Adaptive Robot Coordination: A Subproblem-Based Approach for Hybrid Multi-Robot Motion Planning

Irving Solis, *Graduate Student Member, IEEE*, James Motes, *Member, IEEE*, Mike Qin, *Graduate Student Member, IEEE*, Marco Morales, *Member, IEEE*, and Nancy M. Amato, *Fellow, IEEE*

***Abstract*—This work presents Adaptive Robot Coordination (ARC), a novel hybrid framework for multi-robot motion planning (MRMP) that employs local subproblems to resolve inter-robot conflicts. ARC creates subproblems centered around conflicts, and the solutions represent the robot motions required to resolve these conflicts. The use of subproblems enables an innovative, cost-effective hybrid exploration of the multi-robot planning space by dynamically coupling and decoupling necessary subsets of robots only when required and in specific physical locations. This allows ARC to adapt the levels of coordination efficiently by planning in decoupled spaces, where robots can operate independently, and in coupled spaces, where coordination is essential. ARC is probabilistically complete, can be used for any robot, and produces cost-efficient solutions in reduced planning times. Through extensive evaluation across representative scenarios with different robots requiring various levels of coordination, ARC demonstrates its ability to provide simultaneous scalability and precise coordination. ARC is the only method capable of solving all the scenarios and is competitive with coupled, decoupled, and hybrid baselines.**

***Index Terms*—Path planning for multiple mobile robots or agents, multi-robot systems, motion and path planning.**

I. Introduction

MULTI-ROBOT systems (MRS) have gained significant usage in various applications, including payload transportation and manufacturing, due to their ability to enhance productivity and reduce operational costs. This letter addresses the problem of *Multi-robot Motion Planning* (MRMP), a critical component for utilizing MRS. The complexity of MRMP typically arises from the quantity of robots in the problem and the degree of coordination required to tackle the problem. Some applications require low levels of coordination and can be addressed by online, decentralized methods such as [1], [2], [3], however, other applications need high levels of coordination such as multi-manipulator assembly or scenarios with congested regions of mobile robots where online, decentralized methods struggle with problems such as deadlock and livelock, motivating the use of offline, centralized methods for these problems. Here, we consider offline, centralized MRMP to address problems containing high levels of coordination and seek to minimize planning time to reduce the impact of offline planning.

In the existing MRMP literature, algorithms are classified as coupled, decoupled, and hybrid. Coupled methods offer higher coordination but struggle with larger teams and have slower planning times. Decoupled methods are efficient with larger teams and faster in planning but fail when high coordination is needed. Hybrid methods aim to combine the benefits of both while minimizing their weaknesses. Some problems have consistent coordination complexity, and one approach or another can be used appropriately, but often the required coordination varies, leading to planning times being constrained by the highest level of required coordination, often requiring more expensive coupled methods.

In this letter, we present Adaptive Robot Coordination (ARC), an offline, centralized MRMP method designed to minimize planning times by adapting the level of coordination necessary throughout the planning process by dynamically coupling and decoupling necessary subsets of robots only for relevant times and locations. Rather than choosing a single approach to fit the instance of highest coordination, our approach defines subproblems and finds the cheapest method capable of solving the subproblem. Subproblems consist of a subset of the robots, a subset of the environment, and a local start and goal. It is often impossible to know where these subproblems are needed apriori, so we lazily introduce them as we discover a need. We start by generating individual robot paths, and then use any potential conflicts in these optimistic paths to define a subproblem. The solution of a subproblem is used to patch the conflicting paths and resolve the conflict in the proposed motion plan (Fig. 1). We iteratively find and resolve conflicts in this manner until we generate a valid motion plan for the entire MRS.

Prior hybrid MRMP methods can be split into two categories: hybrid search and hybrid representation. Hybrid search methods such as CBS-MP [4] and ECBS-CT [5] utilize decoupled representations (PRM and state-lattice graphs respectively) and

Manuscript received 11 January 2024; accepted 1 June 2024. Date of publication 28 June 2024; date of current version 8 July 2024. This letter was recommended for publication by Associate Editor A. H. Qureshi and Editor A. Bera upon evaluation of the reviewers' comments. The work of Marco Morales was supported by Asociación Mexicana de Cultura A.C. This work was supported in part by the IBM-Illinois Discovery Accelerator Institute and in part by the Center for Networked Intelligent Components and Environments (C-NICE) at the University of Illinois. *(Corresponding author: Nancy M. Amato.)*

Irving Solis is with the Department of Computer Science and Engineering, Texas A&M University, College Station, TX 77840 USA, and also with the University of Illinois Urbana-Champaign, Champaign, IL 61820 USA (e-mail: irvingsolis89@tamu.edu).

James Motes, Mike Qin, and Nancy M. Amato are with the Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL 61801 USA (e-mail: jmotes2@illinois.edu; yudiqin2@illinois.edu; namato@illinois.edu).

Marco Morales is with the Department of Computer Science, University of Illinois Urbana-Champaign, Urbana, IL 61801 USA, and also with the Department of Computer Science, Instituto Tecnológico Autónomo de México (ITAM), Ciudad de México 01080, México (e-mail: moralesa@illinois.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3420548, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3420548

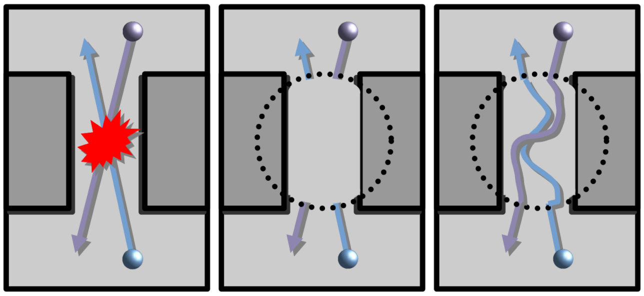


Fig. 1. A simplified overview of our method: (a) Detect a conflict between two robots' paths. (b) Define a local subproblem around the conflict. (c) Use the subproblem's solution to modify the original paths and resolve the conflict.

implicitly search the composite space through an MAPF-like search algorithm. Hybrid representation methods, such as M* [6] and MRd-RRT [7], directly search and construct representations in the composite space of the MRS (or subsets of it) often guided by the decoupled or lower dimensional state space representations. Hybrid representation methods are often able to achieve much higher levels of coordination due to directly searching the composite space (albeit affected by their dependence on lower dimensional representations as we examine in our experiments). Our proposed method, ARC, falls into the hybrid representation category, though we differentiate ourselves through the local search of the composite space defined by our subproblems. This results in less effort on searching the exponentially larger composite space and, thus, faster planning times than the other hybrid representation methods while still achieving high levels of coordination.

