Min-Max Vertex Cycle Covers With Connectivity Constraints for Multi-Robot Patrolling

Jürgen Scherer, Angela P. Schoellig, *Member, IEEE*, and Bernhard Rinner, *Senior Member, IEEE*

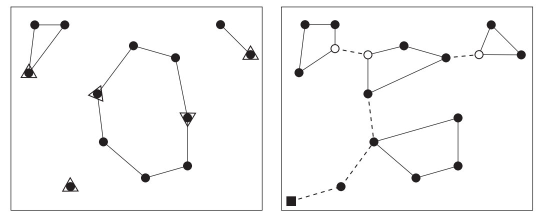
***Abstract*—We consider a multi-robot patrolling scenario with intermittent connectivity constraints, ensuring that robots' data finally arrive at a base station. In particular, each robot traverses a closed tour periodically and meets with the robots on neighboring tours to exchange data. We model the problem as a variant of the min-max vertex cycle cover problem (MMCCP), which is the problem of covering all vertices with a given number of disjoint tours such that the longest tour length is minimal. In this work, we introduce the minimum idleness connectivity-constrained multi-robot patrolling problem, show that it is NP-hard, and model it as a mixed-integer linear program (MILP). The computational complexity of exactly solving this problem restrains practical applications, and therefore we develop approximate algorithms taking a solution for MMCCP as input. Our simulation experiments on 10 vertices and up to 3 robots compare the results of different solution approaches (including solving the MILP formulation) and show that our greedy algorithm can obtain an objective value close to the one of the MILP formulations but requires much less computation time. Experiments on instances with up to 100 vertices and 10 robots indicate that the greedy approximation algorithm tries to keep the length of the longest tour small by extending smaller tours for data exchange.**

***Index Terms*—Multi-robot systems, path planning for multiple mobile robots or agents, surveillance robotic systems.**

I. Introduction

MANY multi-robot patrolling applications (e.g., search and rescue, surveillance) require a set of robots to travel around in the environment and visit points of interest (*sensing locations* [SL]) periodically. The *idleness* of an SL at a certain time instant is defined as the time that passed since the last visit of the SL by a robot. The goal is to minimize the *worst idleness* defined as the maximum idleness over all SLs over an infinite time horizon.

In some surveillance applications, it is not only important that SLs get visited periodically but also that the data captured by the robots arrives at a base station (BS) in due time, e.g., in disaster response scenarios [1]–[3]. If the communication range of the wireless transceivers is smaller than the surveillance area or is limited by obstacles, robots need to coordinate when and where to meet for data exchange.



(a) (b)

Fig. 1. Legend: △ Start position of a robot, ● sensing location (SL), ○ vertex, ■ base station (BS), —— tour edge, − − − connectivity edge. (a) Solution for MIMP with 5 robots. The center tour is patrolled by 2 robots. The upper right tour consists of 2 SLs, and the robot travels back and forth along the edge. The lower left SL is a tour on its own where the robot does not move. (b) Solution for MICP with 5 robots. Additional vertices are added to the tours such that robots can communicate with other robots. The selected vertices, which are not SLs, and connectivity edges ensure that data from each robot reaches the BS. Note that the connectivity edges form a tree of tours (denoted as *tour tree*) including the BS.

In this work we consider two problems: *minimum idleness mixed strategy multi-robot patrolling* (MIMP) and *minimum idleness connectivity-constrained multi-robot patrolling* (MICP). MIMP describes the problem of partitioning a graph and finding tours on the partitions such that the worst idleness is minimized. The term *mixed* describes that the solution also defines how many robots should patrol a tour, which can result in a solution using two different strategies: a single robot on a tour or multiple robots on a tour traveling in the same direction.[1] MICP describes the problem of partitioning a graph and finding tours on partitions such that robots can also communicate and the data arrives at a BS. This might require introducing vertices, which are not SLs, to tours such that robots are within communication range. In contrast to MIMP, only one robot is assigned to a tour. Fig. 1 depicts the MIMP and MICP problems. Although we define and describe both problems formally, this work focuses on complexity, algorithm design, and simulation experiments for MICP only.

The problems considered in this work are variants of the rootless (without a common start/end depot for all robots) min-max vertex cycle cover problem (MMCCP) [4]. The objective is to find closed vertex disjoint paths of minimum maximal length covering all the vertices of a graph with a given upper bound on the number of paths. This problem is NP-hard [5].

[1]Special cases that can arise are robots that are assigned to a single vertex and do not move or robots moving back and forth between vertices

Manuscript received 24 February 2022; accepted 6 July 2022. Date of publication 22 July 2022; date of current version 2 August 2022. This letter was recommended for publication by Associate Editor X. Yu and Editor M. A. Hsieh upon evaluation of the reviewers' comments. *(Corresponding author: Bernhard Rinner.)*

Jürgen Scherer is with the Silicon Austria Labs, 8010 Graz, Austria (e-mail: juergen.scherer@silicon-austria.com).

Angela P. Schoellig is with the Institute for Aerospace Studies and Vector Institute, University of Toronto, Toronto, ON M5S 1A1, Canada (e-mail: angela.schoellig@utias.utoronto.ca).

Bernhard Rinner is with the Institute of Networked and Embedded Systems, University of Klagenfurt, 9020 Klagenfurt am Wörthersee, Austria (e-mail: bernhard.rinner@aau.at).

Digital Object Identifier 10.1109/LRA.2022.3193242

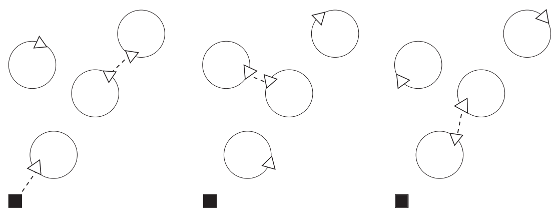


Fig. 2. Legend: △ Robot, ■ BS, – – – connectivity, ○ tour. Example of a multi-robot patrolling scenario with data forwarding towards the BS at three time instances from left to right. The robots move along tours and exchange data with robots on neighboring tours. Positions and directions of a robot on its tour is indicated by the tip of the triangle.

MICP requires scheduling of the robots on the tours such that they meet at particular locations and can exchange information. To ensure that data from each robot eventually arrives at the BS, the data from each robot must travel on a store-and-forward path toward the BS. For this purpose, a *tour tree* has to be selected, which uniquely defines which robots should meet where. In Fig. 1 a the tours are linked with connectivity edges such that a tour tree emerges. In such a tour tree, each robot has a unique robot closer to the BS to which it sends its data. Fig. 2 illustrates the scheduling of the robots on the tours.

