Robotica (2024), 1–11
doi:10.1017/S0263574724000894

RESEARCH ARTICLE

RH-ECBS: enhanced conflict-based search for MRPP with region heuristics

Zhangchao Pan[1], Runhua Wang[1], Qingchen Bi[1], Xuebo Zhang[1] and Jingjin Yu[2]

[1]College of Artificial Intelligence, Nankai University, Tianjin, China
[2]Rutgers University New Brunswick, Computer Science New Brunswick, New Brunswick, USA
Corresponding author: Runhua Wang; Email: wrunhua@nankai.edu.cn

Received: 23 December 2023; **Revised:** 11 April 2024; **Accepted:** 1 May 2024

Keywords: mobile robots; multi-robot system; multi-robot path planning; conflict-based search

Abstract
This paper proposes a novel two-layer framework based on conflict-based search and regional divisions to improve the efficiency of multi-robot path planning. The high-level layer targets the reduction of conflicts and deadlocks, while the low-level layer is responsible for actual path planning. Distinct from previous dual-level search frameworks, the novelties of this work are (1) subdivision of planning regions for each robot to decrease the number of conflicts encountered during planning; (2) consideration of the number of robots in the region during planning in the node expansion stage of A*, and (3) formal proof demonstrating the nonzero probability of the proposed method in obtaining a solution, along with providing the upper bound of the solution in a special case. Experimental comparisons with Enhanced Conflict-Based Search demonstrate that the proposed method not only reduces the number of conflicts but also achieves a computation time reduction of over 30%.

1. Introduction
Multi-robot path planning (MRPP) (or alternatively, multi-agent path planning, MAPF) is a complex combinatorial optimization problem that plans collision-free paths for multi-robot systems. MRPP is important for many applications, such as warehouse automation, traffic management, and multi-robot autonomous exploration [1, 2]. In comparison to single-robot path planning, MRPP exhibits the following characteristics: 1) high dimensionality. MRPP involves finding feasible paths for multiple robots simultaneously, leading to a higher dimensional problem; 2) NP-completeness. It has been proven to be an NP-complete problem [3, 4]; and 3) temporal and spatial entanglement. In MRPP, the coordination and synchronization of multiple robot movements in time and space must be considered simultaneously to avoid collisions and deadlocks.

Comprehensive research has been carried out on MRPP due to its numerous applications and challenges. Centralized methods have garnered substantial research attention due to their ability to guarantee high-quality solutions. Among these, search-based methods constitute the mainstream, wherein A* [5], independence detection (ID) framework [6, 7], M* [8, 9], and conflict-based search (CBS) [10] are recognized for their ability to ensure both optimality and completeness of solutions. CBS is a method based on increasing cost tree search [11], which has garnered significant attention from the research community due to its rapid search capabilities. There are many improvements based on CBS, such as improved CBS (ICBS) [12] and ICBS-h [13]. While these methods ensure optimality, they are often associated with lower efficiency. To expedite the search process, researchers have proposed bounded suboptimal search methods such as Enhanced Conflict-Based Search (ECBS) [14] and Explicit Enhanced Conflict-Based Search (EECBS) [15]. These methods have demonstrated improved search speeds while still providing reasonably good solutions. In addition to the above methods, another notable method in the search-based method is the use of Rubik Tables [16, 17]. While suboptimal, it delivers $1.x$ guarantees on high-density grid environments.

Centralized methods rely on globally available information from all robots, but they may encounter challenges when applied to dynamic environments. Recognizing these drawbacks has spurred the development of distributed methods. Among these, velocity obstacles (VO) [18], reciprocal velocity obstacles (RVO) [19], and optimal reciprocal collision avoidance (ORCA) [20] are prominent methods focusing on real-time collision avoidance for multiple robots in dynamic environments by interaction and perception. There are several improved variants and extensions of these methods, including non-holonomic ORCA [21] and hybrid RVO [22].

With the growing popularity of machine learning methods, researchers are also utilizing deep learning and reinforcement learning to address MRPP problems. For example, Yang Yang et al. have explored the application of deep Q-networks (DQN) to solve MRPP challenges [23]. Qingbiao Li et al. combine convolutional neural network and graph neural network for MRPP problem [24]. In addition, researchers have employed deep reinforcement learning methods to solve the MRPP problem [25, 26].

In this study, we propose a novel regional heuristic framework over ECBS for MRPP, namely **RH-ECBS**. Before solving the MRPP problem, the map undergoes a regional partition. The objective of this region division is to alleviate the robot density within each region, achieving a more uniform distribution of robots. This approach significantly reduces the incidence of conflicts, enhancing the overall efficiency of the planning process. To summarize, the main contributions and novelties of this work are as follows:

1. A novel regional heuristic planning framework is proposed. The region of each robot is divided before planning so that the number of conflicts and deadlocks encountered during planning can be decreased.
2. In the planning stage, the number of robots in each subregion is considered when performing the A^* search. Thus, the robot is guided to plan in the region with a small number of robots.
3. Formally proving the nonzero probability of obtaining a solution using the proposed method. Besides, the upper bound of the solution in a special case is given.
4. The comparative experiments were conducted with different numbers of robots, comparing RH-ECBS against the ECBS algorithm. The experimental results demonstrate that RH-ECBS outperforms ECBS in terms of efficiency while achieving comparable solution quality.

