Model of Surveillance in Complex Environment Using a Swarm of Unmanned Aerial Vehicles

Petr Stodola[1](✉), Jan Drozd[2], and Jan Nohel[1]

[1] Department of Intelligence Support, University of Defence, Brno, Czech Republic
petr.stodola@unob.cz
[2] Department of Tactics, University of Defence, Brno, Czech Republic

Abstract. This paper examines the model of autonomous surveillance using a swarm of unmanned aerial vehicles with the simultaneous detection principle. This model enables to specify a number of sensors needed to detect an object of interest located in the area of interest; objects are detected only if scanned by the specified number of sensors simultaneously. The model plans deployment of individual vehicles in the swarm during the surveillance operation in a such a way that the surveillance is performed in the maximum quality; the quality is measured as a percentage of the area of interest that is covered during the operation. Furthermore, the surveillance is assumed to be conducted in the complex area of operations (including urban environments, build-up areas, or mountain environments with very uneven terrain) where occlusions caused by obstacles or terrain may occur often. For solution, the metaheuristic algorithm based on the simulated annealing is proposed. This algorithm deploys the number of waypoints, from which the monitoring is performed, maximizing the surveillance quality and taking the simultaneous detection principle into consideration. The algorithm is verified by a set of experiments based on typical surveillance scenarios.

Keywords: Surveillance · Unmanned aerial vehicles · Simultaneous detection · Urban environment · Metaheuristic algorithm · Experiments

1 Introduction

Contemporary armed conflicts are different than armed conflicts twenty years and more ago. One of the most significant characteristics of contemporary armed conflicts is changeable situation on the battlefield as well as countless information flow from different sources with different reliability. Moreover, most of the contemporary operations are conducted within specific environment such as urban and build-up areas (Siberia, Ukraine), etc., which significantly limit ordinary way of reconnaissance and surveillance. Such environment requires new approaches to gather all necessary information and process them in order to support Military decision process (in case of battalion level and above) or Troops leading procedure (in case of company level and below).

One of the crucial steps of the commanders' decision making is surveillance. It is possible to state, that surveillance is a continuous process which begins during the planning and decision making procedure. It provides critical information for commander's

J. Mazal et al. (Eds.): MESAS 2020, LNCS 12619, pp. 231–249, 2021.
https://doi.org/10.1007/978-3-030-70740-8_15

decision making. Ordinarily, it is conducted by special teams deployed in the depth of the enemy area. Apparently, deployment of such a team or teams is demanding on their training and preparation. Moreover, information flow from such teams are delayed and does not have to be precise, which could have a tremendous impact on the mission. In contemporary operations, new technologies such as unmanned aerial vehicles (UAVs) are used in order to gather almost online information and support commanders' decision making. The use of UAVs has tremendous impact on speed and quality of decision making. In addition to that, this information gathering save human resources. More information to this issue can be found for example in [1–7].

This article proposes the model of autonomous surveillance using a swarm of small UAVs (sUAV). The goal is to cover as large area of interest as possible by sensors of UAVs in the swarm. Each UAV is deployed in the exact location (waypoint) in the area of operations monitoring a portion of the area of interest. The model also allows the situation when more than one sensor is needed for the detection of some object of interest (further on referred to as the simultaneous detection). Furthermore, the surveillance is assumed to be conducted in complex area of operations (including urban environments, build-up areas, or mountain environments with very uneven terrain) where occlusions caused by obstacles or terrain may occur often.

Countless scientific works are focused on the use of swarms of UAVs for many purposes. There are several topics important to solve such complex problems like reconnaissance or surveillance conducted with an UAV swarm. Path planning for such a task is one of the crucial issue. Yao et al. [8] proposed a hybrid approach based on the Lyapunov Guidance Vector Field (LGVF) and the Improved Interfered Fluid Dynamical System (IIFDS), to solve the problems of target tracking and obstacle avoidance in three-dimensional cooperative path planning for multiple UAVs. Lamont et al. [9] designed and implemented a comprehensive mission planning system for swarms of UAVs. This system integrates several problem domains including path planning, vehicle routing, and swarm behavior as based upon a hierarchical architecture. Shanmugavel et al. [10] examined the problem of path planning for simultaneous arrival on a target.

Another crucial topic connected with UAVs is their reliability and failure protection. Military commanders must be ready to fulfil the mission in any unexpected situation. The use of a swarm of UAVs for surveillance tasks is a very important issue regarding precise critical information gathering. There are no specific scientific works focusing on this topic, however, there are several interesting papers which should be taken into the consideration. Triharminto et al. [11] developed an algorithm for moving target intercept with obstacle avoidance in 3D. The algorithm which is called L+Dumo Algorithm integrates a modified Dubins Algorithm and Linear Algorithm. Such approach could be modified in order to diminish impact of a UAV failure to finish the surveillance mission. Sampedro et al. [12] focused on scalable and flexible architecture for real-time mission planning and dynamic agent-to-task assignment for a swarm of UAVs. The proposed mission planning architecture consists of a Global Mission Planner (GMP) which is responsible of assigning and monitoring different high-level missions through an Agent Mission Planner (AMP), which is in charge of providing and monitoring each task of the mission to each UAV in the swarm. Sujit et al. [13] addressed the problem of generating feasible paths from a given start location to a goal configuration for multiple

UAVs operating in an obstacle rich environment that consist of static, pop-up and moving obstacles. The path planning system in the environment with pop-up and moving obstacles provides an inspiration to solve UAV swarm failure during the surveillance mission in a complex environment including build-up areas or mountain terrain.

2 Problem Formulation

In this section, the problem examined in this article is defined and formulated. In the first part, the model of the aerial surveillance is revised; this model is transformed from the reconnaissance model proposed during the previous research of the authors [14]. Then, the model is extended by the simultaneous detection principle.

2.1 Model of Aerial Surveillance

Let AoI be the area of interest to be monitored during the surveillance operation. AoI is represented by a polygon deployed in the area of operations ($AoI \subseteq AoO$). The goal of the surveillance operation is to monitor as much portion of AoI as possible using a swarm of UAVs.

