

Heterogeneous Team Coordination on Partially Observable Graphs with Realistic Communication

Yanlin Zhou, Manshi Limbu, Xuan Wang, Daigo Shishika, and Xuesu Xiao

Abstract—Team Coordination on Graphs with Risky Edges (TCGRE) is a recently proposed problem, in which robots find paths to their goals while considering possible coordination to reduce overall team cost. However, TCGRE assumes that the *entire* environment is available to a *homogeneous* robot team with *ubiquitous* communication. In this paper, we study an extended version of TCGRE, called HPR-TCGRE, with three relaxations: Heterogeneous robots, Partial observability, and Realistic communication. To this end, we form a new combinatorial optimization problem on top of TCGRE. After analysis, we divide it into two sub-problems, one for robots moving individually, another for robots in groups, depending on their communication availability. Then, we develop an algorithm that exploits real-time partial maps to solve local shortest path(s) problems, with a A*-like sub-goal(s) assignment mechanism that explores potential coordination opportunities for global interests. Extensive experiments indicate that our algorithm is able to produce team coordination behaviors in order to reduce overall cost even with our three relaxations.¹

I. INTRODUCTION

Team Coordination on Graphs with Risky Edges (TCGRE) is a recently proposed problem [1], in which robots need to schedule trajectories while considering possible coordination on the move to reduce overall team cost. Compared with the classical Multi-Agent Path Finding (MAPF) problem [2]–[8], where robots attempt to avoid collisions (and interactions) on their journeys, TCGRE strives to exploit such opportunities, in terms of support between agents. This new problem possesses great practical value because it empowers robots with teamwork potential, which is especially useful in environments too complicated for a single robot to travel through, or in tasks so demanding that one robot may need assistance from others.

Most solutions to TCGRE assume that the *entire* environment is available to a *homogeneous* robot team, with *ubiquitous* communication. However, these assumptions may not hold in real-world scenarios. Robots may be deployed in unfamiliar environments and possess different capabilities to support or to be supported, while communication cannot happen anytime anywhere at no cost. Therefore, in this paper, we study an extended version of TCGRE, called HPR-TCGRE, which relaxes all three assumptions: 1) **Heterogeneous team**: different types of robots are involved, inducing a different edge cost, support cost, and risk reduction for each (pair of) robot type; 2) **Partial observability**: each robot can only observe the graph around it by sensors, while building up

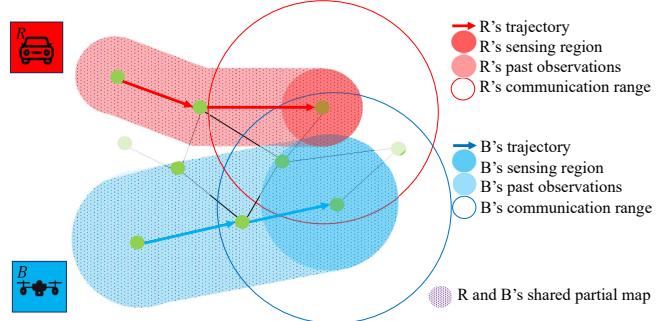


Fig. 1: An HPR-TCGRE Example: A ground and an aerial robot gradually sense the environment as they traverse the graph, while exchanging knowledge and coordinating with each other when they are in communication range.

the entire graph in their memory as they move in the environment; and 3) **Realistic communication**: robots coordinate with one another via local communication channels to make coordination decisions and exchange environment knowledge only with their neighbors.

We first complete a mathematical formulation of this problem, as a combinatorial optimization problem similar to TCGRE [9]. After analysis, we base our solution on dividing the problem into two sub-problems: 1) Individual movement: robots operate on their own when there are no nearby teammates; 2) Coordination: robots coordinate on the run when teammates are available. For the individual movement sub-problem, we then break it into two parts: i) each individual robot schedules a path to its sub-goal, solvable by any shortest path algorithm. ii) the robot decides a new sub-goal every time its partial map is updated, where A* can provide a good solution. Similarly, the coordination sub-problem is also separated into two parts: i) robots in the same partial map schedule their paths to their sub-goals, which is almost TCGRE, solvable by algorithms for TCGRE. ii) every robot in the same partial map decides a new sub-goal every time its partial map is updated, for which we develop a new algorithm that solves it as a multi-choice knapsack problem. Extensive experimental results are presented and discussed to evaluate the performance of our proposed algorithm.

II. RELATED WORK

The original TCGRE problem has attracted some attention from researchers. Limbu et al. [1] proposed a method that constructs a Joint State Graph (JSG) and converts the problem into a single-agent shortest path problem, solvable by Dijkstra's algorithm. They then developed the Critical Joint

All authors are with George Mason University {yzhou30, klimbu2, xwang64, dshishik, xiao}@gmu.edu.

¹Full Version: <https://cs.gmu.edu/~xiao/papers/hpr-tcgre.pdf>

State Graph (CJSG) approach that removes many unnecessary edges in the JSG and mitigates the curse of dimensionality while maintaining optimality. To improve practicality of the solution so it works with more robots and larger graphs, Reinforcement Learning (RL) [10] is one way to scale up, but optimality is no longer a guarantee. Zhou et al. [9] then reformulated the problem as a combinatorial optimization problem, elaborated on the importance of problem decomposition via mathematical analysis, and provided three classes of solutions: JSG-based approaches are optimal but computationally expensive when the number of agents increases; coordination-based methods are optimal under an assumption that one coordination behavior only occurs for a limited number of times, therefore efficient in terms of the number of agents, but not the number of coordination behaviors; Receding-Horizon Optimistic Cooperative A* (RHOC-A*) search algorithms are efficient but sacrifice performance due to a limited horizon and limited robot pair choices.

