IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Multi-UAV Surveillance with Minimum Information Idleness and Latency Constraints

Jürgen Scherer and Bernhard Rinner[1]

Abstract— **We discuss surveillance with multiple unmanned aerial vehicles (UAV) that minimize information idleness (the lag between the start of the mission and the moment when the data captured at a sensing location arrives at the base station) and constrain latency (the lag between capturing data at a sensing location and its arrival at the base station). This is important in surveillance scenarios where sensing locations should not only be visited as soon as possible, but the captured data needs to reach the base station in due time, especially if the surveillance area is larger than the communication range. In our approach, multiple UAVs cooperatively transport the data in a store-and-forward fashion along minimum latency paths (MLPs) to guarantee data delivery within a predefined latency bound. Additionally, MLPs specify a lower bound for any latency minimization problem where multiple mobile agents transport data in a store-and-forward fashion. We introduce three variations of a heuristic employing MLPs and compare their performance with an uncooperative approach in a simulation study. The results show that cooperative data transport reduces the information idleness at the base station compared to the uncooperative approach where data is transported individually by the UAVs.**

I. Introduction

Advances in the field of aerial robotics have increased the interest in the use of unmanned aerial vehicles (UAVs) for various civilian applications including disaster response missions [9], [28], [17].

In this work we consider a path planning problem for multiple UAVs (or other types of mobile robots) that are visiting points of interest (denoted as *sensing locations*, SL), which is typically required in disaster response scenarios for acquiring information from areas of interest. Since the area is potentially large, existing wireless communication technology used on aerial vehicles is not able to establish a fully connected network all the time due to range limitations while the UAVs move to the SLs. Direct communication to the *base station* (BS) can also be prohibited while flying close to the ground or within a building. In these scenarios it is not only important that SLs get visited as soon as possible, but also that the data captured by the UAV arrives at the BS in due time. This allows human mission operators to quickly assess a situation or is a precondition for a prompt analysis of the captured data.

We present a novel cooperative data delivery approach where UAVs visit SLs and transport the captured data in a store-and-forward fashion to the BS. This is different from our previous work on persistent multi-UAV/robot surveillance where permanent [25], [27] or intermittent [26] connectivity to the BS was considered. The paths for the UAVs are planned such that two UAVs meet at points that are within their communication range. The data is sent from one UAV to the other, which travels along a path to meet another UAV. By following so called *minimum-latency paths* (MLP) the data travels towards the BS with a guaranteed latency (see Figure 1 for an illustration).

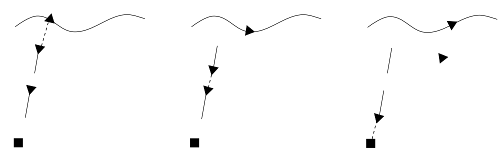

Fig. 1. Three points in time (left to right) with UAVs (triangles) and the BS (square). The top UAV is visiting SLs along a path (waved line) and the other UAVs collaboratively transport the data to the BS on MLPs (straight solid line). Dashed lines indicate that UAVs exchange data.

The rationale behind cooperative data transportation is that UAVs do not need to travel to the BS individually for data delivery but can spend more time on visiting SLs while the previously captured data travels to the BS in a coordinated way over multiple UAVs. This eliminates large detours if the communication range is large compared to the travel speed and SLs are far away from the BS, or no-fly-zones block movement but not communication.

The contributions of this paper are as follows: (i) We define the problem of surveillance with cooperative data transport considering data latency, (ii) model MLPs as shortest path problem with time windows (SPPTW) [8], (iii) present heuristics using MLPs for constraining the latency of data from SLs, and (iv) provide simulation results that show that our heuristics can outperform an approach where UAVs transport the data individually to the BS regarding different metrics.

The paper is organized as follows: Section II reviews the literature. Section III introduces the *minimum information idleness with latency constraints* (MILC) problem. Section IV introduces MLPs and describe three heuristics for MILC. Section V describes the simulation results and Section VI concludes the article.

II. Related Work

While minimizing idleness in surveillance, exploration and patrolling applications is a common optimization goal [21], [10], [23], [2], minimizing latency has received much less attention in literature. Banfi et al. [3] present a MILP (mixed integer linear program) formulation and heuristics for the

[1]Both authors are with the Institute of Networked and Embedded Systems, University of Klagenfurt, Austria. jscherer@edu.aau.at, bernhard.rinner@aau.at

problem of finding a patrolling path for each UAV with the goal to minimize the latency. Each UAV follows a path containing SLs and intermediate detours to communication sites where the data can be transmitted to the BS. A MILP formulation and a heuristic for a similar problem with task revisit constraints are presented in [19]. Acevedo et al. [1] investigates in patrolling considering the propagation of information among the UAVs. A decentralized algorithm maintains a grid shaped partition of the area where each UAV is traveling along a circular path within its subarea. UAVs exchange data on the border of their subareas with each UAV of the neighboring subareas, which minimizes the propagation time of information in this grid shaped partition. Exploration with recurrent connectivity constraints has been considered in [6] where robots are forced to build connected trees from frontiers to the BS recurrently.

A similar problem of exploration considering information update at the BS has been considered in [29]. While the possibility of cooperative data transport is discussed, in the implementation robots move individually to the BS to transmit the data, and the frequency of returns is determined by a number that controls the importance of exploration versus information update at the BS.

In contrast to related work, we consider a cooperative data transport by multiple UAVs to a single BS by investigating which UAVs should meet when and where. Other work focuses on recurrent connectivity without explicitly minimizing data latency [22], [11], [16], [18]. All these works have in common that which robots should meet is determined in advance. Recurrent connectivity of the full robot network, e.g. with the aim for planning and coordination of the next tasks, has been considered in [15], [4].

