

Robot Team Data Collection with Anywhere Communication

Matthew A. Schack[†], John G. Rogers[‡], Qi Han[†], and Neil T. Dantam[†]

Abstract—Using robots to collect data is an effective way to obtain information from the environment and communicate it to a static base station. Furthermore, robots have the capability to communicate with one another, potentially decreasing the time for data to reach the base station. We present a Mixed Integer Linear Program that reasons about discrete routing choices, continuous robot paths, and their effect on the latency of the data collection task. We analyze our formulation, discuss optimization challenges inherent to the data collection problem, and propose a factored formulation that finds optimal answers more efficiently. Our work is able to find paths that reduce latency by up to 101% compared to treating all robots independently in our tested scenarios.

I. INTRODUCTION

Robots teams are an effective way to collect and communicate data from the environment—e.g., in disaster response [1] or environmental monitoring [2]. Combining motion and communication enables a robot team to collect data from various sites, transmit data across areas not physically traversable, and move to circumvent wireless interference (see Fig. 1). Prior work has addressed robot data collection for independent robots [3], [4], [5] or communication at a priori locations [6], [7]. Generalizing such approaches to communicate at arbitrary locations improves achievable latency of robot data collection.

We develop a mixed-integer formulation for robot data collection, analyze its complexity, and produce a more efficiently solvable factored form. Optimal (minimum latency) data collection may require cooperation and communication between team members, and decisions about motion and communication are coupled because wireless communication depends on the team members' positions. Our formulation integrates network routing decisions with convex optimization based path planning [8]. A key feature of many data collection scenarios is the presence of multiple global optima, which poses challenges for efficient optimization. We address the optimization challenge of robot data collection with a factored formulation that improves the tightness of the linear relaxation and reduces the necessary work to resolve multiple optima.

This paper is organized as follows. Sec. II reviews related work in robot data collection. Sec. III formally defines the robot data collection problem. Sec. IV introduces our mixed-integer formulation. Sec. V analyzes this formulation and develops the efficient, factored form. Finally, we evaluate our approach in Sec. VI and show up to 101% improved latency.

* This work was supported in part by the ARL DCIST CRA [W911NF-17-2-0181] and the NSF [CNS-1823245].

[†] Department of Computer Science, Colorado School of Mines, USA. {mschack, qhan, ndantam}@mines.edu.

[‡] United States Army Research Laboratory, Adelphi, MD, USA. john.g.rogers59.civ@mail.mil

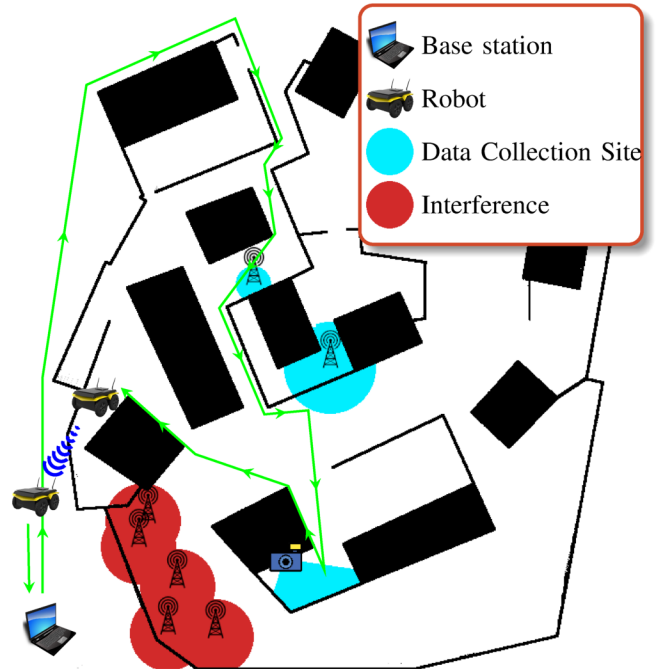


Fig. 1: Lower latency data collection by considering robot-to-robot communication. The robots start at the base station (shown as the computer) and must collect the data from the blue sites. Robot communication is impossible in the red areas. One robot collects all the data from the sites and then transmits it across the wall to another robot team member, resulting in lower latency than robots acting independently.

II. RELATED WORK

We review related works in data collection and similar problems in periodically and constantly connected networks.

In robot data collection, a robot obtain data from static sites and deliver it to a static base station. This problem is a special class of the traveling salesman problem (TSP) called *TSP with neighborhoods* (TSPN) where a mobile robot must visit the areas around sites [9]. Leading approaches have scaled to a large number of sites by combining sites whose regions overlap [3], using heuristics and tangent graphs to find paths that ensure a robot spends enough time at a site to obtain all data [4], or assuming sites can transmit data and only visiting specified sites [5]. These works focus on using a single robot to collect the data, and extensions to use multiple robots [4] create independent paths, rather than having robots communicate with one another. [6] evaluates the effects of robot-to-robot communication through discretization and finds a latency-constrained path. Our work finds a minimum latency path using multiple robots that can communicate with one

another, expanding the possible solution set.

Previous approaches to create a constantly connected network between robots and a static base station [10], [11], [12], [13], [14], [15], [16] can accomplish many tasks, including data collection. While a constantly connected network between sites and the base station may sometimes be optimal, enforcing constant communication as a constraint limits the valid space the entire team can move, potentially resulting in incomplete performance where some sites cannot be visited. In this work, we do not consider communication as a constraint; instead, we use periodic communication to decrease latency.

Approaches for periodically connected robot networks have the team perform a task with periodic connectivity guarantees [17], [18] or introduce rendezvous points where robots may communicate [7], [19]. Using a priori rendezvous points can reduce latency compared to assuming robot independence, but their effectiveness depends on the rendezvous points chosen. We generalize such approaches by permitting robots to communicate not only at a priori rendezvous points, but anywhere within their communication range.

