A Synchronized Task Formulation for Robotic Convoy Operations

Charles Noren[1], Bhaskar Vundurthy[1], Sebastian Scherer[1], Howie Choset[1], and Matthew Travers[1]

***Abstract*—Future ground logistics missions will require multiple robots to travel in a convoy between locations. As each location may require a different number of robots (e.g. resupply vehicles), these missions will require a mutable convoy formation structure that may be divided to meet operational needs at each location. We model this mission type by modifying the vehicle routing problem with multiple synchronizations (VRPMS) to enforce convoy constraints (VRPMS-CC). This centralized approach to organizing and routing convoys is represented as a graph-based routing problem and then solved as a mixed integer program. A solution of the VRPMS-CC forms convoys by ensuring that agents participating in the same convoy remain spatially and temporally coupled, traversing the same edge of the graph simultaneously. We demonstrate our approach through numerical studies, where we route up to six simulated agents through twenty convoying tasks, and on robotic hardware. These demonstrations motivate two further contributions to specialize our approach to robotic systems. We introduce: 1) a warm-starting heuristic that improves solver times by up to eighty-nine percent and 2) an online multi-depot variant of the VRPMS-CC that responds to *a priori* unknown impassable environmental obstacles.**

***Index Terms*—Logistics, multi-robot systems, formation routing, planning, scheduling and coordination**

I. INTRODUCTION

FOR hundreds of years, convoy formations have served as the backbone of large economic and military transport operations [1]. Although prior work has addressed challenges in conducting robotic convoy operations, autonomously generating convoy formations (or "convoy details") and allocating routes to multiple robots remains a daunting challenge for complex missions [2]. In complex missions, the number of required robotic platforms may vary location-to-location (e.g., higher-risk areas), requiring multiple details or modifications to the convoy formation topology. Modeling these missions and developing formation-aware routing algorithms are the next steps toward solving this challenge.

The key challenge for formation-aware routing algorithms is synchronizing the routes of individual agents participating in a formation. Classically, each route is constructed from a sequence of tasks that an agent must visit, but few works require the synchronized action of agents when traveling between tasks. As the optimal allocation of routes to agents is a known NP-hard problem [3], adding additional synchronization constraints can make an already difficult problem even harder.

Manuscript received: December, 31, 2024; Revised April, 2, 2025; Accepted April, 30, 2025.

This paper was recommended for publication by Editor Chao-Bo Yan upon evaluation of the Associate Editor and Reviewers' comments.

[1] All authors are from the Robotics Institute, School of Computer Science at Carnegie Mellon University, Pittsburgh, PA, USA {cnoren; pvundurt; basti; choset; mtravers}@andrew.cmu.edu

Digital Object Identifier (DOI): see top of this page.

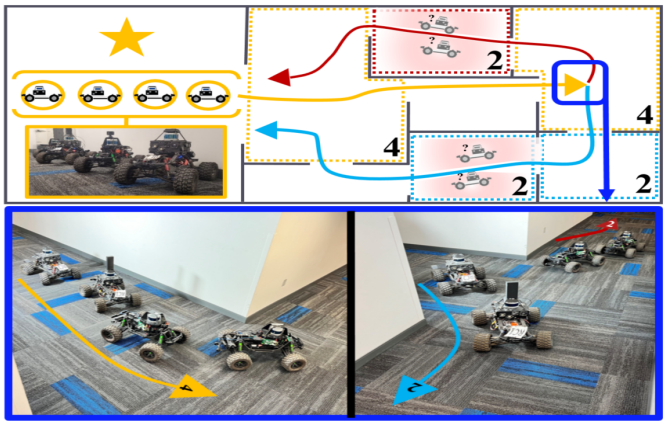


Fig. 1. A team of four agents routing through a sample floor plan (top) where each room in the floor plan may require a different number of agents (black numeral) and the agents must travel as a convoy. The approach forms a singular large convoy (gold) before dividing into two convoys (red, blue) containing two agents each. The blue box depicts two points in time at the location of the division. The image at the earlier time (left) captures the arrival of the four-agent convoy, and the second image (right) depicts the convoy's division into two two-agent convoys.

While multiple works enforce synchronization constraints between a pair of agents, convoy formation topologies contain more than two agents. These topologies add complexity to the convoy operation as they require the synchronization of multiple agents assigned to the formation.

To address this variable formation topology systematically, we introduce a set of "convoy constraints" to the vehicle routing problem with multiple synchronizations (VRPMS). These convoy constraints synchronize the motions of multiple agents into different-sized convoys between locations of interest. Our first contribution is this VRPMS variant, known as the vehicle routing problem with multiple synchronizations - convoy constraints (VRPMS-CC). The solution to an instance of the VRPMS-CC is an optimal set of routes for all participating agents. To our knowledge, this is the first work to employ VRPMS for convoy operations.

The convoy constraints inherently impose a spatial and temporal structure, effectively clustering agents into synchronized groups (treated as distinct depots). This structure allowed us to formulate a multi-depot vehicle routing problem (MDVRP-SV) heuristic, our second contribution, which capitalized on this clustering, resulting in a reduced search space and up to an 89% reduction in solver time.

Finally, these planned routes can face disruptions when *a priori* unknown impassable obstacles arise in multi-robot deployments. To address this, our final contribution is the Dynamic VRPMS-CC (DVRPMS-CC), an approach that dynamically resolves the VRPMS-CC in response to extroceptive stimuli reported by the robots. Demonstrated through hardware trials with our custom fleet (Fig. 1), this approach ensures

robust convoy operations in real-world environments.

II. LITERATURE REVIEW

The core of formation-aware routing approaches for robotic convoys resides in routing-based synchronization [4], where mathematical models enforce coordinated agent movement—the primary focus of this paper. However, to contextualize our approach, we will also examine multi-agent pathfinding (MAPF), market-based synchronization, multi-agent reinforcement learning (MARL), and dynamic grouping. These alternative methods, while advanced, often lack the en-route synchronization and dynamic formation management crucial for convoy operations.