Data latency is defined as the time between capturing data at an SL and its arrival at the BS and is another optimization criterion for MICP. The problem of tour finding and scheduling robots on their tours such that the latency is minimized can be decoupled to reduce complexity. MICP addresses the problem of finding tours that ensure that the data arrives at the BS, the minimization of the latency by determining the direction each robot should traverse its tour is described in [6].

The contributions of this work are as follows: *(i)* We formally define the problems MIMP and MICP as mixed-integer linear programs (MILP). To the best of our knowledge, this has not been done for rootless vertex cycle covers (Section III). *(ii)* We show that MICP and the related problem of extending a solution for MMCCP, such that it forms a valid solution for MICP, is NP-hard (Section IV). *(iii)* We formulate an approximation algorithm for MICP based on extending a solution for MMCCP (Section V). In addition, we assess the performance of different algorithms, including solving the MILP formulation, in simulation experiments (Section VI).

II. Related Work

Different strategies for multi-robot patrolling on predefined tours have been investigated in the literature: *(i)* A single path through all SLs is partitioned into segments, and each segment is traversed by a robot moving back and forth along its segment [7]. *(ii)* Robots travel equally spaced along a single closed tour through all SLs in the same direction [8], [9]. *(iii)* The SLs are partitioned into disjoint subsets, one for each robot, and each robot travels along a closed tour through a subset of SLs [8], [9]. Our MIMP approach is able to combine all strategies in a single formulation. Chevaleyre [8] shows that using a single closed tour is not suitable for graphs with long edges and a partitioning of the graph in vertex disjoint subgraphs, one for each robot, is preferable. Our e-MICP heuristic follows this approach and can avoid traveling along long edges if the data gathered can be sent to other robots over a wireless link.

Acevedo *et al.* [10] consider the problem of information propagation in a multi-UAV persistent surveillance scenario. Each UAV follows a closed path in a subarea of a grid-partitioned area, and UAVs exchange information when they are at the borders of their subareas. The graph resulting from the partition of the area is bipartite, which guarantees that the UAVs can be synchronized for data exchange. In [6], we developed a scheduling algorithm for minimum latency data forwarding for robots moving on predefined tours. In this work, we complement our previous work by planning the patrolling tours such that the data can be forwarded to a BS. Other works that consider recurrent connectivity for data propagation have been developed for task visitation [11] and distributed state estimation [12]. However, in these works the subgroups of robots that have to meet are predefined. The MILP for task allocation in [13] also implements data routing policies and transmission schedules for throughput maximization to a BS. In contrast to our work, the problem is solved only for a limited mission horizon and cannot model patrolling tasks with unspecified duration. Single-hop forwarding of data (each UAV sends its data directly to a BS from a site in communication range to the BS) with the aim of latency minimization for patrolling has been considered in [14], [15]. The effectiveness of this approach depends to a large extent on the distance between the BS and the SLs and works best when the communication sites are not too far from the SLs (see [6] for a discussion and experiments on store-and-forward versus single-hop data transport).

MMCCP is NP-hard and several approximation algorithms have been developed. Farbstein and Levin [16] developed a $(4+\epsilon)$-approximation, Xu *et al.* [5] a $(16/3+\epsilon)$-approximation, and Yu and Liu [4] a $(15/3+\epsilon)$-approximation. A rooted version of the problem has been used for multi-robot path planning for bridge inspection [17].

III. Problem Formulation

A set $R := \{1, \dots, r\}$ of robots is available for patrolling an environment by moving between predefined locations. The environment is modeled as a graph with vertex set V defining possible locations of robots and two different sets of edges, $E_M = V \times V$ and $E_C \subseteq E_M$. The complete undirected graph $G_M := (V, E_M)$ has a weight $\omega(i,j)$ associated with each edge $(i,j) \in E_M$ (short ω_{ij}) modeling the time it takes for a robot to travel with unit speed on the shortest path between vertices $i \in V$ and $j \in V$. The edges E_C model the connectivity of the vertices, and we define $G_C := (V, E_C)$. An edge $(i,j) \in E_C$ indicates that robots located at vertices i and j at the same time can communicate with each other. We assume that multiple robots can be at the same vertex at the same time and that robots can traverse edges without collisions. We assume that robots located at the same vertex can communicate with each other, i.e., $(i,i) \in E_C, \forall i \in V$. The set of SLs $V_S \subseteq V$ have to be visited by the robots periodically, and $V_C := V \setminus V_S$ can be used to establish connectivity. We define a vertex induced subgraph $G_M\langle V_S\rangle$ as the graph containing only vertices V_S and the respective edges from the set E_M (and analog for G_C).

Definition 1: (Tour, tour length) An m-tuple, i.e., $|t| = m$, of mutually distinct vertices $t = (v_1, \dots, v_m)$ is called a tour, its length is defined as $\omega(t) := \sum_{i=1,\dots,m-1} \omega(v_i, v_{i+1}) + \omega(v_m, v_1)$.

Definition 2: (Worst idleness) Given a set of r' tours of vertices from V, and the number of robots r_k assigned to each tour $t_k = (v_1^k, \ldots v_{|t_k|}^k)$, for $k = 1, \ldots, r'$, the worst idleness is defined as $I(\{t_k\}, \{r_k\}) := \max_{k=1\ldots r'} \{\omega(t_k)/r_k\}$.

Definition 3: (MIMP) MIMP is the problem of finding a disjoint set $\{t_k\}$ of r' tours on V_S (i.e., each $v \in V_S$ is part of exactly one tour) and the number of assigned robots r_k for each tour t_k (such that $r \geq \sum_{k=1,\ldots,r'} r_k$) with minimum worst idleness $I(\{t_k\}, \{r_k\})$.

Definition 4: (Tour graph, tour tree, meeting point) Two tours t_i and t_j are connected if the tours contain vertices $v \in t_i$ and $w \in t_j$, and $(v, w) \in E_C$. The edge (v, w) is called meeting point. If the tours are interpreted as vertices of a tour graph G_T^C, there is an edge between two vertices of G_T^C if the respective tours are connected. If G_T^C is a tree, it is called a tour tree.

Definition 5: (MICP) MICP is the problem of finding a set $\{t_k\}$ of $r' \leq r$ tours on V, with minimum $I(\{t_k\}, \{r_k\})$ and the additional constraints that each $v \in V_S$ occurs in at least one tour, all $r_k = 1$, and the tours form a tour tree.

The tours are traversed by the robots continuously. For MICP these tours can also contain vertices from V_C to establish connectivity between tours. The robot traversing the longest tour can continuously move while robots traversing shorter tours might have to wait at meeting points to communicate with robots on neighboring tours. However, the worst idleness is not larger than the length of the longest tour (see [6]), which we therefore consider as the minimization criterion in the remainder of this work.

