting the minimum distance of each robot from their destination. Figure 2v shows the major limitation of this existing approach which states that while calculating the distance, it does not consider the obstacle present in between the robot and destination. In Fig. 3, the improvised method of solving this approach is depicted. Here, also subfigures of Fig. 3i–iv depicts the different instances of the proposed approach, and subfigure Fig. 3v shows the solution of the proposed approach. For calculating the distance of each robot from the destination, intersection points have been taken into consideration. Intersection points are the position of two points of the flip side of the obstacle, which helps in calculating the actual distance considering the presence of obstacle lying in between the robot and destination. The distance ratio helps in finding the destination for each robot is shown in Fig. 6. This ratio helps in finding the optimum distance to be travelled by overcoming the first selection, which does not consider the presence of an obstacle while calculating the distance. The distance ratio α helps in finding the collision-free optimal path. The proposed concept is then applied to solve the problem of warehouse application having multiple robots sharing the same workspace and having multiple destinations.

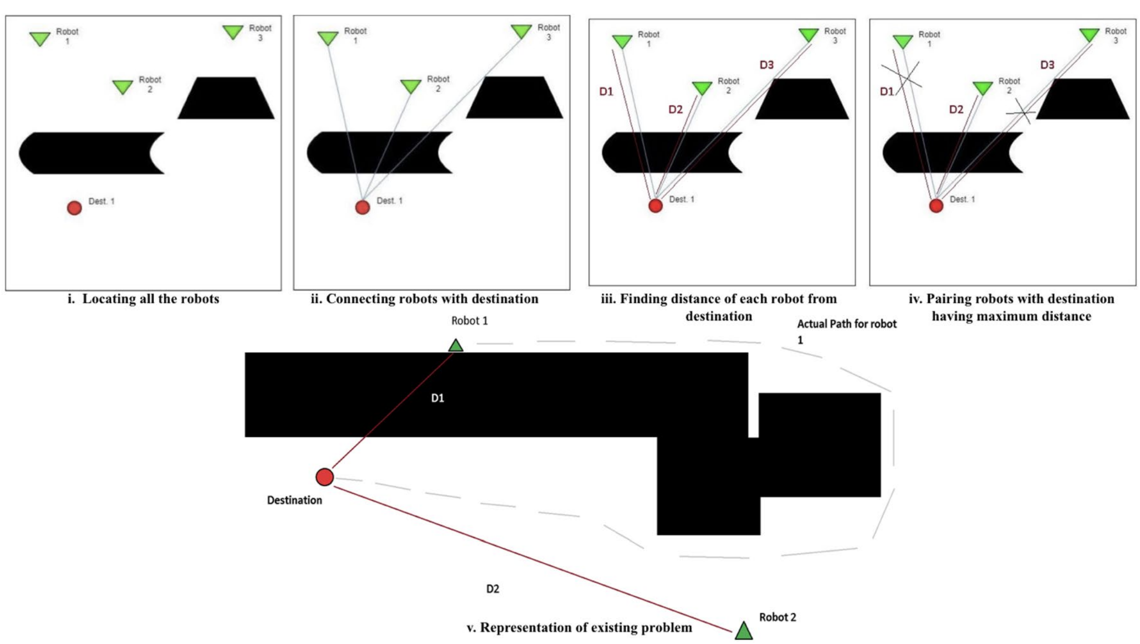

Fig. 2 Different instances of the execution and representing the existing problem of the system