A. Routing-based Synchronization

Routing problems often employ synchronization constraints within their mathematical models to couple the actions of agents. We focus on two primary types: movement (spatial) and operational (temporal) synchronization, both relevant to convoy operations. Movement constraints coordinate agents along the same route, forming dynamic teams. Research on these constraints is limited, with the tractor-trailer routing problem (TTRP) being a notable exception [5]. While the TTRP, which involves heterogeneous tractor and trailer agents, shares similarities with our problem, its tractor-trailer pairing differs from our flexible homogeneous convoy formations. On the other hand, operational constraints synchronize agent arrival times, common to interlogistical applications such as the Traveling Salesman Problem with Drone (TSP-D) [6]. The TSP-D problem seeks to periodically synchronize the motion of a drone with a ground agent. However, unlike our convoy operations, the TSP-D allows for independent operation and reward collection. Column generation and branch-and-cut algorithms [7], [8] are commonly used for exact solutions, while heuristics such as local search and greedy techniques address larger instances [6], [7].

B. Synchronization in multi-agent pathfinding

Synchronization in MAPF is largely represented by two works: 1) multi-agent teamwise-cooperative path finding (MA-TC-PF) [9] and 2) cooperative MAPF (Co-MAPF) [10]. These works seek to assign and plan collision-free paths for teams of agents to different goal locations. MA-TC-PF utilizes a modified conflict-based search (CBS) to optimize actions for pre-organized teams, which makes it unsuitable for our problem, where team formation is a core component. Alternatively, Co-MAPF performs simultaneous task assignment and pathfinding using a modified CBS for synchronization, but limits this to agent pairs at meeting locations, similar to TSP-D solutions. Co-MAPF works primarily focus on intralogistical applications and do not address en-route synchronization needed for convoy operations.

C. Market-based Synchronization

Market-based approaches utilize distributed auctioning protocols for decentralized task assignment, incorporating synchronization constraints within their auction mechanisms [11]. While potentially underperforming traditional routing methods [12], they offer scalability and inherent decentralization. Designing robust auction mechanisms poses challenges due to potential agent misrepresentation and prioritization of individual rewards at the team's expense. Similar to the Co-MAPF, market-based approaches often prioritize operational synchronicity at meeting points rather than continuous en-route synchronization, limiting their applicability in scenarios requiring a maintained formation.

D. MARL Synchronization and Dynamic Grouping

Cooperative MARL learns policies for agent coordination. While scalable and adaptive, these policies often lack guaranteed coordination and may converge to sub-optimal solutions. Given the training complexity, many MARL methods focus on solving sequential rather than combinatorial tasks [13]. Deep reinforcement learning applied to TSP-D [14] uses a hybrid attention long short-term memory model for coordination. However, its effectiveness with larger numbers of agents (only two are considered in [14]) or highly-coupled rewards is unclear. Furthermore, the learned TSP-D policies do not generalize well to basic TSPs, indicating strong problem constraint influence on policy performance. Dynamic grouping, a crowd simulation technique, coordinates simulated agents into dynamic sub-crowds based on relative movements of local agents [15]. This method assumes pre-defined task assignments, similar to MA-TC-PF, and lacks a mechanism to enforce required group sizes.

III. PROBLEM FORMULATION

In a VRP, multiple vehicles (platforms) are tasked with visiting a set of locations in an environment. The platforms usually start and end at a specific location known as a "depot." A classic extension to the VRP is the VRP with Time Windows (VRPTW)s [16], which incorporates two additional time-based constraints. These constraints are: 1) each location must be visited during a specified time window, and 2) each location must be visited for a set amount of time (known as the "service time").

The proposed VRPMS-CC model is a further extension of the VRPTW that adds the requirement that multiple platforms must visit a given location while traveling in a convoy. The number of required platforms is defined as a location's "support value", which may differ between locations. In order to ensure that the agents travel as a single convoy between locations, two additional constraints are placed on the platform's motion: C1) all platforms in a convoy that arrive at a location must depart from the same location, and C2) all platforms in a convoy must arrive at a location together (temporally). A reason for such a motion restriction includes the requirement that convoy elements travel as a formation to defend themselves [17].

For future work, we introduce the concept of "clustering rules" as a generalization of convoy motion constraints C1 and C2. These rules serve as a customizable layer, enabling the VRPMS-CC to be tailored to specific problem contexts (e.g., relaxing C1 could liberate some routes from requiring

convoy formation, while modifying both C1 and C2 could enable compositional team formation at different locations.)

IV. PROBLEM MODELING

In this section, we formulate the VRPMS-CC as a mixed integer linear program (MILP) in which a group of robotic platforms must be assigned to convoy details.

Consider a homogeneous fleet of robotic platforms, $\mathcal{K} = \{k_1, k_2, \ldots, k_{|\mathcal{K}|}\}$, which must complete a set of convoy details that end at locations: $\mathcal{N} = \{n_1, n_2, \ldots, n_{|\mathcal{N}|}\}$. To complete a convoy detail, a convoy containing $v_i \in \mathbb{Z}^+$, $v_i \leq |\mathcal{K}|$ platforms must be assigned to visit the location. We define v_i as the support value of the convoy task that ends at the location n_i and define set $\mathcal{V} = \{v_1, \ldots, v_{|\mathcal{N}|}\}$. All platforms are required to begin and end their routes at a depot location, denoted d^- and d^+, respectively. Thus, for each robotic agent k, the action sequence $r_k = (l_0, l_1, l_2, \ldots, l_g, l_{g+1})$ with $l_0 = d^-$, $l_{g+1} = d^+$, and $L = \{l_1, \ldots, l_g\} \subseteq \mathcal{N}$ not only describes the motion of the platform, but also implicitly encodes the assignment of agent k to a convoy detail. To enforce convoy motion constraints, we propose a routing model that atomizes a convoy detail into elements that may be assigned to an individual vehicle. Convoys are then implicitly modeled by synchronizing the movements of individual vehicles across each vehicle's action sequence.