In the following subsections, we provide MILP formulations for MIMP and MICP, which can be used as input for integer programming optimization software.

A. Minimum Idleness Mixed Strategy Patrolling (MIMP)

This section introduces the MILP formulation of MIMP, which is extended to the MICP formulation in the following section. The formulation makes use of two concepts named *vertex potentials* and *artificial vertices*. Potentials are numbers associated with each vertex on a tour and have been used in formulations of traveling salesperson problems (TSP) and vehicle routing problems (VRP) [18]. The potentials increase along a tour, and the difference between the potentials of two consecutive vertices is at least the respective edge weight in E_M. We introduce artificial vertices that allow to close the tour and decrease the potential at a single vertex on a tour for MICP or for multiple vertices (one for each robot) for MIMP. Together these two concepts allow us to model and minimize the length of a closed tour by minimizing the largest potential (see an illustration in Fig. 3).

For MIMP, we assume that data transfer is either not required or is always possible (e.g., the communication range is large enough). Therefore, no additional vertices need to be included to enable data transfer among the tours, i.e., $V_S = V$.

For every vertex $i \in V$ we introduce an artificial vertex $\bar{i}$ and denote the set of artificial vertices with V'. In the following, an artificial vertex from V' corresponds to the vertex of V with the same symbol, i.e., $\bar{i} \in V'$ corresponds to $i \in V$.

The MILP model contains binary decision variables x_{ij} for arcs of a directed extended graph with vertices $V \cup V'$ and the following arc set $\hat{E}$ with the arc weights $\hat{\omega}_{ij}$:

- $(i, j) \in \hat{E}$ and $(j, i) \in \hat{E}, \forall (i, j) \in E_M$, with $\hat{\omega}_{ij} = \hat{\omega}_{ji} := \omega_{ij}$

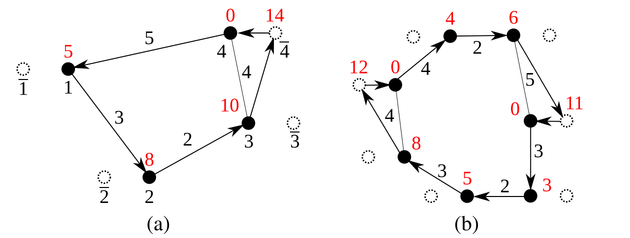


Fig. 3. (a) An example of node potentials of a solution for a closed tour for one robot (MIMP, MICP) of length 14. The vertices and the corresponding artificial vertices are numbered from 1 to 4 and $\bar{1}$ to $\bar{4}$, respectively. The arrows correspond to the selected edges (x_{ij} or x_{ij}^k). The numbers next to the edges depict the weight ω_{ij}, and the red numbers above the nodes depict the node potential u_i of the solution. (b) A solution of a closed tour for two robots (MIMP) with an idleness of 12.

- $(i, \bar{j}) \in \hat{E}, \forall i \in V, \bar{j} \in V'$, with $\hat{\omega}_{i\bar{j}} := \omega_{ij}$ (note that $(i, \bar{i}) \in \hat{E}$ and $\hat{\omega}_{i\bar{i}} := 0$)
- $(\bar{i}, i) \in \hat{E}, \forall i \in V$, with $\hat{\omega}_{\bar{i}i} := 0$

The variable $x_{ij} = 1$ if and only if the edge $(i, j) \in \hat{E}$ is part of a tour. An artificial vertex with an outgoing edge defines the starting point of a robot. The model below allows at most r starting points and at least one starting point on every tour.

A continuous decision variable u_i is defined for every $v \in V \cup V'$ and describes a vertex potential along a tour similar to the Miller-Tucker-Zemlin subtour elimination constraints for traditional TSP or VRP problems [18]. In our case, the purpose is not to eliminate subtours but to describe the worst idleness of a vertex on a tour patrolled by one or multiple robots.

Artificial vertices allow decreasing the node potential along a tour. In the traditional VRP formulations, this is only possible for the depot, which ensures that each tour starts and ends at the depot. In our formulation, at most r artificial vertices can be selected, and an artificial vertex corresponds to the starting position of a robot on a tour. An artificial vertex $\bar{i}$ has only one outgoing edge to i, and a tour entering $\bar{i}$ must continue to i. This formulation allows multiple robots to traverse a tour if the worst idleness can be decreased in this way. The MIMP problem is formally defined as follows:

$$\min\ I \tag{1a}$$

$$s.t. \sum_{j \in (V \setminus \{i\}) \cup \{\bar{i}\}} x_{ji} = 1 \quad \forall i \in V \tag{1b}$$

$$\sum_{j \in (V \setminus \{i\}) \cup V'} x_{ij} = 1 \quad \forall i \in V \tag{1c}$$

$$\sum_{j \in V} x_{j\bar{i}} = x_{\bar{i}i} \quad \forall \bar{i} \in V' \tag{1d}$$

$$\sum_{i \in V} x_{\bar{i}i} \leq r \tag{1e}$$

$$u_i - u_j + M x_{ij} \leq M - \hat{\omega}_{ij} \quad \forall i \in V, j \in (V \cup V') \tag{1f}$$

$$u_i \leq I \quad \forall i \in V \cup V' \tag{1g}$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in \hat{E} \tag{1h}$$

$$u_i \geq 0 \quad \forall i \in V \cup V'. \tag{1i}$$

The objective is to minimize the worst idleness I (line 1a) which is bounded by the potential of each vertex u_i (constraint (1g)). The constraints (1b) and (1c) are degree constraints and force that every vertex from V is part of exactly one tour. The degree constraint (1d) ensures that if there is an outgoing edge from $\bar{i}$, there is also an incoming edge (note that there is only one outgoing edge of $\bar{i}$ in $\hat{E}$, namely $(\bar{i}, i)$). Constraint (1e) ensures that there are at most r vertices $\bar{i} \in V'$ with an outgoing edge. Constraint (1f) forces that the potential u_j of a vertex j is at least $u_i + \hat{\omega}_{ij}$ if the edge (i, j) is part of the tour. M is a number that is large enough such that the constraint has no effect when $x_{ij} = 0$, e.g., $M := \sum_{i,j \in V} \omega_{ij}$. This constraint does not cover edges $(\bar{i}, i)$, i.e., whenever a tour passes through an edge $(\bar{i}, i)$, the potential of i is not forced to be the same as the potential of $\bar{i}$.

