Persistent Multi-UAV Surveillance with Energy and Communication Constraints

Jürgen Scherer and Bernhard Rinner[1]

***Abstract*— Multiple unmanned aerial vehicles (UAVs) are increasingly applied for surveillance tasks such as disaster management and environmental monitoring. Due to limited battery capacity and bounded wireless communication, small-scale UAVs pose fundamental challenges for achieving persistence. We propose an offline path planning algorithm that ensures that the UAVs can always reach the base station to replace their batteries and that each UAV is always connected with the base station via a single or multi-hop link. We focus on heterogeneous UAVs with different flight times. The single base station scenario is compared with several extensions for multiple base stations based on the maximum time a sensing location has not been visited.**

I. Introduction

The use of multiple small-scaled unmanned aerial vehicles (UAVs) for surveillance tasks has gained considerable attention for a variety of applications including disaster management and environmental monitoring [1]. These battery-powered UAVs are equipped with a wireless transceiver with limited communication range and can autonomously follow waypoints. UAVs can send their local data or relay data from neighboring UAVs over the wireless network [2]. In order to achieve *persistent* surveillance, the UAVs must be able to replace their batteries to extent the flight time. Maintaining a durable communication with the base station (via single- or multi-hop) throughout the entire mission allows sending sensor or telemetry data to the base station at any time. This is particularly important for disaster management scenarios where the mission operators have to be aware of the current situation.

There exist various approaches for multi-robot persistent surveillance in literature. In [3] a heuristic for the Continuous Monitoring Problem with Inter-Depot routes with the goal to maximize the visiting frequency of targets is proposed. The traveled distance of a UAV is limited by fuel constraints, but a UAV can refuel at any base station, and the duration of the whole mission is limited by the amount of fuel at the base stations. This is similar to [4] where closed tours through targets and refueling depots for a number of robots are planned such that each target is visited. The Persistent Vehicle Routing Problem (P-VRP) with recharging stations is modeled with temporal logic specifications in [5] and automata-based techniques are applied to derive control policies for the UAVs. The algorithm has exponential time complexity due to the NP-hardness of the VRP. In [6] path planning is decoupled from velocity control. UAVs move on predetermined closed paths and adjust the velocity such that a certain coverage criteria is met. In [7] a closed path with waiting intervals through all targets is generated and the robots move along this path in the same direction with the goal to minimize the highest priority-weighted age. In [8] the agent's controller adjusts direction and speed with the aim to continuously cover more important areas more carefully. The objective is to minimize the coverage error which is the difference between the actual and the desired coverage. In [9] agents choose the next cell to visit based on a weighted sum of distance and age with the aim to minimize the maximum age over all cells. Maintaining communication is a recurring task in robotic applications. In [10] the effect of connectivity on the coverage performance is presented. A distributed controller for maintaining network integrity is proposed in [11]. In [12] the planning of a mission to visit certain targets is done offline exploiting a radio propagation path loss simulator. Planning for periodic connectivity in environments with obstacles is done in [13]. A heuristic for the VRP with communication sites is presented in [14]. In [15] a mathematical programming approach for planning search and rescue missions for different connectivity demands is presented.

In this paper we address persistent multi-UAV surveillance with combined energy and communication constraints and provide a heuristic offline path planning algorithm. In our scenario the UAVs must return to a base station before their energy is depleted and must maintain communication connectivity with the base station throughout the mission. Our algorithm employs the idea of safe paths presented in [16] to ensure that every UAV can reach a base station with the remaining energy. UAVs recharge their batteries when they are at the base station and continue the mission. We extend the idea of selecting next sensing locations from [9] to the case where UAVs are constrained in their movement due to the communication constraints. In contrast to the existing literature, we consider the persistent multi-UAV surveillance with combined energy and communication constraints and focus on heterogeneous UAVs with different maximum flight times. Additionally, we investigate the case where more than one base stations are located within the mission area and compare different extensions to the single base station case.

We define the persistent surveillance problem in Section II and describe the algorithm for the case of one base station in the mission area in Section III and for multiple base stations in Section IV. The setup and the results of the simulations are presented in Section V, and finally, Section VI concludes the paper.