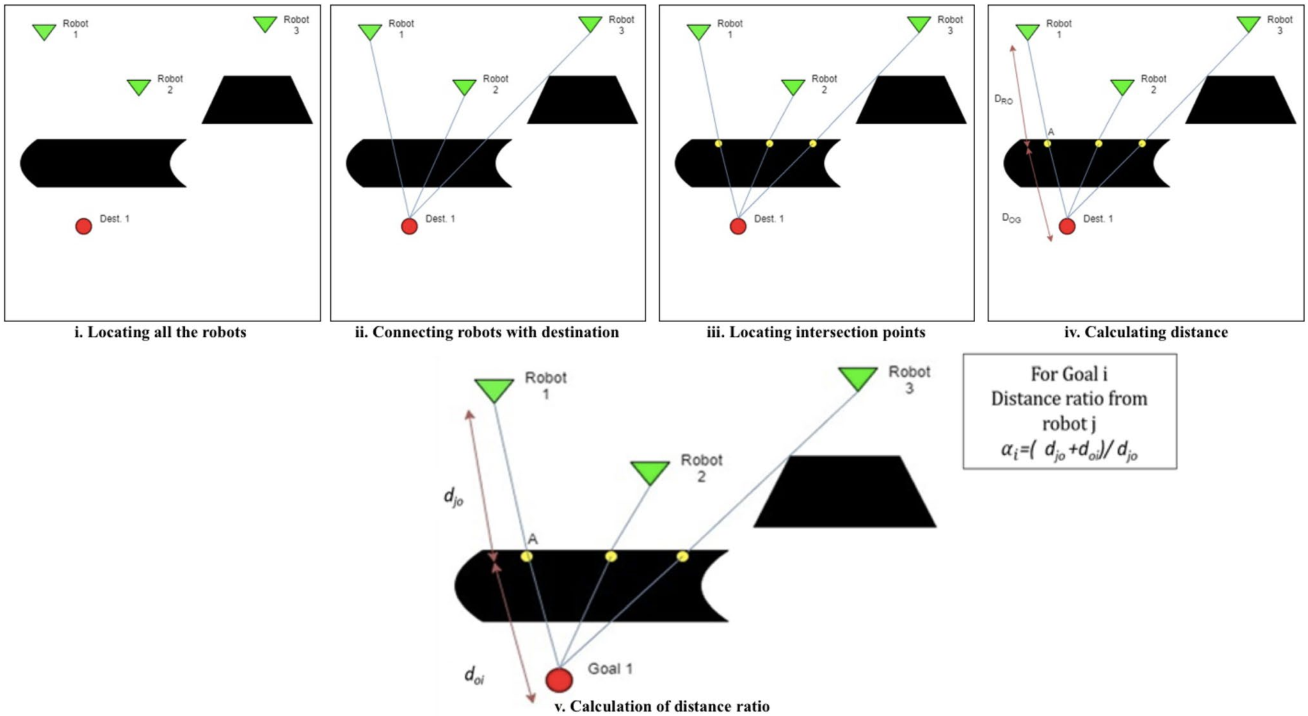

Fig. 3 Different instances of the execution and showing the distance calculation D_{RO} and D_{OG} and the calculation of distance ration α of the proposed approach

3.1 Proposed path planning method

The key steps involved in the solution is first to convert the environment map into binary form. Then, the second step involves the generation of source–destination pairs using the distance ratio for target assignment. The following step involves the usage of the distance metric approach for path planning ensuring collision-free paths are obtained. Here, the mutual collisions among robots are handled by collision avoidance through the reservation table. Initially, the multi-robot environment is generated. A reference map is given as the input to find the path, which is a static warehouse environment map. On conversion, the image obtained contains the black pixels indicate the storage pods that are non-traversable and white pixels denoting traversable region. The start positions are defined for each robot, and the smart selection algorithm is used to assign goals or targets that improve path planning in terms of path length. A goal mapping matrix $'D$ is obtained as a result of smart selection performed that is used for path planning further. According to the smart selection approach, for every goal G_i we will be having many distance ratios α_i. The pair having the smallest value of α will be selected, as the selected robot will be having maximum distance from the obstacle and minimum from the goal position. Thus, it gives a better estimation of free space to be traversed in the path. The process of target assignment in a static warehouse environment is done by estimating the cost of the path. For each goal from the available robots, this cost is obtained by summation of total path length and collisions across the planned path. The goal is assigned to the robot with minimum cost, and the process is repeated for all the goals in the system. A scenario that elucidates the target assignment is shown in Figs. 2 and 3.

Algorithm 1 Path Planning

Initialize S; // Start configurations
Initialize G; // Goal configurations
Initialize $\alpha_i = \infty \ \forall$ i to n // Distance ratio of each goal set to infinity
For each G_i in goal :
For each S_i of robot :
$curr - val = (d_{oj} + d_{oi})/d_{jo}$;
if $curr - val < \alpha_i$ **then**
  α_i = min (α_i , curr-val)
end if
r_{id} = j;
$D[r_{id}]$ [i] =1 ; //matrix value that maps robot with goal

3.2 Proposed coordination method

To avoid robots colliding with other robots and also from the obstacles present in the warehouse, the proposed approach uses the two essential things into consideration to prevent the collision. The first thing is to avoid inter clash of robots with each other. Inter clash of robots is achieved using thresholding function as defined in 2. There are several other coordination techniques described by Yan et al. [21]. Another issue to look after is to avoid the obstacle present in the given workspace. The smart selection technique proposed by the paper handles this issue. An improved smart selection of pairs to find the nearest best possible destination for each robot R_i is applied. This Improved smart selection has an advantage over primitive selection as primitive selection uses the traditional distance metrics, which does not consider obstacles. At the same time, the former will find obstacles. Some more survey on the same issue is presented [55, 56].

3.3 Target assignment

The process of target assignment in a static warehouse environment is done by estimating the cost of the path for each goal from the available robots this cost is obtained by summation of total path length and collisions across the planned path. The goal is assigned to the robot with minimum cost, and the process is repeated for all the goals in the system. A scenario that elucidates target assignment is shown in Fig. 4, which depicts the warehouse multi-robot environment. It consists of robots initial position, goal positions, storage pods and depot pods. Figure 5 depicts that p_{11} is the path from robot R_1 to to G_1 with no collision and having minimum path. Similarly, p_{21} is the path from robot R_2 to G_1 with no collision and p_{31} is the path from robot R_3 to G_1 with no collision. A distance metric approach is used to

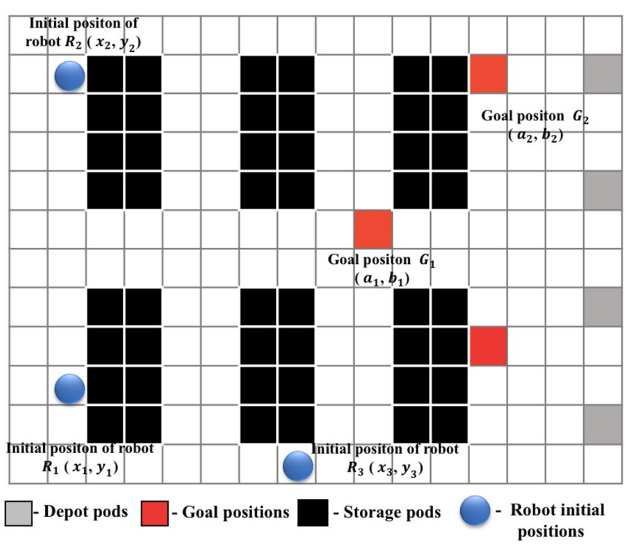

Fig. 4 A pictorial representation of multi-robot system in the warehouse environment

