

On the Covering of a Polygonal Region with Fixed Size Rectangles with an Application towards Aerial Inspection

Sina Sharif Mansouri, George Georgoulas, Thomas Gustafsson and George Nikolakopoulos

Abstract—Unmanned Aerial Vehicles (UAVs) equipped with remote visual sensing can be used in wide range of applications. However, guaranteeing the full coverage of the area and translating this coverage in a path planning problem, it is a quite challenging task. Thus, in this article a well-known and well-investigated family of hard optimization problems, covering a polygonal region (target area) with fixed size rectangles (camera frustrum), is studied. The problem is formulated mathematically and solved using metaheuristic optimization algorithms. The proposed novel algorithmic scheme requires an a priori 2D model of the target area, while it tries to maximize the coverage with a minimum number of fixed size rectangles. Finally, multiple simulation results are presented that prove the efficacy of the proposed scheme.

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) equipped with remote sensing instrumentation are emerging in the last years due to their mechanical simplicity, agility, stability and outstanding autonomy in performing complex manoeuvres etc. [1], [2]. Furthermore, UAVs have the ability to offer numerous opportunities in a variety of applications, such as as mapping [3], landslides [4], search and rescue missions [5], and forest fire inspection [6]. One of the most common remote sensor is the visual sensor, in monocular or stereo versions, while the acquired set of images from the UAV's mission can be analyzed and used to produce sparse or dense surface models, hazard maps, investigate access issues, and other area characteristics [7]. However, the main problem in these approaches is to guarantee the full coverage of the area, a fundamental problem that is directly related to the autonomous path planning of the aerial vehicles.

In the presented approach, it is assumed that during the UAV operation, the later has the ability to retain a closed loop fixed altitude with the downside camera, without loss of generality. In this specific case, the camera frustrum can be modeled by fixed size rectangles, of a $w_i \times h_i \in \mathbb{R}^{2+}$, as it is indicated in Figure 1.

In the case that the altitude of the UAV is varying, then this problem is transformed to the problem of on-line covering a predefined region with rectangles of varying size, a problem that is not addressed in previous works based on the authors' best knowledge. In the related literature, the problem of covering the target area with fixed size rectangles has

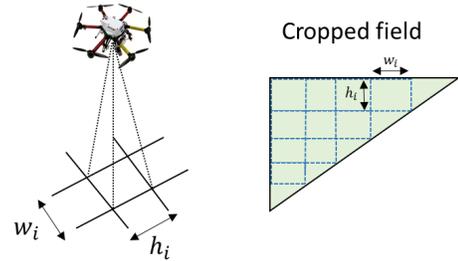


Fig. 1: Schematic of the field of view in the case of UAV.

been addressed multiple times, while this typical problem is known to be NP-complete [8], [9], [10] and one of the several computationally difficult decomposition problems [11]. Additionally, it has a number of important practical applications except from the coverage approaches for visual inspection, such as VLSI layout, pattern recognition, computer graphics, databases, image processing, etc. [12]. Thus, inspired by this vision, the main objective of this article is to establish an algorithm to segment the target area in relation to the camera's field of view in an off-line approach. Based on this partition, the UAV in the sequel will be able to plan its path by e.g. following the center of gravity of the rectangles in the partitioned space.

On the specific topic of covering polygons with rectangles, most of the works considered a varying size of the rectangles' area [11], [13], [14] or the target area is considered to be convex, rectilinear or a union of convex polygons [14]. Additionally, most of the previous contributions formulate the problem mathematically, while the numerical solution to the problem is still lacking, while it should be highlighted that the problem is proven to be NP-hard for the fixed size rectangles [10]. Likewise, in the scope of aerial coverage and visual inspection, it is assumed that a top level procedure handles the area segmentation, and there is no work to consider decomposition of the target area with the relation to the coverage task. Thus, this decoupling of coverage task and segmentation of the area may reduce the generality of the approaches.

