Very Large-scale Multi-Robot Task Allocation in Challenging Environments via Robot Redistribution

Seabin Lee[1], Joonyeol Sim[1], and Changjoo Nam[1,*]

***Abstract*— Multi-robot systems deployed in logistics and inspection domains often suffer from navigation conflicts and deadlocks in dense environments such as warehouses and shopping malls. This paper presents a scalable Multi-Robot Task Allocation via Robot Redistribution Mechanism (MRTA-RM) which integrates robot path information to proactively reduce collisions and alleviate potential deadlocks, thereby enabling faster and more reliable task completion. By constructing a roadmap using a Generalized Voronoi Diagram (GVD), performing demand-supply analysis, and redistributing robots across environment components, MRTA-RM prevents head-on conflicts and reduces navigation congestion. Extensive simulations with hundreds of robots in warehouse-like environments demonstrate significant improvements in makespan, success rate, and scalability compared to state-of-the-art methods. The results highlight the potential of MRTA-RM as a conflict-aware task allocation strategy that significantly mitigates deadlocks, making it suitable for multi-robot logistics and inspection applications.**

I. Introduction

The deployment of multi-robot systems in logistics and inspection has grown rapidly, ranging from automated warehouses to large shopping malls. While task allocation is essential for efficiency, dense environments with narrow passages often lead to navigation conflicts and potential deadlocks. These situations can block progress, delay completion, or even prevent certain tasks from being finished.

Conventional allocation methods such as Hungarian [1] or greedy assignments optimize cost but neglect navigation conflicts. As illustrated in Fig. 1a, in crowded settings a naive assignment can produce head-on encounters between robots (Fig. 1b), resulting in delays or deadlocks. Conflict-Based Search [2] with Task Assignment (CBS-TA) [3], [4] explicitly resolves conflicts by coupling allocation and path planning, but its scalability is limited to tens of robots. Recent approaches in multi-agent pickup and delivery (MAPD) [5], [6] attempt to integrate allocation and path planning, but often rely on discretized environments that oversimplify continuous dynamics.

We propose the Multi-Robot Task Allocation via Robot Redistribution Mechanism (MRTA-RM), a redistribution-based allocation strategy that constructs a roadmap of the environment, analyzes robot–task supply and demand, and reassigns robots to prevent opposite-direction encounters. By aligning robot flows within roadmap sections, MRTA-RM reduces congestion and the likelihood of deadlocks, enabling faster and more reliable task completion (Fig. 1c).

The following are contribution of this work:

[1]Dept. of Electronic Engineering, Sogang University. *Corresponding author: `cjnam@sogang.ac.kr`

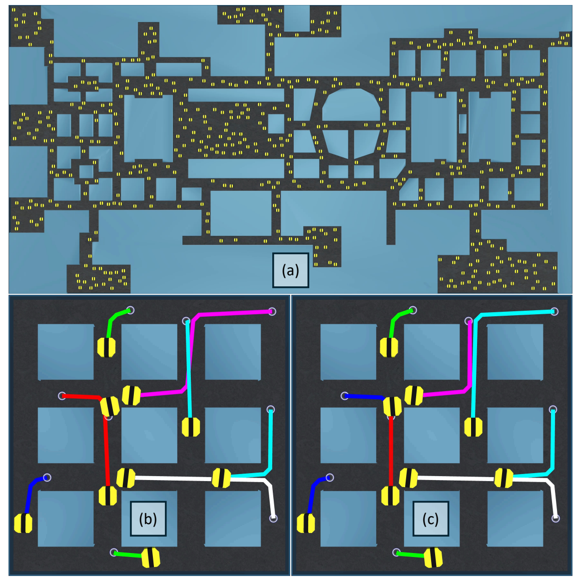


Fig. 1: An example of challenging environments with the effect of considering inter-robot conflicts in the task allocation. (a) 500 robots in a shopping mall with narrow passages, which is one of our test instances incurring frequent conflicts. (b) An allocation found by a conventional method (the Hungarian method) that does not consider conflicts between robots. The colored lines represent expected paths from the robots to their assigned tasks that do not account for collisions between robots. A robot with a red line will have its path blocked by another robot on the path. (c) A result of the proposed method. The robots do not pass the corridors in the opposite direction, which is beneficial to prevent deadlocks.