2. Problem formulation

The environment can be presented by the graph structure $G = \{V, E\}$, where V represents vertices and E represents edge. Multi-robot systems are denoted by $\mathcal{R} = \{R, S, F\}$, where $R = \{R_i\}$ represents the set of robots, $S = \{S_i\}$ represents the set of starting configurations and $F = \{F_i\}$ represents the set of final configurations. The solution of MRPP is defined as $\mathcal{P} = \{P_i\}$, P_i represents the path of robot i.

In the process of multi-robot planning, time is discretized into equal time steps. During each time step, a robot can only move from the current vertex to the surrounding 4-connect or 8-connect vertices, or remain waiting at the current vertex. Additionally, two types of conflicts can occur during the planning process: vertex conflict and edge conflict. Fig 1 shows the two types of conflicts.

According to the above definition, the MRPP problem is defined as follows. Given the environment map $G = \{V, E\}$, the multi-robot system $\mathcal{R} = \{R, S, F\}$. The planner should find a path from S_i to F_i for each robot R_i in R while there should be no conflict between any two paths P_i and P_j in the path set $\mathcal{P}$.

3. Method

The system consists of the following steps. First, the map is built as a two-layer map that consists of the high-resolution and low-resolution maps. The start and goal of the robot need to be mapped on

https://doi.org/10.1017/S0263574724000894 Published online by Cambridge University Press

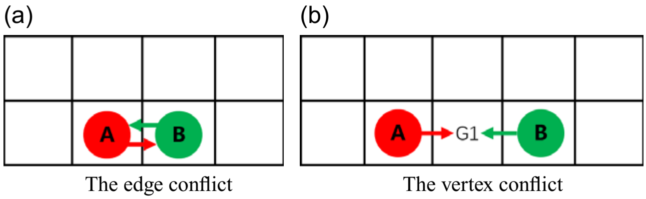

Figure 1. *Two different types of conflict.*

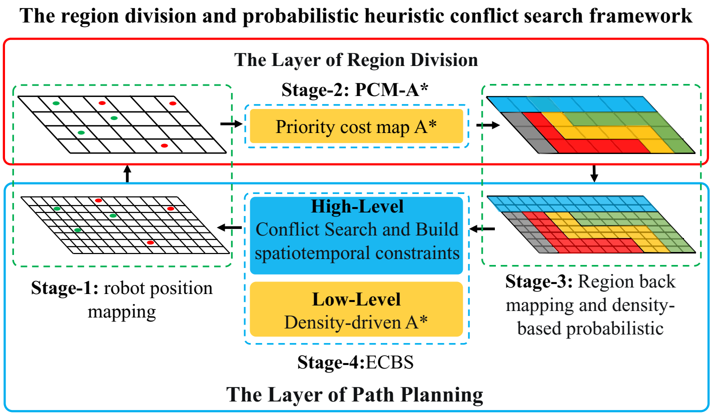

Figure 2. *System flowchart of the framework, where the yellow parts are our contributions.*

the low-resolution map from the high-resolution map. Then, utilizing individual robot path planning to partition regions for each robot in a low-resolution map. The region needs to be mapped back to the high-resolution map. Finally, ECBS is used to plan the path for the multi-robot system in the high-resolution map. When in the low-level search of ECBS, the probability of the search method expanding to the respective regions of each robot is greater. And the probability is relative to the robot density. The diagram of the method is shown in Fig 2. The yellow parts are the novel partition. The detailed method is presented in the rest of this section.

3.1. Two-layer map and region division

In order to segment the grid map into regions, the original high-resolution grid map was downsampled to generate a low-resolution map. At the same time, the position information of the robot is mapped to the low-resolution map. The construction of the high-low-resolution double-layer map and the mapping relationship of the robot position are shown in Fig 3, where the left side is the original high-resolution map and the right side is the low-resolution map after downsampling.