III. Problem description

In this section we formally define the MILC problem. The set of UAVs is denoted as $R = \{1, \ldots, r\}$, $|R| = r$. The problem is modeled with help of a weighted undirected movement graph $G_M = (V, E_M, W^M)$, with a set of vertices V describing locations in the environment, where the UAVs can move from one vertex v to another w within time W^M_{vw} if there is an edge $(v, w) \in E_M$. Vertex $v_0 \in V$ identifies the BS. The set of vertices $V_S \subseteq V$ are SLs which have to be visited by at least one UAV. The set $V_C \subseteq V_S$ contains SLs in communication range of the BS. The undirected communication graph $G_C = (V, E_C, W^C)$ models the communication connectivity between vertices. If there is an edge $(v, w) \in E_C$ then two UAVs or the BS and a UAV can communicate if one is at v and the other is at w at the same time. The edge weights W^C describe the duration of the data transmission for every edge. The latency for a certain SL is defined as the time between the collection of the data at a SL and the arrival of the data at the BS. To collect data at $v \in V_S$ a UAV must be at v but data is not necessarily collected each time a UAV is at v. Data collection at a SL is scheduled such that the latency L^c can be ensured, which requires that data is captured by a UAV on a SL only if it is guaranteed that the data can be

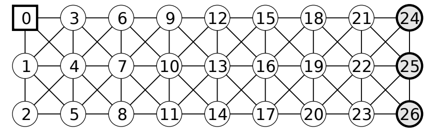

Fig. 2. A small scenario with $V = \{0, \ldots, 26\}$, $V_S = \{24, 25, 26\}$. Here $E_M = E_C$ and the edges are depicted by the lines between the vertices, $W^M_{vw} = 1$, $W^C_{vw} = 0$ for all edges. The BS is the quadratic vertex 0, and the number of UAVs r is 3.

t:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	4	6	9	12	16	18	21	**24**	21	18	$\underline{16}$	14	$\underline{11}$	7
2	0	3	6	3	0	0	0	4	4	0	0	4	8	$\underline{7}$	$\underline{4}$
3	0	4	8	11	14	17	20	23	**25**	**26**	23	$\underline{19}$	15	13	14

Fig. 3. The rows show the sequence of the positions (vertices) of three UAVs for the optimal solution from the sample scenario in Figure 2. The bold numbers indicate visits at the SLs and the underlined numbers when a UAV transmits its data to another UAV or the BS. At $t = 11$, UAV 3 transmits its data to UAV 1, at $t = 13$ UAV 1 transmits the data to UAV 2, and finally at $t = 14$ UAV 2 transmits the data to the BS. For comparison the algorithms presented in Section IV (with $L^c = 6$) achieve the following results for (WII, FII, WL): H1: $(29, 13, 5)$, H2: $(22, 14, 6)$, H3: $(22, 14, 6)$, SH ($L^c = \infty$): $(15, 15, 7)$. In this small example SH is close to the optimal solution.

transmitted on a subsequently scheduled MLP (which will be guaranteed by the algorithms described in Section IV-B).

We discuss the objective function for the MILC problem using a sample scenario with three SLs and three UAVs (Figure 2). If the objective is to visit all SLs as fast as possible, each UAV travels to a different SL on the shortest path starting from the BS, resulting in a solution with objective value 8 (at time 0 the UAVs are at the BS, and the edges length is 1). If individual data delivery to the BS is also considered, the objective value is 15 (a UAV transmits the data from a vertex in communication range to the BS).

On the other hand, if the objective is to minimize the time (from the beginning of the mission) the data from each SL have arrived at the BS (the UAVs can transmit data to each other), the objective value of the optimal solution is 14. Figure 3 depicts the optimal solution where UAVs 1 and 3 visit the SLs and UAV 2 is only transporting data back to the BS. This solution has been determined by solving a MILP for the problem (adaptation of [3] to allow data transmissions between UAVs). When there are many SLs, minimizing the time until the data from all SLs have arrived at the BS might not be the desired objective, because it can happen that the BS does not receive any data for a long period of time. This motivates us to consider path planning with intermediate data transports to the BS by constraining the latency of the data.

We consider three different metrics for the problem: (i) the lag between the start of the mission and the moment when the data *from all* SLs have arrived at the BS (*worst information idleness*, WII), (ii) the first time after the start of the mission until data *from any* SL have arrived at the BS (*first information idleness*, FII), and (iii) the maximum latency (lag between capturing data at a SL and the arrival of the data at the BS) *over all* SLs (*worst latency*, WL). We call the problem of minimizing WII and FII and constraining WL *minimum information idleness with latency* *constraints* (MILC). The example solution in Figure 3 has $WII = FII = 14$ (the unit is time, it takes 1 unit to move along an edge from E_M and 0 units to transmit the data along an edge from E_C), since data does not arrive before time 14 at the BS. The worst latency is determined by the SLs 24 and 25, i.e. $WL = \max\{14-8, 14-8, 14-9\} = \max\{6, 5\}$.

Determining the optimal tour for visiting all SLs as soon as possible on a graph is related to the NP-complete traveling salesperson problem (TSP) [13]. Since MILC with one UAV, a sufficiently large latency bound (e.g. sum of all edge weights), and zero communication range (the UAV can send data to the BS only when it is in close proximity) is equivalent to TSP, MILC is NP-hard too. This requires designing heuristic algorithms that can produce a (suboptimal) solution in an acceptable time. In Section IV we describe heuristic algorithms for solving this problem efficiently.

Graphs are commonly used for modeling motion and connectivity [3], [4], [20]. Obstacles are easily incorporated by omitting the appropriate edges in the graphs. The problem of determining SLs for image acquisition has been considered in [24]. We assume a simple sensor model where data can be captured in neglectable time when the UAV is present at the SL. The communication graph is considered static and captures the communication delay for fixed sized data. Various work on more complex communication and network models for multi-UAV systems are intensively investigated (e.g. [14], [5]) but beyond the scope of this paper. Nevertheless, when UAVs execute the planned paths they can cope with unforeseen deviations from the schedule by simply waiting for the predecessor or successor on the MLP. Additionally, we assume that UAVs are able to avoid collisions such that they can be at the same vertex at the same time.

IV. Algorithm Description

A. *Minimum-latency path*

The problem of transporting the data as fast as possible from a source $s \in V$ to a destination $d \in V$ with a given number of UAVs can be modeled as a shortest path problem with time windows (SPPTW) in a graph $G = (V, A)$ [8]. We refer to this path as MLP, and the problem of finding a MLP is a special case of SPPTW. SPPTW is the problem of finding a shortest path in a graph with a traversal cost W_{vw} and a traversal duration T_{vw} associated with each edge $(v, w) \in A$. The sum of edge traversal times along the path is constrained to be in a time window $[L_v, U_v]$ at $v \in V$.