- **Conflict-aware task allocation**: MRTA-RM explicitly considers robot paths via a roadmap abstraction and redistributes robots between components to reduce navigation conflicts and minimize deadlock situations.
- **Scalability**: The algorithm runs in polynomial time and scales to hundreds of robots even in dense cluttered environments.
- **Comprehensive evaluation**: We evaluate MRTA-RM in three challenging environments under multiple task distributions, comparing against six baseline methods to highlight its advantages.

II. Related Work

Several methods have attempted to jointly address multi-robot task allocation (MRTA) and path planning in order to reduce navigation conflicts. Conflict-Based Search with Task Assignment (CBS-TA) [3] and its variants such as ECBS-TA and TCBS [4] extend the multi-Agent Pathfinding (MAPF) framework to the Task Assignment and Pathfinding (TAPF) problem. These methods search over both assignments and paths to produce conflict-free solutions, but their scalability is limited to tens of robots. In addition, multi-agent pickup and delivery (MAPD) methods [5], [6] integrate task assignment with motion planning in dynamic settings. While effective for online decision making, they require discretization of the workspace and are tested mainly in grid-based or sectorized environments, which limits their applicability in dense continuous spaces.

Heuristic and evolutionary approaches have also been proposed. For example, vitality-driven genetic task allocation [7] applies genetic optimization with conflict-handling strategies, while the goal assignment and planning method [8] uses lexicographic bottleneck assignment [9] and time offsets to avoid collisions. Although these methods can resolve some conflicts, they often rely on discretized environments or handcrafted constraints.

More recent work has investigated large-scale extensions of TAPF. Another work [6] incorporated robot density and travel-time prediction in sector-based environments, which improves scalability but requires manually defined sectors and restrictive passage assumptions.

In contrast, our approach is conflict-aware rather than fully conflict-free. MRTA-RM does not enumerate collision-free paths, but instead reduces the likelihood of deadlocks by eliminating opposite-direction encounters and congestion within sections of a roadmap. This design allows MRTA-RM to operate efficiently in continuous, cluttered environments with hundreds of robots.

III. Methods

We consider a one-shot assignment problem where N interchangeable robots are allocated to M tasks. A centralized allocator has full knowledge of the environment, robot states, and task positions, and aims to minimize the overall makespan (i.e., the time until all tasks are completed). Unlike classical assignment problems, the actual execution cost depends on the navigation paths in a dense environment, where conflicts can lead to significant delays or even deadlocks. Two execution constraints are therefore particularly important:

- **Constraint 1:** avoiding opposite-direction traversal along the same corridor,
- **Constraint 2:** preventing robots assigned to nearby tasks from blocking others that are still en route.

A. Approach Overview

Given an environment, MRTA-RM begins by constructing a roadmap (i.e., a topological graph) to represent traversable space. This roadmap is partitioned into subgraphs called components, and a demand–supply analysis determines which components have robot shortages or surpluses. Redistribution planning then assigns robots to move between components to balance demand and supply, while considering their transfer paths to reduce inter-robot conflicts. Once redistribution is complete, robots are allocated to tasks within each component.

In this work, we assume that all robots remain operational during execution. In real deployments, however, unexpected failures (e.g., battery depletion or mechanical breakdown) may occur. A natural extension of our approach would be to treat failed robots as static obstacles and recompute the roadmap and redistribution plan, allowing the system to adapt and resume task execution with minimal disruption.

B. Roadmap Construction

To represent the traversable space, we construct a roadmap using a Generalized Voronoi Diagram (GVD) [10]. The GVD captures the medial axis of free space and preserves topological relationships between obstacles [11]. After post-processing to remove inaccessible nodes (e.g., too close to obstacles) and to distribute nodes evenly, we obtain a roadmap G as illustrated in Fig. 2.

From the roadmap, we identify *junction nodes* (JC nodes) and *sections*. JC nodes (red dots in Fig. 2) are either terminal nodes or nodes with more than two neighbors. A section is the path between two JC nodes, excluding the nodes themselves. The set of JC nodes and sections together form the roadmap *components*, which serve as the basis for redistribution planning. When obstacles change, only the affected components require updates, making the representation flexible in dynamic environments.