A solution to the MIMP problem can be interpreted as follows. A robot is placed at each vertex $i \in V$ with an incoming edge $x_{\bar{i}i}$, and all robots start to traverse the tour at the same time towards the direction of the edge x_{ij}. In the case that $j = \bar{i}$, the robot does not move and stays on the vertex i. The worst idleness is then I.

B. Minimum Idleness Connectivity-Constrained Patrolling (MICP)

If the vertex induced subgraph $G_C\langle V_S \rangle$ is connected, there are no vertices in addition to the SLs required, and the problem can be solved for $V := V_S$ only. Then for every pair of tours, there exist two vertices $i \in V_S$ and $j \in V_S$, which are not on the same tour, such that $(i, j) \in E_C$.

The formulation for MICP uses three-index variables x_{ij}^k, which describe whether a particular robot $k \in R$ traverses an edge (i, j), and two-index vertex potentials u_i^k. The difference to the MIMP formulation is that every tour is traversed by exactly one robot. The optimization problem is defined as:

$$\min I \tag{2a}$$

$$s.t. \sum_{k \in R} \sum_{j \in (V \setminus \{i\}) \cup \{\bar{i}\}} x_{ji}^k \geq 1 \quad \forall i \in V_S \tag{2b}$$

$$\sum_{j \in (V \setminus \{i\}) \cup \{\bar{i}\}} x_{ji}^k \leq 1 \quad \forall i \in V, k \in R \tag{2c}$$

$$\sum_{j \in (V \setminus \{i\}) \cup \{\bar{i}\}} x_{ji}^k = \sum_{j \in (V \setminus \{i\}) \cup V'} x_{ij}^k$$
$$\forall i \in V, k \in R \tag{2d}$$

$$\sum_{j \in V} x_{j\bar{i}}^k = x_{\bar{i}i}^k \quad \forall \bar{i} \in V', k \in R \tag{2e}$$

$$\sum_{i \in V} x_{\bar{i}i}^k \leq 1 \quad \forall k \in R \tag{2f}$$

$$u_i^k - u_j^k + M x_{ij}^k \leq M - \hat{\omega}_{ij}$$
$$\forall i \in V, j \in (V \cup V'), k \in R \tag{2g}$$

$$u_i^k \leq I \quad \forall i \in V \cup V', k \in R \tag{2h}$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in \hat{E}, k \in R \tag{2i}$$

$$u_i^k \geq 0 \quad \forall i \in V \cup V', k \in R. \tag{2j}$$

Constraint (2b) ensures that each SL is visited by a robot. Constraints (2c)-(2e) are the degree constraints for each tour. Constraint (2f) ensures that at most one artificial vertex is used for a tour.

To describe connectivity, we extent the formulation above by the binary variables $\tilde{x}_{ij}, \forall i, j \in V$ and the constraints:

$$(i, j) \notin E_C \Rightarrow \tilde{x}_{ij} = 0 \quad \forall i, j \in V \tag{3a}$$

$$\sum_{j \in V} \tilde{x}_{ij} \leq \sum_{k \in R} \sum_{j \in V \cup V'} x_{ji}^k \quad \forall i \in V \tag{3b}$$

$$\sum_{k \in R} x_{ij}^k \leq r\,(1 - \tilde{x}_{ij}) \quad \forall i, j \in V. \tag{3c}$$

Constraint (3b) ensures that $\tilde{x}_{ij}$ is zero if i has no incoming edge from a tour. This means that no data can be sent along this edge if i is not visited by a robot. Constraint (3c) states that if there is an edge $\tilde{x}_{ij}$, the corresponding edge (i, j) cannot be part of a tour. The combination of variables $\tilde{x}_{ij}$ and x_{ij}^k allows to model the data flow from each SL to the BS where the information is either carried by a robot k along edge (i, j) ($x_{ij}^k = 1$) or transmitted from one robot to another ($\tilde{x}_{ij} = 1$).

To model the data flow, a tree has to be selected from the combined edge set from $\tilde{x}_{ij}$ and x_{ij}^k. We use the following multi-commodity flow formulation [19] to ensure that the tours are connected:

$$f_{ij}^v \leq \sum_{k \in R} x_{ij}^k + \tilde{x}_{ij} \quad \forall i, j \in V \cup V', v \in V_S \tag{4a}$$

$$\sum_{i \in V \cup V'} f_{ib}^v - \sum_{i \in V \cup V'} f_{bi}^v = 1 \quad \forall v \in V_S \tag{4b}$$

$$\sum_{i \in V \cup V'} f_{iv}^v - \sum_{i \in V \cup V'} f_{vi}^v = -1 \quad \forall v \in V_S \tag{4c}$$

$$\sum_{i \in V \cup V'} f_{iw}^v - \sum_{i \in V \cup V'} f_{wi}^v = 0$$
$$\forall v \in V_S, w \in (V \cup V') \setminus \{v\}. \tag{4d}$$

The continuous flow variable f_{ij}^v models a flow originating from each $v \in V_S$ on the edges for all $i, j \in V \cup V'$. Constraint (4a) ensures that there is no flow if neither any edge x_{ij}^k (for all $k \in R$) nor $\tilde{x}_{ij}$ between i and j is selected. Constraint (4b) restricts the difference between inflow and outflow for each flow of type v at the BS b to one, i.e., the BS is the sink of all flows of type $v \in V_S$. Constraint (4c) restricts the difference between outflow and inflow of a flow of type v to one at $v \in V_S$, i.e., each $v \in V_S$ is the source of one unit of flow of type v. Constraint (4d) states that the amount of inflow equals the amount of outflow at a vertex w for each flow of type v not originating at w. The MICP formulation comprises the objective (2a) and constraints (2b)-(4d).

IV. Complexity Analysis

If (V_S, E_C) is a complete graph, MICP is equivalent to the NP-hard min-max cycle cover problem (MMCCP) [5] and therefore MICP is NP-hard as well. A solution of an MMCCP is defined by a set of tours, one for each robot, such that each SL appears on exactly one tour (this is equivalent to MIMP with all $r_k = 1$). The optimal solution has the smallest possible longest tour which is the cost of the solution. We show that even the problem of

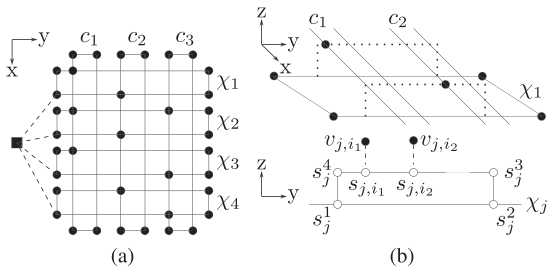


Fig. 4. Legend: • Sensing location (SL), ○ vertex, ■ BS, — tour, ······ potential tour extension, — connectivity edge. Reduction from 3-SAT instance $c_1 \wedge c_2 \wedge c_3 = (\chi_1 \vee \chi_2 \vee \chi_3) \wedge (\overline{\chi}_1 \vee \overline{\chi}_2 \vee \chi_4) \wedge (\chi_2 \vee \overline{\chi}_3 \vee \overline{\chi}_4)$ to e-MICP drawn in 3D space. (a) Tours for clauses c_i and variables χ_j with a meeting point between the variable tours and the BS and potential meeting points between the clause and variable tours. (b) Top: potential extension for each variable on each side ("north" and "south") towards the z-direction to connect clause tours causing an additional length of 1 of the variable tour. Bottom: Details for a tour extension (north or south) of variable χ_j.