III. PROBLEM DEFINITION

We address robot team data collection. Robots must obtain data from multiple sites in the environment, e.g., by using sensors on the robot or downloading from a statically placed sensor at that site. Then, the robots must transfer all collected data to a static base station through a combination of motion and wireless communication, considering both obstacles for motion and wireless interference. We seek a sequence of sites to visit, transmissions between team members, and accompanying paths to route all data to the base station with minimum latency.

Definition 1: A robot data collection problem is the tuple $\Sigma = (\mathcal{X}, \mathcal{R}, \mathcal{S}, \mathcal{O}, \Psi, b, c, u, q)$.

- $\mathcal{X} \subset \mathcal{SE}(2)$ is the environment.
- $\mathcal{R}$ is a finite set of homogeneous robots.
- $\mathcal{S}$ is a set of sites containing data. Each site $s \in \mathcal{S}$ has associated (1) a convex region ($\mathcal{X}_s \subseteq \mathcal{X}$) which the robot occupies to collect data, (2) an amount of data ($d_s^{(0)} \in \mathbb{R}^+$), and (3) a data collection rate ($p_s \in \mathbb{R}^+$).
- $\mathcal{O} \subseteq \mathcal{X}$ is a set of obstacles.
- Ψ is a set of interfering transmitters. Each interfering transmitter $\psi \in \Psi$ has an interference range c_ψ .
- b is the static base station with location $\mathbf{x}_b \in \mathcal{X}$.
- $c \in \mathbb{R}^+$ is the communication range of the robots.
- $u \in \mathbb{R}^+$ is the robot velocity.
- $q \in \mathbb{R}^+$ is the wireless data transfer rate.

Space $\mathcal{X}$ consists of disjoint valid $\mathcal{X}_{\text{valid}}$ and invalid and $\mathcal{X}_{\text{invalid}}$ spaces. Invalid space $\mathcal{X}_{\text{invalid}}$ is obstacle region $\mathcal{O}$, and valid space $\mathcal{X}_{\text{valid}}$ consists of all other points in $\mathcal{X}$.

Similar to other data collection works, we assume that wireless communication within c happens with certainty [3], [4], [5] and that the environments have enough clearance that a robot cannot block a path—i.e., robot-to-robot collision avoidance will not affect the optimality of a robot's path [4]. In addition, we assume communication is blocked with certainty

if either the sender or receiver is inside interference. Finally, we account for the limited bandwidth by restricting a robot to only be able to send and receive from a single source at any time, which is equivalent to sharing equal bandwidth.

We seek a *tour* for each robot which routes all data to the base station—i.e., the sequence of sites $s \in \mathcal{S}$ to visit and team members $r \in \mathcal{R} \cup \{b\}$ to exchange data with—as well as the continuous path to accomplish the tour. We call each discrete step of each robot's tour a *tour leg* and seek to minimize the largest ending time of any robot's last tour leg—i.e., the robot's latency, $l'_r \in \mathbb{R}^+$.

$$\min \max_{r \in \mathcal{R}} l'_r \quad (1)$$

Def. 1 is a generalization of the traveling salesman problem with Neighborhoods (TSPN), which is NP-hard [9]. That is, we may reduce TSPN to an instance of Def. 1 with only a single robot. Thus, Def. 1 must also be NP-hard.

Instances of Def. 1 typically have multiple globally optimal solutions to (1), which increases the challenge of efficiently finding solutions. Two issues cause multiple global optima. First, optima are not unique under reassignment of robots, so given optimal trajectories, we may change which robot follows which path without changing total latency. Second, optimality is determined only by the maximum latency robot in (1), so other robots may take other actions as long as they are completed before the slowest robot. Widely used branch-and-bound optimization techniques are less efficient in the presence of multiple global optima. In particular, such approaches prune sub-optimal variable assignments, but additional global optima cannot be pruned and are instead evaluated to completion.

IV. MIXED INTEGER LINEAR PROGRAM FORMULATION

We first formulate a Mixed Integer Linear Program (MILP) for robot data collection. Solving this MILP determines each robot's tour (i.e., a robot's data collections, receptions, and transmissions) as well as paths to accomplish the tour. Next, in Sec. V, we will analyze this MILP and extend it to a more efficiently solvable factored formulation.

We divide our program into three parts: robot tour constraints (Sec. IV-A); path planning constraints (Sec. IV-B); and timing constraints that determine the latency for a robot based on its path and data transfer times (Sec. IV-C).

Our MILP finds a tour that will route all data to the base station. Each leg of the tour—i.e., a single data collection, reception, or transmission—has an associated robot path to move into communication range or visit a specific site's region. We consider each tour leg as a discrete timestep, and constrain the robots performing the leg to end their path either within communication range or within a site's region. We note that program timesteps are different from actual time and impose only an ordering of actions for associated robots. A timestep means the leg must complete before associated robots can perform another leg with greater timestep. This formulation enables concurrent motion, communication, and collection, despite a solution with actions

at different timesteps. Sec. IV-C describes the rules for timing and concurrency.

An optimal solution will minimize the maximum robot latency (1); however, a MILP cannot directly minimize a maximum. Instead, we define new variable, l_{team} , representing the team's latency, and constrain l_{team} to be greater or equal to every robot's latency.

$$\begin{aligned} \min \quad & l_{\text{team}} \\ & l_r^{(h)} \leq l_{\text{team}} \quad \forall r \in \mathcal{R}, \end{aligned} \quad (2)$$

where $l_r^{(h)}$ is the latency of robot r at the final timestep h .

A. Tour Constraints

Tour constraints determine the sequence of data collections, receptions, and transmissions. We model the tour as a max-flow problem [20]. Later, in Sec. IV-B, we will address the paths required to accomplish the tour.