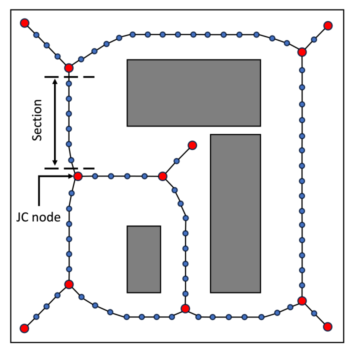


Fig. 2: An example roadmap. Red larger nodes represent junction nodes (JC nodes). A section is defined by non-JC nodes between the JC nodes.

C. Demand-Supply Analysis

After constructing the roadmap, each robot and task is associated with its nearest accessible node, ensuring that connections do not cross obstacles. This mapping allows us to count how many robots and tasks belong to each component (either a section or a junction node).

For each component $z_s \in \mathcal{Z}$, we compute a balance value

$$D_s = |\{\text{robots in } z_s\}| - |\{\text{tasks in } z_s\}|.$$

If $D_s > 0$, the component is *oversupplied* with surplus robots; if $D_s < 0$, it is *undersupplied* and requires additional robots; and if $D_s = 0$, it is balanced. We denote the sets of oversupplied, undersupplied, and balanced components as $\mathcal{Z}^+$, $\mathcal{Z}^-$, and $\mathcal{Z}^\circ$, respectively.

This demand–supply characterization provides the basis for redistribution planning: robots are reassigned from oversupplied to undersupplied components in order to balance workload across the environment.

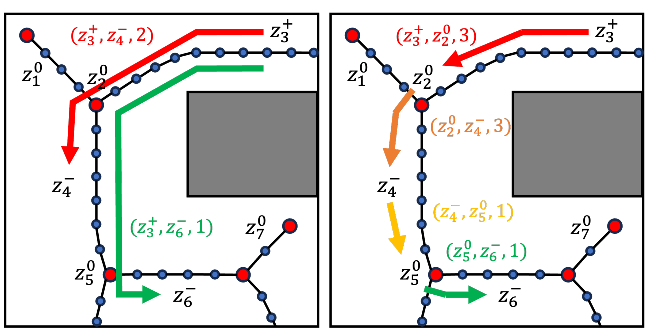
D. Redistribution Planning

Once the demand–supply analysis is completed, robots from oversupplied components are reassigned to undersupplied components. We formulate this as a weighted bipartite matching problem between surplus robots and deficit tasks. Although many assignment methods could be used (e.g., integer programming, auction algorithms), we employ the Hungarian method [1] for efficiency. Unlike approaches that use simple Euclidean distances, we estimate costs using roadmap distances that better reflect actual travel paths in cluttered environments.

To reduce computation, robot-to-task costs are approximated at the component level. For each pair of oversupplied and undersupplied components, we compute a shortest path between their centers on the roadmap and use this length as the cost for all robots/tasks in those components. This significantly reduces the number of path planning queries while still capturing the main congestion structure.

The initial redistribution plan specifies flows of robots from surplus to deficit components. However, executing flows over long paths may cause congestion. Therefore, we revise the plan by decomposing long flows into adjacent component-wise transfers. This yields a set of localized robot transfers that respect the roadmap structure and reduce overall traffic.

An example is shown in Fig. 3, where an initial plan with long flows is revised into a sequence of shorter, adjacent transfers. This decomposition reduces the number of traversed components per robot and alleviates congestion.



(a) Initial redistribution plan (b) Revised redistribution plan

Fig. 3: Example redistribution plan. (Left) Initial long flows from surplus to deficit components. (Right) Revised plan with adjacent component-wise flows, which reduces traffic and congestion.

E. Conflict-aware Assignment

The redistribution flows described above only specify the quantity of robots moving between components, not the execution order or detailed robot–task assignments. We therefore introduce a conflict-aware execution strategy consisting of (i) prioritizing flows and (ii) assigning tasks within components.

Flow prioritization. Components are categorized into sources, sinks, and intermediates depending on whether they only send robots, only receive robots, or both. Flows originating from source components are executed first, followed by flows through intermediate components, and finally those terminating at sinks. This ordering reduces makespan by ensuring that robots which may need to traverse multiple components are dispatched earlier, while robots destined for final sinks move later.