[1]Both authors are with the Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, Austria, {juergen.scherer, bernhard.rinner}@aau.at

II. Problem statement

The mission area is rectangular and divided into a two-dimensional regular grid of square cells. Each cell is identified by x and y coordinates. A subset $\mathcal{S}$ of these cells are sensing locations and a subset $\mathcal{B}$ are base stations. A set of UAVs $\mathcal{U}$ visits sensing locations repeatedly to take measurements. Time is divided into discrete steps, and a UAV can move to one of the 8 neighboring cells or stay at its current position at each time step. The age $a_{l,s}$ of a sensing location s at a certain instant l denotes the number of time steps that passed since the most recent visit of any UAV at s. The flight time of a UAV is limited by the energy (fuel or battery) capacity, which can be recharged at a base station. The energy capacity or maximum flight time is measured in time steps, and a UAV consumes one energy unit at every time step regardless of whether the UAV is hovering or moving to a neighboring cell. This *energy constraint* requires that each UAV has to reach a base station before its energy is depleted. UAVs and base stations are equipped with wireless transceivers with limited communication range R^{com}, and there is a link between two UAVs or a UAV and a base station if the distance between them is less than R^{com}. The distance between two nodes (a node denotes both a UAV and a base station) is the Euclidean distance between the centers of the cells at which the nodes are located. Additionally, the *communication constraint* requires that there must be a single or multi-hop link from every UAV to a base station at every time step.

The problem is to plan a path for every UAV such that the energy and communication constraints are satisfied and the maximum age over all sensing locations is minimized. In the following we assume a limited mission duration of L time steps. The maximum age of the mission with length L is the largest number of time steps between two consecutive visits over all sensing locations.

III. Single base station

The idea of the centralized algorithm is to plan the next move for every UAV at each time step such that the goal of minimizing the maximum age is achieved. This requires a policy for selecting the next sensing location for each UAV. To ensure that every UAV can reach a base station with its remaining energy, the algorithm tries to calculate a safe path for each UAV from its current position to the base station such that the energy and communication constraints among the UAVs are satisfied. If such safe path cannot be computed, some or all UAVs have to move on a previously calculated safe path. The following subsections describe the details of the algorithm which has linear time complexity in the number of time steps and polynomial time complexity in the number of UAVs and sensing locations.

A. Sensing location selection

For the purpose of minimizing the maximum sensing location age, we employ the policy of [9] that assigns a value to each sensing location for each UAV at time step l (see Equation (1)). The value depends on the age of the sensing location s, $a_{l,s}$, the distance between UAV u and s, $\chi_{l,u,s}$, and the minimum distance between any other UAV v to this sensing location. The values ω_0 and ω_1 are weighting values that need to be determined (cp. Section V). The sensing location s with the largest value $v_{u,s}$ is assigned to UAV u, such that $A(u) = s$, where A denotes the assignment vector.

$$v_{u,s} = a_{l,s} + \omega_0 \chi_{l,u,s} + \omega_1 \min_{v \neq u} \chi_{l,v,s} \tag{1}$$

In contrast to [9], a UAV cannot fly directly to the destination in our scenario because of the communication constraint. Instead, a UAV moves to a neighboring cell that is closer to the destination than the current cell if the multi-hop link to the base station does not break. To model this, we use the notion of a communication graph. UAVs and the base station form the nodes of an undirected communication graph with an edge between two nodes if and only if the distance between them is less than R^{com}. There is a multi-hop connection from every UAV to the base station if and only if the communication graph is connected.

In more detail, at every step of the algorithm the procedure do_step (see Algorithm 1) iterates over all UAVs in an arbitrary but fixed order and $calc\_objective$ assigns a value to each neighboring cell c in the 8 neighborhood cells $N(p(u))$ and the current cell $p(u)$ of a UAV u[1]. The objective value of a cell c is the distance between UAV u and its assigned goal $A(u)$ or ∞ if the communication graph would become disconnected by moving to this cell. The UAV then moves to the cell with lowest objective value. Note, that this ensures a connected communication graph at every time step, because a UAV can also stay at its current position (hereafter, a move means an action a UAV executes at a time step which can be moving to a neighboring cell or staying at its current cell). The result of the procedure is a new set of paths P', which comprises one path for each UAV. A path for a UAV is a sequence of cells and every step on a UAV's path represents a move of the UAV.