We demonstrate the ability of ARC to adapt robot coordination efficiently through a comprehensive set of experiments involving mobile robots, planar manipulators, and 3D manipulators in scenarios that demand low, high, and varying levels of coordination. We compare it against decoupled, hybrid, and coupled baselines. Notably, ARC stands out as the sole method that successfully solves all the scenarios. In the low coordination scenario, ARC performs on par with the decoupled baseline. Similarly, in high coordination scenarios, ARC exhibits behavior similar to the coupled baseline. In scenarios that require varying levels of coordination, ARC outperforms all the baselines.

In summary, our contributions are:

- A novel subproblem-based approach to resolve robot conflicts in MRMP problems.
- A MRMP method that exploits this framework to adapt robot coordination by transitioning different levels of (de)coupled spaces to find feasible solutions to problems with many robots requiring high levels of coordination.
- Experimental evaluation of this proposed method with up to 32 mobile robots and 8 manipulators. Results show efficient robot coordination while maintaining efficient planning times for large teams. Our method is the only method able to solve all the scenarios and it exhibits improved performance over existing methods.

II. Preliminaries and Related Work

In this section, we define the multi-robot motion planning (MRMP) problem. We then discuss sampling-based motion planning and review relevant MRMP algorithms, examining how they explore multi-robot state spaces through coupled, decoupled, and hybrid classes. Finally, we contextualize our method within the current literature.

A. Problem Definition

Motion planning is the problem of finding a valid continuous path for a robot from a start to a goal pose within the configuration space ($\mathcal{C}_{space}$) [8], encompassing all possible robot configurations. A configuration encodes the robot's degrees-of-freedom (DOF), including parameters like position, orientation, joint angles, and velocity.

Multi-robot motion planning (MRMP) extends this to finding feasible paths for multiple robots, each with start and goal poses. Each MRMP problem is defined by $(\mathcal{E}, \mathcal{R}, \mathcal{Q})$, where $\mathcal{E}$ is the environment, $\mathcal{R}$ is the set of robots, and $\mathcal{Q}$ is the set of start and goal positions for each robot.

Solutions involve exploring the composite configuration space, the cross product of all individual robot configuration spaces, denoted as $\mathcal{C}_i$ for each robot $r_i \in \mathcal{R}$ and $\mathcal{C}_{\texttt{composite}}$ for the entire system. A valid composite configuration ensures no robot collides with obstacles or other robots. The composite space can be explored in coupled and decoupled manners. The former involves exploring $\mathcal{C}_{\texttt{composite}}$ directly. The latter implicitly considers $\mathcal{C}_{\texttt{composite}}$ by planning in individual configuration spaces. Since decoupled $\mathcal{C}_{space}$ does not explicitly encode other robots' DOFs, detecting and resolving potential *conflicts* between individual paths is necessary to find a valid solution. A *conflict* occurs when two robots, r_i and r_j, interfere at timestep t while traversing their paths, denoted as $< c_i, c_j, t >$, where c_i and c_j are the conflicting configurations. Coupled methods avoid conflicts as they result in invalid configurations in $\mathcal{C}_{\texttt{composite}}$. A MRMP solution is valid if all robot transition from their start states to their goal states through conflict-free paths.

B. Sampling-Based Motion Planning

As the number of DOFs in a motion planning problem increases, representing the $\mathcal{C}_{space}$ becomes intractable [9], [10]. Sampling-based algorithms like the Probabilistic Roadmap Method (PRM) [11] were developed as approximate solutions, trading completeness for probabilistic completeness. PRMs capture $\mathcal{C}_{space}$ connectivity by sampling graphs, known as roadmaps, where vertices represent valid robot states and edges represent valid transitions between states. Paths are obtained by querying these roadmaps.

C. Coupled methods

Coupled approaches explore the composite space directly to find a path from a start to a goal composite configuration, encoding each individual robot's start and goal. These methods often apply single-robot sampling-based approaches (e.g., PRM [11] or RRT [12]) to the composite space [13], [14], [15], [16], [17], providing probabilistic completeness and high levels of coordination that allow them to address complex problems like mobile robots crossing an inlet (Fig. 6(a)) or planning for tangled robotic arms (Fig. 6(b)). However, these methods are suitable only for small robot teams, as the composite space size grows exponentially with the number of robots.

Multi-agent Pathfinding (MAPF) techniques [18], [19] have been adapted for MRMP problem-solving by exploring a composite state space derived from the Cartesian product of individual representations. However, their effectiveness relies heavily on the quality of individual representations and may falter if they lack essential coordination states.

D. Decoupled Methods

Decoupled methods consider individual $\mathcal{C}_{spaces}$, with each robot's path found by exploring its own $\mathcal{C}_{space}$. Sampling-based techniques are used to construct individual representations (e.g., roadmaps) which are then queried for individual paths. Since decoupled representations are constructed separately, single robot search algorithms cannot collectively reason over them. Consequently, additional measures are required to prevent inter-path conflicts. MAPF algorithms are adapted to address this issue, often employing a prioritized planning method [20], [21], [22], [23], [24], [25], [26], assigning priorities to each robot to avoid conflicts.

E. Hybrid Methods

Hybrid approaches seek to leverage the strengths of both coupled and decoupled methods while minimizing their weaknesses. Most existing hybrid methods construct decoupled representations (e.g., roadmaps in individual robot $\mathcal{C}_{spaces}$) and adapt efficient MAPF hybrid techniques originally designed to resolve grid world problems [6], [27] to the roadmap representation. These MAPF algorithms' consideration of the composite space often revolves around conflict resolution. For example, when paths conflict, M* [6] expands the search space's dimensionality to identify the coupled actions needed to resolve the conflict, while CBS [27] imposes constraints on individual robot state spaces to facilitate the re-planning of collision-free paths.

An extension to MRMP [4], [5] adapts CBS to address sampling-based and state-lattice motion planning problems. In [28], optimization-based Mixed Integer-Linear Programming (MILP) is used to compute individual paths and priority-based planning to coordinate agents and avoid inter-agent collisions. In [7], [29], RRT is employed to navigate through the composite configuration space, guiding a decoupled search over individual roadmaps. Yet, all these hybrid approaches primarily engage in coupled exploration solely to guide a search across decoupled representations, which are independently constructed for each robot, disregarding the team as a whole. Consequently, these decoupled planning spaces may lack the composite states needed for executing cooperative robot motions needed to resolve a conflict.