Let $U = \{U_1, U_2, \ldots, U_N\}$ be a finite set of UAVs in the swarm where $N \geq 1$ is their number. The UAVs in the swarm are deployed in their base position in the area of operations and they are prepared to start the surveillance at the beginning of the operation. Each UAV is equipped with a sensor which is able to monitor a portion of the area of interest directly below. All UAVs (and their sensors) in the swarm are assumed to be identical. Each sensor is characterized by two parameters:

- Angular field of view α_{fov}.
- Maximum distance from an object of interest d_{max}.

These parameters specify detection range of the sensor – see the green area in Fig. 1. The ground objects which are located somewhere in this range are detected by an UAV.

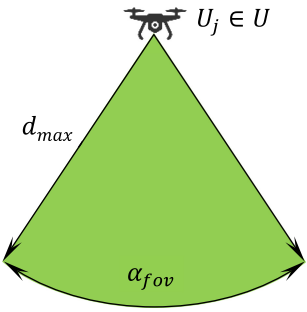


Fig. 1. Detection range of a sensor of an UAV.

Let $W = \{W_1, W_2, \ldots, W_N\}$ be a finite set of waypoints deployed in the area of operations. Each waypoint $W_i \in W$ is defined by:

- Coordinate x_i (x-coordinate).
- Coordinate y_i in the plane (y-coordinate).
- Height above the ground level h_i (z-coordinate).

Each UAV is assigned to its own waypoint from which it participates in the surveillance of the area of interest. Prior to the operation, each UAV flies from their base position to its waypoint. The operation starts when all the UAVs are located at their waypoints and it continues for a specified duration; during this time, all the UAVs are hovering at their waypoints and monitoring the area. At the end of the operation, the UAVs return back to their base position.

Every UAV in the swarm must be a vertical take-off and landing (VTOL) aircraft, i.e., one that can hover, take off, and land vertically. The duration of the operation must comply with the flight parameters of the UAVs, i.e., it must not take longer that the maximum flight time (including the time necessary to fly to their waypoints prior to the operation, and back to their base positions after the operation).

An UAV $U_i \in U$ located at its waypoint $W_i \in W$ during the operation observes some portion of the area of interest. This means that objects of interest are detected if located in the observed area at any time of the operation. However, the terrain and obstacles can cause occlusions. The principle is illustrated in Fig. 2. The green line represents the observed area (further on referred to as visible), the red line shows the area outside the detection range or the area occluded by obstacles.

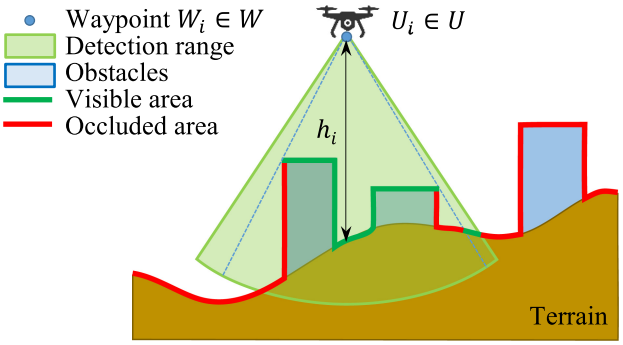


Fig. 2. Principle of monitoring the area by an UAV from a waypoint. (Color figure online)

The height above the ground level of an UAV located at any waypoint can be limited. The minimum and maximum limits (h_{min} and h_{max} respectively) can be set as a tactical requirement of the commander of the operation. The height h_i at waypoint $W_i \in W$ must be within the limits: $h_{min} \leq h_i \leq h_{max}$.

Each UAV $U_i \in U$ observes a portion of the area of interest $V_i \subseteq AoI$. Every point of the visible area V_i must satisfied the conditions as follows:

- Points in V_i must lie in the area of interest ($V_i \subseteq AoI$).
- Points in V_i must be within the detection range of the sensor of UAV U_i (see Fig. 1).

- There must be a visual line of sight (VLOS) between the sensor of UAV U_i and all points in V_i (see Fig. 2).

The principle is illustrated in Fig. 3. It shows the real situation from the top view. The area of interest is enclosed by a blue polygon; grey objects inside are obstacles (buildings in this case). The green area represents the visible area V_i observed from waypoint W_i. The occlusions caused by obstacles can be seen inside the green area.

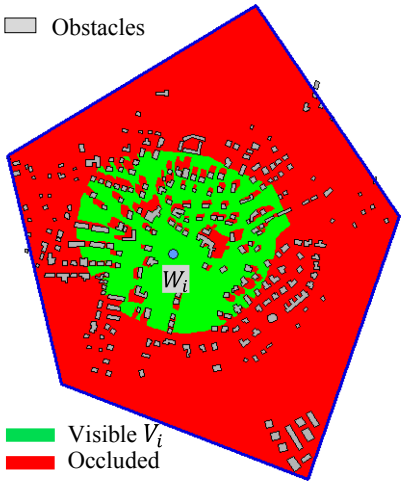


Fig. 3. Area observed from a waypoint.

The entire visible area V observed from all the waypoints is calculated according to formula (1) by unification of visible areas observed from individual waypoints. The coverage of the area of interest is computed according to formula (2) as the proportion of the size of the visible area $|V|$ to the size of the area of interest $|AoI|$.

$$V = \bigcup_{i=1}^{N} V_i \tag{1}$$

$$C = \frac{|V|}{|AoI|} \tag{2}$$

The principle is shown in Fig. 4. It is the same situation as in Fig. 3; this time, however, AoI is observed from five waypoints. The coverage (in percent) in case of using one waypoint is $C_{\%} = 20.07\%$ whereas it is $C_{\%} = 89.38\%$ in case of using five waypoints.

Let X be a particular solution in the state space. This solution is composed of N waypoints: $X = \{W_1, W_2, \ldots, W_N\}$; each waypoint has its exact position in the area of operations: $W_i = \{x_i, y_i, h_i\}$. Thus, a solution is composed of $3N$ independent continuous optimization variables: $X = \{x_1, y_1, h_1, x_2, y_2, h_2, \ldots, x_N, y_N, h_N\}$.