B. Safe paths

A safe path P_s is a set of paths, one for each UAV. A safe path for a UAV is a sequence of cells that starts at the current position of the UAV, ends at the base station and is not longer than the remaining energy of the UAV. Additionally, the safe path has the property that if all UAVs execute a move on its safe path simultaneously, the communication graph stays connected.

The procedure $check\_safepath$ tries to calculate a safe path for the current state of the UAVs (the state of a UAV comprises its positions p and its remaining energy e) and returns a safe path or indicates that such a path could not be found. Similar to the procedure do_step, $check\_safepath$ selects a move for each UAV that decreases the distance to the base station such that the communication graph stays connected. If a safe path was found, do_step returns a new

[1]The outer loop which iterates from 1 to L and calls do_step is not shown

path which is identical to the old path P up to time step l and updated with the new UAV positions at l. The new safe path is stored for the next iteration. If a safe path cannot be calculated, the algorithm tries to repair the situation. Two versions have been implemented. The first one is presented in do_repair_simple (see Algorithm 2) and just returns a new path which is the old path updated with a step on the safe path at index l for every UAV[2]. With a fleet of UAVs having heterogeneous maximum flight times this strategy forces UAVs with more remaining energy time to waste time on the safe path instead of approaching their goal sensing location. For this reason a second repair strategy has been implemented (see Algorithm 3).

The idea is, that only UAVs move along the safe path that have to return to the base station, while others can approach their desired goals. Nevertheless, because the communication graph has to stay connected, the movement of UAVs that do not have to go to the base station may be constrained. To reduce the dependency between the UAVs, a spanning tree (denoted by T_s, T_s', T_s'') for the communication graph is calculated by $check\_safepath$ and returned to do_step. The procedure $check\_safepath$ determines a spanning tree at the current state of the UAVs and tries to find a safe path such that the edges of the spanning tree remain in every subsequent communication graph if the UAVs execute a move on their safe path at the same time. In particular, a minimum spanning tree (MST) is calculated based on the distance between the nodes as edge weights. This spanning tree is passed to do_repair_st, which works as follows. The procedure iterates over all UAVs that have to move to the base station (denoted by the set $fail$). Every UAV in this set moves along its safe path towards the base station. Then, the resulting communication graph G_c (calculated including the position of the base station p_{bs}) is compared with the spanning tree T_s determined by $check\_safepath$. This comparison done by $calc\_broken$ results in all nodes that are incident to edges that are present in the spanning tree T_s but not in the communication graph G_c. These nodes are then added to the set $fail$. In this manner, UAVs that have to return to the base station pull other UAVs along their safe paths if the spanning tree would break. This loop terminates because of the property of the safe path that maintains the spanning tree at each step. If all UAVs have moved on the safe path the spanning tree T_s is a sub-graph of G_c and there are no broken links anymore. Therefore, the set $fail$ is empty. After the loop the set not_moved contains all UAVs that have not moved along the safe path and that can try to approach their goals. Similar to do_step the objective for each neighboring cell is calculated and it is checked by $check\_safepath$ whether the most valuable move is possible.

IV. Multiple base stations

In this section we describe the algorithm extension for multiple base stations where the UAVs can recharge. We

[2]Details about the advancing index for P_s are not shown

Algorithm 1 procedure do_step

Input:
UAV states (p, e), path P, safe path (P_s, T_s), assignment A, current time step l
Output:
extended path P', new safe path (P_s', T_s')

$p' \leftarrow p$
for $u \in \mathcal{U}$ **do**
  $obj \leftarrow calc\_objective(u, p', A(u))$
  $p'(u) \leftarrow \arg\min_c \{obj(c)\}$
$(safe, fail, (P_s'', T_s'')) \leftarrow check\_safepath((p', e))$
if $safe$ **then**
  $(P_s', T_s') \leftarrow (P_s'', T_s'')$
  $P' \leftarrow P$
  $P'(l) \leftarrow p'$
else
  $(P', (P_s', T_s')) \leftarrow$
    $do\_repair((p, e), P, (P_s, T_s), A, fail)$