The task-based model is generated using the location set $\mathcal{N}$ and the support value of each location. Consider a set $\mathcal{S}$ where the elements of $\mathcal{S}$ are sets that describe the atomized elements ("tasks") of a convoy detail. A "task set" for a convoy detail that ends at location n_i is defined as: $\mathcal{S}_i = \{\{s_i^{(1)}, \ldots, s_i^{(v_i)}\}\}$. The same task representation is also made with respect to the depot, with $v_{d^-} = v_{d^+} = v_{|\mathcal{K}|}$. This produces two depot task sets: $\mathcal{S}_{d^-} = D^-$ and $\mathcal{S}_{d^+} = D^+$. The set of all non-depot tasks is $\mathcal{S}_T = \{\bigcup_{i \in \{1,\ldots,|\mathcal{N}|\}} \mathcal{S}_i\}$ and the set of all tasks is $\mathcal{S} = \mathcal{S}_T \cup D^- \cup D^+$.

The task information above may be structured into a connected weighted symmetric task graph $G_t = (V_t, E_t)$, where the vertex set $V_t = \{s | s \in \mathcal{S}_i \in \mathcal{S}\}$ consists of all tasks contained in $\mathcal{S}$. The edge set $E_t = \{e = \{s_i, s_j\} = \{s_j, s_i\} | s_i \in \mathcal{S}_p, s_j \in \mathcal{S}_q, \mathcal{S}_p, \mathcal{S}_q \in \mathcal{S}, p \neq q\}$ has an associated traversal cost $c_{ij}^k \in \mathbb{R}^+ \, \forall \, (i,j) \in E_t$, which is incurred when a platform moves between tasks i and j. Additionally, associated with each convoy detail ending at location n_i (and thus all tasks in the corresponding $\mathcal{S}_i \in \mathcal{S}$) is a time window for completion $TW_i = [a_i, b_i]$ and required service time, st_i, which denotes how long a task takes to complete. Finally, the travel time between tasks $(i,j) \in E_t$ is defined as t_{ij}. Thus, an instance of the VRPMS-CC is uniquely defined by the graph $G_t = (V_t, E_t, c_{ij}^k, TW_i, st_i)$.

Our mixed integer programming model involves three types of decision variables. The first variable is common in three-index platform flow routing problem formulations. For all edges in the task graph $(i,j) \in E_t$ and platforms $k \in \mathcal{K}$, define a binary variable x_{ij}^k, which describes the edge-flow over the task graph for a platform (k).

$$x_{ij}^k = \begin{cases} 1 & \text{if } (i,j) \in E_t \text{ is traversed by platform } k \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The second type of variable is a binary flow variable capturing convoy flow across task sets

$$d_{pq} = \begin{cases} 1 & \text{if } (\mathcal{S}_p, \mathcal{S}_q) \in \mathcal{S}_T \text{ is traversed by a convoy} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The last type of variable is a timing variable $T_{ik}, i \in V_t$ for each subtask and platform that specifies the time when platform k begins subtask i.

We now present the **Vehicle Routing Problem with Multiple Synchronizations - Convoy Constraints (VRPMS-CC)**. This model includes the clustering rule which requires the number of platforms traveling in convoy between two locations be equivalent to the support value at the successor location. The VRPMS-CC is formulated as:

$$\min_{x_{ij}^k, d_{ij}, T_{ik}} \quad J = \sum_{k \in \mathcal{K}} \sum_{(i,j) \in E_t} c_{ij}^k x_{ij}^k \tag{3}$$

$$\text{s.t.} \sum_{d \in D^-} \sum_{j \in \mathcal{S}_T} x_{dj}^k \leq 1 \quad \forall \, k \in \mathcal{K}, \tag{4a}$$

$$\sum_{d \in D^+} \sum_{i \in \mathcal{S}_T} x_{id}^k \leq 1 \quad \forall \, k \in \mathcal{K}, \tag{4b}$$

$$\sum_{i \in \mathcal{S}_T} x_{ij}^k - \sum_{i \in \mathcal{S}_T} x_{ji}^k = 0 \quad \forall \, j \in \mathcal{S}_T, \, k \in \mathcal{K}, \tag{4c}$$

$$\sum_{k \in \mathcal{K}} \sum_{i:(i,j \in E_t)} x_{ij}^k = 1 \quad \forall \, j \in \mathcal{S}_T \tag{4d}$$

$$(T_{ik} + st_i + t_{ij} - T_{jk}) \leq M_t (1 - x_{ij}^k) \quad \forall k \in \mathcal{K}, (i,j) \in E_t, \tag{5a}$$

$$a_i \leq T_{ik} \leq b_i \quad \forall k \in \mathcal{K}, \, i \in \mathcal{S}_T \tag{5b}$$

$$T_{ik} - T_{ik'} = 0 \quad \forall k, k' \in \mathcal{K}, \quad i \in \mathcal{S}_T, \tag{6}$$

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{S}_i} \sum_{j \in \mathcal{S}_j} x_{ij}^k = v_j \, d_{pq} \quad \forall \, (\mathcal{S}_p, \mathcal{S}_q) \in \mathcal{S}_T, \tag{7a}$$

$$\sum_{k \in \mathcal{K}} \sum_{m \in \mathcal{S}_m} \sum_{j \in \mathcal{S}_j} x_{mj}^k \leq M_R (1 - d_{pq}) \quad \forall (\mathcal{S}_p, \mathcal{S}_q) \in \mathcal{S}, \mathcal{S}_m \in \mathcal{S} \setminus (\mathcal{S}_p, \mathcal{S}_q) \tag{7b}$$

$$\sum_{\mathcal{S}_p \in \mathcal{S}} d_{pq} = 1 \quad \forall \, \mathcal{S}_q \in \mathcal{S}, \tag{7c}$$

$$x_{ij}^k \in \{0,1\}, \quad d_{pq} \in \{0,1\}, \quad T_{ik} \in \mathbb{R}^+. \tag{8}$$

Constraints (4a), (4b), and (4c) describe sequence constraints (platforms start at the initial depot, platforms end at the final depot, and platform route-flow constraints). Constraint (4d) ensures that each task is assigned to a single agent. Constraints (5a) and (5b) ensure feasibility with respect to the task time windows. Temporal synchronicity is provided by (6), which states that the start time of any given task is the same across all platforms in a convoy. Movement synchronicity is enforced by (7a), (7b), and (7c). Constraint (7a) ensures that v platforms are used to visit a location, (7b) ensures that all platforms arrive at a successor task from the same predecessor task, and (7c) ensures that all platforms arrive only from one other task group. Hyperparameters M_t and M_R are two "Big-M" values that may be selected by finding the upper-bound of the left-hand side of constraints (5a) and (7b) as outlined in [16], [18].