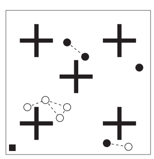


Fig. 5. Legend: • Sensing location (SL), ○ vertex, ■ base station (BS), – – – connectivity edge, + obstacle. Randomly generated scenario with 5 obstacles.

constructing a feasible MICP solution with minimum cost from a solution of MMCCP is NP-hard. We define this problem as:

Definition 6: (e-MICP) Given a solution $\{t_k\}$ for MIMP with all $r_k = 1$, e-MICP is the problem of inserting vertices from V_C into the tours (without changing the relative order of the original vertices), resulting in a set $\{t'_k\}$, such that the tour graph is a tree and such that $I(\{t'_k\}, \{r_k\})$ is minimum.

Theorem 1: e-MICP is NP-hard.

Proof: We describe a reduction from a 3-SAT instance with clauses c_i and variables χ_j to e-MICP (see Fig. 4 for an illustrative example of a reduction). G_M and G_C are constructed such that there is a tour of length L for each clause c_i and each variable χ_j, and there is no edge from E_C between the tours, except between each variable tour χ_j and the BS. There are potential meeting points between a clause tour and the tours of the variables contained in the clause. The meeting points are either on the "north" or "south" part of the tour depending on whether the variable occurs not negated (χ_j) or negated ($\overline{\chi}_j$). For each variable tour, it is possible to extend the tour (i.e., inserting vertices from V_C) in the north or south part resulting in connecting all clause tours with potential meeting points in the north or south part to the variable tour. By construction, each extension increases the tour length by 1, and all other extensions (e.g., the clause tours towards the variable tours) cause the respective tour length to increase by more than 1. Fig. 4(b, bottom) depicts the graph construction for one extension (south or north) of a variable χ_j that appears (negated or not negated) in clauses $c_{i_1}, c_{i_2}, \ldots$. The lengths of the edges (s_j^2, s_j^3) and (s_j^4, s_j^1) in E_M are 0.5. The length of (s_j^3, s_j^4) is larger than 1 and is equal to (s_j^1, s_j^2). The lengths between any s_{j,i_k} and s_j^1 or s_j^2 are larger than the length of (s_j^1, s_j^2). The lengths between v_{j,i_k} and s_{j,i_k} or any other vertex on the variable tour are larger than 0.5 (for clarity not all edges are shown in Fig. 4(b)). A vertex v_{j,i_k} with an edge to s_{j,i_k} from E_C lies on the clause tour c_{i_k}. The question is whether there is a solution with a worst idleness of $L+1$, which is only possible if each variable tour is extended either on the north or south part.

On the one hand, a solution of the 3-SAT instance decides where to extend the variable tours, namely on the north/south part if the variable is $true/false$, resulting in a worst idleness of $L+1$. On the other hand, a feasible solution for MICP with a worst idleness of $L+1$ determines also a solution for 3-SAT by assigning $true/false$ to a variable if a variable tour is extended on the north/south part, respectively. □

V. Approximation Algorithms

The following approximation algorithms take as input a solution for the MMCCP obtained from the original problem containing only the SLs as input. Since the original MMCCP does not include any connectivity constraints between the tours, our proposed algorithms modify the solution to ensure these constraints.

A. MILP Formulation for e-MICP

The MILP formulation for e-MICP is an extension of the MICP formulation from Section III-B. The solution for MMCCP consists of a set of tours $T = \{t_1, \ldots, t_{r'}\}$, with $r' \leq r$, and $t_k = (v_{k_1}, v_{k_2} \ldots, v_{k_{|t_k|}})$. The following constraints are derived from this solution and ensure increasing node potentials u_{k_i} along a tour for each robot k (constraints (5a)-(5c)). These constraints fix the order of the SL vertices to the order given by tour t_k and therefore allow to reuse all the constraints of the MICP formulation:

$$u^k_{k_1} = 0 \quad \forall k \in \{1, \ldots, r'\} \tag{5a}$$

$$u^k_{k_i} - u^k_{k_{i+1}} \leq -\hat{\omega}_{k_i k_{i+1}} \quad \forall k \in \{1, \ldots, r'\}, \forall i \in \{1, \ldots, |t_k| - 1\} \tag{5b}$$

$$u^k_{k_{|t_k|}} - u^k_{\overline{k_1}} \leq -\hat{\omega}_{k_{|t_k|}\overline{k_1}} \quad \forall k \in \{1, \ldots, r'\} \tag{5c}$$

$$x_{\overline{k_1}k_1} = 1 \quad \forall k \in \{1, \ldots, r'\} \tag{5d}$$

$$x^k_{k_i\overline{k_{i+1}}} = 0 \quad \forall k \in \{1, \ldots, r'\}, \forall i \in \{1, \ldots, |t_k| - 1\}. \tag{5e}$$

The constraints force the visit of the artificial vertex $\overline{k_1}$ corresponding to the vertex k_1 on each tour by including the edge $x_{\overline{k_1}k_1}$ (any vertex could be chosen for each tour) and excluding all other edges into artificial vertices (constraints (5d) and (5e)). The e-MICP formulation comprises the objective (2a) and constraints (2b)-(5e) and is a constrained version of MICP of finding the meeting points between tours such that the length of the longest tour is minimized.

B. Greedy Heuristic for e-MICP

Our algorithm EMICP (see Algorithm 1) greedily selects a pair of tours in each iteration which are connected by extending each tour to meeting points (either a single vertex on both tours or a pair of vertices which have a connectivity edge, one on each tour) until the tour tree contains all tours. First, in line 1 a min-max cycle cover problem is solved on the vertex induced subgraph $G_M\langle V_S\rangle$, which results in a set of tours T to which a virtual BS tour containing only the BS (b) is added. In the loop

Algorithm 1: EMICP.