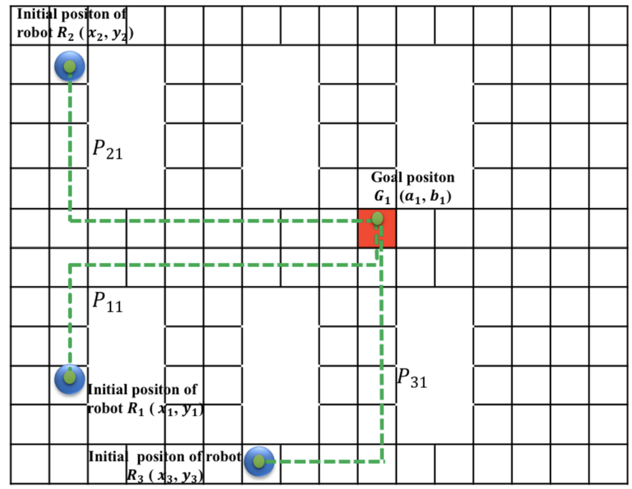

Fig. 5 Target assignment for a goal in the warehouse application

traverses the complete given input image between these two coordinates for finding the best path.

4 Results and discussion

The E-Commerce industry is one of the sectors which is continuously expanding at a very rapid pace, and to cope up with this requirement, the role of their warehouse becomes most important. They are deploying robots to meet immediate and accurate demands. KIVA [57, 58] system is one of the excellent examples of having an intelligent warehouse. This paper has utilized the proposed concept of a smart approach in achieving coordinated path planning in a warehouse application. Usually, in any intelligent warehouse application, after each robot receives the corresponding order task, it needs to plan the path from the current position to the target pod and then from the pod position to the working station. These paths are required to not conflict with

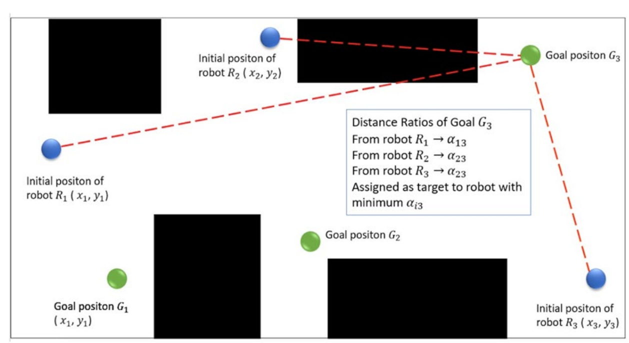

Fig. 6 Figure representing the distance ratio of different robots from a goal

the paths of other robots, and there is no deadlock between the robots during driving.

The proposed approach is applied in a static warehouse environment map with a group of 15 robots and 100 storage pods in the system. In the implementation, the robot is assumed to be a unit sized object and experimental results obtained are displayed in Tables 2, 3, 4, and 5 The execution snap is shown in Figs. 7 and 8. The results obtained by applying the algorithm on a different multi-robot system by varying the number of robots and pods. The results obtained are evaluated using average path length and total computation time elapsed as the parameters.

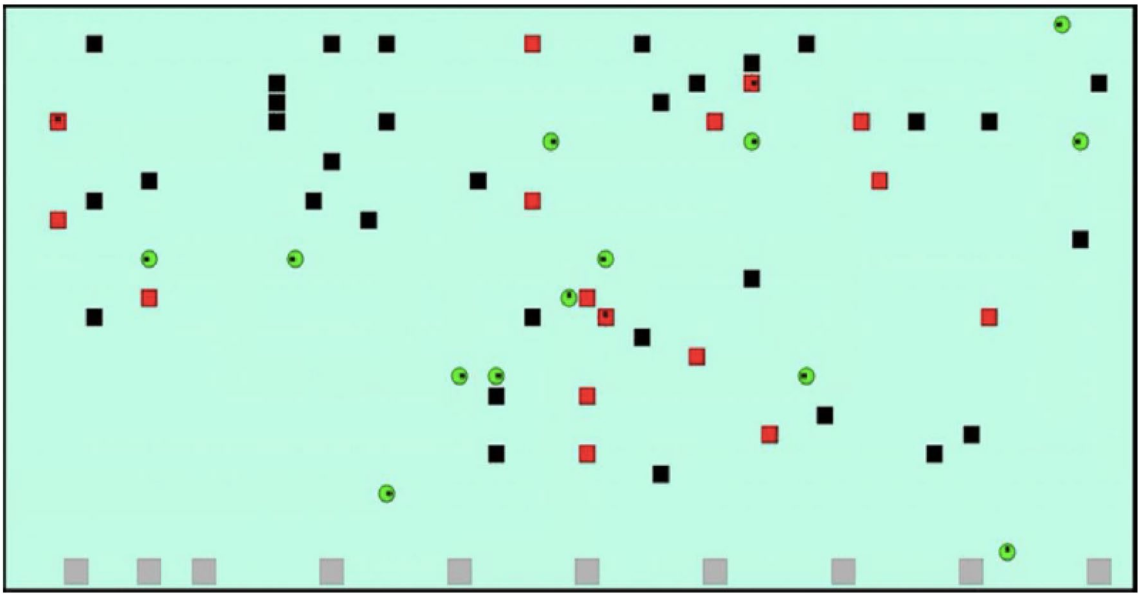

Fig. 7 Simulation sneak peak of warehouse application having 48 pods and 16 robots approaching to reach their respective destinations, green dots in the picture are the robots moving in the workspace