Algorithm 2 procedure do_repair_simple

Input:
UAV states (p, e), path P, safe path (P_s, T_s) current time step l
Output:
extended path P', new safe path (P_s', T_s')

$P' \leftarrow P$
$P'(l) \leftarrow$ next step on $P_s\ \forall u \in \mathcal{U}$
$(P_s', T_s') \leftarrow (P_s, T_s)$

assume that the position of the base stations are fixed and every UAV has to maintain a multi-hop link to any of the base stations. In a variable base station assignment[3], the base station where a UAV recharges and to where it maintains a multi-hop link may change during mission whereas in the fixed assignment this base station is determined for each UAV beforehand and does not change. The latter approach allows to separate the problem into independent sub problems which can be solved in parallel and reduces the size of each problem.

A. Variable base station assignment

In every iteration before do_step, the algorithm tries to find a new assignment of UAVs to base stations based on the distance between UAVs and the base stations. First, each UAV is assigned to its nearest base station. For each base station b the algorithm checks whether there is a safe path that ends in b for all UAVs newly assigned to b. If the algorithm fails to find a safe path for one base station, the new assignment of UAVs to base stations is reverted to the old one. After an assignment has been found the procedure do_step is called for each base station and its assigned UAVs.

B. Fixed base station assignment

In this case we search for an assignment of UAVs to base stations before the actual path planning happens independently for each base station. To achieve this, a Voronoi

[3]Not to be confused with the assignment of sensing locations to UAVs, see Section III-A

Algorithm 3 procedure do_repair_st

Input:
UAV states (p, e), path P, safe path (P_s, T_s), assignment A
failed UAVs $fail$

Output:
extended path P', new safe path (P'_s, T'_s)

$p' \leftarrow p$
$(P'_s, T'_s) \leftarrow (P_s, T_s)$
$not\_moved \leftarrow \mathcal{U}$
while $fail \neq \emptyset$ **do**
  $u \leftarrow$ first in $fail$
  $p'(u) \leftarrow$ next step on $P_s(u)$
  $G_c \leftarrow calc\_conngraph(p' \cup p_{bs})$
  $broken \leftarrow calc\_broken(G_c, T_s)$
  $fail \leftarrow fail \cup broken$
  $fail \leftarrow fail \setminus \{u\}$
  $not\_moved \leftarrow not\_moved \setminus \{u\}$
for $u \in not\_moved$ **do**
  Make move towards goal, check with
    $check\_safepath$ and update p' and (P'_s, T'_s)

partition of the mission area is generated where a grid cell c is in the Voronoi region of base station b if b is the nearest base station of c. Base station b and its assigned UAVs are responsible for covering the sensing locations within the Voronoi region of b (and no other sensing locations). Necessary (but not sufficient) conditions for covering a region are that (i) the number of UAVs is large enough to reach the farthest sensing location within the region and that (ii) the maximum flight time of the UAVs are long enough such that every position along the chain to the farthest sensing location can be reached by a UAV[4] (cp. Fig. 1). This is modeled as an mixed integer linear program (MILP) which assigns a given set of UAVs with certain maximum flight times to the base stations with a given position in the mission area. The objective function should reflect the performance of the UAVs within the region. For this, two assumptions are made: between the base stations the number of UAVs should be proportional to the diameter (distance between base station and farthest sensing location in its Voronoi region) of the Voronoi region and the sum of the maximum flight times of the UAVs should be proportional to the area of the region. The area of the region is defined as the number of sensing locations within a region. To meet both objectives, the program is split into two MILPs. The first one minimizes the deviation of the ratio of the number of UAVs in the assignment and the number of UAVs necessary to cover the region between all base stations, see expressions (2)-(4). Here $x_{u,b}$ is a binary variable which is 1 if and only if UAV u is assigned to base station b, and $\bar{d}_b$ is the number of UAVs necessary to cover the Voronoi region of base station b. Constraint (5) ensures that the number of UAVs is at least the minimum number of UAVs necessary, and constraint (6) ensures that every relay position can be reached by a UAV. The coefficient $d_{u,b,m}$ is 1 if UAV u can reach position m along the chain to the farthest sensing location of the region