Beside the optimization variables, a surveillance operation is characterized by a number of constant parameters. These are as follows:

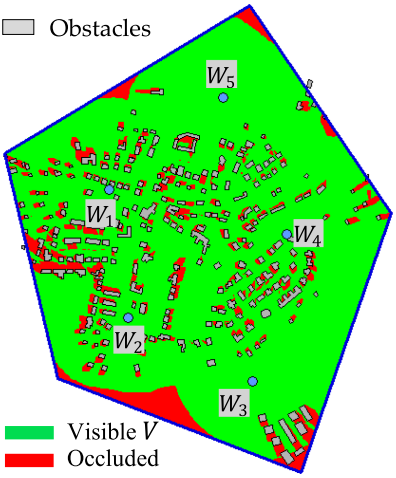


Fig. 4. Coverage of the area of interest.

- The terrain and database of obstacles in the area of operations determined by geographical data.
- Size, shape and position of the area of interest (AoI).
- Number of available UAVs, parameters of their sensors (N, α_{fov}, d_{max}).
- Minimum and maximum height of waypoints above the ground level (h_{min}, h_{max}).

The constant parameters do not change during the whole operation. Thus, the value of coverage C in a particular operation is computed for values of variables in solution X: $C = f(X)$.

The goal of the surveillance problem is to find such a solution X (that means to find values for $3N$ optimization variables) so that the coverage C is as large as possible, i.e., to maximize C – see the optimization criterion in formula (3).

$$\text{maximize}(C) \tag{3}$$

2.2 Model of Aerial Surveillance with Simultaneous Detection

In this section, the model of aerial surveillance is extended by the simultaneous detection principle. This principle ensures planning the surveillance operation in case that more than one sensors are required for the detection of the objects located in the area of interest.

In order to incorporate the simultaneous detection into the model, two new parameters are defined:

- Minimum number of sensors F for detection.
- Minimum permitted distance between waypoints D.

Both parameters are new constant parameters of the particular operation set according to tactical and technical requirements of the commander. Parameter F determines the number of sensors necessary to simultaneously observe all the points inside the AoI ($1 \leq F \leq N$). Parameter D ensures that the minimum space between every pair of sensors is kept – see constraint (4). This parameter is included because the model have a tendency to deploy waypoints close to one another for $F > 1$; this is not always desirable from the tactical point of view.

$$\left|W_i - W_j\right| \geq D \text{ for all } W_i \in W \text{and} W_j \in W;\ W_i \neq W_j \tag{4}$$

The calculation of the visible area V as defined in Sect. 2.1 needs to be reformulated since formula (1) is no longer applicable. Let V^F be a visible area observed simultaneously from at least F waypoints by F sensors. Let S be a set of all visible areas observed from individual waypoints: $S = \{V_1, V_2, \ldots, V_N\}$; set S has N distinct elements. Let S^i be an F-combination of set S (i.e., a subset of F distinct elements of S); $i = 1, 2, \ldots, C(N, F)$. Let S^i_j be an element of set S^i; $j = 1, 2, \ldots, F$. Then, the calculation of V^F is given in formula (5). Let C^F be the resulting coverage of the area of interest computed according to formula (6).

$$V^F = \bigcup_{i=1}^{C(N,F)} \left(\bigcap_{j=1}^{F} S^i_j \right) \tag{5}$$

$$C^F = \frac{\left|V^F\right|}{|AoI|} \tag{6}$$

The principle is shown on an example with $N = 4$ waypoints, i.e., $S = \{V_1, V_2, V_3, V_4\}$. Then, formulae (7) to (10) show the calculations of V^F for $F = 1, 2, 3, 4$.

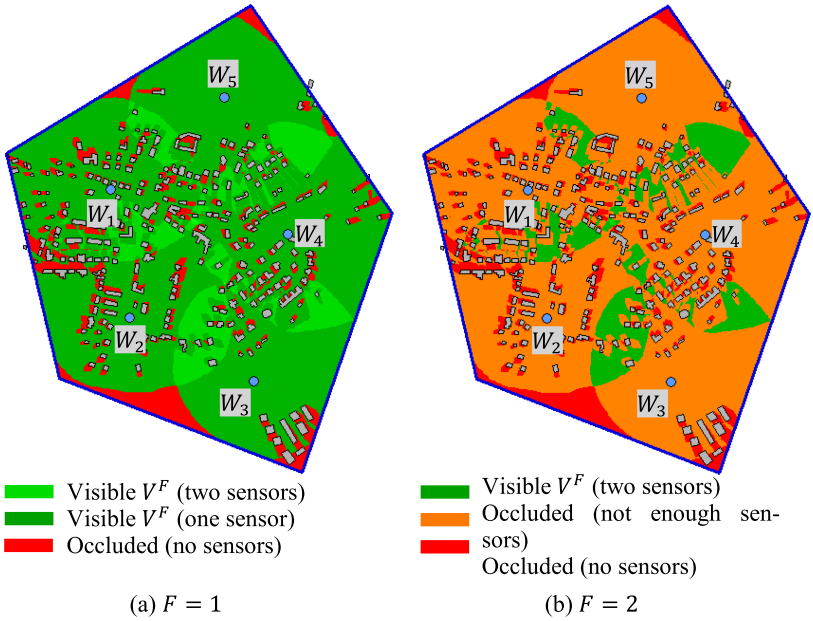
$$V^1 = V_1 \cup V_2 \cup V_3 \cup V_4 \tag{7}$$

$$V^2 = (V_1 \cap V_2) \cup (V_1 \cap V_3) \cup (V_1 \cap V_4) \cup (V_2 \cap V_3) \cup (V_2 \cap V_4) \cup (V_3 \cap V_4) \tag{8}$$

$$V^3 = (V_1 \cap V_2 \cap V_3) \cup (V_1 \cap V_2 \cap V_4) \cup (V_1 \cap V_3 \cap V_4) \cup (V_2 \cap V_3 \cap V_4) \tag{9}$$