In contrast, our proposed method introduces local subproblems, effectively coupling and decoupling robot subsets at appropriate times and workspace locations. This enables planning at different levels of state space compositions, facilitating the discovery of new states essential for coordination.

III. Method

In this section, we present the Adaptive Robot Coordination (ARC) approach to the multi-robot motion planning (MRMP) problem. We first provide an overview of the method and how subproblems are used to resolve conflicts. Then, we detail the creation and adaptation of subproblems. Finally, we discuss the theoretical properties of the approach.

A. Overview

ARC is a hybrid MRMP method that employs subproblems to efficiently address conflicts by exploring relevant sections of the planning space. Given a MRMP problem instance $(\mathcal{E}, \mathcal{R}, \mathcal{Q})$, ARC (Algorithm 1) begins by solving each robot's individual motion planning problem $(\mathcal{E}, \mathcal{R}_i = \{r_i\}, \mathcal{Q} = \{q_i\})$ using probabilistically complete sampling-based techniques in order to obtain the initial set of paths. Paths are represented by a sequence of configurations approximating continuous motion, and $p_i(t)$ is the configuration along p_i at timestep t. They are also discretized into uniform time resolution segments called timesteps. Timesteps may vary in length across different robots' roadmaps, but they take the same duration to traverse according to each robot's velocity.

Because paths are computed in independent robot $\mathcal{C}_{spaces}$, we check them for conflicts through standard collision detection at each timestep (Algorithm 1: lines 6,12). Conflicts between paths p_i, p_j are used to create local subproblems $(\mathcal{E}', \mathcal{R}' = r_i \cup r_j, \mathcal{Q}')$ (Algorithm 1: line 8). $\mathcal{Q}'$ is defined by selecting points along p_i, p_j sufficiently before and after the conflict timestep. $\mathcal{Q}'$ is used to define a local region $\mathcal{E}'$ for the subproblem. Sampling-based MRMP solvers are used to find the subproblem solution, each with specific termination criteria. Local paths from the subproblem solution resolve the conflict and are connected to the rest of the initial paths. This process is repeated until all conflicts are resolved.

Algorithm 1: Adaptive Robot Coordination (ARC).

Input: A MRMP problem with an environment $\mathcal{E}$, a set of robots $\mathcal{R}$, a set of queries $\mathcal{Q}$.
Output: A set of valid paths $\mathcal{P}$.

1: $\mathcal{P} \leftarrow \emptyset$
2: **for** each robot $r_i \in \mathcal{R}$ **do**
3: $\quad p_i \leftarrow$ `MotionPlanning`$(\mathcal{E}, \{r_i\}, \{q_i\})$
4: $\quad \mathcal{P} \leftarrow \mathcal{P} \cup \{p_i\}$
5: **end for**
6: $\mathcal{C} =$ `FindConflict`$(\mathcal{P})$
7: **while** $C \neq \emptyset$ **do**
8: $\quad \mathcal{E}', \mathcal{R}', \mathcal{Q}' =$ `CreateSubProblem`$(\mathcal{C}, \mathcal{P}, \mathcal{E})$
9: $\quad \mathcal{P}' =$ `SolveSubProblem`$(\mathcal{E}', \mathcal{R}', \mathcal{Q}')$
10: $\quad$ **if** $\mathcal{P}'! = \emptyset$ **then** $\quad \triangleright$ conflict resolved
11: $\quad\quad$ `UpdateSolution`$(\mathcal{P}, \mathcal{P}')$
12: $\quad\quad \mathcal{C} =$ `FindConflict`$(\mathcal{P})$
13: $\quad$ **else**
14: $\quad\quad \mathcal{P} = C \leftarrow \emptyset$ $\quad \triangleright$ if conflict not resolved,
15: $\quad$ **end if** $\quad \triangleright$ return empty solution
16: **end while**
17: **return** $\mathcal{P}$

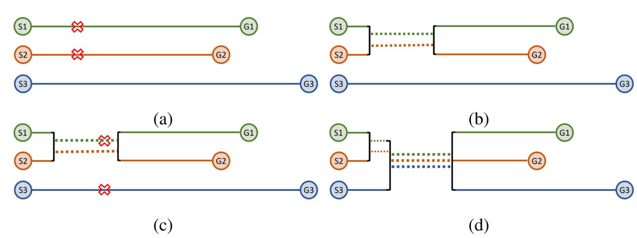


Fig. 2. A three-robot MRMP problem with path timelines indicating conflict times. Dashed lines represent path modifications to resolve conflicts. (a) The initial conflict involves robots 1 and 2, resolved by defining a subproblem. (b) The subproblem's solution resolves the conflict. (c) A second conflict emerges as Robot 3 conflicts with the solution for robots 1-2. (d) A new subproblem, involving robots 1, 2, and 3, is defined and solved.

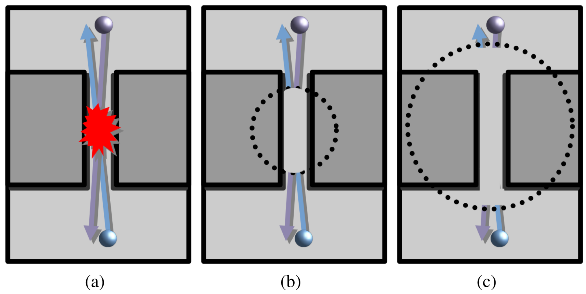


Fig. 3. (a) A conflict between two paths inside a narrow passage. (b) Initial subproblem where no solution exists. (c) Expanding subproblem to find a feasible solution.