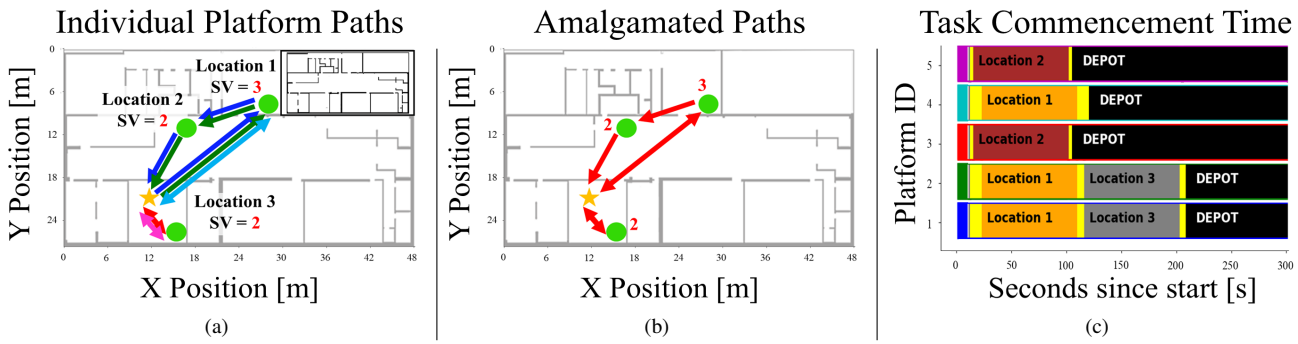


Fig. 2. An example security patrol mission on "Floor Plan A" where a team of five robotic platforms must visit three locations of interest.

V. Numerical Studies

The model presented in Section IV is tested on modified variants of both Solomon's [19] and multi-agent pathfinding [20] (MAPF) benchmarks. These studies motivate the introduction of a warm-start routing heuristic (in Section V-C) that empirically demonstrates a decrease in solver time.

A. Experimental Design

Each VRPMS-CC instance is uniquely defined by the graph $G_t = (V_t, E_t, c_{ij}^k, TW_i, st_i)$. The information utilized to construct this graph is provided to the solver as an environment map, a list of locations, the support value at all locations, and a location's timing information. All instances considered in this manuscript are solvable.

1) Environment Maps: Each VRPMS-CC instance was associated with an environmental map to introduce an idea of environmental obstacle avoidance into the study. These obstacles directly affect the cost associated with traveling between locations. This environment map was represented as a binary 2D occupancy grid that captured information regarding impassable obstacles. The "floor plan" studies were associated with environmental maps that are presented in this manuscript. MAPF environments are referenced by name from the open-source MAPF benchmark [20]. All other trials are associated with environmental maps without obstacles.

2) Task Graph Vertex and Edge Definitions: Graph vertices are populated by creating duplicate tasks at each location equal to the support value associated with that location. Locations were either provided (Solomon's benchmark) or randomly sampled from non-occupied cells (Floor Plan, MAPF). For Solomon's benchmark, six to eight locations were randomly selected from [19]. Tests with designed location layouts will be denoted in the discussion of that test. As neither the MAPF benchmark nor the Solomon benchmark was originally constructed with the concept of support value, unless noted otherwise, a random support value between $S_i \in [2, K]$ was selected for each location.

Graph edges are defined between all vertices. Each edge is weighted by finding the shortest obstacle-free path between the locations (vertices). Vertices located at the same location have a relative distance of zero. (Note that constraints (7a), (7b) ensure that different agents are assigned tasks at the same location). All results are denoted with straight arrows, which represent the obstacle-free paths of the vehicles.

3) Timing information: Unless specified in the benchmark (i.e., Solomon's), locations are assigned time windows with the width of the mission duration. Estimated traversal times were computed by finding the shortest obstacle-free path distance between two locations and then dividing that path distance by an assumed average velocity. Unless indicated otherwise, all locations have a service time set at 10 [s].

4) Solver Information: Each VRPMS-CC instance was solved using the Gurobi Optimizer (Version 10.03) on a mobile workstation with an AMD Ryzen 7 4800H CPU with 8 cores and 16 threads. Each test was conducted twice, once with Gurobi's internal solver presolve accelerations and once without. As Gurobi's presolve accelerations decrease problem solve times by an order of magnitude (average: 98% across all Solomon trials), we provide results with and without the presolve accelerations. Such results are presented not only to empirically demonstrate the computation complexity, but also to provide insight into solver performance when the presolve is not available. For all trials, the solver time limit was set at 7200 [s] and an optimality gap of 0.01. If a solver did not resolve in 7200 [s], it will be marked as "Did Not Finish" or "DNF".

B. VRPMS-CC Numerical Studies

As a solution to an instance of the VRPMS-CC may contain complex convoy routing behaviors, we utilize the floor plan environment as an illustrative example. The individual robotic platform behaviors captured in the floor plan environment are also present in the tabulated solutions.

1) Demonstration: Floor Plan A Routing: Figure 2 depicts a patrol scenario in which five robotic agents must visit three locations of interest with different support values. The routes taken by each platform are shown in Fig. 2a, and the aggregate convoy routes are shown in Fig. 2b. The alignment of the task commencement times in Fig. 2c demonstrates the temporal synchronicity in the movements of the agents.

The agents start at the depot and are initially organized into two separate convoy details of sizes two and three. The size-two detail travels to Location 3 before returning to the depot (marked by a bidirectional arrow). The size-three detail first travels to Location 1 before splitting into a new detail of sizes one and two. The singular agent returns to the depot as it is not needed at Location 2. The size-two detail proceeds to service the task at Location 2 before returning to the depot.