However, the original TCGRE problem is based on three strong assumptions: 1) All robots are the same; 2) The entire graph is known to the team a priori; and 3) any team member can communicate with the central controller anytime anywhere at no cost. All these assumptions are not always realistic in real-world robot applications with limited perception range, different robot types, and constrained communication channels. Therefore, the HPR-TCGRE problem and its solution proposed in this work aim to relax all these three oversimplified assumptions.

III. PROBLEM FORMULATION

In the original TCGRE problem, a team of N homogeneous robots traverse an undirected graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where \mathbb{V} is the set of nodes the robots can traverse to and \mathbb{E} is the set of edges connecting the nodes, i.e., $\mathbb{E} \subset \mathbb{V} \times \mathbb{V}$. The team of robots traverse in the graph from their start nodes $\mathbb{V}_0 \subset \mathbb{V}$ to goal nodes $\mathbb{V}_g \subset \mathbb{V}$ via edges in \mathbb{E} . Each edge $e_{ij} = (V_i, V_j) \in \mathbb{E}$ is associated with a cost. Specially, some edges with high costs are difficult to traverse through, denoted as risky edges $\mathbb{E}' \subset \mathbb{E}$, but with the support from a teammate from a supporting node, their costs can be significantly reduced. In this problem, such coordination behaviors only occur between two robots: one receiving robot receives support while traversing a risky edge, and another supporting robot offers support from some (nearby) location, called support node. Each risky edge $e_{ij} \in \mathbb{E}'$ corresponds to certain support node(s) $\mathbb{S}_{e_{ij}} \subset \mathbb{V}$ ($\mathbb{S}_{e_{ij}} = \emptyset$ if $e_{ij} \notin \mathbb{E}'$). Additionally, the coordination also induces some cost for the supporter. The scheduling of all agents' movement and coordination relies on a central controller.

However, TCGRE may not be applicable in many real-world cases as stated in Sec. I, so in this paper we generalize it to HPR-TCGRE, by relaxing three assumptions as follow:

1) Heterogeneous team: the team of robots involve H types, who have different specialties possibly complementary to each other and thus contribute to better teamwork. As a result, an edge cost is not only dependent on the specific edge e_{ij} , but also the type of the robot traversing it h_n , represented as $c_{ij}^{h_n}$. For coordination, the reduced cost is related to the

types of both robots conducting a coordination behavior h_n and h_m , represented as $\tilde{c}_{ij}^{h_n h_m}$; the coordination cost is related to the type of the supporter h_m , denoted by $c_{ij}^{h_m}$.

2) Partial observability: the environment is unknown in the beginning, and each robot can only use its sensors to observe the graph around it. That is to say, at time $t = 0$, the graph is unknown to each robot n in the team, i.e., $\mathbb{G}_n^0 = (\mathbb{V}_n^0, \mathbb{E}_n^0)$ where $\mathbb{V}_n^0, \mathbb{E}_n^0 = \emptyset$. Each robot n can only sense a region around its current node l_n^t ; the range is dependent on the sensors of different robot types, i.e., $\mathbb{G}^h(l_n^t)$, with a memory of previously collected data. Therefore, each robot's knowledge of the graph, called partial map, is based on past and current observations $\mathbb{G}_n^t = \mathbb{G}_n^{t-1} \cup \mathbb{G}^h(l_n^t)$, as in Fig. 1. Moreover, each node corresponds to a set of coordinates, and each robot knows the coordinates of its goal, but not the path(s) to it, due to partial observability.

3) Realistic communication: without central control, robots rely on communication for coordination, with which they can also exchange their partial maps to facilitate navigation. Inspired by the 3-way handshake [11], we propose a similar process, where each robot n broadcasts a hello message once at each time step; when another robot m detects the message, it replies with another broadcast. If robot n receives the response, it sends out a confirmation message. Then, the actual communication is established, and they start to coordinate after exchanging and combining their own partial maps and their coordination messages, also once per time step, i.e., $\mathbb{G}_n^t, \mathbb{G}_m^t = \mathbb{G}_n^t \cup \mathbb{G}_m^t$ if $\mathcal{C}_{nm}^t = 1$, as shown in Fig. 1. In this mechanism, the actual communication only starts when they get close, within the communication range—the minimum of the two transmission distances (which may be derived from the Friis Equation [12]), i.e., $r_{nm} = \min(D_{nm}, D_{mn})$. (The communication range in Fig. 1 and Fig. 2 are not following the strict definition, only for a better demonstration of the key ideas.) Additionally, robots can communicate with each other via other intermediate robots and form ad-hoc networks [13], called robot groups. Denote any robot group that includes n as \mathbf{RG}_n^t , where $\mathbf{RG}_n^t = \{n | \forall n \in \mathbf{RG}_n^t, \exists m \in \mathbf{RG}_n^t, s.t. d_{nm}^t \leq r_{nm}\}$ and $\mathbf{RG}_n^t, \mathbf{RG}_m^t = \mathbf{RG}_n^t \cup \mathbf{RG}_m^t$ if $m \in \mathbf{RG}_n^t$. So for any robot pair in the same group, their communication is available, i.e., $\mathcal{C}_{nm}^t = 1$ if $m \in \mathbf{RG}_n^t$, and 0 otherwise. In this problem, we assume a constant cost for each communication, which is trivial in the objective function. A more realistic communication cost model will be discussed in future work.

Ultimately, each robot needs to schedule a path to reach their individual goal and coordinate with one another along the way, to minimize the overall cost.