Based on the aforementioned state of the art, the main contribution of this article is double-fold. Firstly, to mathematically formulate the problem and approach the NP-complete problem of covering a polygonal regions with fixed sized rectangles by different metaheuristic techniques such as: pattern search, genetic algorithm and particle swarm optimization. Secondly, this paper addresses the problem of the covering a polygonal region with fixed size rectangles for

The authors are with the Robotics Team, Control Engineering Group, Department of Computer, Electrical and Space Engineering, Luleå University of Technology, Luleå SE-97187, Sweden. Emails: {sinsha, geogeo, thomas.gustafsson, geonik}@ltu.se

This work has received funding from the European Unions Horizon 2020 Research and Innovation Programme under the Grant Agreement No.644128, AEROWORKS.

the coverage and inspection tasks for the first time based on the author's best of knowledge. Finally, it should be noted that the representation can maximize the coverage of the target area while the rectangles are representing the camera frustum in 2D. The method is evaluated with multiple case studies.

The rest of the article is structured as follows. The mathematical establishment of the proposed problem is presented in Section II, followed by the presentation of the selected algorithms for solving the problem in Section III. In Section IV multiple simulation results are presented with a corresponding comparison and discussion on them. Finally the article concludes in Section V.

II. PROBLEM STATEMENT

Let $\Omega \subset \mathbb{R}^2$ be a given region that can have multi-connected components, while the finite set

$$\Lambda = \{R_i : i \in I_n = \{1, 2, \dots, n\}\} \quad (1)$$

of rectangles

$$R_i = \{(x_i, y_i) \in \mathbb{R}^2 : \frac{-w_i}{2} \leq x_i \leq \frac{w_i}{2}, \frac{-h_i}{2} \leq y_i \leq \frac{h_i}{2}\} \quad (2)$$

is considered. Each rectangle can be identified by its position in the lower left corner (x_i, y_i) and the width w_i and the length h_i . The placement of the rectangles R_i is defined by the translation vector $u_i = (x_i, y_i)$, $i \in I_n$, while the set of translated rectangles $R_i(u_i)$ is expressed by $\Lambda(u)$, where $u = \{u_1, u_2, \dots, u_n\} \in \mathbb{R}^{2n}$. The polygonal

$$P(u_i, n) = \bigcup_{i=1}^n R_i(u_i) \quad (3)$$

represents the region covered by the union of the rectangles R_i , while Λ^* is a cover of Ω if there exist a solution such that

$$\Omega \subset P(u_i, n) = \bigcup_{i \in I_n} R_i(u_i) \quad (4)$$

Therefore, the search is to find the position of each rectangle through the translation vector u_i and the total number of rectangles n . Thus, the goal is to define whether there exists a vector $u \in \mathbb{R}^{2n}$, such that $\Lambda(u)$ is a cover of Ω . The general schematic of the problem is presented in Figure 2.

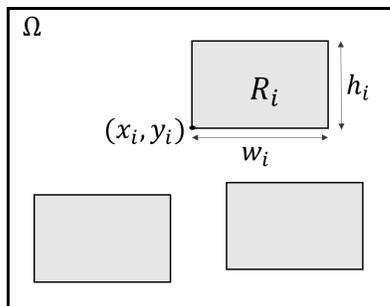


Fig. 2: Schematic of the problem statement.

In order to mathematically formulate the described problem, the following optimization problem is defined:

$$\begin{aligned} \min_{u, n} & \underbrace{|A(P(u_i, n) \cap \Omega) - A(\Omega)|}_{\text{maximizing coverage}} + \underbrace{\alpha \sum_i^n w_i h_i}_{\text{minimizing \# rectangles}} \\ \text{s.t.} & \\ & u_l \leq u \leq u_u \\ & P(u_i, n) = \bigcup_{i=1}^n R_i(u_i) \\ & n \in \mathbb{N} \end{aligned} \quad (5)$$

where A is the area of polygons, u_u and u_l are the upper and lower bounds of the translation vector that is defined by the boundaries of the target area Ω , α is the weight for the number of rectangles in the objective function, $w_i h_i$ is the area of each rectangle and n is the number of rectangles that is member of a natural number \mathbb{N} . In general, the proposed problem statement is considered to be an NP-hard problem thus in the sequel metaheuristic optimization algorithms will be utilized for the solution.

III. TOOLS FOR SOLVING THE PROBLEM

When it comes to solving hard optimization problems, either discrete NP-hard problems or continuous non-convex with many local minima, a very common solution involves the use of metaheuristic optimization algorithms. These algorithms even though do not guarantee optimality, they usually lead to “good enough” solutions, within reasonable amount of time. Therefore they have attracted lot of attention with new algorithms proposed every day. However most of them are variations of well established themes.