Formula (1) describes the mapping relationship of the robot position coordinates between the high- and low-resolution maps,

$$x_{low} = \lfloor \frac{x_{high} - x_o}{m} \rfloor, \quad y_{low} = \lfloor \frac{y_{high} - y_o}{n} \rfloor \tag{1}$$

where $\lfloor \bullet \rfloor$ represents rounding down, (x_{high}, y_{high}) represents the coordinates of the robot under the high-resolution map, (x_{low}, y_{low}) represents the coordinates of the robot under the low-resolution map, (x_o, y_o) denotes the coordinates of the origin of the high-resolution map, m represents the ratio of the resolution in the x-direction between the high-resolution map and the low-resolution map, and n represents the ratio of the resolution in the y-direction between the high-resolution map and the low-resolution map.

https://doi.org/10.1017/S0263574724000894 Published online by Cambridge University Press

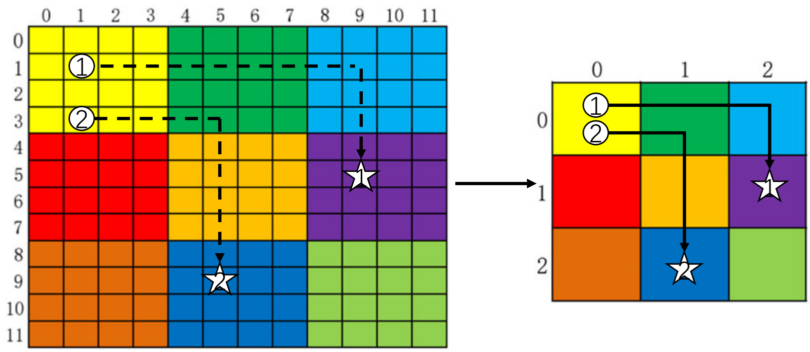

Figure 3. *Mapping relationship between high-resolution map and low-resolution map.*

Algorithm 1. PCM-A*

Require: multi-robot R, cost map C_m

1: **for** each R_i in R **do**

2: path p_i^l =invoking A*(R_i, C_m)

3: **for** each grid g_k in p_i^l **do**

4: $\mathrm{Cm}[g_k] = \mathrm{Cm}[g_k] + 10$

5: update the cost map C_m

6: **for** each grid g_k in grid map **do**

7: calculate the count that is occupied by robots N

8: calculate the $p[g_k] = \frac{N}{R_m+1}$

Ensure: path P^l and probability p

After building a two-layer map and mapping the robot position from the high-resolution map to the low-resolution one, the regions are segmented in the low-resolution map.

The novel priority cost map A* (PCM-A*) is used as the region partition method. The robot may avoid collisions with another robot in the same low-resolution grid as each low-resolution grid comprises multiple high-resolution grids. Consequently, when PCM-A* is employed to partition regions, it does not need to impede the robot with lower priority passing through the path of the robot with higher priority. However, to minimize conflicts among the robots, it is essential to decrease the overlap between the paths of each robot. Therefore, after a high-priority robot has a path, it is necessary to increase the cost of the path. This operation causes other robots to expand their paths to nodes with fewer occupancies during path search, thereby reducing overlap between regions. Lines 1 to line 7 of Algorithm 1 shows the flow. The input to the algorithm is the robot set R and a low-resolution cost map C_m. Besides, the initial value of the C_m is 10. The number of robots in the robot set is R_m.

Take Fig 4 as an example, to plan an optimal path from S_1 to G_1 for robot A, and the result of the planning is assumed to be the yellow grid. Next, the optimal path from S_2 to G_2 is planned for robot B. If the traditional A* is used, the planning result may be the green path in Fig 4(b); thus, there is a possibility of conflict between robot A and robot B. Whereas when PCM-A* is used to plan the path for robot B, the cost of the yellow grid has been added. In this case, the optimal path for robot B is the grid through which blue or green line segments pass. It is assumed that it takes the same time for the robot to pass through each grid. In this case, no conflict will occur when using PCM-A $*$, but conflicts will occur when using A*.

After dividing the region for each robot, there may be no solutions if each robot is only allowed to perform low-level path planning within its respective region. Therefore, a probabilistic heuristic method is designed to solve the problem, where each robot has a higher probability of expanding within its

https://doi.org/10.1017/S0263574724000894 Published online by Cambridge University Press

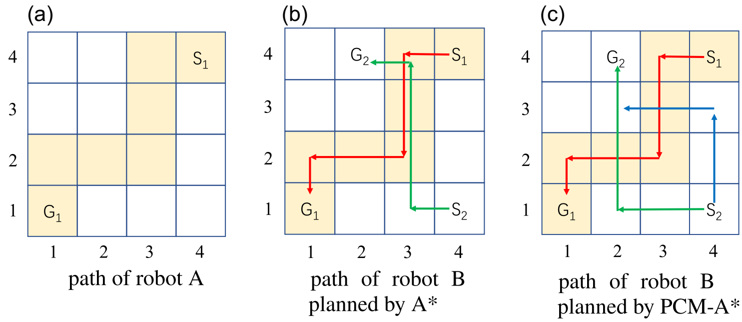

Figure 4. *Comparison of A* with PCM-A* in path planning.*

designated region while still having the possibility to expand into other regions. The method can ensure completeness, which is analyzed in the section completeness analysis and the maximum of solution. The probability is calculated by the formula (2). The physical meaning of formula (2) is that the more times a region is occupied by robots, the probability that a path node expands to this region when searching is lower.

$$Dp = w(1 - p), \tag{2}$$

where p is the output of Algorithm 1, and w is a weight used to calculate the probability Dp. If w is greater, it implies that during the path search process, the probability of path nodes only expanding to the partitioned areas is higher. In scenarios where there is a higher number of robots, this could potentially lead to an inability to find a path solution that meets the constraint conditions in a low-level search. When w is smaller, the region division is meaningless. The algorithm performs better in terms of time consumption when w is between 0.05 and 0.2 during the experiment.