A. Cost Model

Similar to previous work [9], we can represent robots' decision making with two sets of 0/1 variables, a movement decision set \mathcal{M} and coordination decision set \mathcal{S} . Specifically, without coordination, at each time step t , each robot n needs to decide where to go, denoted as $\mathcal{M} = \{M_{ij}^{nt} | \forall i, j, \forall n, \forall t\}$, where $M_{ij}^{nt} = 1$ represents robot n traverses edge e_{ij} at time t and 0 otherwise. The general cost in this case is simply the

cost of the chosen edge, i.e., $C_n^t = c_{ij}^{h(n)}$, where $M_{ij}^{nt} = 1$. Staying is allowed for future coordination, i.e., $c_{ii} = 0$.

When coordination is possible — after exchanging partial maps with other robots via communication, one robot can move to a support node to help another pass through an edge with reduced cost, coordinated via live communication — each robot pair needs to decide whether they are going to coordinate, denoted as $\mathcal{S} = \{s_{nm}^t | \forall n, m, \forall t\}$. The cost is decided by both variables:

$$C_n^t = \begin{cases} c_{ij}^{h_n}, & \text{if } s_{nm}^t = 0; \\ \tilde{c}_{ij}^{h_{nm}}, & \text{if } s_{nm}^t = 1; \\ c_{ij}^{h_m}, & \text{if } s_{nm}^t = -1. \end{cases} \quad (1)$$

where $s_{nm}^t = 1$ means robot n receives support from m , $s_{nm}^t = -1$ indicates robot n offers support to m , and $s_{nm}^t = 0$ suggests no coordination between n and m at t . Such coordination only happens to a robot pair, i.e., $s_{nm}^t + s_{mn}^t = 0$, and only to two different robots, i.e., $s_{nn}^t = 0$. Additionally, coordination is only possible when robots know the existence of other robots by communication, i.e., $s_{nm}^t = 0$ if $\mathcal{C}_{nm}^t = 0$. Last, there is no point for every robot to stay still, i.e., $\sum_{\forall n} \sum_{\forall i \neq j} M_{ij}^{nt} \neq 0$.

B. Problem Definition

Given the node set \mathbb{V} , the edge set \mathbb{E} , support nodes for each edge $\mathbb{S}_{e_{ij}}$, N robots with their starts \mathbb{V}_0 and goals \mathbb{V}_g , their sensing regions $\mathbb{G}^h(\cdot)$ and communication range $\{r_{nm}\}$ (to determine \mathcal{C}_{nm}^t), cost of each edge without and with coordination $c_{ij}^{h_n}, \tilde{c}_{ij}^{h_{nm}}, c_{ij}^{h_m}$, coordination cost $c_{ij}^{th_{nm}}$, optimize the movement and coordination decisions \mathcal{M} and \mathcal{S} , in order to minimize the total cost for each agent to traverse from its start to its goal within a time limit T . Formally, the problem can be represented as

$$\begin{aligned} \min_{\mathcal{M}, \mathcal{S}} & \sum_{t=0}^{T-1} \sum_{n=1}^N C_n^t; \\ \text{s.t. } & \sum_{\forall m \in \{1, 2, \dots, N\}} |s_{nm}^t| \leq 1, \end{aligned} \quad (2)$$

$$\forall n \in \{1, 2, \dots, N\}, \forall t \in \{0, 1, \dots, T-1\}. \quad (3)$$

$$s_{nm}^t, s_{mn}^t \in \{-1, 0, 1\}, s_{nm}^t + s_{mn}^t = 0, \quad (4)$$

$$\forall n, m \in \{1, 2, \dots, N\}, \forall t \in \{0, 1, \dots, T-1\}. \quad (5)$$

$$s_{nm}^t = 0, \text{ if } \mathcal{C}_{nm}^t = 0. \quad (6)$$

$$\mathbb{G}_n^t = \mathbb{G}_n^{t-1} \cup \mathbb{G}^h(l_n^t). \quad (7)$$

$$\mathbb{G}_n^t, \mathbb{G}_m^t = \mathbb{G}_n^t \cup \mathbb{G}_m^t, \text{ if } \mathcal{C}_{nm}^t = 1. \quad (8)$$

$$l_n^0 = \mathbb{V}_0(n), l_n^T = \mathbb{V}_g(n), \forall n \in \{1, 2, \dots, N\}. \quad (9)$$

$$\sum_{\forall e_{ij} \in \mathcal{N}_{l_n^t}} M_{ij}^{nt} = 1, \sum_{\forall e_{ij} \notin \mathcal{N}_{l_n^t}} M_{ij}^{nt} = 0, \quad (10)$$

$$\forall n \in \{1, 2, \dots, N\}, \forall t \in \{0, 1, \dots, T-1\}. \quad (11)$$

$$\sum_{\forall n} \sum_{\forall i \neq j} M_{ij}^{nt} \neq 0, \forall t \in \{0, 1, \dots, T-1\}. \quad (12)$$

Eqn. (2) suggests the goal of the problem is to minimize the total cost of all agents across all time steps with two decision variables, movement set \mathcal{M} and coordination set \mathcal{S} . Eqn. (3) enforces, at each time step, each robot can participate in at most one coordination behavior. Eqn. (4) regulates that,

at each time step, one coordination behavior only occurs between one robot pair. Eqn. (5) shows that coordination is only possible when communication is available. Eqn. (6) defines the update mechanism of a partial map. Eqn. (7) is for graph knowledge exchange when communication is available. Eqn. (8) set the start and the goal for each robot. Eqn. (9) guarantees, at each time step, a robot can only move to a neighbor node or stay still in the neighbor set $\mathcal{N}_{l_n^t}$ of node l_n^t . Eqn. (10) assures no unnecessary stagnation. Eqns. (2),(5),(6),(7) are elaborated in this problem; others can be found in previous work [9]. Notice that the sensing region $\mathbb{G}^h(\cdot)$ and communication range $\{r_{nm}\}$ are decided by hardware, so we treat them as given inputs here.

IV. MATHEMATICAL ANALYSIS

In this section, we show the NP-hardness and present problem analysis of HPR-TCGRE.