TABLE I
NUMERICAL STUDIES DEMONSTRATE THAT THE VRPMS-CC INSTANCE CAN ROUTE CONVOYS WITH AND WITHOUT TIMING CONSTRAINTS

Map Name	Solve Time: (Normal / Wide) (No Presolve) [s]	Solve Time: (Normal / Wide) (Presolve) [s]
C101	26.7 / **DNF**	1.9 / 43.8
C102	1222.7 / 6772.9	14.1 / 15.1
C201	168.8 / **DNF**	3.9 / 26.8
R101	39.8 / 1062.1	1.5 / 16.6
R102	426.4 / 529.5	12.6 / 18.1
R201	149.7 / 6731.7	2.8 / 19.6

Compared to maintaining a static size-three detail, splitting the convoy details decreases the total distance traveled by 1.3%. The amount of savings increases for every robot that is prevented from making unnecessary trips.

2) Study: Solomon's benchmark: In this work, a non-capacitated version of Solomon's benchmark was used to evaluate the VRPMS-CC. Two variants of each trial were considered, one which maintains Solomon's original timings and one which replaces these timings with wide windows and service times of magnitude zero. While the first variant represents missions or operations with tight time schedules, we have found the second variant to be a representation of many robotic missions where such tight time window constraints may not naturally emerge. Each instance was conducted for a team of six robots with twenty-eight tasks.

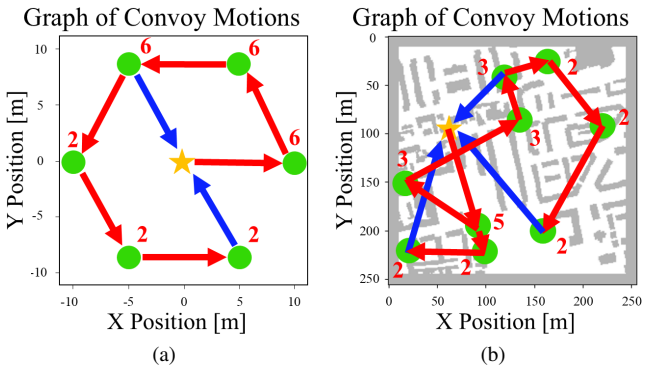
The results of the studies demonstrate that the modeled approach correctly routes convoys in the presence of timing constraints. An interesting trend is that loose timing constraints yield longer solve times than tight timing constraints. Table I shows that both the presolve accelerated and the non-accelerated cases take advantage of the strict timing constraints to order the visitation of the convoys across the environments (e.g., the solve times for wide time windows in *C101* are greater in both columns). On average, wider time windows required an increase in solve time of 77.6% and 80.4% with and without the presolve, respectively. The observation inspires the use of wide time windows in all following investigations in this manuscript.

3) Study: MAPF Benchmark Performance: The MAPF benchmarks are tailored to represent environments in which robotic systems are expected to operate. The required obstacle avoidance behavior considered in these maps is included during the evaluation of the edge weights, as each edge is weighted by the obstacle-free path distance taken by a robotic platform should that edge be included in the route.

We observe from the MAPF tests that, when using the associated pre-solve techniques on problems of the size considered in the simulation section, an empirical worst-case performance of 45.2 [s] is achieved. For our application, which is similar to the DARPA Subterranean Challenge, missions take between 30-60 minutes [21]. Thus, this empirically observed worst-case performance is sufficient for task allocation. Compared to the dynamic convoy topologies determined by the VRPMS-CC, simple fixed convoy topologies equal to the largest support value found at any location demonstrate large increases in total

TABLE II
TESTS ON MAPF BENCHMARKS DEMONSTRATE VRPMS-CC REDUCES TOTAL DISTANCE TRAVELED BY ALL AGENTS IN THE SYSTEM

Name (Robots/Tasks)	Route Length % Decrease	Solve Time [s] (No presolve)	Solve Time [s] (Presolve)
Berlin_0_256 (5/24)	-10.2%	2900.5	15.8
Boston_0_256 (6/22)	-15.2%	1315.8	10.2
Paris_1_256 (6/21)	-20.3%	315.6	7.7
Maze_32_32_2 (6/46)	0.0%	2430.1	31.1
Maze_32_32_6 (6/50)	-56.2%	2543.1	45.2
Hex_CLUSTER (6/24)	-28.6%	338.9	36.2
Hex_MIX (6/24)	-15.7%	7184.0	6.3
Hex_ALT (6/24)	-4.3%	1142.4	4.2



(a) (b)

Fig. 3. Two solutions depicted considered on a "*Hex*" environment (Fig. 3a) and a MAPF environment (Fig. 3b). Support values for each location (green) are indicated in red next to the location. The depot node is indicated by the gold star. Convoy routes which return to the depot are indicated with blue arrows while convoy routes which progress to further nodes are in red.

distance traveled by all agents. In the generated missions, the total convoy travel distance can be decreased by as much as 56% (*Maze_32_32_6*). This large decrease in total traveled distance arises from the structure of the instance, as only two agents are needed at all locations except for a single location, which requires the entire team. This result indicates that the set and relative magnitude of different support values highly influence the resulting convoy behaviors. This observation was also tested on the *Hex* environments, as the locations are held fixed while the support values are changed. Noticeably, trials where similar support values are clustered (*HEX_CLUSTER*) yield lower solve times (without presolve) than randomized support values (*HEX_ALT*). Figure 3 is a visual representation of the determined set of routes plotted on both the "*Hex*" (Fig. 3a) and "*Berlin_1_256*" (Fig. 3b) environments.

C. MDVRP-SV Routing Heuristic

It is clear from Table II that additional solver accelerations could be introduced to improve solver performance. Noticeably, finding an initial feasible solution proves difficult in many of the MAPF trials. If an initial feasible solution could be found, we hypothesize that the data would reflect a decrease in solver time of a magnitude similar to the time it took to find a feasible solution. We introduce a decomposition-based multi-depot warm-start heuristic to achieve this decrease in solver time.