We create a graph representing data flow between sites, robots, and the base station. Graph vertices consist of all robots, sites, and the base station $\mathcal{V} = \mathcal{S} \cup \mathcal{R} \cup \{b\}$. Graph edges exist between every site and robot, every robot pair, and every robot and the base station $\mathcal{V}_e = \{(s, r) \forall s \in \mathcal{S}, r \in \mathcal{R}\} \cup \{(r, r') \forall r, r' \in \mathcal{R}\} \cup \{(r, b) \forall r \in \mathcal{R}\}$, admitting any possible tour. At each step i , each vertex $v \in \mathcal{V}$ has data $d_v^{(i)}$, which changes with the flow $f_{vv'}^{(i)}$ along edges. When we sum over pairs of vertices if the pair is not an edge in $\mathcal{V}_e$ —e.g., $f_{sb}^{(i)} \forall s \in \mathcal{S}$ —we define the variable's value as zero.

$$d_v^{(0)} = 0 \quad \forall v \in \mathcal{V} \setminus \mathcal{S}, d_b^{(h)} = \sum_{s \in \mathcal{S}} d_s^{(0)} \quad (3)$$

$$d_v^{(i+1)} = d_v^{(i)} + \sum_{v' \in \mathcal{V}} \left(f_{vv'}^{(i)} - f_{vv'}^{(i)} \right) \quad \forall v \in \mathcal{V}, i < h \quad (4)$$

$$\sum_{v' \in \mathcal{V}} f_{vv'}^{(i)} \leq d_v^{(i)} \quad \forall v \in \mathcal{V}, i \leq h \quad (5)$$

We add a binary variable indicating the specific tour leg happening at timestep i , allowing us to conditionally enforce motion (described in Sec. IV-B) if the tour leg is a part of the overall tour. Tour leg $t_{vv'}^{(i)}$ is true if and only if vertices v and v' transfer data at timestep i . We disallow simultaneous data transfers, since we only traverse one leg per timestep, and force $t_{vv'}^{(i)}$ to be true if $f_{vv'}^{(i)}$ is positive.

$$\sum_{v, v' \in \mathcal{V}_e} t_{vv'}^{(i)} \leq 1 \quad \forall i \leq h \quad (6)$$

$$t_{vv'}^{(i)} \sum_{s \in \mathcal{S}} d_s^{(0)} \geq f_{vv'}^{(i)} \quad \forall (v, v') \in \mathcal{V}_e, i \leq h \quad (7)$$

We allow for parallel motion and data transmissions even though (6) limits the team to one tour leg per timestep since a later timestep does not necessarily mean the leg happens later in continuous time. We discuss the coupling between discrete timesteps and the latency for every robot in Sec. IV-C.

We add to the classic max-flow problem constraints to define the durations of data transfers, which will be used in Sec. IV-C to determine latency. We represent a data

transfer's duration as $\gamma_r^{(i)}$. Data transfer durations depend on the amount of data ($f_{vv'}^{(i)}$), the data transfer rate (q), or the data collection rate (p_s).

$$\gamma_r^{(i)} \geq \frac{f_{rr'}^{(i)}}{q}, \gamma_{r'}^{(i)} \geq \frac{f_{rr'}^{(i)}}{q} \quad \forall r \in \mathcal{R}, r' \in \mathcal{R} \cup \{b\}, i \leq h \quad (8)$$

$$\gamma_r^{(i)} \geq \frac{f_{sr}^{(i)}}{p_s} \quad \forall r \in \mathcal{R}, s \in \mathcal{S}, i \leq h \quad (9)$$

B. Path Planning Constraints

We create constraints describing a robot's path at each timestep and conditionally constrain timestep ending position if involved with a tour leg at that timestep. This conditional constraint ensures each tour leg has the appropriate motion to accomplish it. A robot on a tour leg must end its path within either a site's region or within communication range (Sec. IV-B.1), and we additionally ensure that communication occurs outside of interference in Sec. IV-B.2.

Optimization-based path planning is an established technique, and we adapt the approach of [8], which describes a tight MILP formulation. However, our requirements differ from this prior work in that the goal for a robot's path at timestep i is not known a priori but instead, depends on the decision variables for the tour leg $t^{(i)}$. Additionally, we sequence multiple path planning problems where the starting location at timestep i is the ending location at $i-1$. We briefly review the MILP formulation of [8] (equations (10) to (12)) and refer the reader to [8] for a more in-depth discussion of the formulation. Then, we describe our extensions for interaction between a robot's tour and the corresponding path.

The formulation in [8] requires the environment to be decomposed into convex regions. Any environment's valid space—even non-convex environments—is decomposable into convex shapes with space decomposition methods such as Delaunay Triangulations [21] or Voronoi Diagrams [22].

Let the convex region decomposition be Ω where $\bigcup_{\omega \in \Omega} \omega = \mathcal{X}_{\text{valid}}$, the prospective cone of each region be $\bar{\Omega}$, the edge set between convex regions be $\mathcal{E}_{\omega} = \mathcal{E}_{\omega}^{\text{in}} \cup \mathcal{E}_{\omega}^{\text{out}} \forall \omega \in \Omega$, binary variables describing what edges are traversed be $y_{re}^{(i)}$, and the auxiliary variables describing the location of the robot be $\mathbf{z}_{re}^{(i)}$ and $\mathbf{z}'_{re}^{(i)}$. The robot may only move through valid space according to flow and inclusion constraints.

$$\sum_{e \in \mathcal{E}_{\omega}^{\text{in}}} \left(\mathbf{z}'_{re}^{(i)}, y_{re}^{(i)} \right) = \sum_{e \in \mathcal{E}_{\omega}^{\text{out}}} \left(\mathbf{z}_{re}^{(i)}, y_{re}^{(i)} \right) \quad \forall \omega \in \Omega, r \in \mathcal{R}, i \leq h \quad (10)$$

$$\sum_{e \in \mathcal{E}_{\text{start}}^{\text{out}}} y_{re}^{(i)} = \sum_{e \in \mathcal{E}_{\text{end}}^{\text{in}}} y_{re}^{(i)} = 1 \quad \forall r \in \mathcal{R}, i \leq h \quad (11)$$

$$\left(\mathbf{z}_{re}^{(i)}, y_{re}^{(i)} \right) \in \bar{\Omega}_u, \left(\mathbf{z}'_{re}^{(i)}, y_{re}^{(i)} \right) \in \bar{\Omega}_v \quad \forall e = (u, v) \in \mathcal{E}, r \in \mathcal{R} \quad (12)$$

We sequence paths from timestep to timestep, so the start location of the robot's i^{th} path is the ending location of its path at $i-1$. We also enforce that the zero-th path starts at

the base station.

$$\mathbf{z}_{re}^{(0)} = \mathbf{x}_b y_{re}^{(0)} \quad \forall r \in \mathcal{R}, e \in \mathcal{E}_{\text{start}}^{\text{out}} \quad (13)$$

$$\mathbf{z}_{r,(\text{start},\omega)}^{(i)} = \mathbf{z}_{r,(\omega,\text{end})}^{(i-1)} \quad \forall r \in \mathcal{R}, \omega \in \Omega, 1 \leq i \leq h \quad (14)$$

Next, we describe our extensions to [8] coupling ending positions at each timestep and the tour obtained in Sec. IV-A.