A. NP-Hardness

By constraining the number of robot types to one, and the sensing region of any robot and communication range between any robot pairs to be large enough to cover the entire graph, the HPR-TCGRE problem becomes exactly TCGRE. That is to say, TCGRE is a specific version of HPR-TCGRE; since TCGRE is already proven NP-hard [9], HPR-TCGRE is also NP-hard.

B. Problem Analysis

1) *Cost Model*: Similar to previous work [9], since the coordination only occurs in a pair, then the supporting cost can be moved to the receiving robot, as we only care about the overall cost, so Eqn. (1) becomes

$$C_n^t = \begin{cases} c_{ij}^{h_n}, & \text{if } s_{nm}^t = 0; \\ \tilde{c}_{ij}^{h_{nm}}, & \text{if } s_{nm}^t = 1; \\ 0, & \text{if } s_{nm}^t = -1, \end{cases} \quad (13)$$

$$\text{where } \tilde{c}_{ij}^{h_{nm}} = \tilde{c}_{ij}^{h_{nm}} + c_{ij}^{h_m}.$$

2) *Distributed Goals*: Our fundamental goal is to utilize communication and coordination among robots to reduce the overall cost of all robots for reaching their goals. However, only when the communication channel is established, will there be any opportunity to coordinate, so the cost is only possible to be reduced with an active channel (Eqn. (5)). Thus, the objective function can be rewritten as

$$\min_{\mathcal{M}, \mathcal{S}} \sum_{t=0}^{T-1} \sum_{n=1}^N (1 - \mathcal{C}_{nm}^t) c_{ij}^{h_n} \quad (14)$$

$$+ \sum_{t=0}^{T-1} \sum_{n=1}^N \mathcal{C}_{nm}^t [(1 - s_{nm}^t) c_{ij}^{h_n} + s_{nm}^t \tilde{c}_{ij}^{h_{nm}}], \quad (15)$$

where Eqn. (14) refers to independent operation of each individual robot when no teammates are detected, while Eqn. (15) involves communication and coordination.

V. SOLUTION

Based on the mathematical analysis, the decision variable s_{nm}^t is only in Eqn. (15) but not Eqn. (14), one simple solution is to divide the problem into two sub-problems: independent movement (Eqn. (14)) and coordination (Eqn. (15)), depending on whether other robots are detected. Regarding detection, since the communication range is usually much

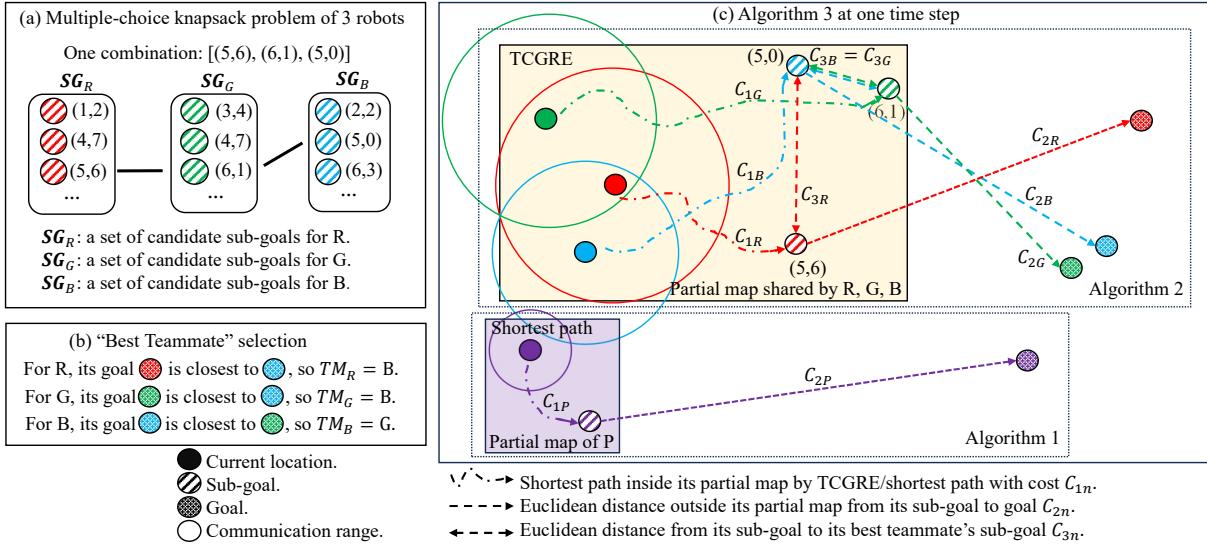


Fig. 2: Solution Overview.

larger than the sensing range, we use communication as the indicator, i.e., $\mathcal{C}_{nm}^t = 0/1$.

A. Individual Movement

When a robot detects no teammates, i.e., $\mathcal{C}_{nm}^t = 0, \forall m$, no coordination is available, so it approaches its goal individually. The objective of all robots in Eqn. (12) can be distributed to each individual robot n as

$$\min_{\mathcal{M}} \sum_{t=0}^{T-1} c_{ij}^{hn}. \quad (14)$$

If the entire graph is provided, it is a shortest path problem. However, partial observability (Eqn. (6)) makes the nodes and edges outside its partial map \mathbb{G}_n^t invisible. Hence, the shortest path algorithm is only applicable locally inside the partial map, outside which we use Euclidean distance to estimate the real cost. Specifically, to run the shortest path algorithm, we need to assign a goal for each robot in its partial map, called sub-goal. Moreover, the partial map grows as it moves, so the sub-goal also changes constantly along its way. The idea is *if all the sub-goals selected for an individual robot are on its optimal path to its final goal, the solution is optimal*. That is to say, this sub-problem can be equivalently transformed into deciding a sub-goal for each partial map.