The problem is NP-hard in general, but there exist dynamic programming algorithms for the problem instance with integer T_{ij} [8]. The dynamic programming approach converts the problem to a shortest path problem in a graph with at most $\sum_{v \in V}(U_v - L_v + 1)$ vertices. In our case, W_{ij} is the time it takes to traverse the edge $(v, w) \in E_M$, and T_{vw} is the number of UAVs necessary to traverse the edge (1 for every edge $(v, w) \in E_C$), the lower limit is $L_v = 0$ and the upper limit is $U_v = r - 1$ for all vertices.

Figure 4a depicts an example graph. If there are two types of edges (from E_M and E_C) between two vertices v and w,

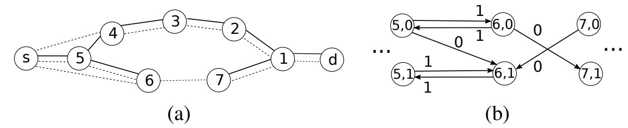

Fig. 4. (a) Graph with edges from E_M (solid lines, $W^M_{vw} = 1$ for all edges) and edges from E_C (dashed lines, $W^C_{vw} = 0$ for all edges). (b) Converted graph from (a) for the dynamic programming approach of SPPTW with $r = 2$. Only vertices and edges corresponding to vertices $5, 6, 7$ are shown.

there are two possible ways to transport the data from v to w. Either one UAV moves from v to w, which takes W^M_{vw} time, or one UAV is placed at v and the other at w and the data is transmitted in time W^C_{vw}. In the latter case the edge "consumes" one UAV. In our example, with $W^M_{vw} = 1$ and $W^C_{vw} = 0$ for all $(v, w) \in A$, there are different optimal paths in terms of the latency depending on the number of available UAVs. If $r = 1$, the MLP from s to d is $(s, 5, 4, 3, 2, 1|)$ with a latency of 5. This notation means that the UAV moves from s to 1 along the solid edges and stops at vertex 1 to transmit the data to the destination d. If $r = 2$, the MLP is $(s, 5, 6|7, 1|)$ or $(s|4, 3, 2, 1|)$ with a latency of 3. In the first case the first UAV moves from s over 5 to 6 and stops there to transmit the data to the second UAV waiting at 7. This UAV finally transports the data to 1. In the second case the first UAV does not move and transfers the data to the second one waiting at 4. Finally, if $r = 3$, the optimal path is $(s|6|7, 1|)$ with a latency of 1.

This description of a MLP, assumes that a UAV j is at the appropriate vertex (which is the start vertex of its subpath on the MLP) when the predecessor i on the MLP arrives at the end vertex of its subpath where i transmits the data to j. In the first example with $r = 2$ above, the first UAV starts at s (which is a SL), and the second UAV is already at 7, when the first UAV arrives at 6. In the path planning algorithm (Section IV-B) waiting times will have to be introduced to guarantee the latency of the gathered data.

Figure 4b shows parts of the converted graph for $r = 2$. The vertex $(5, 0)$ on a valid path from s to d represents the fact that only one UAV has been used so far. From this vertex the path can continue over $(6, 0)$, which means that the edge $(5, 6) \in E_M$ is used. If the path continues to vertex $(6, 1)$, then the edge $(5, 6) \in E_C$ is used, which means that the first UAV stops at 5 and transmits the data to 6. From $(6, 1)$ there is no path to $(7, 0)$ or $(7, 1)$ because $(6, 7) \notin E_M$ and two UAVs have already been used. The edge length in the converted problem is 1 for a movement on an edge from E_M or 0 for transmitting the data over an edge from E_C. The problem of finding the shortest latency path reduces to finding the shortest path in the converted graph.

Our implementation for computing a MLP with r UAVs, $|V|$ vertices, and the edges E_C and E_M given as adjacency matrices has time complexity $O(|V|^2 \cdot r)$ for generating the adjacency matrix for the new graph (with $|V| \cdot r$ vertices), and $O(|V|^2 \cdot r^2)$ for calculating the shortest path (Dijkstra's algorithm [7]) in this new graph.

Algorithm 1 Heuristic for MILC (MILC-H1)

Input:
graphs G_M, G_C, SLs V_S, number of UAVs r, latency bound L^c
Output:
subtours $t_1, \ldots, t_k$, MLP and schedule $(st(v), sv_v, ev_v, st_v) \quad \forall v \in V_S$

1: $\gamma_v \leftarrow \infty \; \forall v \in V_S$
2: **for** $i = r$ **downto** 1 **do** // All MLPs from all $v \in V_S$ to the BS with i UAVs:
3: $\quad (d_1, \ldots, d_{|V_S|}) \leftarrow min\_latency(V_S, v_0, i, G_M, G_C)$
4: $\quad$ **if** $d_v \leq L^c$ **then** $\gamma_v \leftarrow i, \forall v \in V_S$
5: **if** $\exists v \in V_S : \gamma_v > r$ **then** exit "Problem is infeasible!"
6: $\gamma \leftarrow \max_{v \in V_S \setminus V_C} \{\gamma_v\}$
7: $k \leftarrow \lfloor r/\gamma \rfloor$
8: $T \leftarrow solve\_tsp(V_S, G_M)$
9: $(t_1, \ldots, t_k) \leftarrow split\_tour(T, k)$
10: **for** $i = 1$ **to** k **do**
11: $\quad R_i \leftarrow \{(i-1) \cdot \gamma + 1, \ldots, i \cdot \gamma\}$
12: $\quad v' \leftarrow v_0$
13: $\quad$ **for** each v on path t_i **do**
14: $\quad\quad (sv_v, ev_v, st_v, et_v) \leftarrow mlp(v, v_0, \gamma_v, G_M, G_C)$
15: $\quad\quad$ **for** $l = 1$ **to** $\gamma_v, \forall m \in R_i$ **do**
16: $\quad\quad\quad A_{lm} \leftarrow st(v') + et_{v'}(m) + dist_{G_M}(ev_{v'}(m), sv_v(l))$
17: $\quad\quad M \leftarrow minmax\_matching(A)$
18: $\quad\quad$ **for** $m \in R_i$ **do**
19: $\quad\quad\quad sv_v(m) \leftarrow sv_v(M(m)); \; ev_v(m) \leftarrow ev_v(M(m))$
20: $\quad\quad\quad st_v(m) \leftarrow st_v(M(m)); \; et_v(m) \leftarrow et_v(M(m))$
21: $\quad\quad st(v) \leftarrow st(v') + \min_{m \in R_i} \{et_{v'}(m) + dist_{G_M}(ev_{v'}(m), sv_v(m))\}$
22: $\quad\quad v' \leftarrow v$