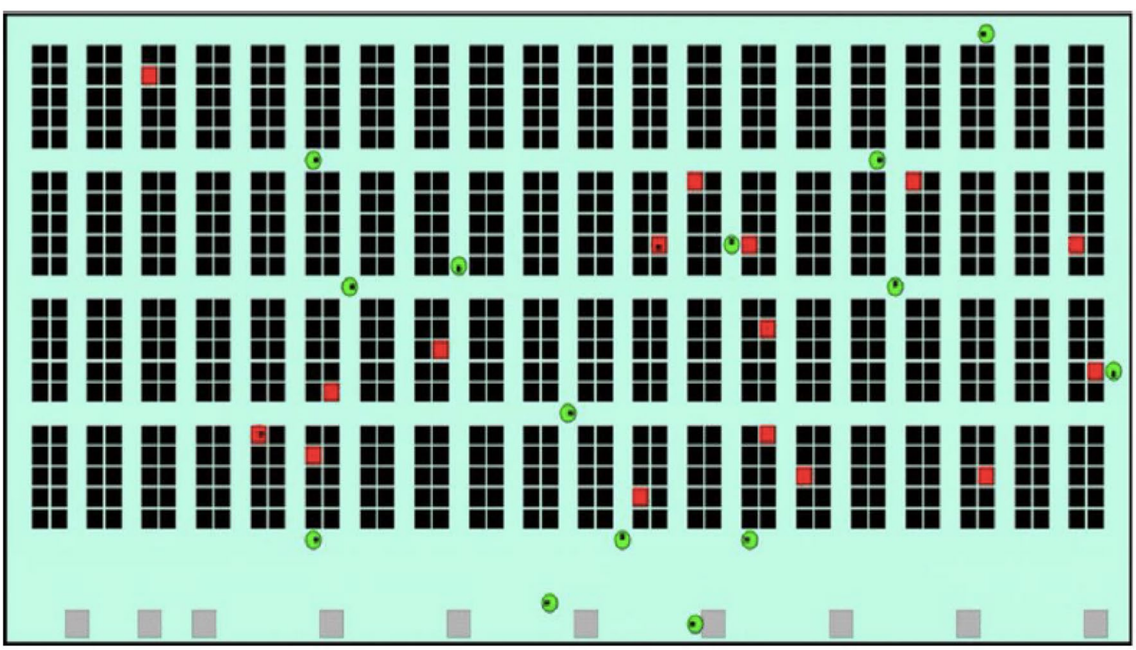

Fig. 8 Simulation sneak peek of warehouse application having 800 pods and 16 robots moving in the warehouse workspace and approaching to reach their respective destinations

Table 2 Experimental results obtained for multi-robot system in warehouse environment with 1-to-1 Pod using proposed approach

S.no	No. of pods	No. of robots	Source(S)	Destination(S)	Path Length	Elapsed time (Sec)	A* Algo (Sec)	ILP Algo (Sec)
1	1	1	(36, 36)	(45, 45)	18	0.363433	0.4179	0.4070
2	2	2	(17, 40) (29, 11)	(10, 54) (14, 44)	21, 36	2.369620	2.7250	2.6539
3	4	4	(27, 13) (11, 49)	(8, 8) (17, 23)	24, 32	4.941293	5.6823	5.5341
			(25, 46) (19, 6)	(8, 2) (15, 15)	61, 13			
4	8	8	(21, 22) (29, 16)	(5, 50) (18, 44)	44, 39	8.208342	9.4395	9.1932
			(27, 30) (25, 36)	(5, 3) (24, 27)	49, 10			
			(7, 35) (20, 19)	(10, 41) (22, 35)	09, 18			
			(13, 16) (9, 58)	(4, 47) (10, 2)	42, 57			
5	16	16	(7, 6) (3, 46)	(20, 30) (8, 29)	37, 22	17.501243	20.1264	19.6013
			(13, 25) (29, 53)	(6, 50) (24, 47)	32, 11			
			(21, 43) (29, 43)	(20, 48) (9, 39)	06, 24			
			(22, 61) (19, 13)	(2, 27) (24, 53)	64, 45			
			(4, 46) (25, 6)	(21, 15) (23, 50)	48, 46			
			(19, 48) (12, 22)	(2, 50) (5, 39)	19, 24			
			(5, 40) (25, 12)	(17, 18) (5, 44)	34, 52			
			(12, 37) (3, 52)	(17, 54) (5, 17)	22, 37			

Table 3 Experimental results obtained for multi-robot system in warehouse environment with doublings pods using proposed approach