Task assignment within components. Once redistribution brings robots into a component, tasks are allocated to robots based on their entry directions and positions (Fig. 4). If robots enter from both ends of a section, tasks are split into two groups (front and back) to prevent cross-traffic. When both incoming and native robots are present, tasks are divided into three groups (front, middle, back), assigning each robot to the nearest group to minimize interference. This grouping ensures that robots avoid blocking each other's paths and mitigates the chance of local deadlocks. For JC nodes, tasks form a single group. Each robot is then given a path consisting of reference junction nodes as waypoints, which guide it to its assigned task while allowing local controllers to handle obstacle avoidance.

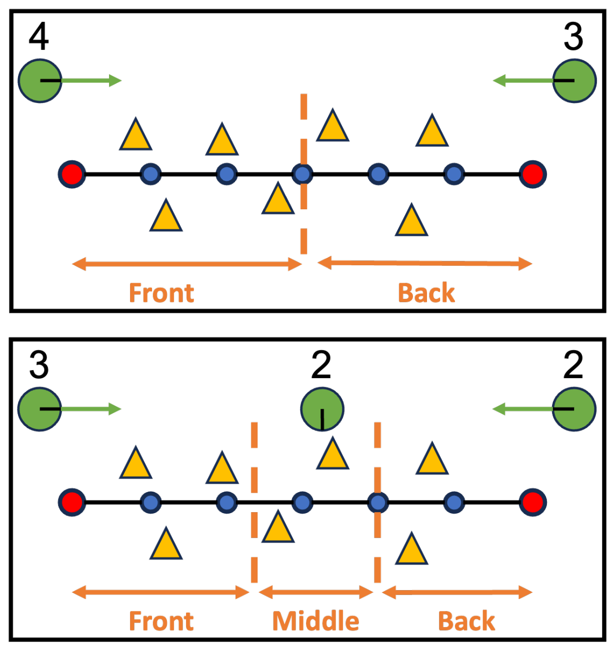


Fig. 4: Examples of task grouping in a section. Yellow triangles mark task positions within the section, and green circles represent robots. Incoming robot directions are indicated by green arrows, while orange dashed lines highlight the boundaries between task groups. Top: tasks split into two groups when robots enter from both ends. Bottom: tasks split into three groups when both incoming and native robots are present.

F. Method Properties

The proposed MRTA-RM framework is designed to satisfy the two execution constraints introduced earlier (Constraints in III). These properties make MRTA-RM conflict-aware: while it does not guarantee globally collision-free paths, it systematically reduces congestion and the likelihood of deadlocks in dense environments.

In terms of scalability, the algorithm executes in polynomial time with overall complexity on the order of $O(N|\mathcal{V}| + \max(N^3, |\mathcal{Z}|^3))$, where N is the number of robots (and tasks), $|\mathcal{V}|$ is the number of roadmap nodes, and $|\mathcal{Z}|$ is the number of roadmap components (JC nodes and sections). This enables MRTA-RM to scale effectively to hundreds of robots and tasks in large, cluttered environments.

IV. Experiments

We evaluate MRTA-RM in three types of environments (shopping mall, warehouse, clutter; Fig. 5) under two task distributions (random, separated). Robots navigate with Unity NavMesh and local RVO [12] collision avoidance, ensuring realistic dynamics while still allowing deadlocks to occur in narrow passages.

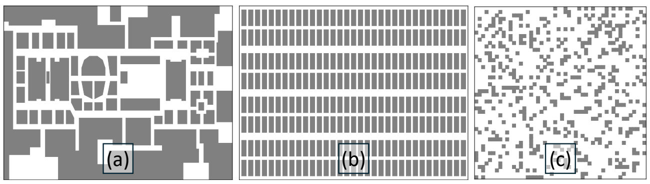


Fig. 5: Environments used for evaluation: (a) Shopping mall (1760 × 900 units), (b) Warehouse (2200 × 880 units), (c) Cluttered (1000×1000 units)

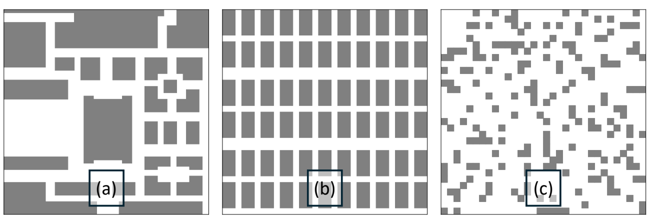


Fig. 6: Part of the environments for small-scale experiments comparing CBS-based methods. The size of all environments is 640×640 units.