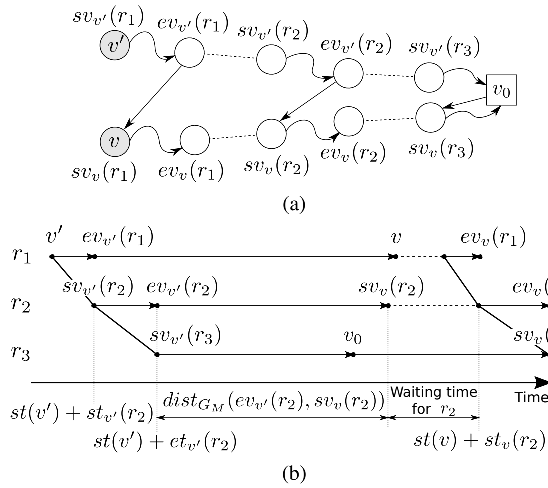

Fig. 5. (a) The MLPs for two consecutive SLs v' and v on a tour and three UAVs r_1, r_2 and r_3. The waved lines depict a path in G_M and the dashed lines edges in G_C. The straight arrows show the transition from the end vertex $ev_{v'}$ on the MLP for v' to the start vertex sv_v for the MLP of v for each UAV. (b) Timing diagram for the scenario above. Horizontal solid lines denote the movement, horizontal dashed lines denote that a UAV is waiting at a vertex, dots denote the start and end vertex of a UAV on the MLP, and oblique lines depict the MLPs (transmission times W^C are 0 in the example). The values on the time axis are labeled only for UAV r_2.

B. Heuristics for MILC

The basic idea of the heuristics is that for each $v \in V_S$ UAVs move to the initial positions along the MLP from v to the BS v_0 and wait, if necessary, for its preceding UAV on the MLP that transmits the data captured at v. It then moves to its final position on the MLP to transmit the data to its successor on the MLP. The order at which SLs are visited is determined by the TSP tour (a high-level view of the resulting behavior is shown in Figure 1). The outputs of the algorithms are mappings from UAVs to its start vertex (sv_v), end vertex (ev_v), and start time (st_v) on a MLP for every vertex $v \in V_S$. The start time determines the time that it has to wait for its preceding UAV on the MLP to transmit the data of v.

The first heuristic MILC-H1 (Algorithm 1, "H1" for short) determines the minimum number of UAVs necessary to achieve L^c for each SL v and stores the value in γ_v (Line 4). Function $min\_latency$ returns the minimum latencies that can be achieved with a given number of UAVs i for all paths from vertices $v \in V_S$ to the BS v_0. The variable γ_v is the minimal number of UAVs for a MLP starting at v such that the latency can be achieved along the MLP. If there is a vertex v with $\gamma_v > r$, the problem is infeasible because it is not possible to transport the data with the available number of UAVs within time L^c to the BS (Line 5). Given the number of UAVs necessary, a TSP tour is split into multiple subtours (Line 6 to Line 9). For splitting the tour k-SPLITOUR from [12] can be used, which tries to minimize the length of the largest subtour.

The rational for splitting the tour is the idea that multiple subtours can be traversed by groups of UAVs in parallel to reduce the time for visiting all SLs. The subtours are then assigned to different groups of UAVs (loop in Line 10). For every vertex $v \in V_S$ on a subtour the MLP is calculated with $mlp()$, which returns the start and end vertices (sv_v and ev_v) and the start and end times (st_v and et_v) for every UAV along the MLP (Line 14).

Which UAV should actually move to which start vertex on the MLP for $v \in V_S$ is determined by a matching calculated based on its end vertex $ev_{v'}$ and end time $et_{v'}$ on the MLP for the predecessor $v' \in V_S$ of v on the subtour. The value $st(v)$ determines the time the first UAV on the MLP for v can start to move from v to its end position ev_v and is measured from the beginning of the mission. The start time st_v and end time et_v are relative to the start of the first UAV ($st_v = 0$ for the first UAV). The element A_{lm} of the weight matrix A, calculated in the loop starting in Line 15, is the earliest time UAV m can arrive at the potential new starting vertex $sv_v(l)$ after moving from $sv_{v'}(m)$ over $ev_{v'}(m)$ to $sv_v(l)$. The matching between UAVs and start vertices minimizes the latest time a UAV can be at its start vertex and is calculated with $minmax\_matching()$ (Line 17). Finally, the mappings are updated according to the matching (Line 18), and the start value $st(v)$ is calculated based on the latest time all UAVs can be at their start vertex (Line 21). Function $dist_{G_M}(s, d)$ returns the length of the shortest path from s to d in G_M.

Figure 5a depicts an example situation with two consecutive SLs v' and v on a TSP tour (i.e. v gets visited after v') and their MLPs for 3 UAVs r_1, r_2 and r_3. For v' UAV r_1 starts at $v' = sv_{v'}(r_1)$ and moves to $ev_{v'}(r_1)$ where it transmits the data to r_2, which starts at $sv_{v'}(r_2)$ and moves to $ev_{v'}(r_2)$. There, r_2 transmits the data to r_3. The timing diagram is shown in Figure 5b. The start time $st(v)$ (and therefore the time r_1 can start at v) is determined by r_3, because r_1 has to wait such that it does not arrive at $ev_v(r_1)$ before r_2 has arrived at $sv_v(r_2)$. UAV r_2 in turn has to wait for UAV r_3. Note that r_1 and r_2 can start before r_3 reaches $sv_v(r_3)$, and r_2 will arrive at $ev_v(r_2)$ and transmit the data to r_3 exactly when the latter one arrives at $sv_v(r_3)$. Since the first UAV has to wait at $v \in V_S$, the latency bounds are met because v can be considered as visited right before the first UAV leaves v, and the corresponding data will arrive within the bound at the BS.

The second heuristic H2 is similar to the first one. For every SL the number of UAVs γ_v is calculated such that the latency cannot be decreased on a MLP with additional UAVs (which is different from the loop in Line 2 in Algorithm 1). This is equivalent to minimizing the length (latency) of the MLP and therefore the latency for each SL. If there is a number of UAVs available that is at least a multiple k of $\max_{v \in V_S} \gamma_v$, then the tour is split into k subtours. The algorithm then tries for each subtour to visit as many SLs as possible with one UAV along the TSP tour before transporting the data with help of the others to the BS such that L^c is not violated. This is different from H1 where UAVs transport the data immediately to the BS after visiting a SL (cf. Line 13 in Algorithm 1). Another difference is that in each subtour the same UAV is visiting all SLs (i.e. it is not part of the matching, cf. Line 17 in Algorithm 1).