[4]Flight time has to include the return to the base station

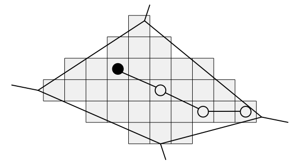

Fig. 1. Base station b (black circle) and its Voronoi region. In this case there are 3 UAVs (white circles) necessary to reach all sensing locations (assuming that the rightmost cell is a sensing location), thus $\bar{d}_b = 3$.

of base station b. Constraint 7 ensures that each UAV gets assigned to exactly one base station.

$$\min \Delta^d \tag{2}$$
$$\delta_i^d - \delta_j^d \leq \Delta^d, \quad \forall i,j \in \mathcal{B} \tag{3}$$
$$\sum_{u \in \mathcal{U}} \frac{1}{\bar{d}_b} x_{u,b} = \delta_b^d, \quad \forall b \in \mathcal{B} \tag{4}$$
$$\sum_{u \in \mathcal{U}} x_{u,b} \geq \bar{d}_b, \quad \forall b \in \mathcal{B} \tag{5}$$
$$\sum_{u \in \mathcal{U}} d_{u,b,m} x_{u,b} \geq 1, \quad \forall b \in \mathcal{B}, \forall m = 1, \dots \bar{d}_b \tag{6}$$
$$\sum_{b \in \mathcal{B}} x_{u,b} = 1, \quad \forall u \in \mathcal{U} \tag{7}$$

The result of this program is an assignment $X_{u,b}$ which is used as input for the second MILP. Here, the deviation of the ratio of the sum of the flight times and the area A_b between the base stations is minimized (expressions (8)-(10)), and the number of UAVs is fixed according the outcome of the previous MILP (constraint (11)). For brevity, the constraints (6) and (7) are not repeated here.

$$\min \Delta^a \tag{8}$$
$$\delta_i^a - \delta_j^a \leq \Delta^a, \quad \forall i,j \in \mathcal{B} \tag{9}$$
$$\sum_{u \in \mathcal{U}} \frac{e_u}{A_b} x_{u,b} = \delta_b^a, \quad \forall b \in \mathcal{B} \tag{10}$$
$$\sum_{u \in \mathcal{U}} x_{u,b} = \sum_{u \in \mathcal{U}} X_{u,b}, \quad \forall b \in \mathcal{B} \tag{11}$$

V. Simulation results

The size of a grid cell and the duration of one time step in a real world mission depend on different factors. For example, in case of imaging, the area covered on the ground by a camera and therefore the cell size depends on the altitude of the UAV. If the cell size is defined such that the cell cannot be covered with a single measurement, the definition of one time step has to take into account the time it takes to cover a cell and to travel to a neighboring cell. We provide simulation results on abstract mission specifications (cp. Subsection V-A) to get insights into the performance of the approaches. The mission specifications are also a tradeoff between computation time and mission size in terms of mission area and mission length. The reason for this is that the parameters of Equation (1) are determined by

TABLE I

UAV CONFIGURATIONS FOR THE SINGLE BASE STATION SCENARIO. (THE VALUES HAVE TO BE MULTIPLIED BY 10.)

4 UAVs		6 UAVs		8 UAVs		10 UAVs	
(1)	2,4,8,15	(7)	2,4,4,8,8,15	(13)	2,2,4,4,8,8,15,15	(19)	2,2,4,4,4,8,8,8,15,15
(2)	4,4,8,15	(8)	4,4,4,8,8,15	(14)	4,4,4,4,8,8,15,15	(20)	4,4,4,4,4,8,8,8,15,15
(3)	4,8,15,30	(9)	4,8,8,15,15,30	(15)	4,4,8,8,15,15,30,30	(21)	4,4,8,8,8,15,15,15,30,30
(4)	8,8,15,30	(10)	8,8,8,15,15,30	(16)	8,8,8,8,15,15,30,30	(22)	8,8,8,8,8,15,15,15,30,30
(5)	8,15,30,60	(11)	8,15,15,30,30,60	(17)	8,8,15,15,30,30,60,60	(23)	8,8,15,15,15,30,30,30,60,60
(6)	15,15,30,60	(12)	15,15,15,30,30,60	(18)	15,15,15,15,30,30,60,60	(24)	15,15,15,15,15,30,30,30,60,60

TABLE II

UAV CONFIGURATIONS FOR MULTIPLE BASE STATION SCENARIOS. (THE VALUES HAVE TO BE MULTIPLIED BY 10.)