In this article, representative members of three characteristic families of metaheuristic optimization, are investigated and compared for their utility for this specific problem, namely: the Pattern Search (PS), the Genetic Algorithm (GA) and the Particle Swarm Optimization (PSO). PS is one of the oldest metaheuristic methods that does not rely on the use a population of solutions, rather it evolves a single one using a systematic approach, and it was selected to serve as the base comparison method. GA and PSO both belong to the family of population based methods, which evolve a number of solutions simultaneously. GA is probably the most used one method from the family of evolutionary methods that relies on the concept of combining and altering potential solutions using operations that were inspired from actual genetic operations, with numerous applications in engineering applications. Nowadays, the GA has become the benchmark algorithm to compare against any new proposed metaheuristic method. PSO is the newest of the three investigated methods. It is also a population based method, relying on a completely different approach than the GA. The popularity of PSO stems from the fact that apart from its efficiency, it is relatively easy to be implemented.

A. Pattern Search

PS [15] is a family of numerical optimization methods that do not require the gradient of a function. Lets consider the following optimization setting:

$$\min_{u \in U} f(u) \quad (6)$$

where $f : U \rightarrow \mathbb{R} \cup \{\inf\}$, $U = \{u \in \mathbb{R}^{2n} : U_l \leq Au \leq u_u\}$, $A \in \mathbb{R}^{m \times n}$ is a real matrix, $u_l, u_u \in (\mathbb{R} \cup \pm \inf)^n$, and $l < u$. A PS is a direction-based method, where convergence theory is based on searching in directions that form a positive spanning set. Thus, a finite set of positive spanning directions D is used to construct a mesh M_k . The mesh M_k is defined as:

$$M_k = \{u_k + \Delta D z \in U : z \in \mathbb{Z}_+^{|D|}\} \quad (7)$$

where u_k is the current iterate, $\Delta_k > 0$ is a parameter for fitness of the mesh, and \mathbb{Z}_+ is the set of nonnegative integers. The Algorithm for PS is presented in 1. Moreover a full reference on PS formulations and algorithms can be found in [16].

Algorithm 1 Pattern Search

Require u_0

- 1) Calculate $M_0 \subset U$ by $\Delta_0 > 0$ and u_0
 - 2) Repeat for $k = 0, 1, 2, \dots, k_{max}$
 - a) *Search Step* seeking for an improved mesh point ($f(u_{k+1}) < f(u_k)$)
 - b) *Poll Step* if $f(u_{k+1}) > f(u_k)$, improve mesh point u_{k+1}
 - c) *Update* if both (a) and (b) steps failed update u_{k+1} and set $\Delta_{k+1} \geq \Delta_k$ according to $\Delta_{k+1} = \tau^{m_k^+} \Delta_k$ else, set $u_{k+1} = u_k$ and set $\Delta_{k+1} < \Delta_k$ according to $\Delta_{k+1} = \tau^{m_k^-} \Delta_k$
-

B. Genetic Algorithm

The GA [17] is an optimization and search technique based on the principles of genetics and natural selection. Overall, the algorithm is presented in 2.

Algorithm 2 Genetic Algorithm

- 1) randomly initialize population u
 - 2) repeat until best individual is good enough or maximum number of generations has been reached
 - a) **Fitness** determine fitness of population $f(u)$
 - b) **Selection** select parents from population according to their fitness
 - c) **Crossover** perform crossover on parents creating population
 - d) **Mutation** With a mutation probability mutate new offspring
 - e) **Accepting** Place new offspring in a new population
-

C. Particle Swarm Optimization

PSO is probably the most prominent member of the swarm intelligence family [18]. The Basic PSO algorithm consists of the velocity and position equation:

$$\begin{aligned} v_i(t+1) &= \phi(t)v_i(t) + \eta_1 r[p_i - u_i(t)] + \eta_2 r[p_{g(t)} - x_i(t)] \\ u_i(t+1) &= u(t) + v_i(t) \end{aligned} \quad (8)$$

where i is the particle index, $v_i(t)$ is the current velocity of the i^{th} particle, $\phi(t)$ is an inertia function (usually a linearly decreasing one), $u_i(t)$ is the current position of the i^{th} particle, p_i is the position with the best fitness value, visited by the i^{th} particle, $g(t)$ is the particle with the best fitness among all particles (best position found so far - global version of the PSO [19]), r is a positive constant called acceleration constant and η_1, η_2 are random numbers uniformly distributed in $[0, 1]$. Overall, the algorithm is presented in 3.