The intuition for this heuristic arises from the structure of the convoy constraints. The constraints provide routing restrictions that may be exploited to form a valid set of initial motions for each convoy detail. Specifically, the clustering rules described in Section III suggest that only a single convoy can visit one location from any other location. The consequence of this clustering rule is that convoys of agents can only be split from a larger convoy and not combined with smaller convoys. Thus, a convoy is largest when it leaves the depot, and all agents in the convoy are assigned tasks at locations with the highest support values first.

Our approach takes the form of a decomposition-based heuristic in which a set of multi-depot vehicle routing problems (MDVRP) are solved sequentially in order of descending support value. While it is clear that requiring the solution of an MDVRP inside the MDVRP-SV heuristic procedure indicates the heuristic is itself NP-hard, the advantage of posing the MDVRP-SV heuristic on the location space rather than the task space is that the location space is a smaller space by construction. Effectively, this solves for the motion of entire formations instead of any particular platform. This process is detailed using Python syntax in Algorithm 1.

Algorithm 1 MDVRP-SV Heuristic

1: **Define:** MDVRP-SV($\mathcal{N}, \mathcal{V}, d^-, d^+$)
2: $L \leftarrow \emptyset$, $SV \leftarrow \emptyset$ $D \leftarrow d^-$, $N \leftarrow \emptyset$
3: $SV \leftarrow$ sorted(set($\mathcal{N}$), reverse=True)
4: **for** $sv \in SV$ **do**
5: **for** $v_i \in \mathcal{V}$ **do**
6: **if** $v_i = sv$ **then**
7: $N \leftarrow \mathcal{N}[i]$
8: $L \leftarrow$ VRP(N, D) ▷ Perform VRP on sv locations
9: L.pop() ▷ Remove Depot Return
10: $D \leftarrow \emptyset$, $N \leftarrow \emptyset, D \leftarrow L[-1]$
11: $L \leftarrow d^+$

Given the set of locations, $\mathcal{N}$, the support value set, $\mathcal{V}$, and the initial depot, the heuristic first searches through the support value list to extract all unique support values (Line 3). Next, the heuristic iterates through the unique support value list L_v to build partial routes at each support value level. This procedure is completed by first storing all locations with a particular support value (Line 7) in a set N. Following this, a procedure that solves the multi-depot Vehicle Routing Problem (VRP in Line 8) on set N starting from one or more depots stored in D. The result of solving the VRP is used to populate the sequence of locations for each formation containing sv platforms to visit, denoted L. The last element in L is removed (Line 9) to extract the last location visited by the formation with sv platforms (Line 10). After cycling through all support values, the final locations in L are connected to the return depots d^+.

The heuristic itself is designed for instances of the VRPMS-CC that are not time-constrained, as the expectation is that the convoy movement constraints and not the timing constraints drive the solution of the problem. For a problem with $|SV|$ unique support layers with a max of V nodes at any layer, the heuristic can be solved $|SV|$ times using both Yang's [22] multi-depot multiple traveling salesman transformation and the Held-Karp Algorithm [23] to yield an exponential worst case complexity of $\mathcal{O}(|SV| \cdot (V^2 2^V))$. Note that the heuristic still retains its exponential complexity in the number of nodes. Alternatively, to overcome scalability challenges, the MDVRP-SV could be solved via heuristic or metaheuristic methods. Note that the heuristic is myopic to each layer, as the solved MDVRP does not ensure that the last location in the previous layer is necessarily closer to the first location in the next layer. Consequently, symmetrically placed locations in the previous layer can adversely affect the optimality of the heuristic solution.

TABLE III
VRPMS-CC BENCHMARKS DEMONSTRATE FORMATION-BASED ROUTING HEURISTIC DECREASES SOLVER TIME

Map Name	Solver Time / % Change (No presolve) [s]	Solver Time / % Change (Presolve) [s]
Berlin_0_256	1381.9 / -52.4 %	21.2 / 33.5 %
Boston_0_256	1264.6 / -51.4 %	4.3 / -43.2 %
Paris_1_256	319.3 / 1.2 %	6.5 / -15.5 %
Hex_CLUSTER	157.8 / -53.5 %	2.7 / -92.5 %
Hex_MIX	758.5 / >-89.4 %	5.8 / -7.5 %
Hex_ALT	251.3 / -78.0 %	4.1 / -2.4 %

The influence of the heuristic on the solver performance was demonstrated on both the MAPF benchmark and the Hexagon (*Hex*) environments. All warm-starts were solved in less than 0.02 [s]. The first solution produced by Algorithm 1 (suboptimal or not) was used. The results of the rerun numerical studies can be seen in Table III. Of particular note is that almost all maps display a marked decrease in solver time for tests both with and without the presolve. The only exceptions are the non-presolve *Paris_1_256* and presolved *Berlin_0_256* cases. Both warm-starts provided in these tests are highly suboptimal (<40%). The greatest improvement arises from the *Hex_12* case, which achieves an improvement in the solver time of more than 89%. This performance improvement directly rises from the warm-start, providing the optimal solution.

VI. FOUR-PLATFORM TEAM HARDWARE TRIALS

An important concern regarding the use of operations research models is whether the solved problem instance accurately describes the modeled robotic system(s) or environment. We illustrate the validity of this concern by considering how an unmarked environmental feature (a closed door) yields an *a priori* unknown infeasible set of routes. We then address this concern by demonstrating that an agent's onboard sensing can augment the representation in order to populate an instance of the VRPMS-CC.