3.2. Two-layer division and probability heuristic conflict search framework

ECBS is currently one of the most popular and high-performing algorithms for efficiently handling conflicts among multiple robots. It achieves this by generating bounded-suboptimal paths with fewer collisions with the paths of the other robots on the low level and expanding bounded-suboptimal CT nodes that contain fewer collisions on the high level [27]. In the low-level path planning phase, the A* method is used to find the path for each robot on a known map, which satisfies constraints and avoids collisions with known obstacles. In the high-level conflict detection and constraint generation phase, the paths of all robots are considered and checked for conflicts. For existing conflicts, constraints are added to resolve the conflicts between robots. The low-level path planning is then called until a nonconflicting solution is found. To further improve the algorithm, probabilistic heuristic search is introduced in low-level path planning to reduce the conflicts and improve efficiency when planning the path, which is called Ph-A*. The probability is obtained by the PCM-A*. The Ph-A* algorithm is described as Algorithm 2. The function *CheckConstraints*() in Line 8 checks whether the node satisfies the constraints of the high-level search build and whether it is occupied by obstacles.

3.3. Analysis of the maximum value of the solution and the probability of obtaining the solution

Theorem 1. *If a solution exists, the probability of RH-CBS obtaining a solution is nonzero.*

Proof. Consider the path nodes for the final solution of agent i denoted as $N_1, N_2, \ldots, N_m$. Each node in the high-resolution map is subject to expansion with a specific probability, denoted as $p_1, p_2, \ldots, p_m$. Thus, the probability of adding node N_m to the *OpenList* of A* can be expressed as follows:

https://doi.org/10.1017/S0263574724000894 Published online by Cambridge University Press

Algorithm 2. Ph-A*

Input: start s and goal g;

insert start to Open list: $\mathcal{L}_o$
while $\mathcal{L}_o != \emptyset$ **do**
  Node $N = \mathcal{L}_o$.top()
  put N to close list
  for each node N_i of adjacent nodes **do**
    if $N_i == g$ **then**
      break
    if CheckConstraints(N_i) **then**
      continue
    else insert N_i to $\mathcal{L}_o$ with probability $Dp[N_i]$
Find path $\mathcal{P}$ from g to s

Ensure: $\mathcal{P}$

Algorithm 3. RH-ECBS

Require: multi-robot instance

1: build two-layer map: low-resolution map and high-resolution map;
2: pose map from the high-resolution map to low-resolution map;
3: invoke the PCM-A* and get the Dp;
4: for each robot invoke ECBS-low-level(Ph-A*) to get path;
5: invoke ECBS-high-level to check conflict and build constraints;
6: **if** conflicts is $\emptyset$ **then**
7:   $\mathcal{P} = p$
8: **else** go to 4

Ensure: path $\mathcal{P}$

$$\begin{aligned} P(N_m) =& P(N_m|N_{m-1}) = p_m \cdot P(N_{m-1}) \\ =& p_m \cdot P(N_{m-1}|N_{m-2}) = p_m \cdot p_{m-1} \cdot P(N_{m-1}) \\ \cdots& \\ =& \prod_{i=1}^{m} p_i. \end{aligned} \tag{3}$$

Since the $p_i > 0$, the $P(N_m) > 0$. In addition, the probability of the method entering the loop is as follows:

$$P_{loop} = P_i^n. \tag{4}$$

The P_i is the probability of one of the paths with conflicts. And $0 < P_i < 1$, so when the $n \to \infty$, the $P_{loop} \to 0$. Therefore, RH-ECBS does not get stuck in a loop when solving MAPF problems.