Algorithm 3 Particle Swarm Optimization

- 1) randomly initialize population u
 - 2) repeat until best individual is good enough or maximum number of iterations has been reached
 - a) **Velocity** update velocities of all particles
 - b) **Position** update positions of all particles
 - c) **Best** perform crossover and global best positions and values
-

IV. SIMULATION RESULTS

In this Section different scenarios are presented in order to evaluate the performance of the proposed method and for comparison of the three different metaheuristic algorithms adopted. The initial positions of the rectangles are generated randomly, while the initial value for the number of rectangles is $\lceil A(\Omega)/A(R) \rceil$. Moreover, it should be highlighted that the number of rectangles is inversely related to the term α in the objective function. A larger value of α results in a lower value for the number of rectangles. Furthermore, from the obtained result, in each scenario, the path for the UAV can be obtained as the one that passes through the center of rectangles. In all cases the algorithm stops if the average relative change in the best fitness function value is less than threshold and for illustration purposes the value of first term of the objective function is shown in the following. Due to the stochastic nature of the algorithms, each experiment was repeated ten times.

In the first scenario, the target area is considered to be a rectangle with a size of 20×10 that can be filled by 8 rectangles with a fixed size of 5×5 . Figure 3 presents the obtained results and the corresponding path is depicted for the inspection of the area. Additionally, Figures 3 and 5 present the value and the average of the first term of the objective function respectively.

In the second scenario, the non-convex target area is covered with four fixed size rectangles. The corresponding results from this optimization are presented in Figure 6,

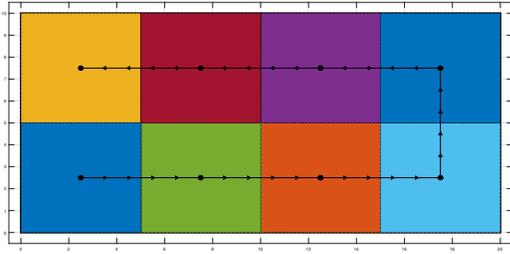


Fig. 3: Covering a convex polygons with 8 rectangles.

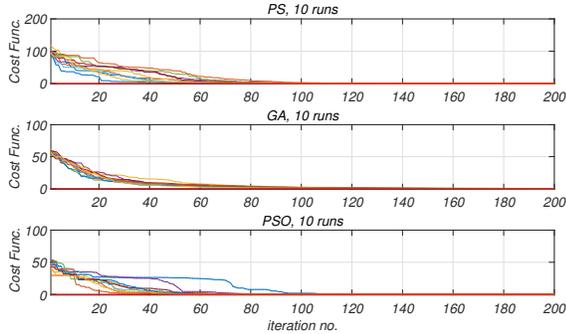


Fig. 4: Objective function value of maximizing coverage of each method for 10 run in first case.

extended with an indicative example of the path that can be followed by the UAV. Furthermore, the value and average of the first term of the objective function in 10 runs is depicted in Figures 7 and 8. In this case all three investigated methods managed to cover the whole area. Moreover, it should be additionally mentioned that different initial positions, for each algorithm, resulted in different convergence rate of the objective function while this does not effect the final solution.

In the third scenario, the octagonal polygon with an area of 181 is covered by 9 rectangles. The result of the coverage and the value and average of the first part of the objective function are depicted in Figures 9, 10 and 11 respectively. Likewise, a path through the center of the rectangles is

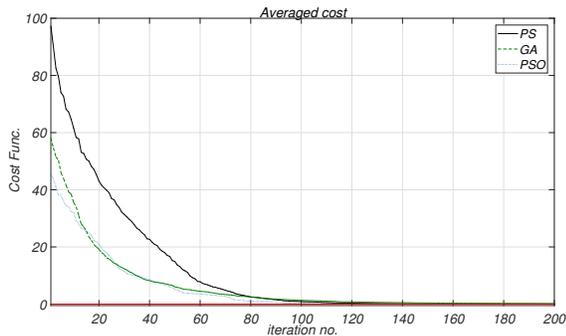


Fig. 5: Average objective function of maximizing coverage for each metaheuristic method in first scenario.