A. Hardware Trial Environment Overview

All hardware trials were carried out in the indoor multi-hallway environment depicted in Fig. 4. Of note are two terrain features (doors) that, in certain configurations (closed), are impassable to the agents. The existence of these features is

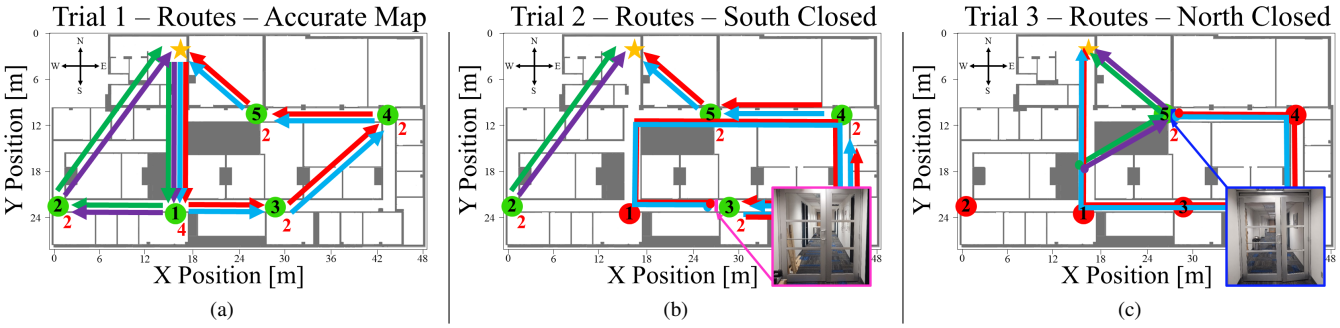


Fig. 4. A diagram showing the final routes performed by each convoy in the hardware trials. Figure 4a depicts the convoy routes when no impassable terrain features are present. In Fig. 4b, the routes taken by the light blue and red agents are replanned once the doors (in pink) are observed and block forward progress. Fig. 4c demonstrates a similar scenario where the doors (in blue) are closed, requiring the green and purple agents to finish the last task.

not denoted in the initially provided representation (occupancy map). The locations of these doors are highlighted in Fig. 4b (south) and Fig. 4c (north). Clutter present in the environment was not dense enough to completely obstruct the motion of any agent.

1) Hardware Trial Robotic Agent Autonomy System: The team of four robotic agents (Fig. 1) was tasked with completing the convoy operation. Each agent maintains its own local motion-primitive-based obstacle avoidance planner and its own local receding-horizon tracking controller. Environmental mapping and agent odometry were performed using a Simultaneous Localization and Mapping (SLAM) algorithm [24]. While the original testing environment does not contain any supporting communication infrastructure, we utilize an agent-based communication network construction technique described in our prior work [25], [26] to maintain network connectivity and communications coverage during system operation. The construction behavior ensures all agents were connected to each other and to the base station (located at the gold star in Fig. 4) via a wireless communication network. The route of each agent was determined by solving an instance of VRPMS-CC on the base station. This solution was then provided to an agent-based convoy formation controller, which formed the agents into a convoy and conducted the mission. Each agent provides map and odometry information back to the base station. More information about the multi-agent convoy formation architecture is available in our prior work [26], [27].

B. Dynamic VRPMS-CC (DVRPMS-CC)

If the initial environmental representation is inaccurate, the initial route allocation may be suboptimal or infeasible. In this work, the inaccurate representation produces an inaccurate measure of c_{ij}^k. As this representation does not have an indication of the impassable feature, we utilize reactive "dynamic" vehicle routing problem (DVRP) techniques to address the potential infeasibility [16].

Our extension to a Dynamic VRPMS-CC (DVRPMS-CC) introduces a process that addresses two aspects common to DVRPs. The first aspect is that (4a) must be modified such that the platforms start from the locations they currently occupy rather than the original depot. Redefining the starting depot set as $\mathcal{D}^- = \{d^{k_1^-}, d^{k_2^-}, \ldots, d^{k_k^-}\}$, this modification yields a new route that starts from the agent's initial position ($d^{k_k^-}$ for platform k). The second modification relates to updating the environment representation for accurate measures of c_{ij}^k. The agent's SLAM system provides real-time mapping capabilities that may be compared to the occupancy grid used to initially generate the routes. In the presence of an impassible obstacle, the local planner does not yield a viable forward path for the system. This information is provided to the base station, which can then construct a new VRPMS-CC instance. The base station receives a map update from the agents over the communication network and then uses the updated map to recompute c_{ij}^k and populate the new VRPMS-CC instance. Solving the new VRPMS-CC instance yields an updated set of routes for the agents.

C. Accurate Environment Representation (Fig. 4a)

Consider an application of the VRPMS-CC framework to a patrol mission consisting of five locations (Fig. 4, in green), where one of the locations requires four robotic agents and all other locations require only two agents. Agents must visit all locations while minimizing the total distance traveled by all agents. The service time for each task is negligible, and the time window for task completion is the mission duration.

Given an accurate environmental representation, only a single instance of the VRPMS-CC must be solved on the base station. As shown in Fig. 4a, the agents first form a four-vehicle convoy formation that then splits into two two-vehicle convoy formations (green-purple and blue-red). The agents collectively traveled a total distance of 358.1 [m].

D. Inaccurate Environment Representation (Fig. 4b and 4c)

If either the north or the south set of doors is closed in the test environment, the initial routes prove to be infeasible and are either re-planned (i.e., the same agents perform the task) or re-allocated (i.e., a different set of agents perform the task). In Trial 2, the southern doors obstruct the motion of the robotic agents toward Tasks 3-5 in Fig. 4a. The DVRPMS-CC process outlined above produces the new set of routes shown in Fig. 4b. The total distance traveled by all vehicles increased by 46.7% as a result of the re-solve. In Trial 3, instead of the southern door being closed, the northern door was closed. The new routes re-allocate the task at Location 5 to the convoy that completed the task at Location 2, causing an increase in

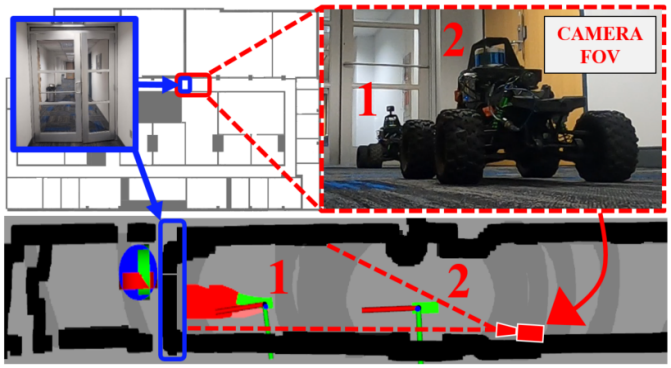


Fig. 5. An image from Trial 3 depicting a convoy encountering the northern closed door (blue). The bottom figure shows the onboard occupancy map, reflecting that the front agent has no forward paths (red) due to the door.