8 UAVs	10 UAVs	12 UAVs
$7 \times 2, 1 \times 60$	$9 \times 2, 1 \times 60$	$11 \times 2, 1 \times 60$
$7 \times 4, 1 \times 120$	$9 \times 4, 1 \times 120$	$11 \times 4, 1 \times 120$
$7 \times 4, 1 \times 60$	$9 \times 4, 1 \times 60$	$11 \times 4, 1 \times 60$
$7 \times 8, 1 \times 120$	$9 \times 8, 1 \times 120$	$11 \times 8, 1 \times 120$
$7 \times 8, 1 \times 60$	$9 \times 8, 1 \times 60$	$11 \times 8, 1 \times 60$
$7 \times 15, 1 \times 120$	$9 \times 15, 1 \times 120$	$11 \times 15, 1 \times 120$

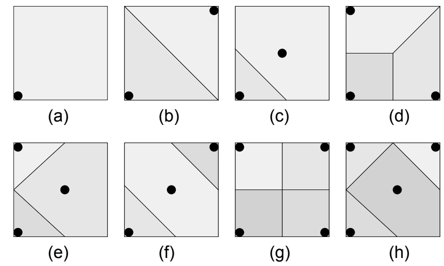

Fig. 2. Base station configurations with Voronoi partitions. Black circles denote base stations.

the non-gradient based optimization algorithm *patternsearch* provided by Matlab which searches for a local optimum. This algorithm systematically samples the parameter space and plans for a whole mission to determine the objective value for a specific set of parameters. We do this optimization for each single scenario to get comparable results. In real world applications these parameters can be looked up in tables.

A. Single base station

The simulations are conducted for a limited time horizon of $L = 1200$ steps on an area of 20×20 cells where every cell is a sensing location. The communication range $R^{com} = 8$ is chosen such that 4 UAVs can reach every sensing location. In the single base station case the base station is located at the lower left corner of the area. The flight times of the UAVs are drawn from factors of 1200 between 20 and 600 for different number of UAVs. The specific flight times of the UAVs measured in time steps are listed in Table I (hereafter, a set of UAVs with specific flight times is called UAV configuration).

In Fig. 3 the maximum age over all sensing locations is plotted for every UAV configuration. The first data row represents the simple safe path and the second the ST approach (cp. Subsection III-B). The ST approach gains a considerable improvement over the simple approach if the shortest maximum flight time of the UAVs is small compared to the mission area. This effect decreases with increasing number and flight times of the UAVs.

B. Multiple base stations

For the multiple base station case the simulation is conducted for 8 base station configurations (a)-(h) (cp. Fig. 2) where the configuration (a) is the single base station case. For this simulations the ST approach is chosen. An assignment of UAVs to base stations is calculated before the simulation for a certain UAV and base station configuration. The upper graph of Fig. 4 shows average values over 18 UAV configurations in total for every base station configuration. The UAV configurations include (13)-(24) from Table I and 6 similar configurations for 12 UAVs. Additionally, the 18 configurations listed in Table II are used to investigate in the difference between the two fixed assignments described below (cp. lower graph of Fig. 4). The first data row shows the dynamic base station assignment. It can be seen that the performance is better in scenarios where there is a base station in the center of the mission area that serve as kind of range extender for the UAVs. The second data row shows the fixed base station assignment derived by the two MILPs described in Section IV-B. It shows a worse performance than the dynamic assignment in scenarios where the variation in the area of the Voronoi regions is large. This is due to the fact that the MILPs overrate the effect of the number of UAVs in a region. The third data row shows a fixed base station assignment where only the variation of the area is taken into account (one MILP with objective (8), (9), and (10) and constraints (5)-(7) is solved). Here it can happen that the number of UAVs is not distributed evenly to the regions which causes a slightly worse performance in (b) and (d), especially when the deviation of the flight times is large, as for the scenarios in Table II.