1) *Leg Completion Constraints*: We conditionally constrain a robot's ending position if it is involved in a tour leg.

We do not initially know the tours when creating the MILP, which means we do not know where robots will start or end paths at each timestep. Therefore, we must have edges $\mathcal{E}_{\text{end}}^{\text{in}}$ and $\mathcal{E}_{\text{start}}^{\text{out}}$ to every convex region of space in Ω , allowing the robot to start or end its path anywhere. Rather, we add additional constraints limiting a robot's final position by introducing a *big-M* [23]. We use the big-M to increase the size of all destinations—i.e., site regions or within communication range—where a tour leg is *not* occurring to the size of the entire space, leaving just the destination where the tour leg is occurring at the proper size.

A robot must be within a site's region $\mathcal{X}_s, s \in \mathcal{S}$ to receive data from that site—i.e., $t_{sr}^{(i)} = 1$. We enforce this conditional requirement by representing $\mathcal{X}_s$ in $Ax \leq b$ form—i.e., $A_s x \leq b_s \forall x \in \mathcal{X}_s \wedge A_s x \not\leq b_s \forall x \notin \mathcal{X}_s$, and adding the big-M to conditionally enable the constraint. We note that due to (11) there exists only one edge in $\mathcal{E}_{\text{end}}^{\text{out}}$ for any robot and timestep pair that is non-zero, so $\sum_{e \in \mathcal{E}_{\text{end}}^{\text{out}}} \mathbf{z}_{re}^{(i)}$ is the robot's position at the end of the path.

$$\begin{aligned} A_s \sum_{e \in \mathcal{E}_{\text{end}}^{\text{out}}} \mathbf{z}_{re}^{(i)} &\leq b_s + M \left(1 - t_{sr}^{(i)}\right) \\ \forall s \in \mathcal{S}, r \in \mathcal{R}, i &\leq h \end{aligned} \quad (15)$$

The transmitter and receiver must be within the communication range c for robot-to-robot or robot-to-base communication. We represent c to arbitrary precision using the same linear decomposition as [24], which inscribes an N sided equilateral polygon into a circle of radius c .

$$\begin{aligned} \left(\sum_{e \in \mathcal{E}_{\text{end}}^{\text{out}}} \mathbf{z}_{re}^{(i)} - \mathbf{x}_b \right) \begin{bmatrix} \sin\left(\frac{2\pi n}{N}\right) \\ \cos\left(\frac{2\pi n}{N}\right) \end{bmatrix}^\top &\leq c + M \left(1 - t_{rb}^{(i)}\right) \\ \forall r \in \mathcal{R}, n &\in [1, 2, \dots, N] \end{aligned} \quad (16)$$

$$\begin{aligned} \left(\sum_{e \in \mathcal{E}_{\text{end}}^{\text{out}}} \mathbf{z}_{re}^{(i)} - \sum_{e \in \mathcal{E}_{\text{end}}^{\text{out}}} \mathbf{z}_{r'e}^{(i)} \right) \begin{bmatrix} \sin\left(\frac{2\pi n}{N}\right) \\ \cos\left(\frac{2\pi n}{N}\right) \end{bmatrix}^\top &\leq c + M \left(1 - t_{rr'}^{(i)}\right) \\ \forall n \in [1, 2, \dots, N], r, r' \in \mathcal{R}, r &\neq r', i \leq h \end{aligned} \quad (17)$$

Similar to before, the additional $M \left(1 - t_{rr'}^{(i)}\right)$ expands the communication range if no data is transmitted between r and r' at timestep i , removing the constraint.

2) *Interference Constraints*: Environmental interference prevents wireless transmission, so the final positions for robot-to-robot and robot-to-base communication must be outside of interference. A robot is outside the range of each interfering

transmitter ψ if there exists at least one n that satisfies:

$$\begin{aligned} \left(\mathbf{x}_j - \sum_{e \in \mathcal{E}_{\text{end}}^{\text{out}}} \mathbf{z}_{re}^{(i)} \right) \begin{bmatrix} \sin\left(\frac{2\pi n}{N}\right) \\ \cos\left(\frac{2\pi n}{N}\right) \end{bmatrix} &\geq c_\psi, \\ \forall r \in \mathcal{R}, i &\leq h \exists n \in [1, 2, \dots, N]. \end{aligned} \quad (18)$$

Unfortunately, we cannot directly evaluate a $\exists$ operator in MILP formulations, so we add a binary variable $\alpha_{\psi rn}^{(i)}$ allowing us to turn off at most $N - 1$ instances of (18) if robot r is involved in a transmission at timestep i .