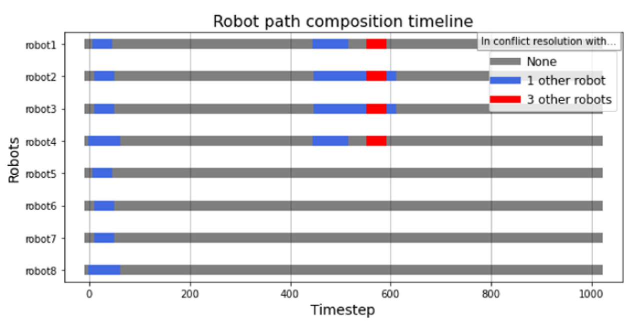


Fig. 4. Solution paths for the scenario shown in Fig. 7(b)–(c). Timelines illustrate each robot's path, with colors indicating segments planned in spaces of different dimensions. Mostly, paths are computed independently within their respective state spaces (gray), but they transition to higher-dimensional spaces occurs when more coordination is needed to resolve conflicts. At certain points, the planning transitions to 2-robot state spaces (blue) to address conflicts between pairs of robots of the same color (see Fig. 7(b)–(c)). Around timestep 600, the planning transitions to a 4-robot state space to resolve a more complex conflict involving four differently colored robots at the center.

If the probabilistically complete, sampling-based planner fails to find a solution within some bounded effort (e.g., time or number of samples), the local subproblem $(\mathcal{E}', \mathcal{R}', \mathcal{Q}')$ is expanded by pushing $\mathcal{Q}'$ further from the conflict on p_i, p_j. $\mathcal{E}'$ is expanded accordingly (Algorithm 2: line 10, Fig. 3), and the algorithm then attempts to solve this expanded subproblem.

The local solution $\mathcal{P}'$ to $(\mathcal{E}', \mathcal{R}', \mathcal{Q}')$ resolves the conflict in p_i, p_j. If $\mathcal{P}'$ conflicts with another subproblem solution, we introduce a new local subproblem that accounts for all conflicting robots (Algorithm 1: line 8, Fig. 2) as discussed in Section III-B.

The final solution yields a set of paths $\mathcal{P}$, where each path p may consist of segments from different planning spaces of various robot compositions (Fig. 4). Each $p_i \in \mathcal{P}$ has a start and end timestep $p_i.t_{\texttt{start}}, p_i.t_{\texttt{end}}$ and configuration $p(p_i.t_{\texttt{start}}), p(p_i.t_{\texttt{end}})$. There exists a $p_{R_i}, p'_{R_i} \in P$ such that $R_i = \{r_i\}$ for all $r_i \in \mathcal{R}$ and $p_{R_i}(0) = q_i.\texttt{start}$ and $p'_{R_i}(t_{\texttt{final}}) = q_i.\texttt{goal}$. If r_i is never found to be in conflict with another robot, then $p_{R_i} = p'_{R_i}$.

Each robot r_i should exist in exactly one path at every timestep except from those where a robot is part of a state space composition change. At every change of robot state space composition from a set of paths $P_{\texttt{pre}}$ to a set of paths $P_{\texttt{post}}$, the end configuration of the pre-transition paths $\prod_{p_i \in P_{\texttt{pre}}} p_i(p_i.t_{\texttt{end}})$ is equivalent to the start configuration of the post-transition paths $\prod_{p_i \in P_{\texttt{post}}} p_i(p_i.t_{\texttt{start}})$.

Algorithm 2: SolveSubProblem.

Input: A subproblem with an environment $\mathcal{E}'$, a set of robots $\mathcal{R}$, a set of queries $\mathcal{Q}'$. A set of MRMP solvers $\mathcal{S}$.
Output: A set of valid local paths $\mathcal{P}'$.

1: $\mathcal{P}' \leftarrow \emptyset$
2: **while** $\mathcal{P}' == \emptyset$ **do**
3:   **for** s **in** $\mathcal{S}$ **do**
4:     $\mathcal{P}' \leftarrow$`SolveMRMP`$(s, \mathcal{E}', \mathcal{R}', \mathcal{Q}')$
5:     **if** $\mathcal{P}' \neq \emptyset$ **then** ▷ subproblem solved
6:       **return** $\mathcal{P}'$
7:     **end if**
8:   **end for**
9:   **if** $\mathcal{E}' \neq \mathcal{E}$ **or** $\mathcal{Q}' \neq \mathcal{Q}$ **then**
10:     `AdaptSubProblem`$(\mathcal{E}', \mathcal{R}', \mathcal{Q}')$
11:   **else**
12:     **return** $\emptyset$ ▷ if subproblem not solved,
13:   **end if** ▷ return empty local solution
14: **end while**

B. Subproblem Creation and Adaptation

Given a set of paths P, and a conflict (c_i, c_j, t) between $p_{R_i}, p_{R_j} \in P$, we define a local subproblem $(\mathcal{E}', \mathcal{R}', \mathcal{Q}')$ around the conflict (Algorithm 1: line 8) where $\mathcal{R}' = R_i \cup R_j$ merges the involved robots. The local query $\mathcal{Q}'$ defines the local start and goal configurations for each robot and is obtained by taking the corresponding configurations located in a time window before and after the conflict timestep $t - \texttt{window}, t + \texttt{window}$, where the time window consists of an initial number of timesteps.

The local region $\mathcal{E}'$ is defined by a $\mathcal{C}_{space}$ boundary encapsulating $\mathcal{Q}'$. This allows the planning methods to focus the $\mathcal{R}'$ composite space search on a local region around the conflict.

If a solution is not found for $(\mathcal{E}', \mathcal{R}', \mathcal{Q}')$, the local problem is adapted by expanding $\mathcal{Q}'$ and $\mathcal{E}'$(Algorithm 2: line 10). This involves the continuous advancement of the query points along their respective paths, which in turn expands the local environment. This expansion is repeated until a solution is found or $\mathcal{E}' = \mathcal{E}$ and $\mathcal{Q}' = \mathcal{Q}$ at which point the method terminates with no solution (Algorithm 2: line 12).

If additional robots need to be incorporated to resolve the current conflict, ARC expands $\mathcal{R}'$ to account for all the involved robots, ensuring a feasible resolution for all of them. This can occur in instances where resolving one conflict invalidates a prior conflict resolution.

C. Subproblem Planning

Subproblems focus computational effort on conflict resolution, and since not all conflicts require the same level of coordination, ARC adapts the method's complexity accordingly. We employ a hierarchy of MRMP methods (Algorithm 2, line 3), resorting to more expensive methods only when necessary. The framework is designed to incorporate various strategies for different levels of coordination, and we have chosen the simplest ones to demonstrate its functionality. Our experiments use the following hierarchy:

- Prioritized Query (no sampling)
- Decoupled PRM (sampling individual robot $\mathcal{C}_{spaces}$)
- Composite PRM (sampling composite $\mathcal{C}_{space}$)

Initially, conflicts are resolved using decoupled queries of existing representations, often involving robots waiting for others. If the current representation is insufficient, Decoupled PRM expands the individual roadmap representations, allowing robots to find new paths to avoid collisions. When these methods fail, Composite PRM builds and searches a representation of the composite space directly to find more coordinated motions. If this final layer fails, the local subproblem is expanded.