A. Comparison with CBS-based Methods

We first compare MRTA-RM with conflict-resolving task allocation methods including CBS-TA and ECBS-TA, as well as continuous variants TA-CCBS and TA-CECBS. These experiments are conducted in the cropped environments shown in Fig. 6, where the robot density is kept high to stress-test conflict resolution methods.

As shown in Fig. 7, MRTA-RM consistently solves all tested instances within 0.2 seconds, while CBS-based methods become intractable beyond 20–30 robots. In the more challenging separated scenario, CBS-based methods frequently fail to compute any solution within the time limit, whereas MRTA-RM still achieves success rates above 90% in most settings.

In terms of execution performance, MRTA-RM achieves shorter makespan and lower sum-of-costs (SoC) across all environments. On average, it reduces makespan by up to 30% compared to ECBS-TA and over 20% compared to CBS-TA, while also improving SoC by more than 30%. Compared to TA-CCBS and TA-CECBS in continuous space, MRTA-RM demonstrates at least 20–30% reductions in both makespan and SoC.

These results confirm that while CBS-style methods can provide collision-free guarantees in small instances, they scale poorly and incur high computation times. In contrast, MRTA-RM balances scalability and conflict-awareness, yielding efficient allocations that remain effective in dense, continuous environments.

B. Comparisons with the methods without conflict resolution

To evaluate the scalability of our method in comparison to others that do not explicitly resolve conflicts, we include Greedy-TA and Hungarian-TA. These large-scale experiments are conducted in the full environments shown in Fig. 5, where the number of robots ranges from hundreds to five hundred.

Greedy-TA assigns tasks to robots based on the lowest-cost pairing, whereas Hungarian-TA uses an optimal assignment strategy via the Hungarian method. Both methods are efficient and scale to large instances but do not consider conflicts during task execution, making them fundamentally different from our proposed MRTA-RM.

As shown in Fig. 8, MRTA-RM requires slightly longer computation time than Greedy-TA and Hungarian-TA, yet the difference is within a few seconds even in the largest cases ($N = 500$). In return, MRTA-RM achieves substantially higher success rates in navigation. While the baselines often suffer from deadlocks (success rates below 50% in many separated scenarios), MRTA-RM consistently achieves above 60–100% across all settings.

In terms of execution performance, MRTA-RM also outperforms the baselines in makespan and SoC. For successful instances, MRTA-RM reduces makespan by up to 90% compared to Greedy-TA and up to 70% compared to Hungarian-TA. SoC is also improved by at least 10–30% in most scenarios.

These results highlight the trade-off between computation time and execution quality: although MRTA-RM spends slightly longer in allocation, its conflict-aware redistribution greatly improves the robustness and efficiency of task execution at scale.

C. Integration with MAPF solver

To assess the benefit of combining task allocation with a continuous-space MAPF solver, we evaluate MRTA-RM, Hungarian-TA, and Greedy-TA coupled with ST-RRT* [13]. Experiments follow the full-scale settings of Sec. IV-B, with a 5-minute limit for planning.

MRTA-RM consistently yields higher success rates than the baselines. For instance, in cluttered environments with 200 robots under random distributions, MRTA-RM achieves 95% success while Hungarian-TA drops to 0% and Greedy-TA to 70%. In separated scenarios with 100 robots, MRTA-RM maintains 70–90% success, compared to 50–60% for Hungarian-TA and below 65% for Greedy-TA.

In computation time, MRTA-RM + ST-RRT* reduces planning time by at least 8–11% compared to Hungarian-TA and by 7–8% compared to Greedy-TA. In terms of path quality, MRTA-RM generally achieves lower makespan and SoC; while Hungarian-TA marginally outperforms in one