$$\begin{aligned} \left(\mathbf{x}_j - \sum_{e \in \mathcal{E}_{\text{end}}^{\text{out}}} \mathbf{z}_{re}^{(i)} \right) \begin{bmatrix} \sin\left(\frac{2\pi n}{N}\right) \\ \cos\left(\frac{2\pi n}{N}\right) \end{bmatrix} &\geq c_\psi - M \alpha_{\psi rn}^{(i)}, \\ \left(\mathbf{x}_j - \sum_{e \in \mathcal{E}_{\text{end}}^{\text{out}}} \mathbf{z}_{r'e}^{(i)} \right) \begin{bmatrix} \sin\left(\frac{2\pi n}{N}\right) \\ \cos\left(\frac{2\pi n}{N}\right) \end{bmatrix} &\geq c_\psi - M \alpha_{\psi r'n}^{(i)}, \\ \forall n \in [1, 2, \dots, N], \psi \in \Psi, r, r' \in \mathcal{R}, r &\neq r', i \leq h, \\ \sum_{n \in [1, 2, \dots, N]} \alpha_{\psi rn}^{(i)} &\geq t_{rr'}^{(i)}, \quad \sum_{n \in [1, 2, \dots, N]} \alpha_{\psi r'n}^{(i)} \geq t_{rr'}^{(i)}, \\ \forall n \in [1, \dots, N], \psi \in \Psi, r, r' \in \mathcal{V}, i &\leq h \end{aligned} \quad (19)$$

When $t_{rr'}^{(i)} = 0$, all $\alpha_{\psi rn}^{(i)}$ can be one, removing (19) from the MILP. When $t_{rr'}^{(i)} = 1$, there must be at least one $\alpha_{\psi rn}^{(i)} = 0$ satisfying (18), meaning the robot is out of interference.

Finally, robot motion takes time, which affects the latency. We define the time for robot r to move from the start region to the goal region during timestep i as $\delta_r^{(i)}$ and constrain it by the distance and velocity u .

$$\delta_r^{(i)} = \sum_{e \in \mathcal{E}} \frac{|\mathbf{z}_{re}^{(i)} - \mathbf{z}_{re}^{(i-1)}|}{u} \quad \forall r \in \mathcal{R}, e \in \mathcal{E}, i \leq h \quad (21)$$

C. Timing Constraints

Timing constraints ensure that the required time is allotted for robot motion and data transfer. These time requirements determine each robot's latency.

Robot r 's start and end times ($l_r^{(i)}$ and $l_r'^{(i)}$) at step i are defined by motion time, $\delta_r^{(i)}$, and data transfer time, $\gamma_r^{(i)}$.

$$l_r'^{(i+1)} \geq l_r^{(i)} + \gamma_r^{(i)} \quad \forall r \in \mathcal{R} \cup \{b\}, i < h \quad (22)$$

$$l_r^{(i)} \geq l_r'^{(i-1)} + \delta_r^{(i)} \quad \forall r \in \mathcal{R}, 1 \leq i \leq h \quad (23)$$

$$l_r^{(0)} \geq \delta_r^{(0)} \quad \forall r \in \mathcal{R} \quad (24)$$

$$\begin{aligned} l_r^{(i)} &\geq l_r'^{(i)} - M(1 - t_{rr'}^{(i)}), \quad l_r'^{(i)} \geq l_r^{(i)} - M(1 - t_{rr'}^{(i)}), \\ \forall r \in \mathcal{R}, r' \in \mathcal{R} \cup \{b\}, r &\neq r', i \leq h \end{aligned} \quad (25)$$

Constraint (22) ensures that the robots spend the full time collecting or transmitting data, (23) and (24) ensure that robots finish their path before they start transmitting or collecting data, and (25) ensures that both the transmitter and receiver have arrived before any data transfer happens.

Constraints (22)-(25) allow for simultaneous data collection or transfers by only constraining one robot's start time to account for another robot's start time if they are communicating with one another. As a result, increasing timesteps may not

be associated with later continuous time as robots work in parallel and only wait for each other when they communicate.

We construct robot r 's tour by finding the sequence of timesteps where r collects data or communicates—i.e., $r \in (v, v') \wedge t_{vv'}^{(i)} = 1$. We construct r 's path by finding the sequence of edges traversed during a step (i.e., $y_{re}^{(i)} = 1$) and positions from $\mathbf{z}_{re}^{(i)}$ and $\mathbf{z}'_{re}^{(i)}$.

We have presented a MILP for tours and paths that transfers all data to the base station (Sec. IV-A). Each tour leg constrains the positions of associated robots (Sec. IV-B). Using the motion and data transfer times, we compute the latency for each robot and the team as a whole Sec. IV-C.

V. ANALYSIS

We analyze our formulation from Sec. IV and discuss challenges for efficient optimization. Furthermore, we show that cyclic network routes are sometimes optimal, meaning we cannot remove additional global optima by preventing network cycles. We address these challenges with a factored optimization approach that improves efficiency.

A. Optimization Challenges

The formulation in Sec. IV is challenging to solve for two reasons: (1) multiple global optima and (2) a loose linear relaxation. These issues increase the time for branch-and-bound optimization due to increased variable assignments that must be evaluated before pruning a branch.

Three situations cause multiple global optima: (1) robots moving to different regions of space when they are not part of a tour leg; (2) the team following the same paths, but switching which robot follows which path; and (3) a team following the same tour, but switching the discrete timesteps where they perform tour legs. We remove the first cause with,

$$y_{r(\omega, \omega')}^{(i)} \leq \sum_{v \in V} t_{rv}^{(i)} + \sum_{v \in V} t_{vr}^{(i)} \quad \forall r \in \mathcal{R}, \omega, \omega' \in \Omega, \quad (26)$$

which allows r to move to a different region only if part of a leg. Unfortunately, the latter two causes of multiple optima are more difficult to completely prune due to the different, but still potentially uniquely optimal, permutations of t .

Path planning using [8] produces a tight formulation, but the tour constraints (Sec. IV-A) have a comparatively loose linear relaxation. The looseness is primarily due to indicator constraint (7) which accomplishes data transfer in the linear relaxation with low, but non-zero, t . Removing this looseness and the multiple global optima would yield a tight formulation and a single optimum, which would be easier to solve.

B. Cyclic Network Routes

A common way to reduce multiple optima is to remove cyclic network routes—e.g., r_1 transmits to r_2 and then later on r_2 transmits to r_1 —which are always suboptimal for a single robot or pair of robots. However, we show that with three or more robots, cyclic routes could have the minimum latency, meaning we *cannot* disallow tours with a cyclic route.

Optimal cyclic routes depend on robots being able to transmit data when they would otherwise be idle, which may reduce latency when data transfer times are non-negligible.