As both sampling methods are probabilistically complete, termination criteria are necessary to proceed to the next approach. This can be adjusted to control the effort each method spends. For decoupled approaches, we found it is better to fail fast and escalate the coordination level or subproblem scope. For Composite PRM, we use metrics from [30] to develop adaptive termination criteria that assess progress in exploring the composite space and set limits on new state exploration. In Section V, we explore scenarios in which each local method provides the desired level of coordination to resolve different types of conflicts.

We use this hierarchy of MRMP methods with increasing coupling as a straightforward approach to demonstrate the adaptable coordination. First, ARC increases the coordination the local method used, then, if it fails to solve the subproblem, ARC expands subset of the robots' environment, effectively increasing coordination with respect to $\mathcal{C}_{space}$ composition. Coordination is then decreased after the resolution of a conflict. We plan to develop heuristics to choose an appropriate method for the local features of the subproblem considering more specialized strategies like [31], [32], which excel in congested environments with many mobile and manipulator robots, respectively.

D. Theoretical Properties

When resolving a conflict, if the method fails to find a solution to a local subproblem, the subproblem is expanded. In the worst case, it is expanded with respect to robots, query, and environment until it matches the original problem. At this point, the completeness of the approach depends on the methods in the planning hierarchy. If the hierarchy includes a probabilistically complete method (e.g., Composite PRM) and the termination criteria allow continued searching, the approach is probabilistically complete.

ARC lacks an optimality guarantee, even with an asymptotically optimal method in the hierarchy, due to local conflict resolution. If a better resolution exists outside the local subproblem, no method in the hierarchy will find it.

IV. Experiments

In our experiments, we evaluate three types of coordination problems: high coordination, low coordination, and mixed coordination. For problems requiring exclusively high or low coordination, ARC adapts to the appropriate level and performs on par with existing (de)coupled methods tailored to those problem classes. For mixed coordination problems, only ARC finds solutions within the allotted time by adapting the coordination level throughout the planning process. Additionally, we provide a brief analysis of how ARC adapts to the local features of the environment by showing how the distribution of the number of robots involved in a subproblem changes depending on environmental characteristics.

A. Experimental Setup

For each of the three problem classes, we evaluate two scenarios: one involving mobile robots and the other featuring manipulators. We compare against decoupled (Decoupled PRM [13]), hybrid (MRdRRT [7], CBS-MP [4]), and coupled (Composite PRM [13]) baselines. Decoupled PRM operates on a set of decoupled roadmaps and employs a prioritized planning approach with random priority ordering to discover feasible paths. If no path is found, roadmaps are refined. Likewise, MRdRRT initially samples individual roadmaps for each robot, combining them into a tensor roadmap to represent the composite space. It then employs specific heuristics to guide the exploration of the composite space towards the goal to find solutions faster. CBS-MP also samples individual roadmaps and then utilizes the CBS framework to query them. As detailed in [4], it balances exploration of the Conflict Tree and roadmap refinement to enhance performance. Composite PRM constructs an explicit composite roadmap.

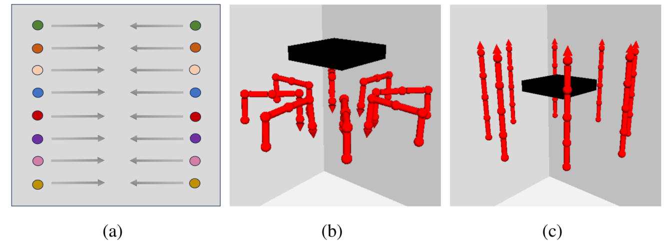


Fig. 5. Low coordination scenarios. (a) Pairs of robots on the same row must switch positions (scaled up for visibility.) (b) Manipulators' start configurations. (c) Manipulators' goal configurations.

We conducted 33 random trials for each scenario, allotting 1000 seconds per trial for planning; any trial exceeding this limit was considered a failure. We compared the planning time to find the first solution for each method. To focus on planning time, we did not implement the rewiring component of the MRdRRT algorithm, as the additional computation was unnecessary for finding a solution. We also reported the solution cost to provide a comprehensive view of the algorithm behaviors. Solution cost, calculated as the sum of all path timesteps including waiting time, assumes that robots move at the same velocity for ease of implementation.

B. Scenarios

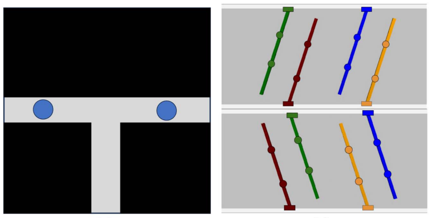
1) Scenario I: Low Coordination: In this scenario, we examine instances that require minimal or no coodination. ARC proves competitive against decoupled approaches, showcasing effective scalability in these contexts.

For mobile robots, we examine a scenario where robots must switch positions horizontally, leading to numerous conflicts that do not demand high coordination (Fig. 5(a)). The problem is scaled by doubling the number of robots on each side.

For manipulator robots, we address the challenge of untangling the robots from a start (Fig. 5(b)) to a goal position (Fig. 5(c)). The problem is scaled by doubling the number of manipulators in a ring pattern.

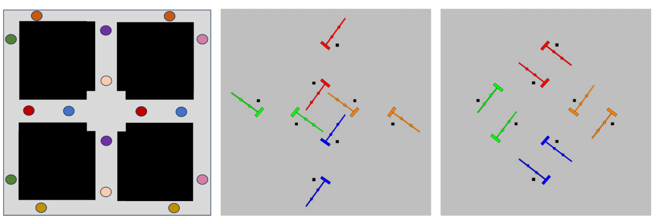
2) Scenario II: High Coordination: In this scenario, we investigate situations requiring higher levels of coordination. We showcase that ARC provides the necessary coordination and competes effectively against a pure coupled approach, which excels in these scenarios.

For mobile robots, we examine a scenario with two robots swapping positions in a narrow passage with a central inlet



(a) (b)

Fig. 6. High coordination scenarios. (a) Two robots need to switch positions in the inlet scenario (scaled up for visibility.) (b) Pairs of manipulator arms needs to swing from left to right/right to left to reach their goal positions which requires one to move out of the way first.