In this paper we propose an A*-like algorithm, as in Alg. 1, that decides a sub-goal, using the sum of 1) C_1 : its minimal cost from its current node to a potential sub-goal (line 5), and 2) C_2 : the Euclidean distance between this sub-goal and the final goal (line 6). If the Euclidean length of an edge is no greater than its cost, this heuristic is admissible, as the first minimal cost is optimal and the second Euclidean distance is an underestimate of the minimum cost between the sub-goal and final goal.

B. Coordination

While robot n detects some teammates, i.e., $\mathcal{C}_{nm}^t = 1, \exists m$, directly or indirectly via intermediate robots, they need to consider coordination to minimize their total cost, almost equivalent to TCGRE, except with heterogeneous robots. In a robot group \mathbf{RG} , all robots communicate and share partial maps. Similar to Sec. V-A, the problem becomes deciding

Algorithm 1: Individual $(\mathbb{G}_n^t, l_n^t, \mathbb{V}_g[n])$

```

1 Generate a candidate sub-goal list,  $SG_n^t$ , in  $\mathbb{G}_n^t$ ;
2 if  $\mathbb{V}_g[n] \in \mathbb{G}_n^t$ , then  $SG_n^t = [\mathbb{V}_g[n]]$ ;
3  $min\_cost \leftarrow \infty$ ;  $min\_Solution \leftarrow \emptyset$ ;
4 for  $subgoal \in SG_n^t$  do
5    $\mathbf{P}, C_1 = ShortestPath(\mathbb{G}_n^t, l_n^t, subgoal, \mathbf{c})$ ;
6    $C_2 = d(subgoal, \mathbb{V}_g[n])$ ;
7   if  $C_1 + C_2 < min\_cost$  then
8      $min\_cost = C_1 + C_2$ ;
9      $min\_Solution = [\mathbf{P}, C_1]$ ;
10 return  $min\_Solution$ 

```

sub-goals for each \mathbf{RG} . In addition, they also need to consider future coordination opportunities; the main idea is that, *for any robots in a robot group whose goals are close, they are potentially good teammates, and thus their sub-goals should also be close*. In other words, in Alg. 2, (in line 12) we approximate Eqn. (13) with

$$\sum_{t=0}^{T-1} \sum_{n=1}^N C_{1n} + C_{2n} + \alpha_n C_{3n}, \quad (15)$$

where $\sum_{t=0}^{T-1} \sum_{n=1}^N C_{1n}$ is Eqn. (13) inside a partial map, solvable by any algorithms for TCGRE [9], with one more dimension (robot type) in edge costs (lines 6 and 8); C_{2n} is the Euclidean distance between each robot n 's sub-goal and final goal (line 9); C_{3n} is the sub-goal distance between each robot n and its best teammate TM_n (line 11), whose final goal is closest to robot n 's (line 10), to encourage robots whose final goals are close to each other to stay close along their journey so they have high chances of supporting each other to reduce cost. C_{3n} is weighed by a coefficient α_n with respect to C_{1n} and C_{2n} . Note that $\mathbf{L}_{\mathbf{RG}}^t = \{l_n^t | \forall n \in \mathbf{RG}\}$; $\mathbf{c} = \{c_{ij}^{hn} | \forall e_{ij} \in \mathbb{G}_{\mathbf{RG}}^t, \forall h_n\}$; $\hat{\mathbf{c}} = \{\hat{c}_{ij}^{h_{nm}} | \forall e_{ij} \in \mathbb{G}_{\mathbf{RG}}^t, \forall h_n, h_m\}$.

We treat Eqn. (15) for each \mathbf{RG} as a multiple-choice knapsack problem [14] with special weights as in Fig. 2, where we need exactly one item (sub-goal) from each class (candidate sub-goals of each robot) of a k -class knapsack, with a cost

calculated by Eqn. (15). Minimizing cost can be regarded as maximizing cost reduction, similar to previous work [9]. For this problem, we use Cartesian product to produce all sub-goal combinations (line 3) and run an exhaustive search (line 4) to find the best combination (lines 12-14).

Algorithm 2: Coordination ($\mathbb{G}_{\text{RG}}^t, \mathbb{L}_{\text{RG}}^t, \text{RG}, \mathbb{V}_g$)

```

1 Generate a list of candidate sub-goal lists,
 $\mathbb{SG}^t = \{\mathbb{SG}_n^t, \forall n \in \text{RG}\}$ , in  $\mathbb{G}_{\text{RG}}^t$ ;
2 if  $\mathbb{V}_g[n] \in \mathbb{G}_{\text{RG}}^t$ , then  $\mathbb{SG}_n^t = [\mathbb{V}_g[n]]$ ;
3  $min\_cost \leftarrow \infty$ ;  $min\_Solution \leftarrow \emptyset$ ;
4  $\mathbb{SSG}^t = Product(*\mathbb{SG}^t)$ 
5 for  $\mathbb{SG}^t \in \mathbb{SSG}^t$  do
6    $\mathbb{P}, \mathbb{C}_1 = \text{TCGRE}(\mathbb{G}_{\text{RG}}^t, \mathbb{L}_{\text{RG}}^t, \mathbb{SG}^t, \mathbf{c}, \hat{\mathbf{c}})$ ;
7   for  $n \in \text{RG}$  do
8      $C_{1n} = \mathbb{C}_1[n]$ ;
9      $C_{2n} = d(\mathbb{SG}^t[n], \mathbb{V}_g[n])$ ;
10     $TM_n = \operatorname{argmin}_{m \in \text{RG}} d(\mathbb{V}_g[n], \mathbb{V}_g[m])$ ;
11     $C_{3n} = d(\mathbb{SG}^t[n], \mathbb{SG}^t[TM_n])$ ;
12     $C = \sum_{n \in \text{RG}} C_{1n} + C_{2n} + \alpha_n C_{3n}$ ;
13    if  $C < min\_cost$  then
14       $min\_cost = C$ ;
15       $min\_Solution = [\mathbb{P}, \mathbb{C}_1]$ ;
16 return  $min\_Solution$ 