We give an example of such a data transfer in Fig. 2a to prove that cyclic routes may be optimal.

Theorem 1: The minimum latency tour may include a transmission from robot r to r' and later a transmission from r' to r . That is, $t_{rr'}^{(i)} \neq 0 \Rightarrow \neg t_{r'r}^{(j)} \quad \forall j > i$

Proof: The optimal tour for environment Fig. 2a with three robots contains a cycle. Environment Fig. 2a has two sites, s_t and s_r . Robots r_t , r_m , and r_b follow the top, middle, and bottom paths from the base station respectively. The optimal tour and path to collect data from just s_t is $t_{s_t r_t}^{(0)}, t_{r_t r_m}^{(1)}, t_{r_m r_b}^{(2)}$, with r_t and r_m traversing their respective paths then transmitting the data over walls. Similarly, the optimal tour and path for data collection from s_r is $t_{s_r r_m}^{(0)}, t_{r_m r_t}^{(1)}, t_{r_t r_b}^{(2)}, t_{r_b r_b}^{(3)}$ with robots following each path and then transmitting the data over the walls. By environment design, r_t and r_m can route all the data in s_t to the base station and still arrive at their positions to transmit the s_r 's data before r_b arrives. Furthermore, if r_t collects data and then attempts to follow any other path, r_t will arrive last, increasing the latency. Therefore the optimal tour is to route each site's data in sequence: $t_{s_t r_t}^{(0)}, t_{r_t r_m}^{(1)}, t_{r_m r_b}^{(2)}, t_{s_r r_m}^{(3)}, t_{r_m r_t}^{(4)}, t_{r_t r_b}^{(5)}, t_{r_b r_b}^{(6)}$. This route involves a cycle $(t_{r_t r_m}^{(1)}, t_{r_m r_t}^{(4)})$ proving $t_{rr'}^{(i)} \neq 0 \Rightarrow \neg t_{r'r}^{(j)} \quad \forall j > i$. ■

The potential for optima with cyclic network routes means we cannot include constraints to remove additional optima due to cycles. However, cyclic routes may only be optimal when data transfer times are non-zero. Otherwise, there must always exist another global optimum avoiding a cyclic network route by removing the cycle's first data transfer. Thus, when data transfer times are negligible compared to robot motion time, we may disallow cyclic network routes and retain optimality.

Algorithm 1: Our factored optimization algorithm.

Input: Σ // Def. 1

Output: $\mathcal{T}, \sigma$ // Robot tours and paths

```

1  $\sigma, \mathcal{T} \leftarrow \text{NULL}$ 
2  $l_{\text{team}} \leftarrow \text{MAX\_INT}$  // latency
3  $\text{bnd} \leftarrow 0$  // Lower bound
4  $\text{dist} \leftarrow \text{createDistanceBounds}(\Sigma)$ 
5  $v \leftarrow \text{TourSelector}(\Sigma)$ 
6 while  $v.\text{hasTour}()$  do
7    $\mathcal{T}' \leftarrow v.\text{getNextTour}()$ 
8    $\text{bnd} \leftarrow \text{lowerBound}(\Sigma, \mathcal{T}', \text{dist})$  // Alg. 2
9   if  $\text{bnd} > l_{\text{team}}$  then break // Optimum found
10   $\sigma', l'_{\text{team}} \leftarrow \text{optimize}(\Sigma, \mathcal{T}')$ 
11  if  $l'_{\text{team}} < l_{\text{team}}$  then  $\sigma, \mathcal{T}, l_{\text{team}} \leftarrow \sigma', \mathcal{T}', l'_{\text{team}}$ 
12 return  $\mathcal{T}, \sigma$ 
```

C. Factored Optimization

We improve optimization efficiency by factoring tour generation—i.e., finding each $t_{vv'}^{(i)}$ —from the MILP in Sec. IV. Factoring avoids the expensive branching to evaluate additional optima by instead checking for equivalent tours before path planning. Additionally, the linear relaxation given a tour is tighter than the relaxation without, implying that branch-and-bound methods will be more efficient.

Algorithm 2: Our lower bound latency heuristic

Input: $\Sigma, \mathcal{T}, \text{dist}$ // Def. 1, tour, min site distances
Output: l'_{team} // Lower bound on latency
 1 $l'_r, d_r \leftarrow 0, 0 \forall r \in \mathcal{R}$ // Robot latency and data
 2 $x_r \leftarrow b \forall r \in \mathcal{R}$ // Current robot positions
 3 **foreach** $t_{vv'}^{(i)} \in \mathcal{T}$ **do** // Iterate in order
 4 **if** $v, v' \in \mathcal{R}$ **then** // robot-to-robot communication
 5 $l'_v, l'_{v'} \leftarrow \max(l'_v, l'_{v'}) + \frac{d_v}{q}$
 6 $d_{v'}, d_v \leftarrow d_{v'} + d_v, 0$
 7 **else if** $v' \in \mathcal{R}$ **then** // Robot visits site
 8 $l'_{v'} \leftarrow l'_{v'} + \frac{\text{dist}(x_{v'}, v)}{u} + \frac{d_v}{q}$
 9 $d_{v'}, x_v \leftarrow d_{v'} + d_v^{(0)}, v$
 10 **else** // Robot delivers data to the base station
 11 $l'_v \leftarrow l'_v + \frac{\text{dist}(x_v, v')}{u} + \frac{d_v}{q}$
 12 $d_v, x_v \leftarrow 0, v'$
 13 **return** $\max(l'_r \forall r \in \mathcal{R})$ // Maximum latency

Our factored approach finds the global optimum by determining a lower bound on latency and terminating when no tours exist with a lower bound less than the optimum. We iterate through tours ordered by lower bound until we find a tour with a lower bound exceeding the optimum (Alg. 1).