(a) (b) (c)

Fig. 7. Adaptive coordination scenarios where conflicts involving varying numbers of robots are likely to occur. (a) Mobile robots of the same colors must switch positions (scaled up for visibility.) (b) Start and (c) goal configurations (the blocks at the end represent the bases) for the first manipulator scenario.

(Fig. 6(a)). Only one robot can go through the passage at a time, thus a solution demands precise coordination.

For manipulator robots, we examine 3-dof planar manipulators positioned oppositely (Fig. 6(b)), with the top manipulators moving right and the bottom ones moving left. Precise coordination is required, as robots need to contract themselves, creating enough space to avoid collisions. We consider scenarios with both 2 and 4 manipulators.

3) Scenario III: Adaptive Coordination: In real-world applications, the required coordination between robots will likely be unknown. This scenario depicts realistic conditions where different coordination levels are needed at different stages of the problem.

In the case of mobile robots, we examine 16 robots in a warehouse with narrow passages (Fig. 7(a)). The number of robots involved in conflicts changes, necessitating varying levels of coordination. Two-robot conflicts typically arise along passages, while four-robot conflicts are more likely to occur in the center.

Regarding manipulators, we examine a scenario involving eight 3-dof planar manipulators with start and goal poses depicted in Fig. 7(b) and (c). Initially, the four central manipulators encounter a 2-robot conflict with those on the outer rim. As the inner manipulators rotate towards the center, all four become involved in a conflict that requires higher coordination for resolution.

4) Study of Coordination Adaptation to Local Features: In this study, we examine how ARC adapts robot coordination based on environmental features. We aim to show that ARC increases coupling in areas with higher robot congestion. Using three environments that induce different congestion levels, and the same start and goal positions as the Low Coordination scenario with 32 robots (Fig. 5(a)), we conducted successful 100 trials for each environment. We measured the total conflicts and the distribution of coupled robots needed for resolution.

TABLE I
SUCCESS RATES FOR ALL EVALUATED SCENARIOS

Method	Succes Rate per Scenario (%)												
	Low								High			Adaptive	
	Mob					Man			Mob	Man		Mob	Man
	2	4	8	16	32	2	4	8	2	2	4	16	8
ARC	100	100	100	100	100	100	100	100	100	100	100	100	100
MRdRRT	100	59	-	-	-	100	68	-	21	100	100	-	-
CBS-MP	100	100	100	36	-	100	100	100	66	-	-	-	-
Decoupled PRM	100	100	100	100	100	100	100	100	-	-	-	-	-
Composite PRM	100	100	100	24	-	100	100	27	100	100	100	-	-

C. Results

1) Scenario I: Low Coordination: For mobile robots, ARC and Decoupled PRM improved scalability, being the only methods able to solve all trials for 32 robots. (Fig. 8). This is because ARC is decoupled most of the time, as conflicts usually involve only a few robots. We provide a deeper study about this behavior at the end of this section. Given its thorough search for optimal solutions, CBS-MP successfully completed all trials for 8 robots but only achieved a 36% success rate for trials with 16 robots. Likewise, due to its coupled behavior, Composite PRM can successfully plan for all trials involving 8 robots but achieves only a 24% success rate for trials with 16 robots (Table I). MRdRRT successfully plans for two robots but only achieves a 59% success rate for four robots (Table I). This lower success rate is due to its indirect exploration of the composite space. MRdRRT queries a tensor roadmap composed of individual representations, sampled independently, which may not contain the team solution and require further refinement. Even when roadmaps are sufficient, the greedy heuristics in MRdRRT often drive the search back to conflict within the existing representation, as conflict resolution essentially involves taking a random step off the path before resuming greedy behavior.

For manipulators, ARC and CBS-MP exhibit the best planning times, as they can provide better coordination and scalability (Fig. 9). CBS-MP produces better solution costs due to its optimality, while ARC's solutions, though slightly more expensive, remain competitive. DecoupledPRM can also solve all the trials but at a slower pace, as it requires the exploration of more decoupled states. CompositePRM can only solve a 27% of the trials for eight robots due to its coupled behavior (Table I). MRdRRT successfully solved all instances for two robots, but only 68% out of the trials for four robots. Once more, this is attributed to the mostly greedy exploration of the composite space through the tensor roadmap (Table I).

2) Scenario II: High Coordination: For mobile robots, ARC and CompositePRM are the only methods that solved all trials (Fig. 10(a)). They directly explore the composite space, enabling the discovery of necessary coupled transitions. ARC produces better cost solutions by focusing exploration in the inlet region. CBS-MP solved only 66% of trials because its decoupled roadmaps sometimes lack necessary states for coordinating robots at the inlet (Table I). Similarly, MRdRRT, exploring a tensor roadmap of individual representations, solved only 21% of trials due to missing necessary states over the inlet. DecoupledPRM failed to solve all trials due to its lack of coordination (Table I).

For manipulators, ARC, CompositePRM, and MRdRRT are the only methods capable of solving the trials. ARC and CompositePRM demonstrate comparable performance in planning time and cost solutions, with ARC having slightly higher planning times due to resolving conflicts before integrating the four

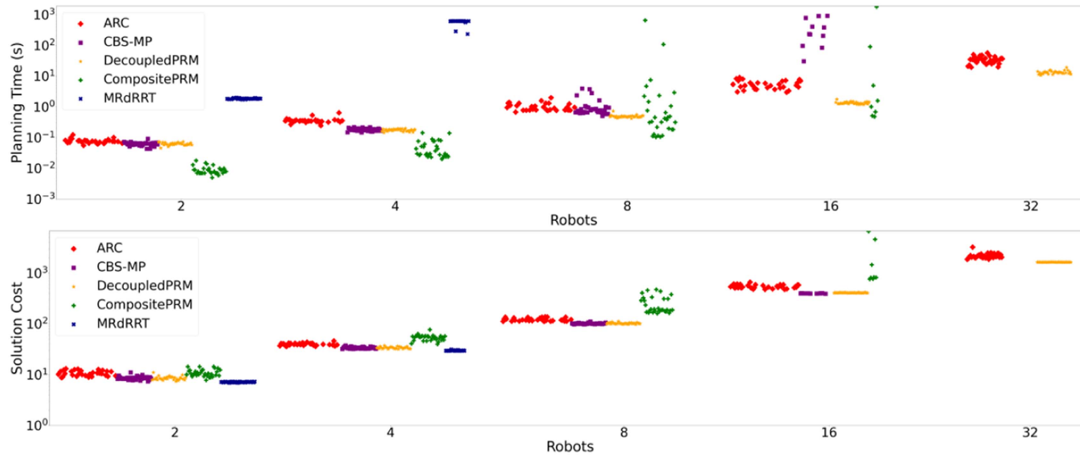


Fig. 8. Results for scenario II: low coordination mobile robots.