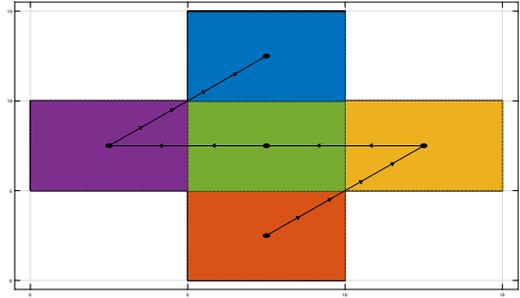


Fig. 6: Covering a non-convex area with 4 rectangles.

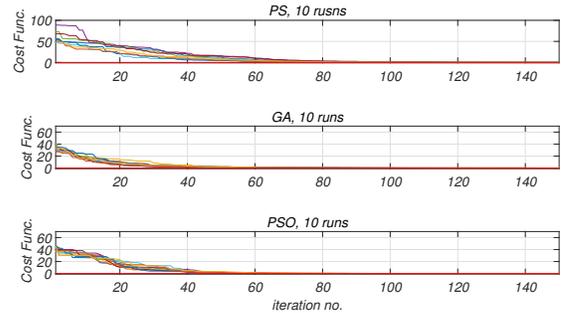


Fig. 7: Objective function value of maximizing coverage of each method for 10 runs in the second case.

obtained.

Furthermore, Figures 12, 13 and 14 present the final placement and the value and average of the first term of the objective function respectively. In this case a complex non-convex polygon is considered and it is filled by 8 rectangles.

Table I summarizes the mean percentage of coverage of the target area. As it can be seen the coverage is higher than 96 % for all four scenarios and all three metaheuristic methods.

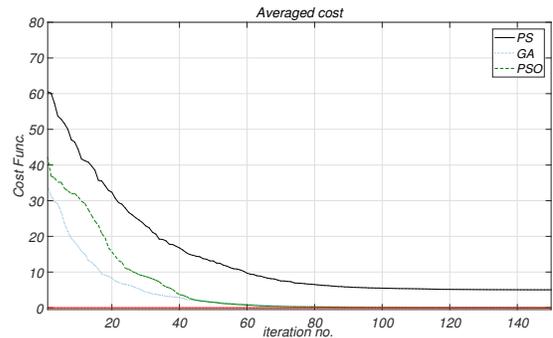


Fig. 8: Average of objective function value of maximizing coverage for each metaheuristic method in second scenario.

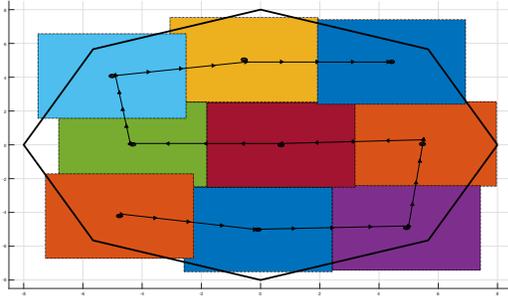


Fig. 9: Covering an octagonal polygons with 9 rectangles .

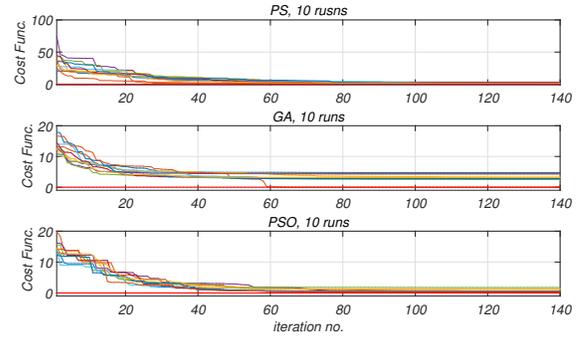


Fig. 13: Objective function value of maximizing coverage of each method for 10 runs in the fourth case.