```

C. Overall Solution

Since we already have Alg. 1 and Alg. 2 for the two sub-problems, now we only need to apply them in the corresponding scenarios. The overall algorithm is shown in Alg. 3. For each time step, we search each robot's communication channel and build robot groups \mathbb{RG}^t : one robot group RG includes a group of robots who are communicating directly or indirectly (sharing the same partial map). For each RG , if it only has one robot, it means the robot is moving individually and we run Alg. 1 (lines 8-15); otherwise, the robot group may need coordination and we run Alg. 2 (lines 16-25). Note that in Alg. 3, we build RG for individuals too ($|\text{RG}| = 1$) for code efficiency; in concept individuals don't form RG s.

Specifically, to better demonstrate the key ideas of the algorithm, some details are not shown in the algorithm definitions, which have great impact on the algorithm performance and efficiency: 1) In both Alg. 1 and Alg. 2, the candidate sub-goals are only nodes on the frontier of each partial map that are not dead ends. To improve efficiency, candidates may be limited to nodes near its current node, at the expense of some performance loss. 2) In Alg. 2, while calculating $\sum_n C_{3n}$ for all combinations of one RG , we only need to sum the counterpart for each cluster, without going through every combination; a cluster is a minimal set of robots that includes all "best teammates" of the robots in the cluster. 3) In Alg. 3, Alg. 1 or Alg. 2 only needs to be called when the partial map is updated or some members of the RG are changed. 4) In both Alg. 1 and Alg. 2, when deciding sub-goal(s), an ϵ -greedy model [15] can be applied to escape local optima: we select the "best" sub-goal(s) with a probability ϵ , and other candidates with an even chance.

Algorithm 3: HPR-TCGRE ($\mathbb{G}, \mathbb{V}_0, \mathbb{V}_g$)

```

1 Initialize paths for all robots  $\mathbb{P}_{min} = [\emptyset] * N$ ;
2 Initialize total cost  $total\_cost = 0$ ;
3 Initialize every robot's partial map  $\mathbb{G}_n^0 = \emptyset, \forall n$ ;
4 Initialize the list of all robot groups  $\mathbb{RG}^0 = \emptyset$ ;
5 for  $t = 1$  to  $T - 1$  do
6   Update  $\mathbb{RG}^t$  and partial maps  $\mathbb{G}_n^t, \forall n$ ;
7   for  $\text{RG} \in \mathbb{RG}^t$  do
8     if  $|\text{RG}| = 1$  then
9        $\mathbb{P}, \mathbb{C} = Individual(\mathbb{G}_n^t, l_n^t, \mathbb{V}_g[n])$ ;
10       $edge = \mathbb{P}[0]$ ;  $cost = \text{cost of edge}$ ;
11       $\mathbb{P}_{min}[n].append(edge)$ ;
12       $total\_cost = total\_cost + cost$ ;
13      Update its coordinate  $l_n^t = edge[1]$ ;
14      if  $l_n^t = \mathbb{V}_g[n]$  then
15        Ignore  $n$  while updating  $\mathbb{RG}^{t+1}$ ;
16    else
17      Share partial maps  $\mathbb{G}_{\text{RG}}^t = \bigcup_{n \in \text{RG}} \mathbb{G}_n^t$ ;
18       $\mathbb{P}, \mathbb{C} = Coordination$ 
 $(\mathbb{G}_{\text{RG}}^t, \mathbb{L}_{\text{RG}}^t, \text{RG}, \mathbb{V}_g)$ ;
19      for  $n \in \text{RG}$  do
20         $edge = \mathbb{P}[n][0]$ ;  $cost = \text{cost of edge}$ ;
21         $\mathbb{P}_{min}[n].append(edge)$ ;
22         $total\_cost = total\_cost + cost$ ;
23        Update its coordinate  $l_n^t = edge[1]$ ;
24        if  $l_n^t = \mathbb{V}_g[n]$  then
25          Ignore  $n$  while updating  $\mathbb{RG}^{t+1}$ ;
26 return  $\mathbb{P}_{min}, total\_cost$ 

```

VI. RESULTS

We conduct a simple case study to explain the algorithms and experiments on a variety of large graphs of the same size but different structures to evaluate the effect of number of robot types, size of sensing region, and size of communication range on the algorithm performance. We also conduct ablation experiments to evaluate each part of the algorithm.

A. Experiment Settings

We vary the number of robot types (labeled "#" in the results), sensing factor (the factor multiplying the graph length is the sensing distance, within which the nodes and edges can be detected), and communication factor (similar to sensing factor), which represent the three relaxations we have in this problem. The two sensing/communication factors are common for every robot; the common factors need to multiply another unique coefficient for each robot type to simulate the definition of an edge cost in the problem. For each setting, we run the algorithms in 100 graphs. Every graph is randomly generated with 20 nodes, 50% edges with base cost in [40,60], half among which are risky edges with 50% cost reduction if coordination happens, and random starts and goals for 10 robots. Since HPR-TCGRE is a new and complex problem, there is no state-of-the-art algorithm to compare against, and we conduct an ablation experiment with the following baselines: 1) The naive approach where every robot is moving individually without coordination

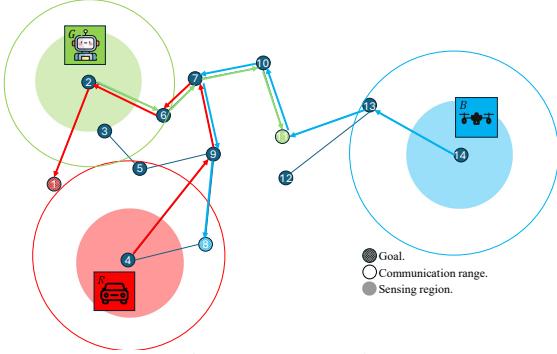


Fig. 3: Case Study.