total distance traveled by all vehicles of 48.3%. Note that the difference in total distances traveled for Trial 2 and Trial 3 is due to navigation around environmental clutter. An image from the trial is shown in Fig. 5

VII. CONCLUSIONS

This work utilizes the structure of routing problems alongside synchronicity constraints to solve a multi-platform convoy coordination problem. The presented constraints enable a team of vehicles to coordinate their routes spatially and temporally to travel as a convoy. We utilize these synchronization constraints to define the VRPMS-CC, which is solved using off-the-shelf mixed integer programming solvers. Numerical studies and hardware results demonstrate the applicability of the model in both simulated and real-world cluttered environments, but inherits the challenges associated with scalability common to combinatorial search problems. We further contribute the MDVRP-SV heuristic, which improves solve times of off-the-shelf commercial solvers, and the DVRPMS-CC for online replanning when the initial environmental representation is inaccurate.

Although the results demonstrate the effective routing of a small team of robotic agents, future work must incorporate additional realistic constraints and address solver performance limitations, particularly for large-scale problem instances. Investigating clustering rules that allow compositional teaming and constraints derived from the limitations of the communications system (e.g., range, drop-out) are critical priorities. In particular, the incorporation of information-transfer rendezvous points seems particularly relevant [28]. The performance of the heuristic implies that additional information regarding the problem structure (e.g., unique support value(s)) and the homogeneous nature of the agents (symmetry breaking) may lead to improved solve times of the VRPMS-CC. Furthermore, hybrid approaches that approximately solve the VRPMS-CC offline and use an alternative sub-optimal method (e.g., reinforcement learning [14]) for online reactivity plays to the strength of both approaches.

REFERENCES

[1] D. A. G. (MAJ), "The future of autonomous ground logistics: Convoys in the department of defense," School of Advanced Military Studies (SAMS), Tech. Rep., 2011.

[2] S. Nahavandi *et al.*, "Autonomous convoying: A survey on current research and development," *IEEE Access*, vol. 10, 2022.

[3] O. Cheikhrouhou and I. Khoufi, "A comprehensive survey on the multiple travelling salesman problem: Applications, approaches and taxonomy," *Computer Science Review*, vol. 100369, no. 40, 2021.

[4] M. Drexl, "Synchronization in vehicle routing - a survey of vrps with multiple synchornization constraints," *Transportation Science*, 2012.

[5] I.-M. Chao, "A tabu search method for the truck and trailer routing problem," *Computers and Operations Research*, 2002.

[6] R. Soares, A. Marques, P. Amorim, and S. Parragh, "Synchronisation in vehicle routing: Classification schema, modelling framework and literature review," *European Journal of Operational Research*, 2023.

[7] S. Ropke and J.-F. Cordeau, "Branch and cut and price for the pickup and delivery problem wiht time windows," *Transportation Science*, vol. 43, 2009.

[8] H. H. Doulabi, G. Pesant, and L.-M. Rousseau, "Vehicle routing problems with synchronized vists and stochastic travel and service times: Applications in healthcare," *Transportation Science*, 2020.

[9] Z. Ren, C. Zhang, S. Rathinam, and H. Choset, "Search algorithms for multi-agent teamwise cooperative path finding," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[10] N. Greshler, O. Gordon, O. Salzman, and N. Shimkin, "Cooperative multi-agent path finding: Beyond path planning and collision avoidance," in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 2021, pp. 20–28.

[11] H.-L. Choi, A. K. Whitten, and J. P. How, "Decentralized task allocation for heterogeneous teams with cooperation constraints," in *Proceedings of the 2010 American Control Conference*, 2010.

[12] R. Patel *et al.*, "Decentralized task allocation in multi-agent systems using a decentralized genetic algorithm," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[13] H. Fu, M. You, H. Zhou, and B. He, "Closely cooperative multi-agent reinforcement learning based on intention sharing and credit assignment," *IEEE Robotics and Automation Letters*, 2024.

[14] A. Bogyrbayeva *et al.*, "A deep reinforcement learning approach for solving the traveling salesman problem with drone," *Transportation Research Part C: Emerging Technologies*, 2023.

[15] L. He, J. Pan, S. Narang, W. Wang, and D. Manocha, "Dynamic group behaviors for interactive crowd simulation," 2016.

[16] P. Toth and D. Vigo, *Vehicle Routing*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2014.

[17] D. Dominique, *The Original Tactical Convoy Handbook*. Wounded Warrior Publications, 2015.

[18] W. Winston, *Operations Research: Applications and Algorithms*. Belmont, CA: Thomson Learning, 2004.

[19] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, 1987.

[20] R. Stern *et al.*, "Multi-agent pathfinding: Definitions, variants, and benchmarks," *Symposium on Combinatorial Search (SoCS)*, 2019.

[21] DARPA, "Defense advanced research project agency, subterranean (subt) challenge (archived): Subterranean challenge final event." [Online]. Available: https://www.darpa.mil/research/challenges/subterranean

[22] Y. GuoXing, "Theory and methodology: Transformation of multidepot multisalesmen problem to the standard travelling salesman problem," *European Journal of Operational Research*, 1995.

[23] M. Held and R. M. Karp, "A dynamic programming approach to sequencing problems," *Journal of the Society of Applied Mathematics*, vol. 10, no. 1, March 1962.

[24] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, "Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8729–8736.

[25] S. Scherer *et al.*, "Resilient and modular subterranean exploration with a team of roving and flying robots," *Field Robotics*, 2022.

[26] P. Sriganesh *et al.*, "Modular, resilient, and scalable system design approaches – lessons learned in the years after darpa subterranean challenge," 2024.

[27] N. Bagree, C. Noren, D. Singh, M. Travers, and B. Vundurthy, "Distributed optimal control framework for high-speed convoys: Theory and hardware results," in *IFAC-PapersOnLine*, 2023.

[28] A. R. da Silva, L. Chaimowicz, T. C. Silva, and M. A. Hsieh, "Communication-constrained multi-robot exploration with intermittent rendezvous," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2024.