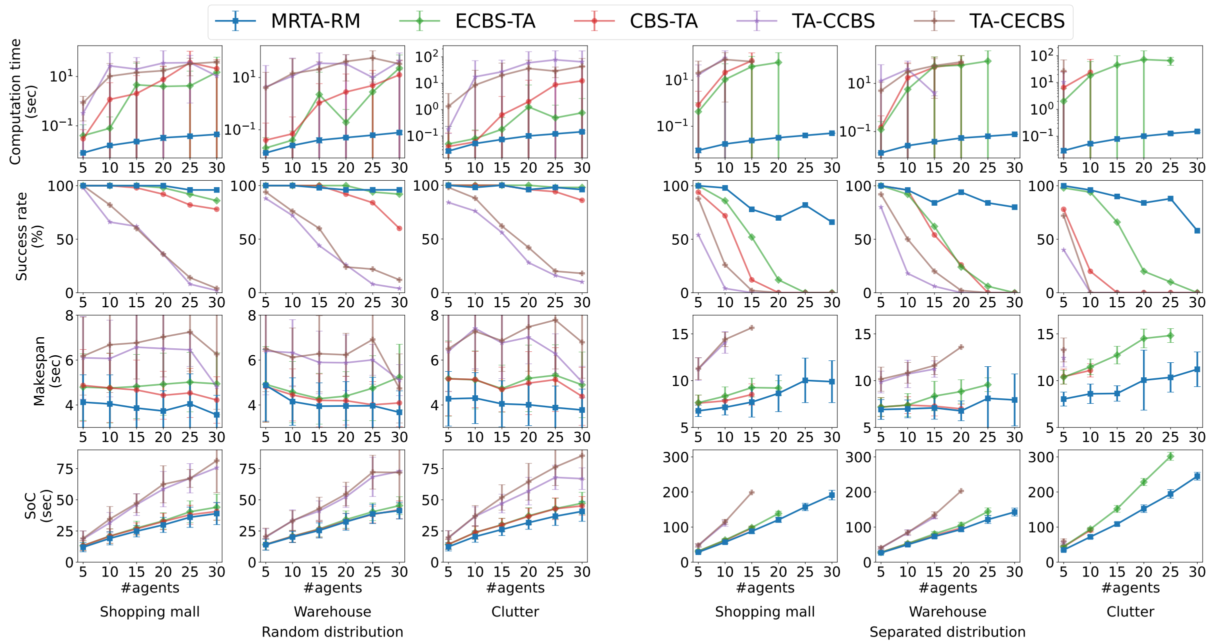


Fig. 7: Performance comparison in the small-scale environments (Fig. 6). MRTA-RM achieves consistently faster computation, higher success rates, and lower makespan and SoC than CBS-TA, ECBS-TA, and their continuous variants (TA-CCBS, TA-CECBS). While CBS-based methods often fail to compute solutions within the time limit as the number of robots increases, MRTA-RM maintains high success rates and superior execution efficiency.