(labeled "Naive"). 2) Alg. 3 without consideration of future coordination (labeled "w/o. C_3 "). 3) Alg. 3 with an ϵ -greedy model while deciding sub-goal(s) (labeled " $w/. \epsilon$ "), where $\epsilon = 0.1$. Furthermore, because we are experimenting on large graphs with a lot of risky edges, we are only using RHOCA [9] for TCGRE in Alg. 2. As heuristic algorithms sometimes get stuck in the dead ends, so we additionally set that a node can be only visited at most 5 times in "Alg. 3" and "w/o. C_3 ", in contrast to the probability model in " $w/. \epsilon$ ".

B. Results

1) *Case study*: We first use a simple graph as a case study to explain how each algorithm works, as in Fig. 3. To make it easier to explain, we assume 1) it is an undirected graph with no risky edge, 2) if a robot senses an edge, it can also sense both endpoints; 3) if two robots' communication ranges overlap, they share partial maps; 4) edge cost is the edge length. In this graph, G starts from node 2 and attempts to reach node 11, R from 4 to 1, and B from 14 to 8.

G and B start sharing partial maps in the beginning. At the moment, communication does not help navigation because their sensing regions are not connected, so "w/o. C_3 " and Naive will give the same first step — G reaches 6, R reaches 8, and B reaches 13 — to minimize their estimated individual cost $C_1 + C_2$. Next, "w/o. C_3 " and Naive will both move R back to 4, for a minimal $C_1 + C_2$. Without limited revisit, R will be stuck in 4 and 8 back and forth; even with limited revisit, R will go for $9 \rightarrow 5 \rightarrow 3$ before moving to 7, for the same reason. Similarly, B will move to 12 before reaching 11. G will reach its goal smoothly because of the graph structure.

However, Alg. 3 assumes G and R are best teammate in the beginning and tries to make their sub-goals closer to minimize $C_1 + C_2 + C_3$, so R will reach 9 instead of 8; G and B will still move to 6 and 13. Then, G and R can both sense 7, which will make their sensing regions connected, which will provide R with a path to 2 and thus avoid the dead end, node 3. Meanwhile, B at 13 will also be able to communicate with R at 9, and R becomes B's best teammate, which also helps B move to 11 instead of 12. As a result, Alg. 3 produces paths as shown in Fig. 3. " $w/. \epsilon$ " is likely to follow the same greedy strategy as Alg. 3, but also has a chance of ϵ to select alternative nodes in each step.

Yet, this is only one case for explanatory purpose, not representative of the general case, so we conduct more experiments with the setting stated in the Sec. VI-A.

s factor	c factor	# types	avg cost	avg runtime
0.2	0.3	2	176	1.0366
0.2	0.3	4	158.2	1.8264
0.2	0.3	6	188	1.6522
0.2	0.3	8	137.4	0.4232
0.2	0.3	10	169	1.0004

TABLE I: Average cost and runtime with # robot types.

# line	s factor	c factor	Alg. 3	w/. ϵ	w/o. C_3	Naive
1	0.1	0.3	113	159	129	241
2	0.1	0.4	100	149	104	237
3	0.1	0.5	119	124	131	244
4	0.1	0.6	122	121	119	240
5	0.1	0.7	123	108	121	245
6	0.15	0.3	112	122	118	245
7	0.15	0.4	123	115	118	230
8	0.15	0.5	108	125	118	246
9	0.15	0.6	104	131	116	234
10	0.15	0.7	100	96	119	241
11	0.2	0.3	100	115	101	237
12	0.2	0.4	117	112	116	237
13	0.2	0.5	110	116	118	235
14	0.2	0.6	107	109	105	235
15	0.2	0.7	98	110	102	235
16	0.25	0.3	98	106	97	238
17	0.25	0.4	110	116	88	236
18	0.25	0.5	82	120	111	229
19	0.25	0.6	97	106	116	234
20	0.25	0.7	111	103	108	230
21	0.3	0.3	100	97	92	229
22	0.3	0.4	108	99	93	226
23	0.3	0.5	91	101	111	233
24	0.3	0.6	96	89	102	224
25	0.3	0.7	96	125	92	235

TABLE II: Average cost w/. sensing/communication factors.

2) *Robot types*: Theoretically, the runtime does not change with the number of robot types, because the computation time to construct and search on JSG [1] for TCGRE remains the same. JSG is not implemented because the large size of our graphs. We conduct some small experiments to verify our theory, as shown in the Table. I.

The table is the average results of our proposed algorithm in 10 random graphs with 20 nodes and 10 robots. As is shown, there is no particular pattern for the average cost and runtime as the number of robot types increases, because the parameters for each robot type are randomly generated. Therefore, in the next section, we fix its value at 2.

3) *Sensing and communication factors*: We use factors instead of the actual sensing and communication ranges in the experiments because their significance varies with the graph size. Instead, a factor is directly related to how many information exchanges a robot potentially needs on average to know the whole graph or communicate with other robots. The range of communication factor is larger than that of the sensing factor as is the most cases in the real world.