VI. CONCLUSION

We present an offline path planning algorithm for the problem of persistent surveillance with multiple UAVs under communication and energy constraints. We compare different safe path approaches for a single base station scenario and different approaches for multi base station scenarios. For the first case, determining a MST for safe paths can greatly reduce performance if UAVs have heterogeneous flight times and the shortest flight times are short compared to the mission area. For the latter case, we describe an extension of the single base station scenario with a variable base station assignment and compare it with fixed assignments derived from solutions of MILPs. It can be seen that the performance is comparable to the variable assignment in most of the investigated scenarios and the problem can be split into independent sub problems. The proposed approach can cause deadlocks (mutual blocking in movement) because UAVs try to reach their assigned goals independently of each other. Additionally, uncertainties that lead to different velocities of the UAVs will prohibit maintaining the communication links and therefore limit the practical use of an offline path planning algorithm. Nevertheless, if path planning is done before the mission starts, online (distributed) coordination during the mission is limited to synchronization of the UAVs

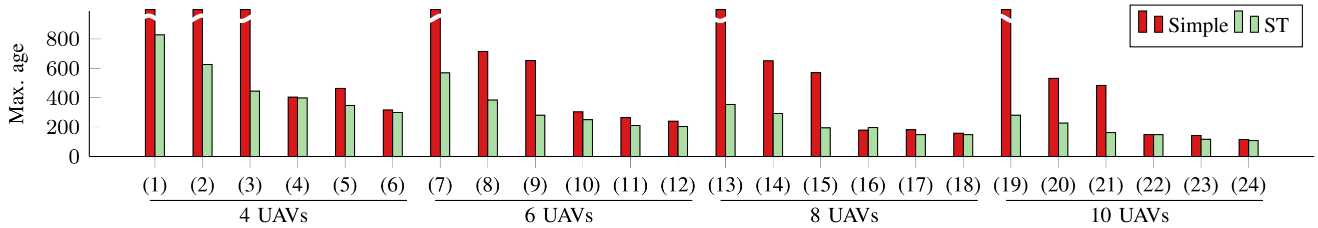

Fig. 3. Comparison of simple (first data row) and ST (second data row) safe path approach for 24 scenarios from Table I.

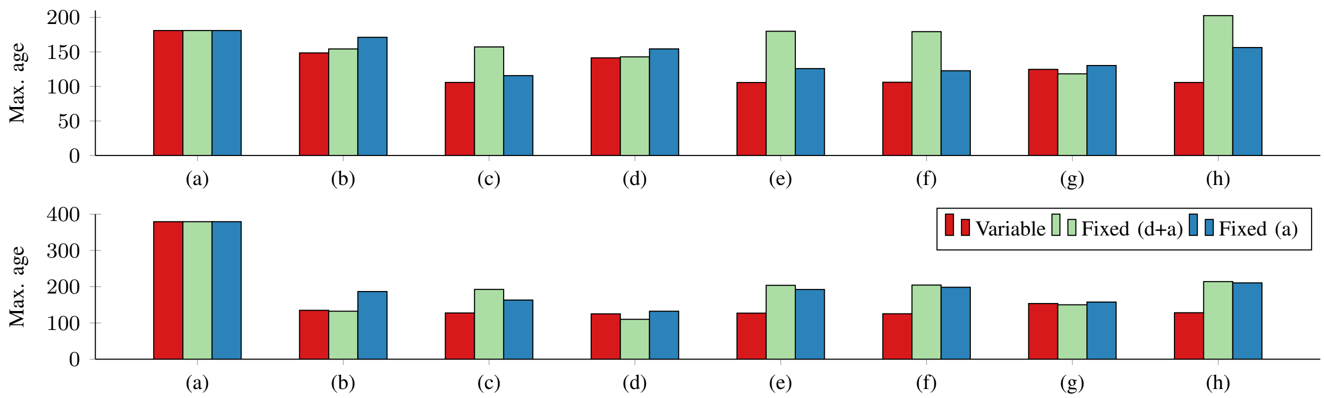

Fig. 4. Comparison of variable (first data row) and fixed (second and third data row) base station assignment. The values are averaged over scenarios (13)-(24) from Table I and 6 similar configurations for 12 UAVs (upper) and the scenarios listed in Table II (lower). In the second data row diameter and area of a Voronoi region is considered whereas in the third data row only the area is considered.