S.no	No. of pods	No. of robots	Source(S)	Destination(S)	PathLength	Elapsed time (Sec)	A* Algo (Sec)	ILP Algo (Sec)
1	2	1	(19, 60)	(11, 45)	23	0.298725	0.3435	0.3345
2	4	2	(4, 34,) (26, 43)	(9, 56) (15, 18)	27, 36	0.664236	0.7638	0.7439
3	8	4	(28, 28) (8, 16)	(14, 21) (5, 2)	21, 17	0.919797	1.0576	1.0300
			(2, 13) (26, 38)	(9, 21)(21, 42)	15, 09			
4	16	8	(1, 39) (8, 40)	(17, 3) (17, 11)	52, 38	17.594425	20.2335	19.7057
			(10, 31) (7, 50)	(4, 11) (10,32)	26, 21			
			(27, 53) (25, 11)	(20, 32) (24, 51)	28, 41			
			(23, 4) (11, 43)	(21, 44) (5, 57)	42, 20			
5	32	16	(19, 46) (11, 58)	(9, 56) (5, 5)	20, 59	79.338232	91.2389	88.8587
			(14, 31) (25, 56)	(6, 12) (17, 42)	27, 22			
			(9, 19) (5, 31)	(15, 14) (8, 54)	11, 26			
			(25, 47) (20, 43)	(22, 26) (6, 39)	24, 18			
			(29, 18) (25, 58)	(17, 54) (21, 57)	48, 05			
			(19, 1) (27, 44)	(10, 54) (6, 56)	62, 29			
			(3, 61) (7, 17)	(18, 2) (3, 57)	74, 44			
			(25, 8) (26, 11)	(22, 53) (21, 42)	48, 36			

A comparison of the proposed approach with the A* algorithm when implemented on the warehouse system with 800 pods and 16 robots is done. A graphical representation indicating a comparison of average path length and computation time is shown in Tables 6 and 7 and in Figs. 9 and 10. On comparing average path length, the proposed approach shows little better performance of up to 2.5 % while in terms of computation time, a useful improvement of around 13 % is found. The elapsed time factor of the proposed approach is compared with two other approaches in Table 2, 3, 4, and 5. The first approach which is being compared is A* in association with the GA (genetic algorithm), and another one is the ILP (integration linear programming) method. ILP is an integer linear programming method, which appears to be one of the fastest optimal solver's. Here, the hybrid combination of A* & GA is considered; GA is used for task allocation, while A* was used for finding the collision-free optimal path. ăThe proposed work results better in terms of elapsed time.

In a large-scale warehouse environment, the position of the pod often changes since the replenishment tasks and the picking tasks are continuously in progress. The pods placed

Table 4 Experimental results obtained for multi-robot system in warehouse environment with tripling pod using proposed approach

S.no	No. of pods	No. of robots	Source(S)	Destination(S)	Path length	Elapsed time (Sec)	A* Algo (Sec)	ILP Algo (Sec)
1	3	1	(27, 39)	(24,59)	23	0.545130	0.6268	0.6105
2	6	2	(21, 19) (27,13)	(23, 60) (17, 6)	43, 17	3.711996	4.2686	4.1573
3	12	4	(27, 56) (19, 25)	(24, 57) (16, 3)	4, 25	11.287250	12.9802	12.6416
			(13, 38) (20, 13)	(15, 3) (17, 53)	37, 43			
4	24	8	(29, 57) (7, 32)	(5, 23) (14, 9)	48, 30	17.132150	19.7019	19.1879
			(13, 28) (3, 49)	(11, 26) (8, 12)	4, 42			
			(11, 28) (6, 25)	(20, 15) (4, 33)	22, 10			
			(6, 61) (26, 29)	(6, 42) (17, 3)	19, 35			
5	48	16	(14, 13) (2, 46)	(23, 32) (18, 38)	28, 24	57.878143	66.5598	64.8234
			(17, 22) (23, 61)	(22, 42) (6, 47)	25, 31			
			(18, 46) (18, 22)	(10, 29) (11, 3)	25, 26			
			(29, 37) (7, 20)	(16, 54) (9, 48)	30, 30			
			(20, 58) (28, 26)	(2, 29) (15, 32)	47, 19			
			(1, 26) (16, 43)	(4, 41) (16, 33)	18, 10			
			(20, 25) (25, 1)	(6, 39) (20, 32)	28, 36			
			(15, 34) (4, 16)	(15, 8) (6, 3)	26, 15			

Table 5 Experimental results obtained for multi-robot system in warehouse environment with 800 pod using proposed approach

S.no	No. of pods	No. of robots	Source(S)	Destination(S)	PathLength	Elapsed time (Sec)	A* Algo (Sec)	ILP Algo (Sec)
1	800	1	(20, 49)	(15, 5)	49	1.395674	1.6049	1.5630
2	800	2	(10, 43) (28, 59)	(21, 5) (22,3)	49, 62	3.421114	3.9342	3.8316
3	800	4	(18, 61) (27, 25)	(8, 35) (8, 41)	36, 35	7.951920	9.1446	8.9061
			(29, 17) (6, 61)	(15, 14) (18, 6)	17, 67			
4	800	8	(19, 42) (26, 16)	(12, 56) (9, 12)	21, 21	11.1265	12.7954	12.4616
			(26, 43) (12, 1)	(17, 17) (15, 51)	35, 53			
			(15, 43) (19, 20)	(12, 23) (15, 48)	23, 32			
			(23, 16) (29, 53)	(24, 20) (10, 39)	5, 33			
5	800	16	(24,58) (5, 61)	(23, 35) (3, 8)	24, 55	75.545151	86.8768	84.6105
			(28, 48) (13, 1)	(21, 17) (22, 44)	38, 52			
			(2, 37) (7, 12)	(11, 36) (20, 14)	10, 15			
			(7, 30) (1, 60)	(11, 59) (17, 60)	33, 16			
			(1, 31) (11, 28)	(16, 24) (11, 41)	22, 13			
			(29, 57) (24, 1)	(8, 38) (15, 42)	40, 50			
			(3, 4) (17, 16)	(22, 54) (20, 42)	69, 39			
			(28, 52) (26, 50)	(8, 50) (18, 18)	22, 40			