The average cost of all 25 lines in Table. II would be 106 for Alg. 3, 115 for " $w/. \epsilon$ ", 110 for "w/o. C_3 ", and 236 for Naive, so Alg. 3 with limited revisit outperforms all other baselines. In the experiment setting, there are 50% risky edges among all edges, each with 50% cost reduction after coordination, which leads to an expected overall cost

s factor	Alg3↓	w/. ϵ ↓	w/o. C_3 ↓	Naive↓
0.1	115	132	121	242
0.15	109	118	118	240
0.2	106	112	108	236
0.25	100	110	104	234
0.3	98	102	98	230

TABLE III: Average cost with sensing factor.

reduction rate of 25% from coordination. However, Alg. 3 reduces 55% cost compared to the Naive approach, which indicates the algorithm modifies path selection by a large percentage. Surprisingly, limited revisit works much better than an epsilon greedy model here, as Alg. 3 and "w/o. C_3 " both beat "w/. ϵ ". (More unlisted experiment results without limited revisit also point to the same conclusion.) Moreover, the fact that Alg. 3 tops "w/o. C_3 " verifies that our best teammate prediction method in Alg. 2 is also effective.

By averaging lines from Table. II, we can observe the impact of each factor, as in Tables. III and IV. In Table III, it is clear that the average cost decreases with increasing sensing factor, which is reasonable, since the larger region a robot can sense, the more information it can gather to help navigation. However, to our surprise, this is not the case for the communication factor in Table. IV, likely due to the limited size of the graph. As there are 10 robots in a 20-node graph, a communication factor of 0.3 is already large enough for every robot to communicate with one another via multi-hops. Increasing the value does not make any difference in a 20-node graph, and the fluctuation in Table. IV is caused by random graph structure. A fix would be either decreasing its value or increasing the graph size. However, decreasing this value will force a smaller sensing factor, which makes it harder for robots to find candidate sub-goals; it is also very difficult to increase the graph size, as the runtime will increase dramatically, unless we reduce the number of robots. In the future, we will try to improve our algorithm efficiency and expand the experiments into larger graphs.

VII. CONCLUSIONS AND DISCUSSIONS

We study a more general version of TCGRE [9], called HPR-TCGRE, by relaxing three assumptions with Heterogeneous robots, Partial observability, and Realistic communication. We construct a new combinatorial optimization problem based on the previous one [9]. (Some equations from previous work are not elaborated due to page limit.) After analysis, we break down the problem into two sub-problems: individual movement and coordination. The coordination sub-problem is then broken into TCGRE problems inside their shared local partial maps of robot groups. Globally for the group we solve the multi-choice knapsack problem with an estimated cost model considering future coordination.

Our HPR-TCGRE problem is difficult as the robots are moving in an unknown environment without prior knowledge about the environment in the beginning. We apply a communication model to help them explore the graph and enable coordination. But the model is not completely realistic, because we are not considering the transmission cost function and the transmission time; both involve how

c factor	Alg3	w/. ϵ	w/o. C_3	Naive
0.3	105	120	107	238
0.4	112	118	104	233
0.5	102	117	118	238
0.6	105	111	112	234
0.7	106	108	108	238

TABLE IV: Average cost with communication factor.

much graph information they need to exchange at once and at what distances they should exchange the information, which may also introduce a data filtering mechanism to decide which nodes and edges are worth going to. Furthermore, we use an estimated cost model that considers future coordination, because robots fall into individual moving mode and coordination mode interchangeably, and they are not really two independent sub-problems. However, the estimation may not work very well in complex environments, as it is a simple linear model. In the future, it may be necessary to develop a teammate position prediction mechanism, with which the problem can be solved in a more comprehensive manner.

REFERENCES

- [1] M. Limbu, Z. Hu, S. Oughourli, X. Wang, X. Xiao, and D. Shishika, "Team coordination on graphs with state-dependent edge costs," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 679–684, IEEE, 2023.
- [2] J. L. Adler and V. J. Blue, "A cooperative multi-agent transportation management and route guidance system," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 5-6, pp. 433–454, 2002.
- [3] Y. Hu, M. Chen, W. Saad, H. V. Poor, and S. Cui, "Distributed multi-agent meta learning for trajectory design in wireless drone networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3177–3192, 2021.
- [4] C. Yu, A. Velu, E. Vinitksy, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative, multi-agent games," 2022.
- [5] P. Surynek, "An optimization variant of multi-robot path planning is intractable," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 24-1, pp. 1261–1263, 2010.
- [6] J. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. H. Torr, P. Kohli, and S. Whiteson, "Stabilising experience replay for deep multi-agent reinforcement learning," in *International conference on machine learning*, pp. 1146–1155, PMLR, 2017.
- [7] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8–12, 2017, Revised Selected Papers 16*, pp. 66–83, Springer, 2017.
- [8] B. Liu, X. Xiao, and P. Stone, "Team orienteering coverage planning with uncertain reward," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9728–9733, IEEE, 2021.
- [9] Y. Zhou, M. Limbu, G. J. Stein, X. Wang, D. Shishika, and X. Xiao, "Team coordination on graphs: Problem, analysis, and algorithms," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5748–5755, IEEE, 2024.
- [10] M. Limbu, Z. Hu, X. Wang, D. Shishika, and X. Xiao, "Team coordination on graphs with reinforcement learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024.
- [11] C. Hunt, *TCP/IP network administration*, vol. 2. " O'Reilly Media, Inc.", 2002.
- [12] H. Friis, "A note on a simple transmission formula," *Proceedings of the IRE*, vol. 34, no. 5, pp. 254–256, 1946.
- [13] J. Wu and I. Stojmenovic, "Ad hoc networks," *Computer*, vol. 37, no. 2, pp. 29–31, 2004.
- [14] H. Kellerer, U. Pferschy, D. Pisinger, H. Kellerer, U. Pferschy, and D. Pisinger, "The multiple-choice knapsack problem," *Knapsack problems*, pp. 317–347, 2004.
- [15] R. S. Sutton, "Reinforcement learning: An introduction," *A Bradford Book*, 2018.