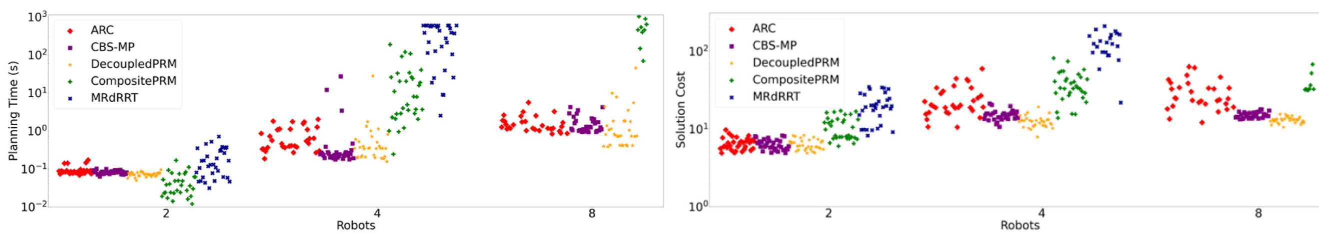
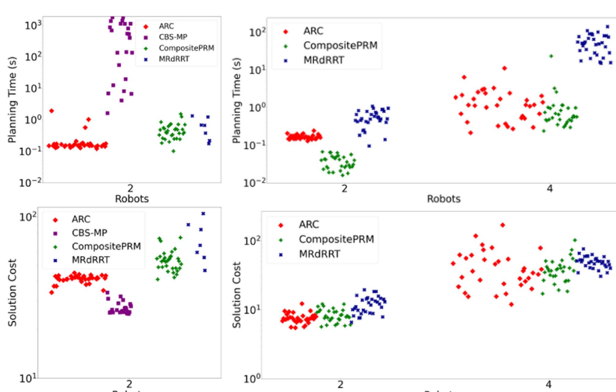


Fig. 9. Results for scenario II: low coordination manipulator robots.



(a) (b)

Fig. 10. Results for Scenario III: (a) High Coordination Mobile, (b) High Coordination Manipulator.

robots, while CompositePRM plans them simultaneously. The three methods using decoupled roadmaps all struggle. MRdRRT solves the problem through its tensor product roadmap search, though with longer planning times than methods directly sampling the composite space. CBS-MP's iterative conflict tree expansion takes too long to search decoupled representations, and Decoupled PRM fails due to its incomplete nature, limiting feasible coordination.

3) Scenario III: Adaptive Coordination: In all adaptive scenarios (Table II), only ARC generates feasible solutions. This is because ARC dynamically adapts robot coordination to resolve diverse conflicts, each requiring different levels of coordination. Other methods fail due to coordination deficiencies or their inability to scale well with a larger number of robots. In Fig. 4, we illustrate how ARC's final paths (Fig. 7(b)–(c)) result from transitioning through path segments planned in state spaces of different dimensionalities.

TABLE II
RESULTS FOR THE ADAPTIVE COORDINATION SCENARIOS

Robot type	Number of robots	Method	Planning Time (s)		Solution Cost		Success rate
			Avg	Std dev	Avg	Std dev	
Mobile	16	ARC	24.5	20.5	450.1	19.6	100%
Manipulator	8	ARC	9.4	6.1	316.5	121.5	100%

In mobile robot scenarios, ARC's planning time varies due to conflicts requiring high coordination. The use of a sampling-based method for conflict resolution introduces randomness, increasing variability as more conflicts are addressed. In manipulator robot scenarios, ARC's solution costs vary because ARC stops local exploration when conflict resolution is found. Due to the complexity of manipulator planning, achieving lower-cost conflict resolutions requires deeper exploration of the planning space, inevitably increasing planning time, which is not the focus of this work.

4) Study of Coordination Adaptation to Local Features: Our study reveals that ARC is highly effective and reactive to local environmental features. Results shown in Fig. 11 demonstrate that ARC effectively adapts robot coordination by considering a higher number of robots in areas with high congestion, coupling up to 18 robots in such environments. Most importantly, ARC maintains efficiency in conflict resolution by generally requiring only two robots, reducing computational and operational load. Remarkably, across all congestion levels, approximately 70% of conflicts are resolved by only two robots, underscoring ARC's efficiency in managing most conflicts with minimal robot coupling.

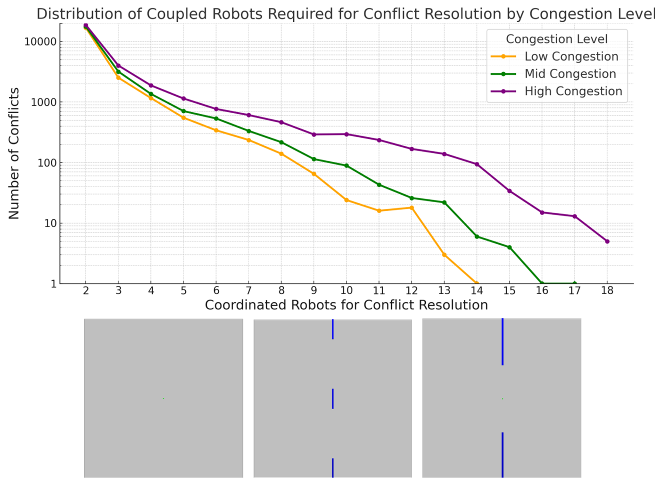


Fig. 11. Environments inducing various levels of robot congestion: (Left) Low congestion, where robots can pass anywhere; (Center) Mid congestion, with passage restricted to two corridors; (Right) High congestion, with passage restricted to one corridor.