Table 6 Multi-robot system using proposed approach in warehouse environment with a set of 16 robots and varying the number of pods

Warehouse Specification	Comp. Time	Average Path
48 pods and 16 robots	37.577	28.32
800 pods and 16 robots	165.545	32.34

Table 7 Experimental results obtained for Multi-robot system using A* algorithm and proposed approach in warehouse environment with a set of 16 robots and 800 pods

Name of Algorithm	Comp. Time	Average Path
A* Algorithm	190.79	33.1875
Proposed Approach	165.545	32.34

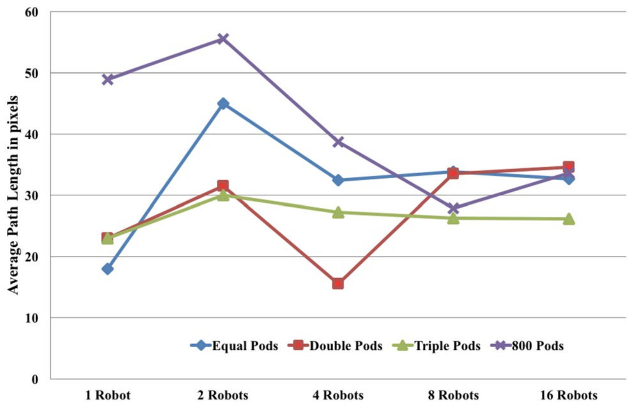

Fig. 9 Average path length variations graph of four different multi-robot systems with varying pods in number

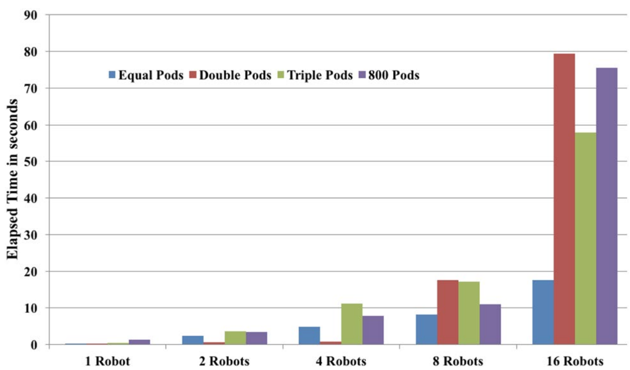

Fig. 10 Elapsed time variations graph of four different multi-robot systems with varying pods in number

in the storage area are transported by robots to the replenishment stations or picking stations for corresponding operations. Thus, the original storage position of the pod can be free to travel by other robots. The thresholding function will avoid any obstacle in the path. The proposed distance metric based smart approach of self-identifying the destination will help each robot in making the system more efficient.

Figures 7 and 8 display the picture of the experiment carried out for a multi-robot system on warehouse application. The green color symbol in the figure shows the robot moving inside the warehouse, the black square box is showing the pod, and the red square box is the destination where the robot is destined to reach.

5 Conclusion and future work

This paper addressed the issue of path planning with coordination in a multi-robot system. Both the problem is quite critical in its own. Combining these two to solve simultaneously becomes an arduous task to carry-on. Only coordination can be achieved in a much more straightforward way by keeping aside the issue of path length. Similarly, by flipping the issue, optimizing the path length of individual robots can be done if there are no constraints for robots not to collide with each other or share their sections of the path with other robots. However, in a multi-robot system, these two problems are adjacent to each other and are required to deal parallely to automate any application efficiently. Each parameter is needed to handle acutely while optimizing the other. Here, in this paper, we have taken the warehouse application to solve the problem. The proposed approach of smart selection is applied to observe and validate the results.

The primary objective is to find an optimal, collision-free path using smart selection for target assignment and distance metric path planning and coordination. A custom algorithm for multi-robot path planning in a static warehouse environment is created, which facilitates completeness and coordination for the system. The algorithm is further modified to generate source and destination pairs out of given configurations based on distance ratio values and generate path accordingly, maintaining coordination. The proposed approach shows the improved results experimented in two different conditions of warehouse application. The warehouse application having 48 pods along with 16 robots, and another scenario with 800 pods along with 16 robots are considered and tested. Computation time and average path length are observed, which showed better performance and is also compared with the A* algorithm and ILP method in a tabular form in the result section.

Future prospects include path planning and coordination in a more complex environment. The proposed approach considers the distance ratio using a single dimension. While in the case of a map with a larger obstacle having width more than length. Then, this approach would fail to find the optimal robot target pair. The scenario considered to solve the path planning problem is a static warehouse, while in real-time, warehouses are seldom static. There's an ever-changing aspect to any regular warehouse, and this can be reflected in future work. Also, the workspace is restricted to a limit of 800 pods which is still smaller than an average warehouse limit in real-time. In future, efforts can be extended to include larger warehouses which require higher computing and processing power. This leads to a more significant problem, equipping each robot with a processor powerful enough to simulate the real-time warehouse environment and use it for the computation required. Thus, moving to a centralized cloud system would be pragmatic in the coming time and would be the future extension of this work.