We find a lower bound on latency by simulating a tour using per-leg lower bounds (Alg. 2). The robot with the greatest latency bound determines the team's bound. Initially, we pre-compute minimum latency paths dist between all pairs of sites and the base station, considering starting and ending *anywhere* in each region. When a robot visits a site (line 7) or the base station (line 10), the robot's latency is increased based on the minimum path and data transmitted. When robots communicate with other robots, (line 4), each robot's latency is set to the pair's maximum plus the data transmission time. We return the greatest latency for any robot, which is a lower bound on the team's latency.

VI. EXPERIMENTS

We evaluate our MILP and factored approach to finding minimum latency routes and trajectories for robot team data collection. We compare against a state-of-the-art TSP solver LKH [25], which assumes robot independence, and we analyze the performance in a variety of environments. Each environment captures a specific feature: Fig. 2a has an optimum with a cyclic network route, Fig. 2b has many walls and long paths, Fig. 2c has non-convex obstacles, and Fig. 2d is based on an actual outdoor environment.

We use LKH [25] configured to solve the Vehicle Routing Problem, which attempts to minimize the sum of every robot's path cost. LKH does not account for regions, so we used the center-to-center path length between every static area (sites and the base station) for the path cost. However, we also compare against using LKH to find a robot tour and optimizing that tour using our MILP (called "LKH+Opt") to show latency reduction from considering regions.

Method	Routes (Fig. 2a)	Maze (Fig. 2b)	Non-Convex (Fig. 2c)	Outdoor (Fig. 2d)
LKH	1490	3782	842	5762
LKH+Opt	1174	3135	760	4013
MILP	882	3137	761	4015
Factored	791	2062	628	2861
Optimal	791	2062	[294, 614]	[2060, 2842]

TABLE I: Latency for the team in seconds (lower is better). Our factored approach quickly reasons about the effects of robot-to-robot communication, resulting the least latency of any method. The true optima are shown for comparison, and the ones with intervals are the bounds on the optimum after running the factored approach for 12 hours.

Method	Routes (Fig. 2a)	Maze (Fig. 2b)	Non-Convex (Fig. 2c)	Outdoor (Fig. 2d)
MILP	77.9%	100%	100%	100%
Factored	0%	18.4%	53.6%	28.1%

TABLE II: The final optimality gap ($\frac{\text{solution} - \text{bound}}{\text{solution}}$) (lower number is better). Our factored method has a tighter bound on optimality than the MILP. Neither baseline calculates a bound, meaning they do not have an optimality gap.

We use A* to generate possible tours. States in the search are locations of data at robots, sites, and the base station. The goal is all data at the base station. We use the admissible heuristic from Alg. 2, which is a lower bound on latency. A* generates tours ordered by the lower bound on latency, so we have found the optimum once the next tour's lower bound exceeds the current best result. Additionally, we calculate the optimality gap ($\frac{\text{solution} - \text{bound}}{\text{solution}}$) to evaluate the tightness of our lower bound—i.e., the quality of Alg. 2 as a heuristic compared to the linear relaxation of Sec. IV.

We performed all experiments on an Intel Xeon processor at 3.70GHz and use a ten-minute timeout. All environments used a three-robot team. The site regions and base station for each environment are shown in Fig. 2. We used Gurobi [26] for optimization. We decomposed space into greedily created rectangles in Fig. 2a and Fig. 2b and used Triangle [27] to perform spacial decomposition in all other environments. We use unit velocities and data transfer times for each environment. Additionally, we seeded our MILP with the answer provided by LKH as an initially valid solution, and we used a time horizon of $(\|\mathcal{R}\| + 1)(\|\mathcal{S}\|)$, which is the maximum number of time steps required for any possible tour. We measure latency improvement as $\frac{\text{ours} - \text{baseline}}{\text{ours}}$.

Table I shows the latency for each method. Our factored method has up to a 101% improvement compared to methods assuming robot independence, showing the importance of robot-to-robot communication. Environments with many walls (Fig. 2a, Fig. 2b, and Fig. 2d) saw a larger improvement in latency from robot-to-robot communication than more open environments (Fig. 2c); transmitting data over walls saves time robots would spend moving around walls, so the difference in improvement is expected. Additionally, the large difference in LKH+Opt and LKH in Fig. 2d is due to LKH+Opt optimizing to reach any point in the goal area, rather than the center.

Table II shows the optimality gap of the factored approach and MILP, which indicates the bound tightness. The factored

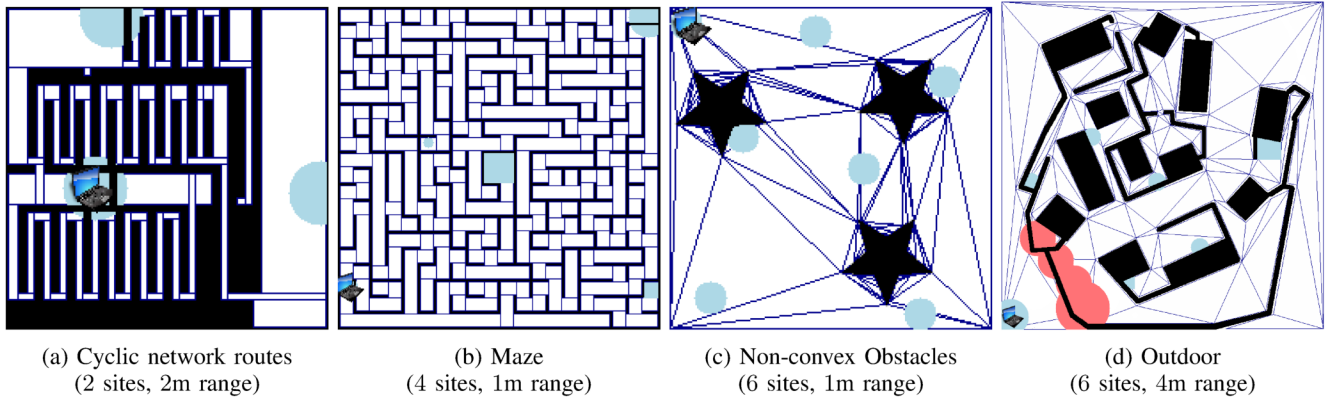


Fig. 2: The different testing environments with the number of sites and robot communication ranges. The base station shares the robot's communication range and is shown as the computer, the site regions are shown as blue convex areas, the spatial decomposition is shown by the purple lines, and areas with environmental interference are shown as red circles.

approach used the heuristic lower bound described in Sec. V-C, while the MILP used the lower bound from branch-and-bound. We improved the optimality gap by factoring tour generation from the MILP, suggesting that separating data collection problems into two sub-problems (tour generation and path optimization) improves efficiency.