V. Conclusion

In this letter, we introduce ARC, a novel hybrid MRMP method that employs a subproblem-based approach for resolving robot conflicts. ARC efficiently explores the extensive multi-robot planning space by introducing local subproblems, enabling a cost-effective exploration of relevant regions within the composite space. The solutions to these subproblems depict the appropriate robot motions for conflict resolution, allowing ARC to rapidly adapt subproblems to plan for the necessary robots and physical space. The results demonstrate ARC's ability to offer simultaneous scalability and coordination across various scenarios. In comparison to the decoupled baseline, ARC exhibits competitive scalability and effectively competes in terms of coordination against the coupled approach. In scenarios featuring a large number of robots with varying coordination requirements, ARC stands out as the only method capable of finding solutions, thanks to its capacity to quickly adapt subproblems for resolving diverse types of conflicts.

References

[1] I. Draganjac, D. Miklić, Z. Kovačić, G. Vasiljević, and S. Bogdan, "Decentralized control of multi-AGV systems in autonomous warehousing applications," *IEEE Trans. Automat. Sci. Eng.*, vol. 13, no. 4, pp. 1433–1447, Oct. 2016.

[2] I. Draganjac, T. Petrović, D. Miklić, Z. Kovačić, and J. Oršulić, "Highly-scalable traffic management of autonomous industrial transportation systems," *Robot. Comput.- Integr. Manuf.*, vol. 63, 2020, Art. no. 101915.

[3] F. Pratissoli, R. Brugioni, N. Battilani, and L. Sabattini, "Hierarchical traffic management of multi-AGV systems with deadlock prevention applied to industrial environments," *IEEE Trans. Automat. Sci. Eng.*, early access, May 24, 2023, doi: 10.1109/TASE.2023.3276233.

[4] I. Solis, J. Motes, R. Sandstrom, and N. M. Amato, "Representation-optimal multi-robot motion planning using conflict-based search," *IEEE Robot. Automat. Lett.*, vol. 6, pp. 4608–4615, Jul. 2021.

[5] L. Cohen, T. Uras, T. S. Kumar, and S. Koenig, "Optimal and bounded-suboptimal multi-agent motion planning," in *Proc. 12th Int. Symp. Combinatorial Search*, 2019, vol. 10, pp. 44–51.

[6] G. Wagner and H. Choset, "Subdimensional expansion for multirobot path planning," *Artif. Intell.*, vol. 219, pp. 1–24, 2015.

[7] R. Shome, K. Solovey, A. Dobson, D. Halperin, and K. E. Bekris, "dRRT*: Scalable and informed asymptotically-optimal multi-robot motion planning," *Auton. Robots*, vol. 44, pp. 443–467, 2019.

[8] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Commun. ACM*, vol. 22, pp. 560–570, Oct. 1979.

[9] J. F. Canny, *The Complexity of Robot Motion Planning*. Cambridge, MA, USA: MIT Press, 1988.

[10] J. H. Reif, "Complexity of the mover's problem and generalizations," in *Proc. IEEE Symp. Foundations Comput. Sci.*, 1979, pp. 421–427.

[11] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Trans. Robot. Automat.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.

[12] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Iowa State Univ., Ames, IA, Res. Rep. 9811, 1998.

[13] G. Sanchez and J.-C. Latombe, "Using a PRM planner to compare centralized and decoupled planning for multi-robot systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 2112–2119.

[14] S. Carpin and E. Pagello, "On parallel RRTs for multi-robot systems," in *Proc. Italian Assoc. AI*, 2002, pp. 834–841.

[15] D. Ferguson, N. Kalra, and A. Stentz, "Replanning with RRTs," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2006, pp. 1243–1248.

[16] M. Čáp, P. Novák, J. Vokrínek, and M. Pěchouček, "Multi-agent RRT*: Sampling-based cooperative pathfinding," in *Proc. Int. Conf. Auton. Agents Multi-Agent Syst.*, 2013, pp. 1263–1264.

[17] M. Saha and P. Isto, "Multi-robot motion planning by incremental coordination," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 5960–5963.

[18] J. Pearl, *Heuristics: Intelligent Search Strategies for Computer Problem Solving*. Upper Saddle River, NJ, USA: Addison-Wesley Longman Publishing Co., Inc., 1984.

[19] P. Švestka and M. H. Overmars, "Coordinated path planning for multiple robots," *Robot. Auton. Syst.*, vol. 23, no. 3, pp. 125–152, 1998.

[20] R. Regele and P. Levi, "Cooperative multi-robot path planning by heuristic priority adjustment," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 5954–5959.

[21] J. P. van den Berg and M. H. Overmars, "Prioritized motion planning for multiple robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2005, pp. 430–435.

[22] R. Cui, B. Gao, and J. Guo, "Pareto-optimal coordination of multiple robots with safety guarantees," *Auton. Robots*, vol. 32, no. 3, pp. 189–205, 2012.

[23] M. Cáp, P. Novák, M. Selecky, J. Faigl, and J. Vokffnek, "Asynchronous decentralized prioritized planning for coordination in multi-robot system," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 3822–3829.

[24] V. R. Desaraju and J. P. How, "Decentralized path planning for multi-agent teams with complex constraints," *Auton. Robots*, vol. 32, no. 4, pp. 385–403, 2012.

[25] A. Geramifard, P. Chubak, and V. Bulitko, "Biased cost pathfinding," in *Proc. 2nd Artif. Intell. Interactive Digit. Entertainment Conf.*, 2006, pp. 112–114.

[26] M. Muthiah and A. Saad, "Multi robot path planning and path coordination using genetic algorithms," in *Proc. SouthEast Conf.*, 2017, pp. 112–119.

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

[28] J. Chen, J. Li, C. Fan, and B. C. Williams, "Scalable and safe multi-agent motion planning with nonlinear dynamics and bounded disturbances," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 11237–11245.

[29] K. Solovey, O. Salzman, and D. Halperin, "Finding a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning," *Int. J. Robot. Res.*, vol. 35, pp. 501–513, 2016.

[30] M. A. Morales A, R. Pearce, and N. M. Amato, "Metrics for analyzing the evolution of C-Space models," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2006, pp. 1268–1273.

[31] C. McBeth, J. Motes, D. Uwacu, M. Morales, and N. M. Amato, "Scalable multi-robot motion planning for congested environments with topological guidance," *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 6867–6874, Nov. 2023.

[32] Y. Shaoul, I. Mishani, M. Likhachev, and J. Li, "Accelerating search-based planning for multi-robot manipulation by leveraging online-generated experiences," in *Proc. Int. Conf. Automated Plan. Scheduling*, 2024, pp. 523–531.