$$V^4 = (V_1 \cap V_2 \cap V_3 \cap V_4) \tag{10}$$

The example in Fig. 5 shows the same situation as in Fig. 4; there are five waypoints deployed in the area of operations for five UAVs in the swarm. Figure 5(a) presents the coverage C^F for $F = 1$ (i.e., one sensor is sufficient for detection); the intensity of the green color distinguishes between the number of sensors observing the area. Figure 5(b) shows the coverage for the same deployment of waypoints for $F = 2$ (i.e., at least two sensors are needed for detection); the orange color represents the area which is observed by too low number of sensors (less than F). In the former case, the coverage is $C^1_{\%} = 89.38\%$ whereas in the latter it is only $C^2_{\%} = 11.50\%$. The reason for this high reduction in the quality of the surveillance operation is that the positions of waypoints were optimized for $F = 1$.



(a) $F = 1$ (b) $F = 2$

Fig. 5. Coverage for the deployment of waypoints optimized for $F = 1$.

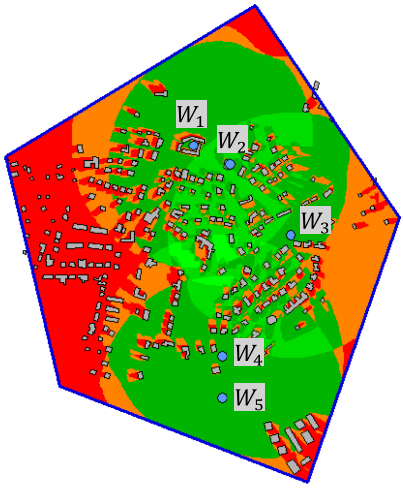


Fig. 6. Coverage for the deployment of waypoints optimized for $F = 2$.

Figure 6 presents the same surveillance operation with the same number of available UAVs in the swarm $N = 5$, but this time the positions of waypoints are optimized for higher number of sensors for detection: $F = 2$ and $D = 100$m. The new deployment of waypoints provides the increase in the coverage from $C^2_{\%} = 11.50\%$ (see Fig. 5(a)) to

$C^2_{\%} = 57.21\%$ (see Fig. 6). If bigger coverage is required, more sensors (i.e., UAVs in the swarm) are necessary.

3 Solution Algorithms

For the deployment of sensors, a metaheuristic algorithm based on the simulated annealing principle was proposed. The simulated annealing is a probabilistic technique inspired by the process of controlled cooling of a material in metallurgy to increase the size of its crystals and reduce their defects.

The simulated annealing is one of many possible stochastic approaches to solve complex optimization problems. The metaheuristic methods are often problem dependent; a method could be very successful in solving a certain type of a problem, nevertheless it does not cope well with others. The reason why the authors chose the simulated annealing is that it proved to be very effective in deployment problems, see for example [15, 16].

3.1 Evaluation of a Solution

The most critical part of the optimization from the time consumption point of view is the evaluation of a particular solution $X = \{x_1, y_1, h_1, x_2, y_2, h_2, \ldots, x_N, y_N, h_N\}$, i.e., calculation of $C^F = f(x_1, y_1, h_1, x_2, y_2, h_2, \ldots, x_N, y_N, h_N)$. It includes testing all the points laying inside the area of interest: they must be within the detection range of any sensor (or a number of sensors for $F > 1$) and the VLOS must exist between the point and this sensor (sensors).

There are an infinite number of points laying inside the area of interest; such a number of points cannot be evaluated from the practical point of view. Therefore, a rasterization using the Sukharev grid was chosen as an approach to estimate the value of C^F. Only points laying in the middle of rasterization squares are evaluated. Then, the value of C^F is calculated as a proportion of the number of visible points in rasterization squares to the total number of these squares. The precision of the estimated value depends on the size of the rasterization step d_{rast}.

A lot of effort was put into the effectiveness of the solution evaluation. Several approaches were used to make the algorithm as efficient as possible. The implemented algorithm manages to evaluate tens of millions of points per second on computers with common configurations. These main approaches were implemented:

- The VLOS between a sensor and a point is tested using the floating horizon algorithm. This algorithm is able to evaluate a number of points laying on the straight line between the point and the sensor with the linear computational complexity.
- Operations with integers only are employed in the most critical parts of the algorithm.
- The evaluation of points can be parallelized. Thus, it is distributed on cores of a multi-core processor.

3.2 Deployment of Waypoints

The algorithm for the waypoints deployment works in iterations. The idea behind the simulated annealing is in gradual cooling of the temperature in successive iterations. The temperature influences the probability of accepting newly generated solutions; even worse solutions can be accepted. This principle extends the search space and prevents to be stuck in some local optimum.

The algorithm employs a set of parameters controlling its behavior – see Table 1. The algorithm is presented in Fig. 7 in pseudocode. The algorithm calls several standalone functions as follows:

Table 1. Parameters of the algorithm for the deployment of waypoints.

Parameter	Description
T_{max}	Initial temperature (used in the first iteration)
T_{min}	Lower limit of temperature (termination condition)
γ	Cooling coefficient
n_{max}	Maximum number of transformations per iteration
m_{max}	Maximum number of replacements per iteration

- **Generate_Random_Solution**: generation of an initial (first) solution.
- **Evaluate_Solution**: calculation of $C^F = f(X)$ (see Sect. 3.1).
- **Transform_Solution**: transformation of the current solution into the new solution.
- **Calculate_Probability**: calculation of the probability to replace the current solution by the transformed solution.

The algorithm starts by generating an initial solution X_{cur} and its evaluation (lines 1 and 2). Each variable of the solution vector is set by the pseudo-random number generator with uniform distribution in the permitted range (given by the width and height of the area of interest for variables x_i and y_i, and between h_{min} and h_{max} for variables h_i). Then, the current temperature is set to its higher limit (line 3).