Input: SLs V_S, movement graph G_M, connectivity graph G_C, BS vertex b, number of robots r
Output: tour tree G_T
1: $T \leftarrow MMCCP(G_M\langle V_S\rangle, r) \cup \{(b)\}$
2: $G_T \leftarrow$ empty tour tree, $r' \leftarrow |T|$
3: **while** $|E(G_T)| < r'$ **do**
4: $G_T^C \leftarrow MINCONNECTIVITYTOURGRAPH\,(G_M, G_C, T)$
5: $t_1, t_2 \leftarrow SELECTEDGE\,(G_T^C, G_T)$
6: $t_i' \leftarrow$ insert m_i in t_i **if** $m_i \notin t_i, i \in \{1,2\}$
7: $T \leftarrow (T \cup \{t_i'\}) \setminus \{t_i\}, i \in \{1,2\}$
8: add (t_1, t_2) to $E(G_T)$

starting in line 3, $r'-1$ (the number of actual tours excluding the BS tour) edges are added to the edge set $E(G_T)$ of an initially empty tour tree G_T. To this end, a tour graph G_T^C containing an edge for each pair of current tours from T is computed (line 4) where the weight of the edge is the minimal possible increase in the maximum tour length when connecting two tours at potential meeting points. From this tour graph an edge, i.e., a pair of tours (t_1, t_2), is selected such that G_T remains a tree (line 5). The meeting vertices are denoted by (m_1, m_2) and are inserted in each tour (at the position determined in line 7 of Algorithm 2, i.e., between v_i and w_i for the corresponding element in D_i) if they are not part of the respective tour already (line 8).

Algorithm EMICP makes use of the routines listed in Algorithms 2 and 3, respectively. `MinConnectivityTourGraph` iterates over all pairs of consecutive vertices v_1, w_1 and v_2, w_2 on all pairs of tours t_1, t_2, respectively, and computes the increase in tour length if a pair of potential meeting vertices m_1 and m_2 (either there is an connectivity edge between them or $m_1 = m_2$) is inserted between the consecutive vertices on each tour, respectively (line 7 and 8). The minimum increase of the length required to connect each pair of tours is recorded by selecting the minimum from the set D_i for each tour t_i (line 9). The respective weights $\xi(t_1, t_2), \xi(t_2, t_1)$ in the tour graph G_T^C are updated accordingly (line 10).

`SelectEdge` deletes the edges of the tour graph G_T^C computed by `MinConnectivityTourGraph` that would result in a cycle in the tour tree G_T or are already in the tour tree by setting the edge weight ξ' to ∞. Then, the minimum weight edges from the remaining edges are collected in the set M (line 4). For each edge (t_1, t_2) in M the meeting points m_1 and m_2 computed in Algorithm 2, line 7 are inserted in t_1 and t_2, respectively, and the new tours are added to M' (in loop of line 6). Finally, the pair of tours from M' that have the smallest maximum length are selected as new edge $(\tilde{t_1}, \tilde{t_2})$ for the tour tree G_T (line 9). In the preceding description of the algorithm, the virtual BS tour requires a separate treatment because this tour cannot be extended since it is not traversed by a robot (note that the BS vertex b can be on a tour of a robot as two tours can be connected by a single vertex, i.e., $m_1 = m_2 = b$). We omit this treatment in the pseudo code for brevity and clarity reasons.

Theorem 2: The runtime of Algorithm 1 is in $O(r \cdot (|V|^5 + r^3))$.

Proof: Line 8 in Algorithm 2 is executed $|V|^2$ times for each $(m_1, m_2) \in E(G_C)$ (which has $O(|V|^2)$ elements), in total $|V|^4$ times. Computing the lengths of the tours t_i' requires $O(|V|)$ steps, which results in $O(|V|^5)$ steps. Checking for cycles for each edge in the tour graph in Algorithm 3, line 3 requires $O(r^3)$ steps (using e.g., breadth-first search). ■

Algorithm 2: MinConnectivityTourGraph.

Input: movement graph G_M, connectivity graph G_C, BS vertex b, tours T
Output: tour graph G_T^C with weights $\xi(e), e \in E(G_T^C)$
1: $G_T^C \leftarrow$ tour graph with $V(G_T^C) = T, E(G_T^C) = T \times T$
2: **for** $t_1, t_2 \in T, t_1 \neq t_2$ **do**
3: $D_i \leftarrow \emptyset, i \in \{1,2\}$
4: **for** $(m_1, m_2) \in E(G_C) \cup \{(v,v) | v \in V(G_C)\}$ **do**
5: **for** each consecutive vertices v_1, w_1 on tour t_1 **do**
6: **for** each consecutive vertices v_2, w_2 on tour t_2 **do**
7: $t_i' \leftarrow t_i$ with edge (v_i, w_i) replaced by egdes $(v_i, m_i), (m_i, w_i), i \in \{1,2\}$
8: $D_i \leftarrow D_i \cup \{len(t_i') - len(t_i)\}, i \in \{1,2\}$
9: $\xi(t_1, t_2) \leftarrow \min D_1, \xi(t_2, t_1) \leftarrow \min D_2$
10: update $\xi(t_1, t_2)$ and $\xi(t_2, t_1)$ in $E(G_T^C)$

Algorithm 3: SelectEdge.

Input: tour graph G_T^C with weights $\xi(e), e \in E(G_T^C)$, current tour tree G_T
Output: tours to connect $\tilde{t_1}, \tilde{t_2}$
1: $\xi'(e) \leftarrow \xi(e) \quad \forall e \in E(G_T^C)$
2: **for** $e \in E(G_T^C)$ **do**
3: **if** $e \cup E(G_T)$ has cycle or $e \in E(G_T)$ **then** $\xi'(e) \leftarrow \infty$
4: $M \leftarrow \{e | e \in E(G_T^C)$ with $\xi'(e) = \min_{e' \in E(G_T^C)}\{\xi'(e')\}\}$
5: $M' \leftarrow \emptyset$
6: **for** $(t_1, t_2) \in M$ **do**
7: $t_i' \leftarrow$insert m_i in $t_i, i \in \{1,2\}$ ▷see Algorithm 2, line 7
8: $M' \leftarrow M' \cup \{(t_1', t_2')\}$
9: $(\tilde{t_1}, \tilde{t_2}) \leftarrow \arg\min_{(t_1,t_2) \in M'}\{\max\{len(t_1), len(t_2)\}\}$