Therefore, if a solution exists, the probability of RH-CBS obtaining a solution is nonzero. □

Theorem 2. *When each agent **only** finds a path in its region, traditional A* is used for regional division. The cost of the solution returned by the RH-ECBS is at most $2m/n \cdot C_{min}$, and the m and n represent the number of the grid in high-resolution and low-resolution maps, respectively. C_{min} represents the cost of the optimal solution.*

Proof. The L_i represents the cost of the solution returned by RH-ECBS and the l_i represents the cost of each agent in the optimal solution. Besides, the relationship between L_i and l_i is that $L_i = k_i \cdot l_i$, where k_i

https://doi.org/10.1017/S0263574724000894 Published online by Cambridge University Press

is an indeterminate constant. The cost can be represented as follows:

$$\begin{aligned} cost = \sum L_i &= L_1 + L_2 + \ldots + L_m \\ &= k_1 l_1 + k_2 l_2 + k_m l_m \\ &< \max\{k_i\} \sum l_i \\ &= \max\{k_i\} C_{min}. \end{aligned} \tag{5}$$

Next, the k_{max} can be calculated. Assuming that each agent only finds a path in its region. Because the region is returned by A*, the optimal path of each agent must be in the region. In addition, the worst path is to pass through all the grids in the region. Therefore, the $k_{max} = 2m/n$. So, theorem 2 is proved. □

4. Experiments and discussions

In order to verify the effectiveness of the proposed method, we compared RH-ECBS with the open-source method ECBS [14, 28]. All experiments are performed on an Intel(R) Core(TM) i7-11800H CPU with 2.3 GHz and 16 G of RAM. The experiment scenarios are set as 24×18 resolution maps with no obstacles and random obstacles, respectively.

In order to facilitate a comprehensive comparison, each experimental scenario involves the configuration of four distinct low-resolution maps: 2×2, 4×3, 6×6, and 12×9. Moreover, varying numbers of robots are designated for each resolution namely, 118, 142, 166, 190, and 213 – representing 25%, 30%, 35%, 40%, and 45% of the map size, respectively. There are five different configurations for each number of robots. Given that the proposed method incorporates probability and lacks consistency, each experiment is conducted 20 times. Subsequently, the 10 trials with the smallest discrepancy in solution cost between RH-ECBS and ECBS are chosen. Finally, the average of these 10 trials is computed for comparison with the results obtained using ECBS.

Four distinct low-resolution maps are configured to assess the influence of map resolution on planning outcomes. Specifically, the low-resolution maps are defined as 2×2, 4×3, 6×6, and 12×9, respectively. Comparative experiments are conducted with varying numbers of robots. The average experimental results are presented in Table II and Fig 5. Comparison with ECBS indicates that RH-ECBS exhibits superior efficiency across different robot quantities. Furthermore, the experimental outcomes for the 6×6 low-resolution map are detailed in Table I to more effectively illustrate performance improvements with varying robot numbers.

The results show that compared with ECBS, the number of conflicts solved by the proposed method is reduced by 34.08%, the solving time is reduced by 31.89%, and the cost is only 0.22% worse. It is noted that the percentage reduction refers to the average of the percentage reduction of experiments at each configuration.

It is noteworthy that we also conducted experiments on a low-resolution map of 24×12; however, obtaining a solution within a short timeframe proved unfeasible; hence, it is excluded from the table. Analysis of the results reveals that, with an increase in the resolution of low-resolution maps, the cost of the proposed algorithm's solution also increases. This degradation is attributed to the heightened randomness of RH-ECBS as the resolution increases. However, the significance of regional division diminishes when the low-resolution map's resolution is small, explaining the observed minor efficiency improvement in the 2×2 low-resolution map. Taking into account the solution cost and solving time, efficiency sees more improvement with only a slight increase in cost when the low-resolution map has a resolution of 6×6 or 4×3.

Furthermore, five sets of tests with different numbers of robots are conducted in an environment with obstacles in a low-resolution map of 4×3. The map with random obstacles is shown in Fig 6a. The comparison experiment is set up as before, and the results of experiments are presented in Table IV.

https://doi.org/10.1017/S0263574724000894 Published online by Cambridge University Press