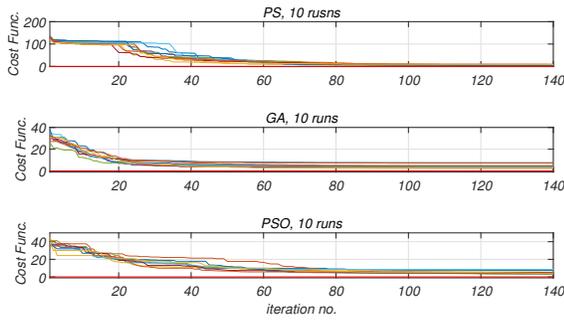


Fig. 10: Objective function value of maximizing coverage of each method for 10 runs in the third case.

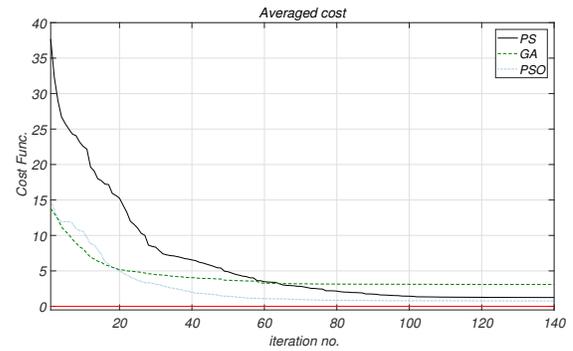


Fig. 14: Average of objective function value of maximizing coverage for each metaheuristic in the fourth case.

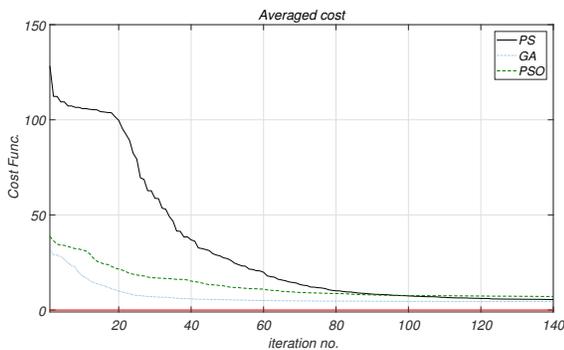


Fig. 11: Average of Objective function value of maximizing coverage value for each metaheuristic in the third case.

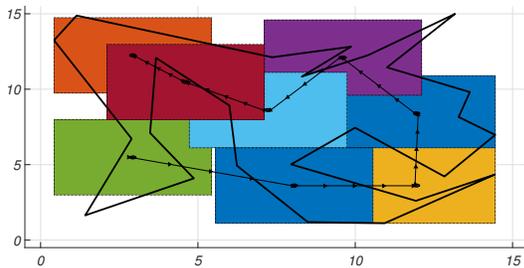


Fig. 12: Covering a complex polygon with 8 rectangles.

V. DISCUSSION AND CONCLUSIONS

This article addressed the NP-hard problem of coverage of polygons with fixed size rectangles. Initially, the problem is formulated mathematically and it is solved by different metaheuristic algorithms using an approximation approach. The presented methods have been tested in different convex and non-convex polygons and in all the examined scenarios, a very satisfactory coverage of a level above 96 % has been obtained. Total coverage could be achieved using more conservative approaches (using more rectangles). For the specific scenarios no single method seems to overwhelmingly outperform all others. This, on one hand, shows that the metaheuristic approaches can provide good solutions in general while, on the other hand suggests that more experiments should be conducted before safer conclusions can be reached.

Finally, from the obtained results, the corresponding in-

TABLE I: Coverage percentage for the different scenarios and methods.

Methods	Cases			
	1	2	3	4
Pattern Search	99.9%	99.9%	96.3%	98.6%
Genetic Algorithm	99.9%	99.9%	97.5%	97.7%
Particle Swarm Optimization	99.9%	99.9%	97.0%	99.0%

dicative path that can be assigned to the aerial vehicles for remote visual sensing is generated. However, this specific path generation problem needs more investigation as depicted in Figure 15, there are multiple choices for the resulting path based on the considered dynamics of the UAV and the corresponding constraints, e.g. minimum energy consumption, minimum time coverage, etc. The future work is related towards the investigation of the effect of changing the UAV's altitude, which can cause time varying areas of camera frustum in an on-line approach, as well as the extended experimentation of the overall suggested scheme.