The third heuristic H3 is a combination of the other two. For every SL the minimum number of UAVs is calculated such that L^c is not violated (similar to H1). If possible, the tour is also split into subtours. The algorithm visits in every subtour as many SLs as possible such that L^c is not violated (similar to H2).

For all heuristics the TSP tour is shortcut if a SL has been visited on the MLP of another SL already. This is valid because the latency constraint is met for these SLs.

Since the communication graph G_C with weights W^C are inputs, we assume a constant transmission time on each edge independent of the amount of transmitted data. This assumption is valid for H1 if the amount of data captured at each SL is constant. We can relax the fixed data assumption for the other heuristics by recalculating the weights W^C after the visit of a SL with the consequence that determining whether a feasible solution can be generated, cannot be done before iterating through the vertices of a calculated path, as it is done in Algorithm 1 (cf. Line 13).

C. Complexity analysis

We briefly analyze the computational complexity of our heuristics. Algorithm 1 starts with the MLP computation from all SLs to the BS for each number of available robots (Line 2). This can be done with the Dijkstra single source shortest path algorithm [7]. The converted graph described in Section IV-A is asymmetric, and a MLP from a particular SL to the BS can be computed by determining a path in the converted graph with the Dijkstra algorithm from a source to a destination vertex representing the SL and the BS, respectively. To compute the MLPs from all SLs to the BS, the single source Dijkstra algorithms has to be used on the reversed graph (all edges in the converted graph are reversed) with the BS as source. With the complexity of the Dijkstra algorithm of $O(|V|^2 \cdot r^2)$ on the converted graph (described in Section IV-A) the loop has the time complexity of $O(|V|^2 \cdot r^3)$. We denote the complexity of solving the TSP on the SLs and splitting the tour with $O(TSP(|V_S|))$ and $O(Split(|V_S|))$, respectively.

The statements within the nested loops in Line 10 and Line 13 are executed $|V_S|$ times, since the subtours are disjunct and the outer loop iterates over the subtours. Assuming that the MLPs have been computed and stored already in the loop in Line 2, the statement in Line 14 is a simple lookup for the MLP of vertex v. Similarly, the shortest paths in G_M (computed by function $dist_{G_M}$) can be determined beforehand with the Floyd-Warshall algorithm in $O(|V|^3)$ [7], and the inner statement of the following loop (Line 15) is executed $O(r^2 \cdot |V_S|)$ times. The complexity of $minmax\_matching$ (solving the linear bottleneck assignment problem, LBAP) is denoted with $O(LBAP(r))$. The statements in the loop Line 18 are executed $O(|V_S| \cdot r)$ times. This gives an overall complexity of Algorithm 1 of $O(|V|^3 + |V|^2 \cdot r^3 + TSP(|V_S|) + Split(|V_S|) + |V_S| \cdot LBAP(r))$, which is the same for H2 and H3.

D. Suboptimality analysis

For a preliminary analysis we consider a simplified algorithm that tries to minimize WII and ignores WL (as considered in the example in Section III where the objective was to minimize WII). We assume that the set of UAVs is partitioned into k sensing UAVs and $r-k$ data transportation UAVs (a predefined value of k may arise from the fact that not all UAVs are equipped with sensors). Furthermore, we assume that the triangular inequality holds in G_M, that G_M is connected, and that $W^C_{ij} = 0$ for all $(i, j) \in E_C$. For the heuristics described above, the value of k is determined by the latency constraint. Here we assume that k is given and leave the determination of the optimal value of k and a suboptimality analysis considering also the latency as future work. The simplified algorithm can be sketched as follows:

1) $G'(V_S \cup \{v_0\}, W') \leftarrow$ compute complete graph with vertices $V_S \cup \{v_0\}$ where the edge weights W' are the lengths of the shortest paths in G_M
2) $T \leftarrow$ solve the TSP in G'
3) $(t_1, \ldots, t_k) \leftarrow$ split T into k subtours with k-SPLITOUR [12], $t_i = (v_0, v^i_1 \ldots, v^i_{l(i)}, v_0), 1 \le i \le k$
4) Each sensing UAV follows its subtour (from v_0) to $v^i_{l(i)}$ where the data is finally transported on a MLP to v_0 with help of $(r-k)/k$ UAVs (for each subtour there is a fixed number of data transportation UAVs).

The cost of a solution produced by this algorithm is denoted with $\hat{C}_k$. Here we assume that the data transportation UAVs are at their starting positions of the MLPs in time such that the sensing UAV i does not have to wait for the data transportation UAVs at the last vertex of its tour, $v^i_{l(i)}$. A lower bound for the optimal solution that minimizes WII (without distinguishing between sensing and transportation UAVs) is determined by a set of r paths starting from v_0 and visiting all SLs such that the longest path is as short as possible. The reason is that WII cannot be smaller than the time it takes to visit all SLs as soon as possible with r UAVs. We denote the cost of such a solution with C_r^{O*} ("O" indicates that the paths are open tours, i.e. do not terminate in v_0). The cost (which is WII) of the optimal solution minimizing WII is denoted with C^*, and the cost of the optimal solution with k sensing UAVs is denoted with C_k^*, therefore $C_r^{O*} \leq C^*$. Note that $C^* = C_r^*$.

First, we establish that

$$\hat{C}_k \leq 2C_k^{O*}\left(e + 1 - 1/k\right), \tag{1}$$

where e is the approximation factor for the TSP heuristic, i.e. $L \leq eC_1^*$, and L is the length of the TSP tour in step 2. The proof is similar to the one in [12] but has to be modified slightly because the triangular inequality does not always hold in the presence of MLPs, and the algorithm replaces paths in G_M by MLPs in the last step. Note that W'_{v,v_0} is an upper bound for the MLP from v to v_0 with any number of UAVs, because in the worst case the shortest path in G_M from v to v_0 is also the MLP, i.e. $c_{max} := \max_{v \in V_S} W'_{v,v_0}$ is an upper bound for all MLPs from any v to v_0.

k-SPLITOUR produces subtours with lengths at most $L/k + 2(1-1/k)c_{max}$. Inequality (1) follows from $L \leq eC_1^*$, $c_{max} \leq C_k^{O*}$, and $C_1^* \leq k(C_k^{O*} + c_{max}) \leq 2kC_k^{O*}$.