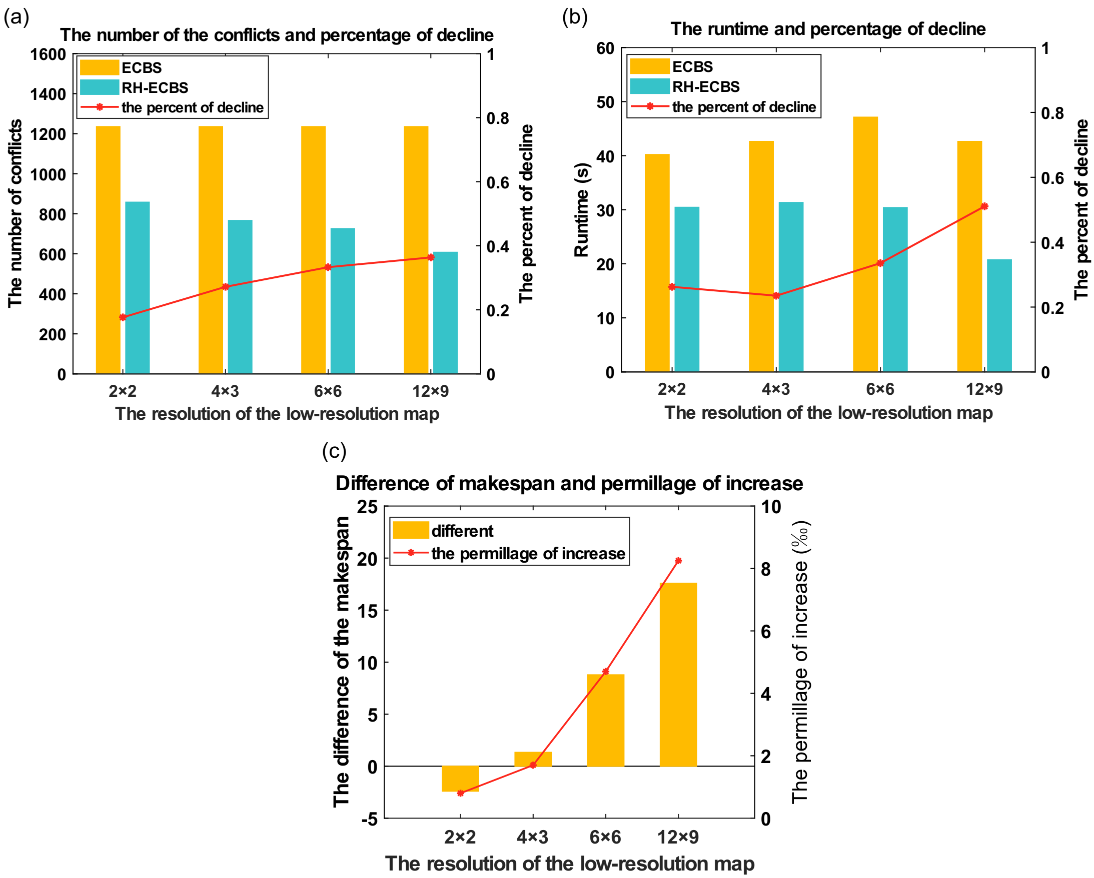

Figure 5. *The average of the experimental results for different numbers of robots.*

The results show that compared with ECBS, the solution time of the proposed method is reduced by more than 50% in the environment with obstacles.

In addition, tests are conducted in a warehousing environment, as shown in Fig 6b. In this environment, path planning was tested for 72 robots, 90 robots, and 108 robots, with five different start-target points selected for each quantity. The average results are shown in Table III. It is worth noting that the efficiency improvement refers to the average improvement under each test condition. Furthermore, the low-resolution map has a resolution of 4×3. The results show that compared with ECBS in a warehousing environment, the RH-ECBS algorithm obtains paths more quickly.

5. Conclusion

This paper introduces a MRPP framework incorporating regional heuristics. This method improves the efficiency of solving MRPP problems by reducing the number of conflicts during planning. Besides, compared with the popular ECBS method, the solving time is decreased by more than 30%. Further proving the nonzero probability of obtaining a solution using this method and determining the maximum cost of the solution in certain cases. However, there are still areas for further improvement in the proposed method. For example, although this method improves efficiency, the path cost increases, so it needs to be further optimized in the future. At the same time, the region division framework is considered to be

https://doi.org/10.1017/S0263574724000894 Published online by Cambridge University Press

Table I. *Experimental comparison results for different numbers of robots on a low-resolution 6×6 map.*

Number	Method	Conflicts	Conflicts reduction	Makespan	Makespan increase	Solving time	Time reduction
25%-118	ECBS	262.2	32.95%	1527.8	0.362%	2.70	20.58%
	RH-ECBS	168.7		1533.2		1.70	
30%-142	ECBS	340.6	36.62%	1977.4	−0.21%	5.80	23.90%
	RH-ECBS	205		1972.56		3.28	
35%-166	ECBS	958.25	26.17%	2360	0.04%	17.24	27.30%
	RH-ECBS	420.35		2361		9.02	
40%-190	ECBS	1111.8	33.92%	2880	0.45%	38.20	36.10%
	RH-ECBS	641.06		2893		22.75	
45%-213	ECBS	3060.4	40.76%	3250	0.47%	157.38	33.55%
	RH-ECBS	1686.54		3265		103.75	

Table II. *Experimental comparison results under maps of different resolutions.*

Map resolution	Method	Conflicts	Conflicts reduction	Makespan	Makespan increase	Solving time	Time reduction
2×2	ECBS	1235.36	17.62%	2393.2	0.21%	40.21	16.77%
	RH-ECBS	857.76		1533.2		30.45	
4×3	ECBS	1235.36	27.19%	2393.2	0.05%	42.63	23.90%
	RH-ECBS	766.44		2394		31.34	
6×6	ECBS	1235.36	33.32%	2393.2	0.36%	47.12	31.80%
	RH-ECBS	725.68		2401.3		30.40	
12×9	ECBS	1235.36	36.37%	2393.2	1.06%	42.63	37.31%
	RH-ECBS	607.55		2410.4		20.74	