The algorithm works in iterations (lines 4 to 18). The temperature does not change within an iteration. The algorithm is terminated when the current temperature drops below the lower limit (line 4). In an iteration, a number of transformations and replacements are performed (lines 6 to 17). This number is controlled by parameters n_{max} (maximum number of transformations per iteration) and m_{max} respectively (maximum number of replacements per iteration) – see the termination condition of an iteration (line 6).

The transformation of the current solution X_{cur} into the new solution X_{new} (line 7) is carried out as follows. The solution vector $X = \{x_1, y_1, h_1, x_2, y_2, h_2, \ldots, x_N, y_N, h_N\}$ is composed of $3N$ independent variables. All the variables remain the same except one which is randomly selected. The selected variable changes its value in its permitted

Deploy_Waypoints()
Output: C^F, X
Constant parameters: $N, F, D, AoI, \alpha_{fov}, d_{min}, h_{min}, h_{max}, d_{rast}$
Algorithm settings: $T_{max}, T_{min}, \gamma, n_{max}, m_{max}$
1. $X = X_{cur}$ = Generate_Random_Solution()
2. $C^F = C^F_{cur}$ = Evaluate_Solution(X_{cur})
3. $T_{cur} = T_{max}$
4. **while** $T_{cur} \geq T_{min}$ **do** // Iterations
5. $n = m = 1$
6. **while** $n \leq n_{max}$ **and** $m \leq m_{max}$ **do** // Transformations
7. X_{new} = Transform_Solution(T_{cur}, X_{cur})
8. C^F_{new} = Evaluate_Solution(X_{new})
9. $p(X_{new} \rightarrow X_{cur})$ = Calculate_Probability($T_{cur}, C^F_{cur}, C^F_{new}$)
10. **with** probability $p(X_{new} \rightarrow X_{cur})$ **do** // Replacements
11. $X_{cur} = X_{new}$
12. $C^F_{cur} = C^F_{new}$
13. $m = m + 1$
14. **if** $C^F_{cur} > C^F$ **then do** // Save the best solution
15. $X = X_{cur}$
16. $C^F = C^F_{cur}$
17. $n = n + 1$
18. $T_{cur} = \gamma \cdot T_{cur}$
19. **return** C^F, X

Fig. 7. Algorithm for the deployment of waypoints

range by adding a random value obtained using the pseudo-random number generator with the normal distribution with the zero mean and the standard deviation σ calculated according to formula (11). *Range* is the difference between the maximum and minimum limits for the selected variable (for x and y it is determined by the size of the area of interest, for h it is the difference between the maximum and minimum permitted heights h_{max} and h_{min}). The current temperature influences the size of the change; the bigger the temperature, the bigger the change. This results in the extensive search in the state space at the beginning of the optimization, and tuning the solution towards its end.

$$\sigma = \frac{(T_{cur} - T_{min}) \cdot \left(\frac{range}{3}\right)}{T_{max} - T_{min}} \tag{11}$$

The transformation may lead to the violation of the constraint in formula (4), i.e., the requirement on the minimum distance D between all combinations of pairs of sensors (if set). In this case, sensors which do not satisfy the distance condition are disabled, i.e., they do not participate on the surveillance.

The new transformed solution replaces the current solution (lines 10 to 13) with probability $p(X_{new} \rightarrow X_{cur})$ computed according to formula (12) using the Metropolis

criterion (line 9). If the transformed solution is better than the original, it is always replaced. Otherwise, the probability depends on the difference between the two solutions (the lower the difference, the bigger the probability) and the current temperature (the bigger the temperature, the bigger the probability).

$$p(X_{new} \rightarrow X_{cur}) = \begin{cases} 1 & \text{for } C^F_{new} \geq C^F_{cur} \\ e^{-\frac{C^F_{cur} - C^F_{new}}{T_{cur}}} & \text{otherwise} \end{cases} \tag{12}$$

When an iteration ends, the current temperature is decreased by the cooling coefficient $0 < \gamma < 1$ (line 18). The best solution found during the optimization is stored (lines 14 to 16) and returned at the end of the optimization (line 19).

4 Experiments and Results

This section presents the results from experiments carried out using simulations. First, it describes the scenarios used as benchmark surveillance operations in experiments. Then, the results of the algorithm achieved on the benchmark operations are presented. Finally, a discussion about the performance and behavior of the algorithm concludes this section.

4.1 Benchmark Surveillance Operations

Five benchmark operations (labelled e01 to e05) for experiments were created based on the typical scenarios of the military surveillance operations. In simulations, real geographic data were provided by the Military Geographic and Hydrometeorologic Office of the Ministry of Defense of the Czech Republic. Two models were used as follows:

- Digital Elevation Model (DEM): a representation of the terrain surface in the form of a heightmap. In its last version, the distance between neighboring elevations is 2.5 m and the elevation precision is 0.3 m.
- Topographic Digital Data Model (TDDM): a database of topographic and other objects represented by a polygon and parameters (e.g., object height). The layer of buildings in this model was used as a database of obstacles which may cause possible occlusions.

Table 2 shows the parameters (terrain and obstacles) of the physical environment for individual operations. The environment varies for individual operations both in parameters describing the terrain (from flat to very uneven surface) and obstacles (from low density to very high density). The variety of operations can be seen on these examples: scenario e04 is a city center located in a relatively flat terrain with high density of tall buildings and narrow streets; opposite to this, scenario e05 is a mountain area with very uneven terrain and low density of obstacles.

Table 2. Parameters of the physical environment of the area of operations.

Operation	Terrain			Obstacles	
	Minimum elevation	Maximum elevation	Elevation difference	Number of obstacles	Avg. height of obstacles
e01	201 m	245 m	44 m	14	14.6 m
e02	412 m	491 m	79 m	266	6.4 m
e03	236 m	365 m	129 m	533	5.1 m
e04	197 m	244 m	47 m	936	10.6 m
e05	950 m	1,604 m	654 m	8	9.4 m

Table 3 presents the parameters of the surveillance operations: number of UAVs in the swarm at the disposal of the commander of the mission, parameters of their sensors, minimum required distance between deployed sensors, minimum and maximum limits of the flight height, and size of the area of interest to be surveilled. The operations suppose that a large swarm of small UAVs are available (120 in case of scenario e04).