References

1. D'Andrea R (2012) Guest editorial: a revolution in the warehouse: a retrospective on kiva systems and the grand challenges ahead. IEEE Trans Autom Sci Eng 9(4):638–639
2. Ng M-K, Chong Y-W, Ko K-M, Park Y-H, Leau Y-B (2020) Adaptive path finding algorithm in dynamic environment for warehouse robot. Neural Comput Appl 32(17):13155–13171
3. Yang Y, Zhang J, Liu Y, Song X (2019) Multi-agv collision avoidance path optimization for unmanned warehouse based on improved ant colony algorithm. International conference on bio-inspired computing: theories and applications. Springer, New York, pp 527–537
4. Wen S, Zheng W, Zhu J, Li X, Chen S (2011) Elman fuzzy adaptive control for obstacle avoidance of mobile robots using hybrid force/position incorporation. IEEE Trans Syst Man Cybern Part C Appl Rev 42(4):603–608
5. Canny J (1988) The complexity of robot motion planning. MIT press, Cambridge
6. Cho D-H, Jang D-S, Choi H-L (2020) Memetic algorithm-based path generation for multiple dubins vehicles performing remote tasks. Int J Syst Sci 51(4):608–630
7. Bakdi A, Hentout A, Boutami H, Maoudj A, Hachour O, Bouzouia B (2017) Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. Robot Auton Syst 89:95–109
8. Song B, Wang Z, Zou L (2017) On global smooth path planning for mobile robots using a novel multimodal delayed pso algorithm. Cogn Comput 9(1):5–17
9. Wu Q, Lin H, Jin Y, Chen Z, Li S, Chen D (2020) A new fallback beetle antennae search algorithm for path planning of mobile robots with collision-free capability. Soft Comput 24(3):2369–2380
10. Song R, Liu Y, Bucknall R (2019) Smoothed a* algorithm for practical unmanned surface vehicle path planning. Appl Ocean Res 83:9–20
11. Mac TT, Copot C, Tran DT, DeKeyser R (2016) Heuristic approaches in robot path planning: a survey. Robot Auton Syst 86:13–28
12. Choset H (2001) Coverage for robotics-a survey of recent results. Ann Math Artif Intell 31(1–4):113–126
13. Brumitt BL, Stentz A (1996) Dynamic mission planning for multiple mobile robots. In: Proceedings of IEEE international conference on robotics and automation, vol 3. IEEE, pp 2396–2401
14. Wawerla J, Sukhatme GS, Mataric MJ (2002) Collective construction with multiple robots. In: IEEE/RSJ international conference on intelligent robots and systems, vol 3. IEEE, pp 2696–2701
15. Cui R, Guo J, Gao B (2013) Game theory-based negotiation for multiple robots task allocation. Robotica 31(6):923–934
16. Roldán JJ, Garcia-Aunon P, Garzón M, DeLeón J, DelCerro J, Barrientos A (2016) Heterogeneous multi-robot system for mapping environmental variables of greenhouses. Sensors 16(7):1018
17. Luo M, Hou X, Yang J (2019) Multi-robot one-target 3d path planning based on improved bioinspired neural network. In: 2019 16th international computer conference on wavelet active media technology and Information Processing. IEEE, pp 410–413
18. Dias MB, Zlot R, Kalra N, Stentz A (2006) Market-based multirobot coordination: a survey and analysis. Proc IEEE 94(7):1257–1270
19. Yuan J, Wang H, Lin C, Liu D, Yu D (2019) A novel gru-rnn network model for dynamic path planning of mobile robot. IEEE Access 7:15140–15151
20. Luna R, Bekris K.E (2011) Efficient and complete centralized multi-robot path planning. In: 2011 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 3268–3275
21. Yan Z, Jouandeau N, Cherif AA (2013) A survey and analysis of multi-robot coordination. Int J Adv Robot Syst 10(12):399
22. Parker LE (2007) Distributed intelligence: overview of the field and its application in multi-robot systems. In: AAAI fall symposium: regarding the intelligence in distributed intelligent systems. pp 1–6
23. Capitan J, Spaan MT, Merino L, Ollero A (2013) Decentralized multi-robot cooperation with auctioned pomdps. Int J Robot Res 32(6):650–671
24. Ota J (2006) Multi-agent robot systems as distributed autonomous systems. Adv Eng Inform 20(1):59–70
25. Nerurkar ED, Zhou KX, Roumeliotis SI (2011) A hybrid estimation framework for cooperative localization under communication constraints. In: 2011 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 502–509
26. Sartoretti G, Wu Y, Paivine W, Kumar TS, Koenig S, Choset H (2019) Distributed reinforcement learning for multi-robot decentralized collective construction. Distributed autonomous robotic systems. Springer, New York, pp 35–49
27. Baxter JL, Burke EK, Garibaldi JM, Norman M (2009) Shared potential fields and their place in a multi-robot co-ordination taxonomy. Robot Auton Syst 57(10):1048–1055
28. Gerkey BP, Mataric MJ (2002) Sold!: Auction methods for multirobot coordination. IEEE Trans Robot Autom 18(5):758–768
29. Chen X, Zhang P, Du G, Li F (2018) Ant colony optimization based memetic algorithm to solve bi-objective multiple traveling salesmen problem for multi-robot systems. IEEE Access 6:21745–21757
30. Han SD, Yu J (2019) Effective heuristics for multi-robot path planning in warehouse environments. In: 2019 International symposium on multi-robot and multi-agent systems (MRS). IEEE, pp 10–12
31. Al-Jarrah R, Shahzad A, Roth H (2015) Path planning and motion coordination for multi-robots system using probabilistic neuro-fuzzy. IFAC-PapersOnLine 48(10):46–51
32. Ni J, Wang K, Cao Q, Khan Z, Fan X (2017) A memetic algorithm with variable length chromosome for robot path planning under dynamic environments. Int J Robot Autom 32(4):414–424
33. Alotaibi ETS, Al-Rawi H (2018) A complete multi-robot path-planning algorithm. Autonom Agents Multi-Agent Syst 32(5):693–740
34. Warren CW (1990) Multiple robot path coordination using artificial potential fields. In: Proceedings., IEEE international conference on robotics and automation. IEEE, pp 500–505
35. Svestka P, Overmars MH (1995) Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In: Proceedings of 1995 IEEE international conference on robotics and automation, vol 2. IEEE, pp 1631–1636
36. Švestka P, Overmars MH (1998) Coordinated path planning for multiple robots. Robot Auton Syst 23(3):125–152
37. Burgard W, Moors M, Fox D, Simmons R, Thrun S (2000) Collaborative multi-robot exploration. In: Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065), vol.1. IEEE, pp 476–481
38. Bennewitz M, Burgard W, Thrun S (2001) Optimizing schedules for prioritized path planning of multi-robot systems. In: Proceedings 2001 ICRA. IEEE international conference on robotics and automation (Cat. No. 01CH37164), vol 1. IEEE, pp 271–276
39. Sanchez G, Latombe JC (2002) Using a prm planner to compare centralized and decoupled planning for multi-robot systems. In: Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292), vol 2. IEEE, pp 2112–2119
40. Kalra N, Ferguson D, Stentz A (2005) Hoplites: a market-based framework for planned tight coordination in multirobot teams. In:

Proceedings of the 2005 IEEE international conference on robotics and automation. IEEE, pp 1170–1177
41. Wurm K.M, Stachniss C, Burgar W (2008) Coordinated multi-robot exploration using a segmentation of the environment. In: 2008 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 1160–1165
42. Hollinger G, Singh S, Djugash J, Kehagias A (2009) Efficient multi-robot search for a moving target. Int J Robot Res 28(2):201–219
43. van Den Berg J, Snoeyink J, Lin MC, Manocha D (2009) Centralized path planning for multiple robots: optimal decoupling into sequential plans. In: Robotics: science and systems, vol 2. pp 2–3
44. Hollinger G, Singh S (2010) Multi-robot coordination with periodic connectivity. In: 2010 IEEE international conference on robotics and automation. IEEE, pp 4457–4462
45. Ding XC, Rahmani AR, Egerstedt M (2010) Multi-uav convoy protection: an optimal approach to path planning and coordination. IEEE Trans Robot 26(2):256–268
46. Wagner G, Choset H (2011) M*: A complete multirobot path planning algorithm with performance bounds. In: 2011 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 3260–3267
47. Otte M, Correll N (2014) Any-com multi-robot path-planning with dynamic teams: multi-robot coordination under communication constraints. Experimental robotics. Springer, New York, pp 743–757
48. Contreras-Cruz MA, Ayala-Ramirez V, Hernandez-Belmonte UH (2015) Mobile robot path planning using artificial bee colony and evolutionary programming. Appl Soft Comput 30:319–328
49. Ma H, Wu X, Gong Y, Cui Y, Song J (2015) A task-grouped approach for the multi-robot task allocation of warehouse system. In: 2015 international conference on computer science and mechanical automation (CSMA). IEEE, pp 277–280
50. Wagner G, Choset H (2015) Subdimensional expansion for multirobot path planning. Artif Intell 219:1–24
51. Claes D, Oliehoek F, Baier H, Tuyls K, et al., (2017) Decentralised online planning for multi-robot warehouse commissioning, In: AAMAS'17: Proceedings of the 16th international conference on autonomous agents and multiagent systems. pp 492–500
52. Araki B, Strang J, Pohorecky S, Qiu C, Naegeli T, Rus D (2017) Multi-robot path planning for a swarm of robots that can both fly and drive. In: 2017 IEEE international conference on robotics and automation (ICRA), IEEE, pp 5575–5582
53. Bolu A, Korçak Ö (2019) Path planning for multiple mobile robots in smart warehouse. In: 2019 7th international conference on control, mechatronics and automation (ICCMA). IEEE, pp 144–150
54. Kumar NV, Kumar CS (2018) Development of collision free path planning algorithm for warehouse mobile robot. Procedia Compu Sci 133:456–463
55. Almadhoun R, Taha T, Seneviratne L, Zweiri Y (2019) A survey on multi-robot coverage path planning for model reconstruction and mapping. SN Appl Sci 1(8):847
56. Cortés J, Egerstedt M (2017) Coordinated control of multi-robot systems: a survey. SICE J Control Meas Syst Integr 10(6):495–503
57. Wurman PR, D'Andrea R, Mountz M (2008) Coordinating hundreds of cooperative, autonomous vehicles in warehouses. AI Mag 29(1):9–9
58. Bogue R (2016) Growth in e-commerce boosts innovation in the warehouse robot market. Ind Robot Int J