To establish a relationship between C_k^{O*} and C_r^{O*} we describe a procedure for iteratively merging paths from an optimal solution with r paths until a solution with at most k paths remains. The procedure starts from the initial solution with cost C_r^{O*} and merges $\lfloor r/2 \rfloor$ pairs of paths by connecting a pair of paths $p_i = (v_0, v_1^i, \ldots, v_{l(i)}^i)$ and $p_j = (v_0, v_1^j, \ldots, v_{l(j)}^j)$ to a new path $(v_0, v_1^i, \ldots, v_{l(i)}^i, v_1^j, \ldots, v_{l(j)}^j)$. The new solutions has a cost of at most $c(p_i) + W'_{v_{l(i)}^i, v_1^j} + c(p_j) \leq 3C_r^{O*}$ because of the triangular inequality in G'. Here $c(p)$ denotes the length of the path p in G'. If an unmerged path remains, it is merged in the next iteration. There are at most $\lceil \log(r) - \log(k) \rceil + 1$ iterations until a solution with at most k paths is reached, which has a cost of $\bar{C}_k^O$. Therefore,

$$\begin{aligned} C_k^{O*} &\leq \bar{C}_k^O \leq \left(2^{\lceil \log(r) - \log(k) \rceil + 1} - 1\right) C_r^{O*} \\ &\leq \left(2^{\log(r) - \log(k) + 2} - 1\right) C_r^{O*} \leq \left(4r/k - 1\right) C_r^{O*}. \end{aligned} \tag{2}$$

Combining (1), (2) and $C_r^{O*} \leq C^*$ results in

$$\hat{C}_k \leq 2\left(e + 1 - 1/k\right)\left(4r/k - 1\right) C^*. \tag{3}$$

It remains to show that there is a constant e such that $L \leq eC_1^*$. Because of $c(MST(G')) \leq C_1^{O*} \leq C_1^*$, the cost of the minimum spanning tree in G', $c(MST(G'))$, is also a lower bound for the cost of the optimal solution with one sensing UAV, C_1^*. Therefore, the factor is the same as for the MST heuristic for TSP with $e = 2$.

The sensing UAVs in H2 and H3, which meet L^c if they can find a solution, have to send the data to a relaying UAV. This introduces a waiting time until the first relaying UAV is at its start vertex of the next MLP or a time span caused by traveling to the end vertex of the MLP and traveling to the next SL on the tour. The delay of such a detour is at most $2c_{max}$ for each data transport after a SL (H2 and H3 visit multiple SLs consecutively before transporting data to the BS). The number of such detours α for each of the k sensing UAVs is at most $\lceil \hat{C}_k/\beta \rceil$, with $\beta := \min\{c_{min}, L^c - c_{max}\}$ (instead of c_{max} the length of the longest MLP from any SL with $(r-k)/k$ UAVs can be used for a lower bound), and $c_{min} := \min_{v,w \in V_S} W'_{v,w}$. Thus, WII of H2 and H3, $\hat{C}_k^H$, is bounded $\hat{C}_k^H \leq 2\alpha c_{max} + \hat{C}_k = 2c_{max} + (2c_{max}/\beta + 1)\hat{C}_k$.

V. Simulation results

In this section we describe the results from simulation experiments with the aim to assess the performance of our three heuristics and compare it with a single-hop (SH) approach. In the single-hop approach the UAVs do not cooperatively transport the data but each UAV has to travel to a location where it can send its data directly to the BS.

The environment is modeled as rectangular grid of cells of unit size, and time is discretized into time steps. A UAV can move from one cell of the grid to one of the 8 neighboring cells (which determines G_M) or stay at the same cell within one time step. The communication range R^C (measured in number of cells) determines which cells are within communication range (and therefore determines G_C). We set W^C to zero for all edges. The BS is in the cell at the lower left corner. The size of the environment is 5×50 cells ($|V| = 250$) where the area of 5×5 cells ($|V_S| = 25$) with the largest distance to the BS are SLs.

In this scenario an optimal open tour through all SLs is a lawn mower pattern on the SL cells. This tour is split evenly into subtours for multiple UAVs. We have implemented our heuristics in Matlab using simple algorithms for solving the TSP and splitting the tour. Solving the LBAP is done optimally with the state-of-the-art solver Gurobi. Alternatively, the assignment can be calculated with the Hungarian method in $O(r^3)$. The runtimes of all heuristics for all problems were less than 3 seconds on a Core i7-6700K.

The single-hop approach is not always able to meet L^c because the time to travel from a SL to a location where the data can be transmitted to the BS is larger than L^c. In this case a UAV travels directly to a location where it can send its data to the BS after visiting a SL and the resulting worst latency WL is recorded for the simulation results. The single-hop approach is equivalent to H3 with a sufficiently large latency bound, e.g. $L^c = \infty$. In this case a subtour for each UAV is generated and a UAV visits SLs on its subtour consequently until the latency would be larger than L^c. This effectively results in the optimal solution for visiting all SLs as soon as possible, and a UAV does not transmit data to the BS before it visited all SLs on its subtour.

Figure 6 (left column) shows a comparison of WII, FII and WL between the heuristics H1, H2, and H3 for varying latency constraint L^c. The number of UAVs n is 6 and the communication range $R^C = 8$. The counter intuitive behavior of H1 and H3, which show an increasing WII with increasing L^c, results from the fact that the algorithms minimize the number of UAVs necessary to meet the latency