Table 3. Parameters of the surveillance operations.

Operation	Number of UAVs N	Sensors		Sensor space D	Flight height		Area of interest
		α_{fov}	d_{max}		h_{min}	h_{max}	
e01	20	90°	100 m	40 m	50 m	150 m	0.1 km^2
e02	40	90°	160 m	20 m	50 m	200 m	0.7 km^2
e03	60	120°	220 m	50 m	50 m	300 m	2.8 km^2
e04	120	120°	220 m	50 m	50 m	300 m	5.3 km^2
e05	40	120°	600 m	100 m	100 m	600 m	6.8 km^2

For the solution evaluation (see Sect. 3.1), the rasterization step d_{rast} was set for individual benchmark operations as a compromise between the optimization speed and the precision of the coverage estimation. The values of d_{rast} are shown in Table 4 together with the resulting number of points inside the area of interest which needs to be processed when computing the coverage C^F for a particular solution X.

4.2 Simulation Results

The optimizations of the deployment of waypoints were conducted via the proposed algorithm using the computer with the configuration: CPU Intel i7-7700 3.5 GHz (4 cores). The parameters of the algorithm as defined in Table 1 were set empirically as follows: $T_{max} = 10^{-2}$, $T_{min} = 10^{-6}$, $\gamma = 0.9$, $n_{max} = 200$ (for e01, e02, e05)/$n_{max} = 500$ (for e03, e04), $m_{max} = 20$ (for e01, e02, e05)/$m_{max} = 50$ (for e03, e04).

Table 4. Size of the rasterization step selected for the surveillance operations.

Operation	Rasterization step d_{rast}	Number of points
e01	2 m	19,375
e02	3 m	75,442
e03	5 m	110,266
e04	5 m	213,891
e05	10 m	68,050

The optimizations of the individual benchmark operations were conducted independently for different number of sensors necessary for the simultaneous detection: $F = 1$ (Table 5), $F = 2$ (Table 6), and $F = 3$ (Table 7). In total, 50 trials per experiment were performed and the best solution, the mean and the standard deviation are recorded. The last columns in Tables 5 to 7 show the average execution time of the algorithm per optimization.

Table 5. Results achieved by the proposed algorithm for $F = 1$.

Operation	Best solution $C^F_{\%}$	Mean $C^F_{\%}$	Standard deviation	Execution time
e01	100%	100%	0%	28 s
e02	99.99%	99.98%	0.01%	57 s
e03	100%	99.99%	0.01%	341 s
e04	99.45%	99.38%	0.04%	628 s
e05	100%	100%	0%	92 s

Table 6. Results achieved by the proposed algorithm for $F = 2$.

Operation	Best solution $C^F_{\%}$	Mean $C^F_{\%}$	Standard deviation	Execution time
e01	91.85%	89.82%	1.36%	32 s
e02	86.55%	84.15%	1.05%	117 s
e03	84.99%	81.94%	1.83%	764 s
e04	87.54%	86.00%	0.87%	1210 s
e05	100%	99.93%	0.08%	120 s

Table 7. Results achieved by the proposed algorithm for $F = 3$.

Operation	Best solution $C^F_{\%}$	Mean $C^F_{\%}$	Standard deviation	Execution time
e01	69.45%	66.39%	2.53%	32 s
e02	53.66%	51.43%	1.29%	109 s
e03	55.94%	49.58%	4.20%	427 s
e04	56.52%	54.43%	1.02%	929 s
e05	89.42%	86.01%	1.35%	147 s

From the experiments, it is clear that there are enough UAVs to cover the whole *AoI* when $F = 1$. The coverage is over 99% in all cases. Moreover, 100% coverage was achieved in all optimization trials in case of experiments e01 and e05. The situation when $F = 2$ shows the reduction in the coverage (expect for operation e05); the coverage exceeds 80% in all cases Another reduction is obvious when $F = 3$; e.g., the number of UAVs are sufficient to cover only about 50% in case of experiments e02, e03, e04.

Figure 8 illustrates the results of the surveillance operation e01. Figure 8(a) shows the layout of the obstacles and the area of interest which is small and laying in the relatively flat terrain; Fig. 8(b) shows the image of the real physical environment; Fig. 8(c), 8(d) and 8(e) present the best solution for $F = 1, 2, 3$.

4.3 Analysis and Discussion

This section analyses various features and behavior of the proposed algorithm. First, the convergence of the algorithm is examined. Figure 9 shows the example of the progress of an average optimization on task e01 in dependence on the value of coefficient F. The convergence curves show the fast improvement of the solution at the initial phases of the optimization and the tuning of the solution in the final phases. Also, the curves differ depending on the value of F; the reason is the influence of the number of sensors available for the task (there are more than enough sensors to cover the entire *AoI* when $F = 1$, enough to cover about 90% when $F = 2$, and less than 70% when $F = 3$). Another aspect causing the worse solutions at the beginning of the optimizations and slower convergence to some local optimum for bigger values of F is the constraint not allowing waypoints to be too close to one another (see formula (4)). This constraint has the effect especially when $F > 1$; the bigger the value of F, the bigger the effect.

The impact of the constraint forcing the minimum space between waypoints is also apparent when comparing the execution times of the algorithm for different values of F. This comparison is shown in Fig. 10. In some tasks (e02, e03), it took more than twice as long for $F = 2$ compared to $F = 1$. This is caused by the different number of transformations resulting in total number of solution evaluations.