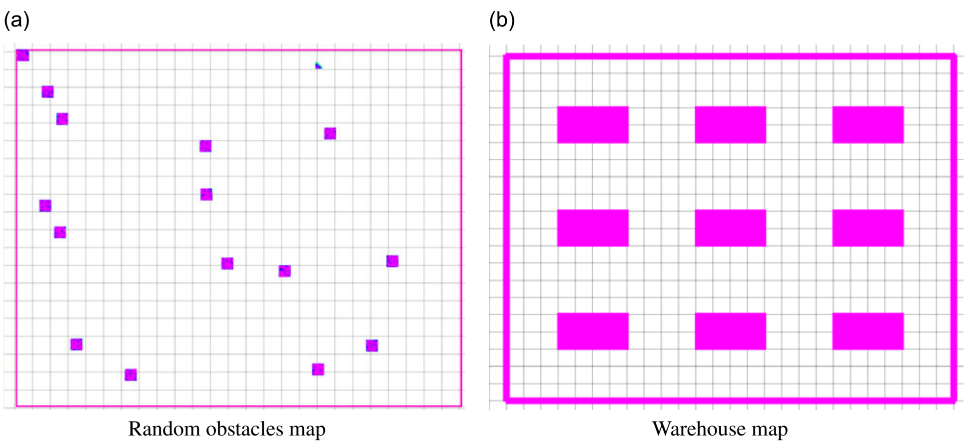

Figure 6. *Simulation maps.*

https://doi.org/10.1017/S0263574724000894 Published online by Cambridge University Press

Table III. *Experimental comparison results with ECBS in warehouse map.*

Number	Method	Conflicts	Conflicts reduction	Makespan	Makespan increase	Solving time	Time reduction
72	ECBS	30.2	21.6%	1060.8	0.6%	0.59	5.18%
	RH-ECBS	23.32		1067.84		0.58	
90	ECBS	240.4	30.27%	1455.4	0.8%	13.18	10.48%
	RH-ECBS	117.7		1467.4		7.51	
108	ECBS	933	50.81%	1676.6	0.03%	29.66	56.40%
	RH-ECBS	351.22		1683		10.44	

Table IV. *Experimental comparison results with ECBS in map with the random obstacle.*

	ECBS	RH-ECBS
Conflicts	1492.2	411.1
Conflict reduction	54.13%	
Makespan	2597	2586
Makespan reduction	0.21%	
Solving time(s)	46.488	15.228
Time reduction	57.68%	

added to other pathfinding methods based on graph search, such as EECBS [15], to further verify the improvement effect of this method.

Author contributions. All authors contributed to the study. Zhangchao Pan was involved in the conceptualization, methodology, software, validation, data curation, formal analysis, investigation, visualization, and writing – original draft preparation; Xuebo Zhang contributed to the conceptualization, writing – review and editing, funding acquisition, resources, project administration, and supervision; Runhua Wang assisted in the methodology, validation; Qingchen Bi was involved in writing – review and editing; and Jingjin Yu contributed to writing – review and editing.

Financial support. This work was supported in part by the National Natural Science Foundation of China under Grant 62293513/62293510, 62303247, and in part by the Natural Science Foundation of Tianjin under Grant 22JCZDJC00810, and in part by the China Postdoctoral Science Foundation under Grant 2020M670628, and in part by the Fundamental Research Funds for the China Central Universities of Nankai University.

Competing interests. The authors declare no competing interests exist.

Ethical approval. None.

References

[1] J. Li, Z. Chen, D. Harabor, P. J. Stuckey and S. Koenig, "MAPF-LNS2: Fast repairing for multi-agent path finding via large neighborhood search," *Proceed AAAI Conf Artif Intell* **36**(9), 10256–10265 (2022).
[2] Q. Bi, X. Zhang, J. Wen, Z. Pan, S. Zhang, R. Wang and J. Yuan, "CURE: A hierarchical framework for multi-robot autonomous exploration inspired by centroids of unknown regions," *IEEE Trans Autom Sci Eng* **99**, 1–14 (2023).
[3] J. Yu and S. LaValle, "Structure and intractability of optimal multi-robot path planning on graphs," *Proceed AAAI Conf Artif Intell* **27**(1), 1443–1449 (2013).
[4] T. Huang, J. Li, S. Koenig and B. Dilkina, "Anytime multi-agent path finding via machine learning-guided large neighborhood search," *Proceed AAAI Conf Artif Intell* **36**(9), 9368–9376 (2022).