VII. CONCLUSION

We developed a Mixed Integer Linear Program (MILP) to minimize latency for robot data collection, analyzed the MILP, and created a more efficient factored form. Our approach enables disconnected robots to rendezvous anywhere in the environment, which reduces latency compared to approaches based on the Traveling Salesman Problem (TSP). Our analysis of the MILP yielded the factored approach, addressing the MILP's two main efficiency challenges: multiple global optima and a loose linear relaxation. We compared our approach against a state-of-the-art TSP solver and showed up to 101% improved latency in our simulated experiments. Our future work will focus on tour generation, which is the current bottleneck for our factored approach.

REFERENCES

- [1] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017.
- [2] M. Rossi and D. Brunelli, "Autonomous gas detection and mapping with unmanned aerial vehicles," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 4, pp. 765–775, 2015.
- [3] C.-F. Cheng and C.-F. Yu, "Data gathering in wireless sensor networks: A combine-tsp-reduce approach," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2309–2324, 2015.
- [4] H. Huang and A. V. Savkin, "Viable path planning for data collection robots in a sensing field with obstacles," *Computer Communications*, vol. 111, pp. 84–96, 2017.
- [5] P. Kumar, T. Amgoth, and C. S. R. Annavarapu, "ACO-based mobile sink path determination for wireless sensor networks under non-uniform data constraints," *Applied Soft Computing*, vol. 69, pp. 528–540, 2018.
- [6] J. Scherer and B. Rinner, "Multi-UAV surveillance with minimum information idleness and latency constraints," *RA-L*, vol. 5, no. 3, pp. 4812–4819, 2020.
- [7] M. A. Schack, J. G. Rogers, Q. Han, and N. T. Dantam, "Optimization-based robot team exploration considering attrition and communication constraints," in *IROS*. IEEE/RSJ, 2021.
- [8] T. Marcucci, J. Umenberger, P. A. Parrilo, and R. Tedrake, "Shortest paths in graphs of convex sets," *arXiv:2101.11565*, 2021.
- [9] B. Yuan, M. Orlowska, and S. Sadiq, "On the optimal robot routing problem in wireless sensor networks," *IEEE transactions on knowledge and data engineering*, vol. 19, no. 9, pp. 1252–1261, 2007.
- [10] Y. Yan and Y. Mostofi, "Robotic router formation in realistic communication environments," *T-RO*, vol. 28, no. 4, pp. 810–827, 2012.
- [11] C. Dixon and E. W. Frew, "Maintaining optimal communication chains in robotic sensor networks using mobility control," *Mobile Networks and Applications*, vol. 14, pp. 281–291, 2009.
- [12] M. A. Schack, J. G. Rogers, Q. Han, and N. T. Dantam, "Optimizing non-Markovian information gain under physics-based communication constraints," *RA-L*, vol. 6, no. 3, pp. 4813–4819, 2021.
- [13] B.-C. Min, Y. Kim, S. Lee, J.-W. Jung, and E. T. Matson, "Finding the optimal location and allocation of relay robots for building a rapid end-to-end wireless communication," *Ad Hoc Networks*, vol. 39, pp. 23–44, 2016.
- [14] R. K. Williams, A. Gasparri, and B. Krishnamachari, "Route swarm: Wireless network optimization through mobility," in *IROS*. IEEE, 2014, pp. 3775–3781.
- [15] J. Stephan, J. Fink, V. Kumar, and A. Ribeiro, "Concurrent control of mobility and communication in multirobot systems," *T-RO*, vol. 33, no. 5, pp. 1248–1254, 2017.
- [16] J. Scherer and B. Rinner, "Multi-robot persistent surveillance with connectivity constraints," *IEEE Access*, vol. 8, 2020.
- [17] Y. Kantaros, M. Guo, and M. M. Zavlanos, "Temporal logic task planning and intermittent connectivity control of mobile robot networks," *T-AC*, vol. 64, no. 10, pp. 4105–4120, 2019.
- [18] X. Yu and M. A. Hsieh, "Synthesis of a time-varying communication network by robot teams with information propagation guarantees," *RA-L*, vol. 5, no. 2, pp. 1413–1420, 2020.
- [19] J. Scherer, A. P. Schoellig, and B. Rinner, "Min-max vertex cycle covers with connectivity constraints for multi-robot patrolling," *RA-L*, vol. 7, no. 4, pp. 10 152–10 159, 2022.
- [20] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Canadian journal of Mathematics*, vol. 8, pp. 399–404, 1956.
- [21] S. Lo, "Delaunay triangulation of non-convex planar domains," *International Journal for Numerical Methods in Engineering*, vol. 28, no. 11, pp. 2695–2707, 1989.
- [22] S. Fortune, "Voronoi diagrams and Delaunay triangulations," in *Handbook of discrete and computational geometry*. Chapman and Hall/CRC, 2017, pp. 705–721.
- [23] J. D. Camm, A. S. Raturi, and S. Tsubakitani, "Cutting big M down to size," *Interfaces*, vol. 20, no. 5, pp. 61–66, 1990.
- [24] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *ACC*, vol. 3. IEEE, 2002, pp. 1936–1941.
- [25] K. Helsgaun, "An extension of the Lin-Kernighan-Helsgaun TSP solver for constrained traveling salesman and vehicle routing problems," *Roskilde: Roskilde University*, vol. 12, 2017.
- [26] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: <https://www.gurobi.com>
- [27] J. R. Shewchuk, "Delaunay refinement algorithms for triangular mesh generation," *Computational geometry*, vol. 22, no. 1-3, pp. 21–74, 2002.