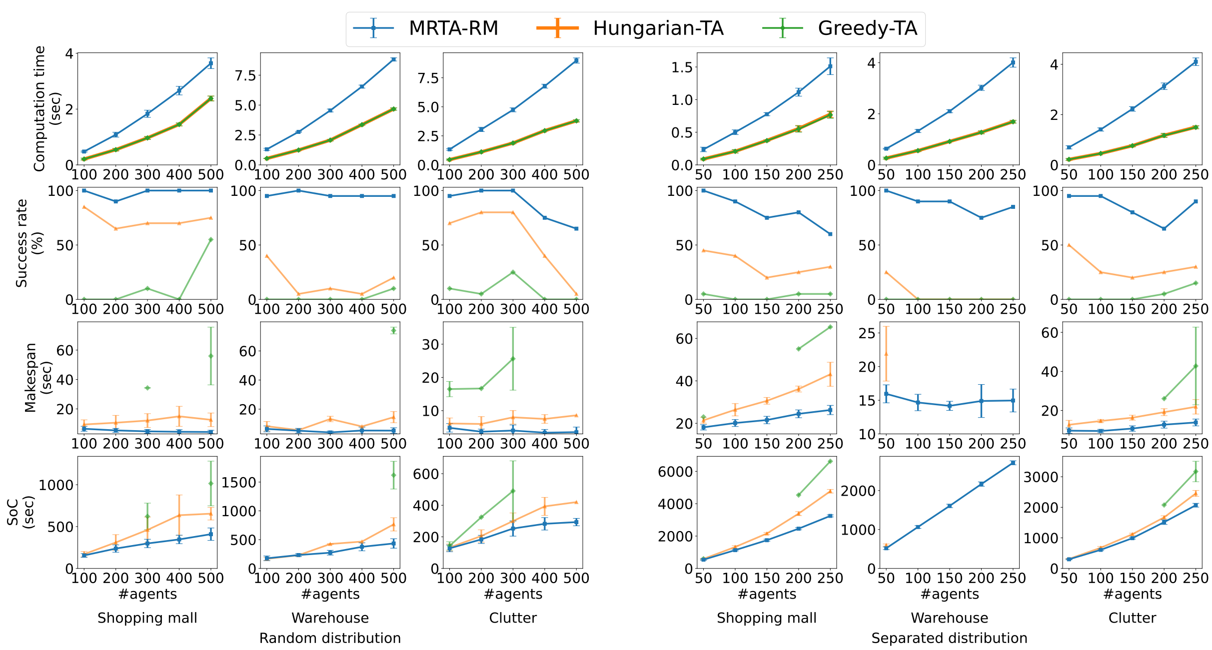


Fig. 8: Performance comparison in the full-scale environments (Fig. 5). MRTA-RM requires slightly longer computation time than Greedy-TA and Hungarian-TA but achieves substantially higher navigation success rates, avoiding deadlocks in dense scenarios. It also yields lower makespan and SoC across most settings, demonstrating the benefits of conflict-aware redistribution even at large scales.

case, MRTA-RM improves makespan by 3–89% and SoC by at least 6% across most settings.

Overall, these results demonstrate that conflict-aware task allocation with MRTA-RM not only facilitates scalable planning but also improves the efficiency and robustness of MAPF execution in dense continuous environments.

V. Conclusion

We presented MRTA-RM, a conflict-aware task allocation framework that balances robot supply and demand across roadmap components through redistribution. This design reduces congestion and deadlocks during execution, enabling scalable allocation for hundreds of robots in dense environments. Dynamic simulation results confirmed that MRTA-RM achieves high success rates and efficient task execution compared to both conflict-resolving and conventional assignment methods.

Future work will focus on (i) integrating controllers that enforce roadmap-following behavior to further increase success rates, and (ii) extending the framework to heterogeneous robot teams by constructing tailored GVD-based roadmaps for robots with different sizes and capabilities.

References

[1] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[2] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, "Conflict-based search for optimal multi-agent pathfinding," *Artificial Intelligence*, vol. 219, pp. 40–66, 2015.

[3] W. Hönig, S. Kiesel, A. Tinka, J. W. Durham, and N. Ayanian, "Conflict-based search with optimal task assignment," in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. IFAAMAS, 2018, pp. 757–765.

[4] C. Henkel, J. Abbenseth, and M. Toussaint, "An optimal algorithm to solve the combined task allocation and path finding problem," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4140–4146.

[5] Z. Chen, J. Alonso-Mora, X. Bai, D. D. Harabor, and P. J. Stuckey, "Integrated task assignment and path planning for capacitated multi-agent pickup and delivery," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5816–5823, 2021.

[6] Z. Liu, H. Wei, H. Wang, H. Li, and H. Wang, "Integrated task allocation and path coordination for large-scale robot networks with uncertainties," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 2750–2761, 2021.

[7] H. Zhang, H. Luo, Z. Wang, Y. Liu, and Y. Liu, "Multi-robot cooperative task allocation with definite path-conflict-free handling," *IEEE Access*, vol. 7, pp. 138 495–138 511, 2019.

[8] M. Turpin, K. Mohta, N. Michael, and V. Kumar, "Goal assignment and trajectory planning for large teams of interchangeable robots," *Autonomous Robots*, vol. 37, no. 4, pp. 401–415, 2014.

[9] R. E. Burkard and F. Rendl, "Lexicographic bottleneck problems," *Operations Research Letters*, vol. 10, no. 5, pp. 303–308, 1991.

[10] H. Choset, "Incremental construction of the generalized voronoi diagram, the generalized voronoi graph, and the hierarchical generalized voronoi graph," in *Proceedings of the CGC Workshop on Computational Geometry*, 1997.

[11] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[12] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2008, pp. 1928–1935.

[13] F. Grothe, V. N. Hartmann, A. Orthey, and M. Toussaint, "St-rrt*: Asymptotically-optimal bidirectional motion planning through space-time," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3314–3320.