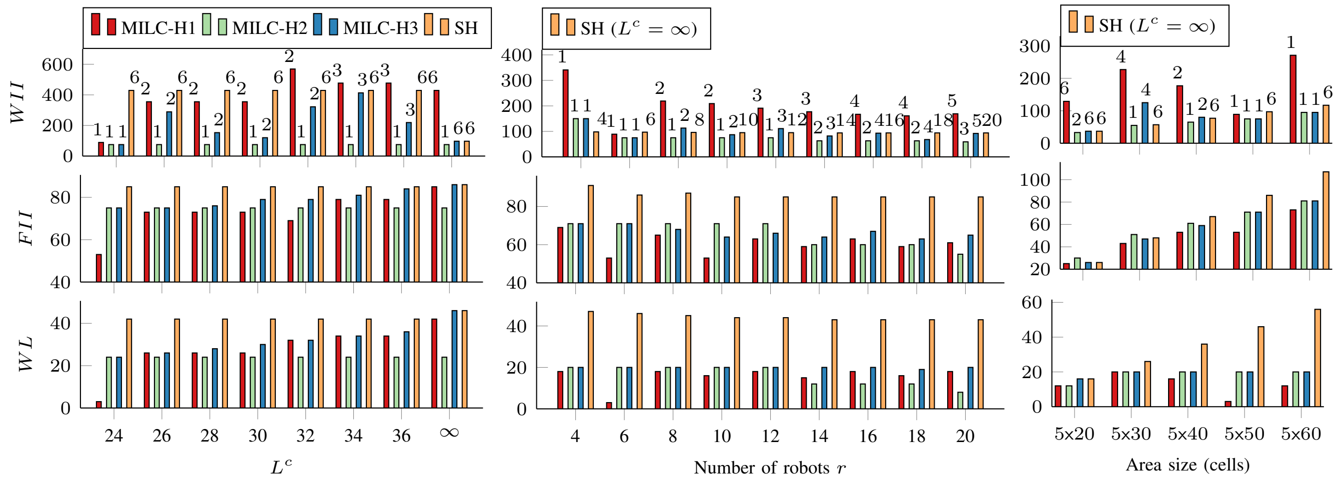

Fig. 6. WII, FII, WL for the MILC heuristics for varying L^c (left, $r = 6$, $R^C = 8$), for varying number of robots r (center, $L^c = 20$, $R^C = 8$), and for varying area sizes (right, $r = 6$, $L^c = 20$ $R^C = 8$). The numbers above the bars show the numbers of subtours.

constraints for each SL. This results in longer durations for the transportation of the data to the BS. The drops in WII happen when the number of subtours increases due to a splitting of the original tour. H2 is not able to split the tour with the available number of UAVs since in all cases the latency could be further decreased with an increasing number of UAVs. For H3 the number of subtours is the same as for H1. In contrast to H1, H3 can benefit from an increasing latency bound, since more SLs can be visited without data transportation to the BS. Figure 6 further shows that L^c directly influences the time when the first data arrives at the BS FII, and necessarily $WL \leq L^c$ for the MILC heuristics.

To justify the cooperative data transport, a comparison between the single-hop approach with effectively unlimited latency and the MILC heuristics is necessary. Figure 6 (left column) shows that FII and WL of the single-hop approach is worse than for any L^c for all three MILC heuristics. The WII of the single-hop approach is larger than for H2 in general, and for H1 and H3 with low latency bounds. With $L^c = \infty$, H3 is equivalent to the single-hop approach, and with low L^c H1 and H3 are equivalent.

Figure 6 (center column) depicts the simulation results for a varying number of UAVs. $L^c = 20$ and $R^C = 8$ have been set such that at least 4 UAVs are necessary to transport the data within L^c to the BS. The increasing WII of H1 and H3 for an increasing number of UAVs results from splitting of the tour into multiple subtours. The number of UAVs per subtour is largest with $r = 6$, resulting in a low latency along the MLP such that the UAV, which is visiting SLs, does not have to wait for the UAVs on the MLP. Because H2 tries to minimize the latency for each SL by using as many UAVs as possible for the MLP, this heuristic does not show the effect of increasing WII. This leads to the behavior that sensing UAVs can make faster progress along their subtours.

Figure 6 (right column) shows a comparison for different area sizes $5 \times w$ (i.e. distances of the block of 25 SLs to the BS) for $w = 20, \ldots, 60$. The benefit of cooperative data transport increases with the distance of the SLs to the BS.

We have further tested the algorithms to environments of size 5×50 where the positions of 25 SLs have been randomly sampled over the whole area. The average values (WII, FII, WL) for 10 runs ($r = 6, L^c = 20$) are: H1: $(109.6, 5.5, 4.5)$, H2: $(94.6, 23.2, 19.4)$, H3: $(89.5, 23, 19.2)$, SH ($L^c = \infty$): $(89.6, 19.9, 46.6)$. Because the SLs are closer to the BS on average, WII and FII for SH ($L^c = \infty$) are comparable to the corresponding values for the MILC heuristics, but WL is much worse.

To summarize our simulations, the simplest heuristic H1 performs worst (regarding WII) in all experiments and serves as baseline, whereas H2 outperforms the other heuristics with the expected behavior of a non-increasing WII with increasing L^c, R^C and r. As shown in our simulation study, cooperative data transport is clearly justified when the SLs are not within the communication range of the BS and the mission objectives require an early arrival of the data from SLs at the BS. These two conditions are relevant for various surveillance applications including large area surveillance and first responder support. Although a manually constructed[1] solution using relay chains performs comparable, relay chains might lead to a large latency and detours for sensing UAVs and might be inferior to MLPs in more complicated environments.

VI. Conclusion

We presented a multi-UAV surveillance problem with cooperative data transport with the aim to minimize the time until data captured by UAVs at SLs arrive at the BS. We achieved prompt data delivery by constraining the latency to a predefined bound. This enforces the UAVs to transport the data cooperatively to the BS in a store-and-forward fashion. The presented heuristics are based on MLPs, which

[1]Manually constructed solution on the 5×50 area with a block of 25 SLs: k UAVs visit the SLs (the assignment of SLs to UAVs and the order in which they are visited is determined manually) and transport the data to a relay chain to the BS established by $6 - k$ UAVs. The values for $k = 1, \ldots, 5$ are: $(75, 75, 29)$, $(73, 64, 27)$, $(72, 72, 26)$, $(80, 79, 35)$, $(86, 86, 41)$.

guarantee the latency bound. We evaluated the performance in simulation experiments, which show that the baseline heuristic H1 performs worst. H2 outperforms the other heuristics with the expected behavior of a decreasing worst idleness with increasing values of the latency bound and the number of robots. Additionally, we show that cooperative data transport can outperform uncooperative data transport with respect to the defined metrics.

This work provides a first theoretical investigation in cooperative data transport with dedicated objectives and requirements. A validation considering further important aspects (e.g. physical properties, advanced communication models) and incorporation of technical limitations (e.g. limited flight time) are still needed for real-world deployment. Scalability and robustness is limited by the fact that a centralized entity has to generate the solution before the mission starts.