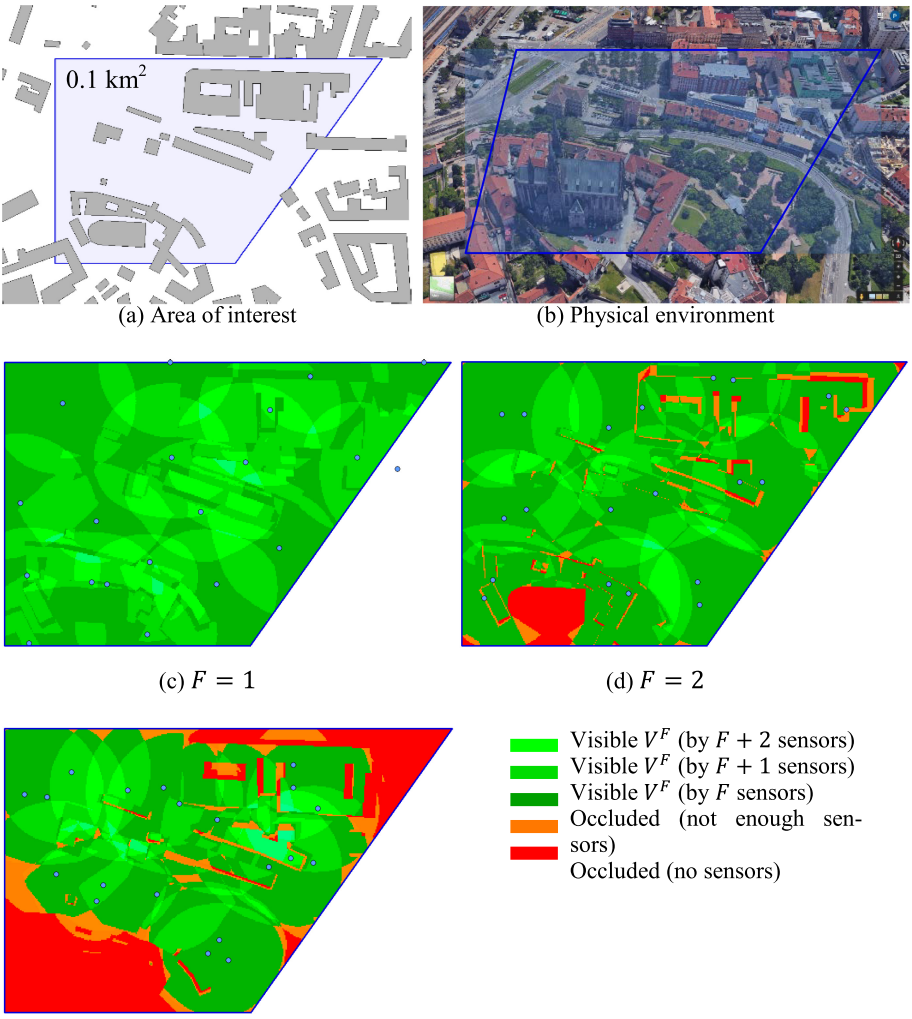


Fig. 8. Experiment e01.

The accumulation of evaluations with the increasing number of iterations is shown on the example of experiment e02 in Fig. 11. In average, the total number of solution evaluations is 6,233 when $F = 1$, 13,140 when $F = 2$, and 11,912 when $F = 3$. The number of evaluations during the later phases of the optimization for $F > 1$ is corresponding to the value of n_{1max} (there are not enough replacements so that an iteration is terminated by exceeding the permitted number of transformations). This is not true for $F = 1$ where it corresponds approximately to half of n_{1max} (an iteration is terminated by exceeding the permitted number of replacements). The reasons for this is the logical tendency to place waypoints close to one another for $F > 1$ (the constraint limiting the

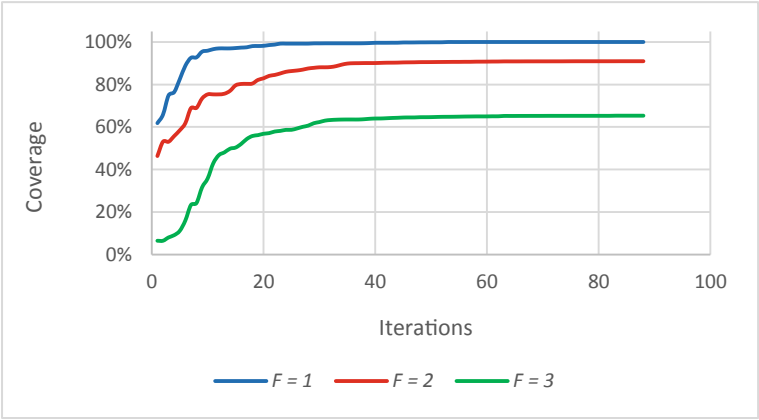


Fig. 9. Optimization progress in an example of experiment e01.

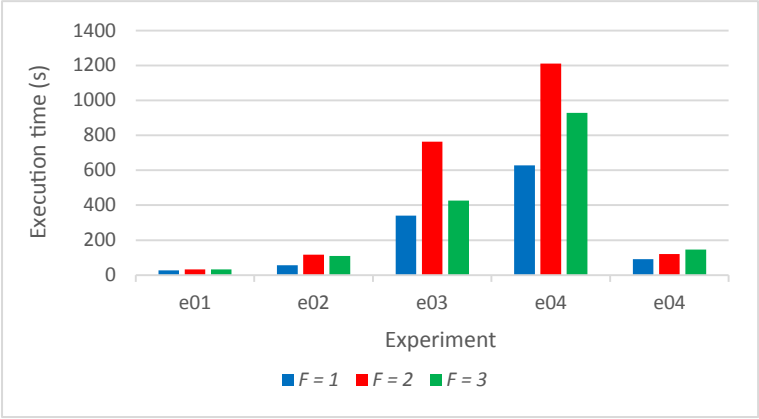


Fig. 10. Execution times of the algorithm.

space between waypoints can be easily violated which leads to disabling those sensors and results in the lower probability to accept the worse solutions); however, this is not the case for $F = 1$. The average time necessary to evaluate a single solution can be calculated; it is about 9 ms in all cases of experiment e02. There are 75,442 points to be evaluated in case of experiment e02 (see Table 4); i.e., more than 8 million points are evaluated per second.