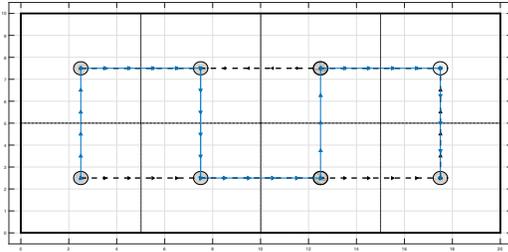


Fig. 15: Multiple path generation.

REFERENCES

- [1] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *Journal of Field Robotics*, vol. 29, no. 2, pp. 315–378, 2012.
- [2] C. Kanellakis and G. Nikolakopoulos, "Survey on computer vision for uavs: Current developments and trends," *Journal of Intelligent & Robotic Systems*, pp. 1–28, 2017.
- [3] L. Zongjian, "Uav for mapping low altitude photogrammetric survey," *International Archives of Photogrammetry and Remote Sensing, Beijing, China*, vol. 37, pp. 1183–1186, 2008.
- [4] U. Niethammer, S. Rothmund, M. James, J. Travelletti, and M. Joswig, "Uav-based remote sensing of landslides," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, no. Part 5, pp. 496–501, 2010.
- [5] P. Doherty and P. Rudol, "A uav search and rescue scenario with human body detection and geolocalization," in *AI 2007: Advances in Artificial Intelligence*. Springer, 2007, pp. 1–13.
- [6] K. Alexis, G. Nikolakopoulos, A. Tzes, and L. Dritsas, "Coordination of helicopter UAVs for aerial Forest-Fire surveillance," in *Applications of Intelligent Control to Engineering Systems*. Springer Netherlands, June 2009, pp. 169–193.
- [7] J. Valente, A. Barrientos, J. del Cerro, C. Rossi, J. Colorado, D. Sanz, and M. Garzón, "Multi-robot visual coverage path planning: Geometrical metamorphosis of the workspace through raster graphics based approaches," in *International Conference on Computational Science and Its Applications*. Springer, 2011, pp. 58–73.
- [8] L. Heinrich-Litan and M. E. Lübbecke, "Rectangle covers revisited computationally," in *International Workshop on Experimental and Efficient Algorithms*. Springer, 2005, pp. 55–66.
- [9] K. Daniels and R. Inkulu, "An incremental algorithm for translational polygon covering," *University of Massachusetts at Lowell Computer Science Technical Report*, no. 1, pp. 1–31, 2001.
- [10] J. C. Culberson and R. A. Reckhow, "Covering polygons is hard," in *Foundations of Computer Science, 1988., 29th Annual Symposium on*. IEEE, 1988, pp. 601–611.
- [11] D. S. Franzblau and D. J. Kleitman, "An algorithm for covering polygons with rectangles," *Information and Control*, vol. 63, no. 3, pp. 164–189, 1984.
- [12] S.-Y. Wu and S. Sahni, "Fast algorithms to partition simple rectilinear polygons," *VLSI Design*, vol. 1, no. 3, pp. 193–215, 1994.
- [13] V. Anil Kumar and H. Ramesh, "Covering rectilinear polygons with axis-parallel rectangles," in *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. ACM, 1999, pp. 445–454.

- [14] Y. G. Stoyan, T. Romanova, G. Scheithauer, and A. Krivulya, "Covering a polygonal region by rectangles," *Computational Optimization and Applications*, vol. 48, no. 3, pp. 675–695, 2011.
- [15] R. Hooke and T. A. Jeeves, "“direct search” solution of numerical and statistical problems," *Journal of the ACM (JACM)*, vol. 8, no. 2, pp. 212–229, 1961.
- [16] M. A. Abramson, "Pattern search algorithms for mixed variable general constrained optimization problems," Ph.D. dissertation, Ecole Polytechnique de Montréal, 2002.
- [17] L. Davis, "Handbook of genetic algorithms," 1991.
- [18] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on*. IEEE, 1995, pp. 39–43.
- [19] J. F. Kennedy, J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm intelligence*. Morgan Kaufmann, 2001.