We identify several directions for future work. First, the heuristics rely on TSP tours through all SLs, which have been generated with traditional algorithms that try to minimize the length of the tour. An open issue is the generation of tours that support the joint minimization of information idleness and latency. Second, the scheduling of UAVs can be improved to minimize the number of idle UAVs (that do neither sensing nor transporting data) at each time instant. The team of UAVs is divided into a fixed partition of teams and only one UAV in a team does sensing while the other UAVs transport the data to the BS although not all of them might be necessary to meet the latency bound. These improvements could also include the design of tree-shaped MLPs where UAVs transport the data from more than one sensing UAV. Third, investigations into mixing the discussed strategies or adjusting the latency for optimizing the metrics would be a promising direction. For example, at the beginning of the mission it could be beneficial to quickly transport data to the BS but more resources could be assigned for visiting SLs in the course of the mission execution. We also envision the situation which allows the mission operator to choose from multiple solutions generated by different algorithms depending on the mission requirements.

References

[1] 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.*, pages 565–574, May 2013.

[2] S. Alamdari, E. Fata, and S. L. Smith. Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations. *Int. J. of Robot. Res.*, 33(1):138–154, Jan. 2014.

[3] J. Banfi, N. Basilico, and F. Amigoni. Minimizing communication latency in multirobot situation-aware patrolling. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 616–622, Sept. 2015.

[4] J. Banfi, N. Basilico, and F. Amigoni. Multirobot reconnection on graphs: Problem, complexity, and algorithms. *IEEE Trans. Robot.*, 34(5):1299–1314, Oct. 2018.

[5] J. Banfi, A. Q. Li, N. Basilico, I. Rekleitis, and F. Amigoni. Multirobot online construction of communication maps. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 2577–2583. IEEE, May 2017.

[6] J. Banfi, A. Quattrini Li, I. Rekleitis, F. Amigoni, and N. Basilico. Strategies for coordinated multirobot exploration with recurrent connectivity constraints. *Auton. Robots*, 42(4):875–894, Apr. 2018.

[7] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer, Berlin, 2007.

[8] J. Desrosiers, Y. Dumas, M. M. Solomon, and F. Soumis. Time constrained routing and scheduling. In *Handbooks in Operations Research and Management Science*, volume 8, pages 35–139. 1995.

[9] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz. Help from the sky: Leveraging UAVs for disaster management. *IEEE Pervasive Comput.*, 16(1):24–32, Jan. 2017.

[10] J. L. Fargeas, B. Hyun, P. Kabamba, and A. Girard. Persistent visitation under revisit constraints. In *Proc. Int. Conf. Unmanned Airc. Syst.*, pages 952–957, May 2013.

[11] E. F. Flushing, M. Kudelski, L. M. Gambardella, and G. A. Di Caro. Connectivity-aware planning of search and rescue missions. In *Proc. IEEE Int. Symp. Saf., Secur., and Rescue Robot.*, 2013.

[12] G. N. Frederickson, M. S. Hecht, and C. E. Kim. Approximation algorithms for some routing problems. In *Proc. Annu. Symp. Found. Comp. Sci.*, pages 216–227, Oct. 1976.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, 1979.

[14] E. I. Grøtli and T. A. Johansen. Path planning for UAVs under communication constraints using SPLAT! and MILP. *Journal of Intelligent & Robotic Systems*, 65(1-4):265–282, Aug. 2011.

[15] G. A. Hollinger and S. Singh. Multirobot coordination with periodic connectivity: Theory and experiments. *IEEE Trans. Robot.*, 28(4):967–973, Aug. 2012.

[16] Y. Kantaros, M. Guo, and M. M. Zavlanos. Temporal logic task planning and intermittent connectivity control of mobile robot networks. *IEEE Transactions on Automatic Control*, 64(10):4105–4120, 2019.

[17] A. Khan, B. Rinner, and A. Cavallaro. Cooperative robots to observe moving targets: Review. *IEEE Trans. Cybern.*, 48(1):187–198, Jan. 2018.

[18] R. Khodayi-mehr, Y. Kantaros, and M. M. Zavlanos. Distributed state estimation using intermittently connected robot networks. *IEEE Trans. Robot.*, 35(3):709–724, June 2019.

[19] 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.*, pages 573–580, June 2017.

[20] V. Mersheeva and G. Friedrich. Multi-UAV monitoring with priorities and limited energy resources. In *Proc. Int. Conf. Autom. Planning Scheduling*, pages 347–355, Apr. 2015.

[21] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian. Control of multiple UAVs for persistent surveillance: Algorithm and flight test results. *IEEE Trans. Control Syst. Technol.*, 20(5):1236–1251, Sept. 2012.

[22] F. Pasqualetti, A. Franchi, and F. Bullo. On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE Trans. Robot.*, 28(3):592–606, June 2012.

[23] D. Portugal, C. Pippin, R. P. Rocha, and H. Christensen. Finding optimal routes for multi-robot patrolling in generic graphs. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 363–369, Sept. 2014.

[24] M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, and B. Rinner. Networked UAVs as aerial sensor network for disaster management applications. *e & i Elektrotechnik und Informationstechnik*, 127(3):56–63, Mar. 2010.

[25] J. Scherer and B. Rinner. Short and full horizon motion planning for persistent multi-UAV surveillance with energy and communication constraints. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pages 230–235, Sept. 2017.

[26] J. Scherer and B. Rinner. Multi-robot patrolling with sensing idleness and data delay objectives. *J. Intell. Robot. Syst.*, Mar. 2020. to be published, https://doi.org/10.1007/s10846-020-01156-6.

[27] J. Scherer and B. Rinner. Multi-robot persistent surveillance with connectivity constraints. *IEEE Access*, 8:15093–15109, 2020.

[28] J. Scherer, B. Rinner, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, and H. Hellwagner. An autonomous multi-UAV system for search and rescue. In *Proc. First DroNet Workshop on Micro Aerial Vehicle Netw., Syst., Appl. Civilian Use*, pages 33–38, May 2015.

[29] V. Spirin, S. Cameron, and J. de Hoog. Time preference for information in multi-agent exploration with limited communication. In *Lecture Notes in Comput. Sci.*, volume 8069, pages 34–45. Springer, Berlin, 2014.