[5] M. R. K. Ryan, "Exploiting subgraph structure in multi-robot path planning," *J Artif Intell Res* **31**, 497–542 (2008).
[6] T. Standley, "Finding optimal solutions to cooperative pathfinding problems," *Proceed AAAI Conf Artif Intell* **24**(1), 173–178 (2010).
[7] T. Standley and R. Korf, "Complete Algorithms for Cooperative Pathfinding Problems," **In:** *IJCAI*, (2011).
[8] G. Wagner and H. Choset, "M*: A Complete Multirobot Path Planning Algorithm with Performance Bounds," **In:** *2011 IEEE/RSJ international conference on intelligent robots and systems. IEEE*, (2011).
[9] C. Ferner, G. Wagner and H. Choset, "ODrM∗ Optimal Multirobot Path Planning in Low Dimensional Search Spaces," **In:** *2013 IEEE international conference on robotics and automation. IEEE*, (2013).
[10] G. Sharon, R. Stern, A. Felner and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artif Intell* **219**, 40–66 (2015).
[11] G. Sharon, R. Stern, M. Goldenberg and A. Felner, "The increasing cost tree search for optimal multi-agent pathfinding," *Artif Intell* **195**, 470–495 (2013).
[12] E. Boyarski, A. Felner, R. Stern, G. Sharon, O. Betzalel, D. Tolpin and E. Shimony, "Icbs: The improved conflict-based search algorithm for multi-agent pathfinding," *Proceed Int Symp Combin Sear* **6**(1), 223–225 (2015).
[13] A. Felner, J. Li, E. Boyarski, H. Ma, L. Choren, T. K. Satish Kumar and S. Koenig, "Adding heuristics to conflict-based search for multi-agent path finding," *Proceed Int Conf Autom Plan Sched* **28**, (2018).
[14] M. Barer, G. Sharon, R. Stern and A. Felner, "Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem," *Proceed Int Symp Combin Sear* **5**(1), 19–27 (2014).
[15] J. Li, W. Ruml and S. Koenig, "EECBS: A bounded-suboptimal search for multi-agent path finding," *Proceed AAAI Conf Artif Intell* **35**(14), 12353–12362 (2021).
[16] M. Szegedy and J. Yu, "Rubik tables and object rearrangement," *Int J Rob Res* **42**(6), 459–472 (2023).
[17] T. Guo and J. Yu, "Sub-1.5 time-optimal multi-robot path planning on grids in polynomial time, "(2022) arXiv preprint arXiv: 2201.08976.
[18] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int J Rob Res* **17**(7), 760–772 (1998).
[19] B. Van den, M. L. Jur and D. Manocha, "Reciprocal Velocity Obstacles for Real-Time Multi-Agent Navigation," **In:** *2008 IEEE international conference on robotics and automation*, (2008).
[20] J. van den Berg, S. J. Guy, M. Lin and D. Manocha, "Reciprocal n-Body Collision Avoidance," **In:** *Robotics Research: The 14th International Symposium ISRR*, (2011).
[21] J. Alonso-Mora, A. Breitenmose, M. Rufli, P. Beardsley and R. Siegwart, "Optimal Reciprocal Collision Avoidance for Multiple Non-Holonomic Robots," **In:** *Distributed autonomous robotic systems: The 10th international symposium*, (2013).
[22] J. Snape, J. van den Berg, S. J. Guy and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Trans Robot* **27**(4), 696–706 (2011).
[23] Y. Yang, L. Juntao and P. Lingling, "Multi-robot path planning based on a deep reinforcement learning DQN algorithm," *CAAI Trans Intell Technol* **5**(3), 177–183 (2020).
[24] Q. Li, F. Gama, A. Ribeiro and A. Porok, "Graph Neural Networks for Decentralized Multi-Robot Path Planning," **In:** *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE*, (2020).
[25] R. Han, S. Chen, S. Wang, Z. Zhang, R. Gao, Q. Hao and J. Pan, "Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards," *IEEE Rob Autom Lett* **7**(3), 5896–5903 (2022).
[26] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang and J. Pan, "Towards Optimally Decentralized Multi-Robot Collision Avoidance Via Deep Reinforcement Learning," **In:** *2018 IEEE international conference on robotics and automation (ICRA). IEEE*, (2018).
[27] S.-H. Chan, J. Li, G. Gange, D. Harabor, P. J. Stuckey and S. Koenig, "ECBS with flex distribution for bounded-suboptimal multi-agent path finding," *Proceed Int Symp Combin Sear* **12**(1), 159–161 (2021).
[28] J. Li, M. Ran and L. Xie, "Efficient trajectory planning for multiple non-holonomic mobile robots via prioritized trajectory optimization," *IEEE Rob Autom Lett* **6**(2), 405–412 (2020).

Cite this article: Z. Pan, R. Wang, Q. Bi, X. Zhang and J. Yu, "RH-ECBS: enhanced conflict-based search for MRPP with region heuristics", Robotica. https://doi.org/10.1017/S0263574724000894