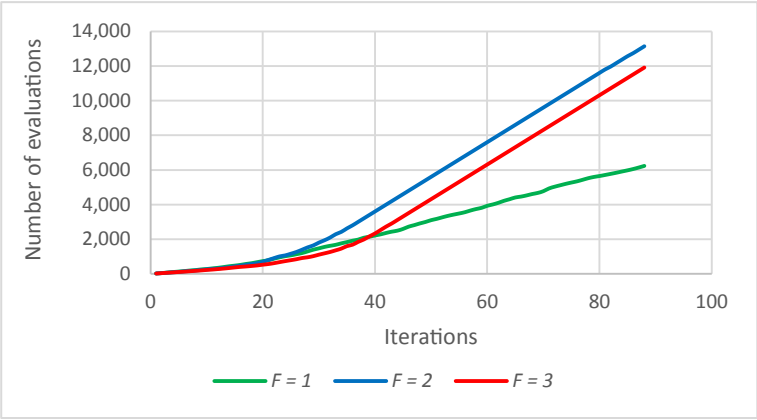


Fig. 11. Accumulation of the number of evaluations in experiment e02.

5 Conclusions

The authors of this article extended their previous research and completed their findings in this area. The particular contributions presented in this article could be summarized in several points as follows:

- The model previously aimed at the aerial reconnaissance using UAVs was modified and transformed to the autonomous aerial surveillance.
- The model was further extended by the new principle called the simultaneous detection. A number of sensors necessary to detect an object can be specified.
- Another new parameter incorporated into the model was the minimum required space between all pairs of sensors.
- Terrain and obstacles, which may occlude the objects of interest, were taken into consideration as well as the parameters of the sensors used.
- The metaheuristic algorithm based on the simulated annealing principles was proposed for deploying waypoints so that as large of the area of interest as possible is monitored by available UAVs in the swarm. New parameters connected with the simultaneous detection were incorporated into the algorithm.
- A set of scenarios was designed for verification. They use the real geographic data and are based on the parameters and features of the real typical surveillance operations.
- The simulation results of the experiments are discussed and some of the parameters of the algorithm are analyzed.

The choice of the metaheuristic method based on the simulated annealing has been supported by the experience of the authors obtained in their previous research. In general, metaheuristic methods are problem dependent; this means that some family of algorithms may be very successful in solving some type of a problem; however, they do not provide good results for another different type of a problem.

The future work of the authors will be focused on implementation of different metaheuristic methods for the solution of the problem examined in this article. Some of

the basic principles which will be considered for implementation are as follows: Particle Swarm Optimization (PSO), genetic algorithms, tabu search, and bio-inspired algorithms (Ant Colony Optimization, Artificial Bee Colony, Bat algorithm). Also, the possibility of their hybridization will be investigated.

References

1. Bruzzone, A.G., Massei, M., Di Matteo, R., Kutej, L.: Introducing intelligence and autonomy into industrial robots to address operations into dangerous area. In: Mazal, J. (ed.) MESAS 2018. LNCS, vol. 11472, pp. 433–444. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-14984-0_32
2. Hodický, J., Prochazka, D., Prochazka, J.: Training with and of autonomous system – modelling and simulation approach. In: Mazal, J. (ed.) MESAS 2017. LNCS, vol. 10756, pp. 383–391. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-76072-8_27
3. Nohel, J., Flasar, Z.: Maneuver control system CZ. In: Mazal, J., Fagiolini, A., Vasik, P. (eds.) MESAS 2019. LNCS, vol. 11995, pp. 379–388. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43890-6_31
4. Otřísal, P., et al.: Testing methods of assessment for the chemical resistance of insulating materials against the effect of selected acids. Materiale Plastice **55**(4), 545–551 (2018)
5. Šilinger, K., Blaha, M.: Conversions of METB3 meteorological messages into the METEO11 format. In: International Conference on Military Technologies, Brno, pp. 278–284 (2017)
6. Stodola, P., Mazal, J.: Tactical decision support system to aid commanders in their decision-making. In: Hodicky, J. (ed.) MESAS 2016. LNCS, vol. 9991, pp. 396–406. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47605-6_32
7. Stodola, P., Nohel, J., Mazal, J.: Model of optimal maneuver used in tactical decision support system. In: International Conference on Methods and Models in Automation and Robotics, Miedzyzdroje, pp. 1240–1245(2016)
8. Yao, P., Wang, H., Zikang, S.: Cooperative path planning with applications to target tracking and obstacle avoidance for multi-UAVs. Aerosp. Sci. Technol. **54**, 10–22 (2016)
9. Lamont, G.B., Slear, J.N., Melendez, K.: UAV swarm mission planning and routing using multi-objective evolutionary algorithms. In: IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, Honolulu, pp. 10–20 (2007)
10. Shanmugavel, M., Tsourdos, A., Zbikowski, R., White, B.A.: 3D path planning for multiple UAVs using pythagorean hodograph curves. In: AIAA Guidance, Navigation, and Control Conference, Hilton Head, USA, pp. 1576–1589 (2007)
11. Triharminto, H.H., Adji, T.B., Setiawan, N.A.: Dynamic UAV path planning for moving target intercept in 3D. In: International Conference on Instrumentation Control and Automation, Bandung, pp. 157–161 (2011)
12. Sampedro, C., et al.: A flexible and dynamic mission planning architecture for UAV swarm coordination. In: International Conference on Unmanned Aircraft Systems, Arlington, USA, pp. 355–363 (2016)
13. Sujit, P.B., Beard, R.: Multiple UAV path planning using anytime algorithms. In: American Control Conference, St. Louis, pp. 2978–2983 (2009)
14. Stodola, P., Drozd, J., Šilinger, K., Hodický, J., Procházka, D.: Collective perception using UAVs: autonomous aerial reconnaissance in a complex urban environment. Sensors **20**(10), 2926 (2020)
15. Stodola, P., Drozd, J., Nohel, J., Hodický, J., Procházka, D.: Trajectory optimization in a cooperative aerial reconnaissance model. Sensors **19**(12), 2823 (2019)
16. Stodola, P., Mazal, J.: Model of optimal cooperative reconnaissance and its solution using metaheuristic methods. Defence Sci. J. **67**(5), 529–535 (2017)