VI. Results

This section compares the different algorithms for e-MICP in terms of worst idleness and computation times. We generate G_M and G_C from randomly sampled vertex coordinates in the two-dimensional plane. The first algorithm MICP-MILP solves the MILP formulation of MICP (Section III-B) with the integer program solver SCIP (we use the parallelized version *fscip*, which is able to use all available CPU cores)[2]. Algorithm EMICP-MILP uses an initial MMCCP solution and the additional e-MICP constraints (Section V-A). Algorithm MICP-MILP-TL (*time limited*) returns the best-found solution given the same computation time EMICP-MILP requires to solve the same instance. We also include these results to get an indication of whether solving the problem from scratch can give better results than solving the problem e-MICP with suboptimal initial tours for MMCCP. We also include the algorithm MMCCP,

[2][Online]. Available: https://www.scipopt.org/

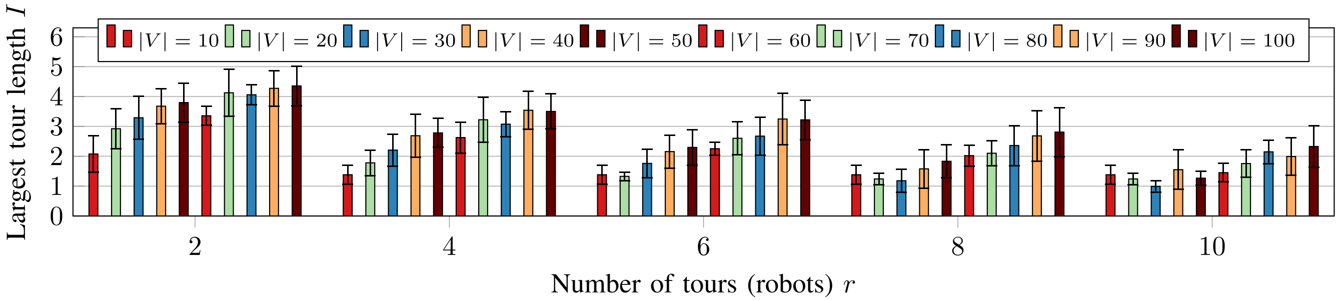


Fig. 6. Mean and standard deviation (narrow vertical lines) over 10 instances of the objective values of EMICP for a larger scenario.

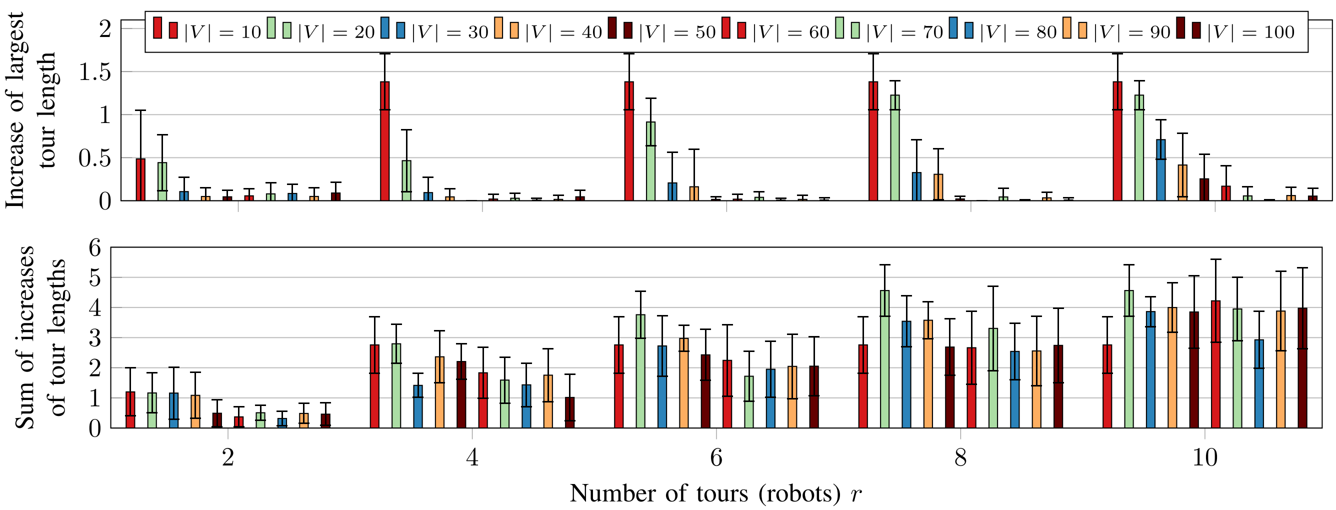


Fig. 7. Comparison of MMCCP and EMICP (mean and standard deviation over 10 instances) for a larger scenario. The upper plot shows the increase of objective values and the lower plot the total increase of tour lengths from MMCCP to EMICP.

which we used to get the initial tours for EMICP (Section V-B) and EMICP-MILP, and which is an implementation of [4]. We implemented all algorithms in Python (we used the package *pyscipopt*[3] as interface to SCIP/fscip) and used a computer with an Intel i7-9700 K (8 cores, 3.6 GHz) CPU and 32 GB of RAM.

For the first scenario, the coordinates for 10 vertices are sampled from the square $[0, 1] \times [0, 1]$, and five of them are randomly selected as SLs. The BS coordinates are fixed at $(0, 0)$. There is a connectivity edge between two vertices with a probability of 0.8 if the Euclidean distance between them is smaller than 0.3. The traveling distance between two vertices is the Euclidean distance. Sampling is repeated 10 times and each instance is solved for $r \in \{1, 2, 3\}$ robots resulting in 30 instances in total. Despite the small scenario, MICP-MILP was not able to find the optimal solution within a time limit of 3 hours (10800 s) per instance for almost all instances with $r \in \{2, 3\}$. EMICP-MILP was able to find the optimal solution within the same time limit for almost all instances. The mean objective values (length of the longest tour) and the computation times with standard deviations are shown in Table I.[4] It shows that the objective values for EMICP-MILP and EMICP are comparable and that solving the problem from scratch results in worse solutions compared to solving the problem from initial tours with the same computation time (see EMICP-MILP and MICP-MILP-TL).