along the generated paths. These shortcomings are subject of ongoing work. Additionally, we plan to implement and test the approaches on our multi-UAV system [17].

ACKNOWLEDGEMENT

The work was supported by the ERDF, KWF, and BABEG under grant KWF-20214/24272/36084 (SINUS). It has been performed in the research cluster Lakeside Labs.

REFERENCES

[1] A. Khan, E. Yanmaz, and B. Rinner, "Information exchange and decision making in micro aerial vehicle networks for cooperative search," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 4, pp. 335–347, Dec. 2015.

[2] E. Yanmaz, S. Hayat, J. Scherer, and C. Bettstetter, "Experimental performance analysis of two-hop aerial 802.11 networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2014, pp. 3118–3123.

[3] V. Mersheeva and G. Friedrich, "Multi-UAV monitoring with priorities and limited energy resources," in *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling*, Apr. 2015.

[4] D. Mitchell, M. Corah, N. Chakraborty, K. Sycara, and N. Michael, "Multi-robot long-term persistent coverage with fuel constrained robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 1093–1099.

[5] C.-I. Vasile and C. Belta, "An automata-theoretic approach to the vehicle routing problem," in *Robotics: Science and Systems X*. Robotics: Science and Systems Foundation, July 2014.

[6] S. L. Smith, M. Schwager, and D. Rus, "Persistent robotic tasks: Monitoring and sweeping in changing environments," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 410–426, Apr. 2012.

[7] F. Pasqualetti, J. W. Durham, and F. Bullo, "Cooperative patrolling via weighted tours: Performance analysis and distributed algorithms," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1181–1188, Oct. 2012.

[8] C. Franco, G. Lopez-Nicolas, C. Sagues, and S. Llorente, "Persistent coverage control with variable coverage action in multi-robot environment," in *Proceedings of the 52nd IEEE Conference on Decision and Control*, Dec. 2013, pp. 6055–6060.

[9] N. Nigam, S. Bieniawski, I. Kroo, and J. Vian, "Control of multiple UAVs for persistent surveillance: Algorithm and flight test results," *IEEE Trans. Contr. Syst. Technol.*, vol. 20, no. 5, pp. 1236–1251, Sept. 2012.

[10] E. Yanmaz, "Connectivity versus area coverage in unmanned aerial vehicle networks," in *Proceedings of the IEEE International Conference on Communications (ICC)*, June 2012, pp. 719–723.

[11] M. Zavlanos, A. Ribeiro, and G. Pappas, "Network integrity in mobile robotic networks," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 3–18, 2013.

[12] E. I. Grøtli and T. A. Johansen, "Task assignment for cooperating UAVs under radio propagation path loss constraints," in *Proceedings of the American Control Conference (ACC)*, June 2012, pp. 3278–3283.

[13] G. A. Hollinger and S. Singh, "Multirobot coordination with periodic connectivity: Theory and experiments," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 967–973, Aug. 2012.

[14] J. Banfi, N. Basilico, and F. Amigoni, "Minimizing communication latency in multirobot situation-aware patrolling," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 2015, pp. 616–622.

[15] E. F. Flushing, M. Kudelski, L. M. Gambardella, and G. a. Di Caro, "Connectivity-aware planning of search and rescue missions," in *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2013.

[16] T. Schouwenaars, J. How, and E. Feron, "Receding horizon path planning with implicit safety guarantees," in *Proceedings of the American Control Conference*, June 2004, pp. 5576–5581.

[17] 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 *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use - DroNet '15*, 2015, pp. 33–38.