TABLE I
COMPARISON OF DIFFERENT ALGORITHMS ON A SMALL SCENARIO WITH MEAN AND STANDARD DEVIATION OVER 10 INSTANCES

	$r = 1$		$r = 2$		$r = 3$	
	mean	std	mean	std	mean	std
Largest tour length I						
MMCCP	2.28	0.64	1.23	0.78	0.65	0.30
MICP-MILP	2.36	0.44	1.40	0.32	1.16	0.36
EMICP-MILP	2.53	0.55	1.63	0.67	1.22	0.26
EMICP	2.53	0.55	1.67	0.67	1.25	0.25
MICP-MILP-TL	3.09	0.79	2.08	0.36	1.93	0.44
Computation time (s) $(\cdot 10^3)$						
MICP-MILP	2.35	1.27	9.63	2.62	9.84	2.93
EMICP-MILP	0.02	0.01	0.40	0.81	1.45	3.16
MICP-MILP-TL	0.01	0.01	0.39	0.81	1.44	3.16

The next scenario contains $10, 20, \ldots, 100$ vertices with randomly sampled coordinates (except the BS with coordinates $(0, 0)$) from the square $[0, 1] \times [0, 1]$ from which 40% are SLs. The scenario contains five cross-shaped obstacles of size 0.3×0.3 (see Fig. 5). There is a connectivity edge between two vertices if they are within the range $0.2, 0.18, \ldots, 0.02$ (depending on the number of vertices) and the straight line between them is not blocked by an obstacle. The distance between two vertices

[3]https://github.com/scipopt/PySCIPOpt

[4]We do not show the computation times for MMCCP and EMICP since the values are considerably below 1 s for each instance.

is the shortest path passing around obstacles. We sample again 10 sets of coordinates for each number of vertices and solve the problem for $r \in \{2, 4, 6, 8, 10\}$. Fig. 6 shows the means and standard deviations of the objective values for each number of vertices and robots. As expected, the objective values tend to decrease with increasing number of robots. Fig. 7 compares the values for the MMCCP solution used as input to EMICP with the values for EMICP. The upper plot shows the differences in the objective values $I_{EMICP} - I_{MMCCP}$. The lower plot shows the sum of the differences in length over the tours which result from inserting vertices to connect the tours. Comparing the two plots, we can see that, especially for the instances with higher number of vertices, EMICP is able to keep the original length of the longest tour by extending shorter tours. EMICP requires at most 3 seconds of computation time for an instance, despite the large exponents in Theorem 2.

VII. Conclusion

We model the problem of multi-robot patrolling with connectivity constraints as a variant of the min-max vertex cycle cover problem (MMCCP), where each robot traverses a closed tour repeatedly, and we introduce algorithms for finding efficient solutions. Our approach provides a framework for continuous multi-robot patrolling applications, which require guaranteed data forwarding. The coordination effort between robots is limited to pairwise regular encounters at meeting points introduced by our algorithms. We show that the problem is NP-hard. Solving the mixed-integer formulation to minimize the longest tour requires large computation power even for the smallest instances and suffers from exploding computation times with increasing problem size. Therefore, we propose a greedy approximation algorithm (EMICP) that starts from an approximate solution for the MMCCP and connects pairs of tours by extending the tours to meeting points until all tours form a tree. Robots can exchange gathered data at meeting points along the tree such that all data can finally arrive at a base station. Our simulation experiments indicate that the algorithm is able to keep the longest tour small by extending smaller tours. EMICP requires a neglectable amount of computation time compared to the MILP-based algorithms (MICP-MILP, EMICP-MILP) but can achieve the same objective values as the MILP-based approximation EMICP-MILP on small problem instances.

Potential future work will focus on investigating alternative heuristics, considering more advanced communication and mobility models, and deriving theoretical guarantees for the approximate solution.

References

[1] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 24–32, Jan./Mar. 2017.

[2] B. Rinner, C. Bettstetter, H. Hellwagner, and S. Weiss, "Multidrone systems: More than the sum of the parts," *IEEE Comput.*, vol. 54, no. 5, pp. 34–43, May 2021.

[3] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative robots to observe moving targets: Review," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 187–198, Jan. 2018.

[4] W. Yu and Z. Liu, "Improved approximation algorithms for some min-max and minimum cycle cover problems," *Theor. Comput. Sci.*, vol. 654, pp. 45–58, Nov. 2016.

[5] W. Xu, W. Liang, and X. Lin, "Approximation algorithms for min-max cycle cover problems," *IEEE Trans. Comput.*, vol. 64, no. 3, pp. 600–613, Mar. 2015.

[6] J. Scherer and B. Rinner, "Multi-robot patrolling with sensing idleness and data delay objectives," *J. Intell. Robot. Syst.*, vol. 99, pp. 949–967, Mar. 2020.

[7] F. Pasqualetti, A. Franchi, and F. Bullo, "On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms," *IEEE Trans. Robot.*, vol. 28, no. 3, pp. 592–606, Jun. 2012.

[8] Y. Chevaleyre, "Theoretical analysis of the multi-agent patrolling problem," in *Proc. IEEE/WIC/ACM Int. Conf. Intell. Agent Technol.*, 2004, pp. 302–308.

[9] A. Almeida et al., "Recent advances on multi-agent patrolling," in *Proc. Adv. Artif. Intell. - Proc. Braz. Symp. Artif. Intell.*, 2004, pp. 474–483.

[10] J. J. Acevedo, B. C. Arrue, J. M. Diaz-Banez, I. Ventura, I. Maza, and A. Ollero, "Decentralized strategy to ensure information propagation in area monitoring missions with a team of UAVs under limited communications," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2013, pp. 565–574.

[11] Y. Kantaros, M. Guo, and M. M. Zavlanos, "Temporal logic task planning and intermittent connectivity control of mobile robot networks," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4105–4120, Oct. 2019.

[12] R. Khodayi-mehr, Y. Kantaros, and M. M. Zavlanos, "Distributed state estimation using intermittently connected robot networks," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 709–724, Jun. 2019.

[13] E. F. Flushing, L. M. Gambardella, and G. A. Di Caro, "Simultaneous task allocation, data routing, and transmission scheduling in mobile multi-robot teams," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 1861–1868.

[14] S. G. Manyam, S. Rasmussen, D. W. Casbeer, K. Kalyanam, and S. Manickam, "Multi-UAV routing for persistent intelligence surveillance & reconnaissance missions," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2017, pp. 573–580.

[15] J. Banfi, N. Basilico, and F. Amigoni, "Minimizing communication latency in multirobot situation-aware patrolling," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2015, pp. 616–622.

[16] B. Farbstein and A. Levin, "Min-max cover of a graph with a small number of parts," *Discrete Optim.*, vol. 16, pp. 51–61, May 2015.

[17] M. Lin and R. J. La, "Miniature robot path planning for bridge inspection: Min-max cycle cover-based approach," in *Proc. IEEE Int. Conf. Automat. Sci. Eng.*, 2020, pp. 365–371.

[18] G. Laporte, "What you should know about the vehicle routing problem," *Nav. Res. Logistics*, vol. 54, no. 8, pp. 811–819, Dec. 2007.

[19] T. L. Magnanti and L. A. Wolsey, "Optimal trees," in *Handbooks in Operations Research and Management Science*, vol. 7. 1995, pp. 503–615.