

Multiobjective evolutionary flight planning of autonomous unmanned aerial vehicles for exploration and surveillance

Sergio Nesmachnow¹, Claudio Paz²,
Jamal Toutouh³ and Andrei Tchernykh⁴

¹ Universidad de la República, Uruguay

² Universidad Tecnológica Nacional, Argentina

³ Massachusetts Institute of Technology, USA

⁴ CICESE, Ensenada, Baja California, México

E-mail: sergion@fing.edu.uy, cpaz@frc.utn.edu.ar, toutouh@mit.edu,
chernykh@cicese.mx

Abstract. This article presents a multiobjective evolutionary approach for computing flight plans for a fleet of unmanned aerial vehicles to perform exploration and surveillance missions. The static off-line planning subproblem is addressed, which is useful to determine initial flight routes to maximize the explored area and the surveillance of points of interest in the zone. A specific flight planning solution is developed, to be applied in low-cost commercial Bebop 2. The experimental analysis is performed in realistic instances of the surveillance problem. Results indicate that the proposed multiobjective evolutionary algorithm is able to compute accurate flight plans, significantly outperforming a previous evolutionary method applying the linear aggregation approach.

1. Introduction

In the last years, Unmanned Aerial Vehicles (UAVs) have developed as useful vehicles to perform different important activities and provide diverse services in smart cities [1]. A wide range of applications have been addressed in logistics and infrastructure inspection, agriculture, photography, rescue and disaster management, and also security and surveillance [2]. Since UAVs can be controlled remotely or programmed to follow specific flight routes, they can be used in situations where manned flight is dangerous. Furthermore, for routine missions, an autonomous flight system can be implemented, without involving human control or a centralized control infrastructure [3]. This paradigm is commonly applied for controlling a fleet of UAVs, by implementing a cooperative approach for surveillance, which provides several benefits: robustness, since multiple agents perform the task at the same time, improvement on the flight time and battery utilization, and improvement of the quality of service of the system (providing an expanded coverage and better surveillance).

Developing an effective cooperative model for exploration and surveillance heavily relies on successfully computing accurate flight plans that account for the problem objectives. The flight route planning problem for UAVs is NP-hard, as it is a variant of the classic Orienteering Problem [4]. Computational intelligence methods have been proposed to generate reliable and fast flight routes for UAV fleets, to fulfill specific missions autonomously [5]. In this context, heuristics and metaheuristics [6] are applied to find accurate routes in reasonable execution times, especially to be implemented in real time missions.

In this line of work, this article presents a Multiobjective Evolutionary Algorithm (MOEA) to solve the off-line problem of route planning for a fleet of UAVs to maximize exploration and surveillance. The problem formulation considers an area to explore and specific points of interest (PoI) to be surveyed periodically. PoI can be static or move across the area, following a pre-determined pattern. The main goal of the proposed MOEA is to compute accurate plans with good compromise between the explored area and the monitoring of PoI. Computed solutions are useful for practical surveillance and can also be extended to address more dynamic situations, e.g. by applying agent oriented programming to perform slight modifications to the route of each UAV to deal with unexpected events.

2. Multiobjective flight planning for UAVs

The considered optimization problem proposes finding a set of routes for UAVs in the fleet to simultaneously maximize the explored area and maximize the surveillance of PoI.

The problem formulation considers the following elements:

- A set of UAVs, $U = \{u_1, \dots, u_{|U|}\}$, able to flight at a maximum speed v_D .
- A mission time period T , discretized in s uniform time steps; $T = \langle t_1, t_2, \dots, t_s \rangle$
- A set of PoI to surveil $O = \{o_1, \dots, o_{|O|}\}$. Two types of PoI are considered: static (do not move) and mobile (move at maximum speed v_O).
- A benefit vector $P = (p_1, p_2, \dots, p_{|O|})$, where p_i is the benefit associated to surveil PoI o_i .
- An object position matrix OP (dimension $|O| \times s$). OP_{ij} indicates the coordinates of PoI o_i in timestep t_j .
- A coverage radius \bar{r}_o and a circumscribed coverage square with side $r_o = 2\bar{r}_o/\sqrt{2}$.
- An area to explore (dimensions $H \times W$), which is considered to be discretized in regions, i.e., squares of length r_o to determine the sequence of flight paths.

The goal of the problem is to determine a flight planning for the fleet of UAVs, i.e., a function $fp : U \times T \rightarrow Q$ that simultaneously maximizes two functions:

- The benefit of monitoring PoIs (as defined by function $\delta(p)$, in Eq. 2). This function takes into account PoIs that are covered, i.e., they are within the coverage radius of each UAV.
- The benefit for exploring (as defined by function $\phi(p)$, in Eq. 2), which accounts for the explored surface of the fleet, defined as the union of the surfaces explored by each UAV in its route in the planned time. The area covered by a UAV at a given time is determined by a circumference of coverage radius that has it as its center, which is directly related to the field view of the UAV on-board camera. The benefit for exploring is evaluated by the Spatial Exploration Ratio metric [7].

$$\delta(p) = \sum_{i=1}^{|U|} \sum_{j=1}^{|T|} \sum_{z=1}^{|O|} D(OP(u_i, t_j), OP(o_z, t_j)) \times p_z, \quad (1)$$

$$\text{with } p(q_i, t_j) = (x, y) / a_{x,y}^j = q_i \text{ and } D(c_1, c_2) = \begin{cases} 1 & \text{if } \text{distance}(c_1, c_2) \leq \bar{r}_o \\ 0 & \text{otherwise} \end{cases}$$

$$\phi(p) = \sum_{x=1}^{\lceil \frac{H}{r_o} \rceil} \sum_{y=1}^{\lceil \frac{W}{r_o} \rceil} \sum_{j=1}^{|T|} E(x, y, j) \times \frac{1}{r_o^2}, \text{ with } E(x, y, t_j) = \begin{cases} 1 & \text{if } a_{x,y}^j \in U \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Without loss of generality, the problem model assumes that all UAVs depart and return for charging to a *base* B , located at coordinates (x_B, y_B) , within the area to explore.

3. The proposed MOEA for flight planning

This section describes the proposed algorithmic approach for multiobjective UAV flight planning.

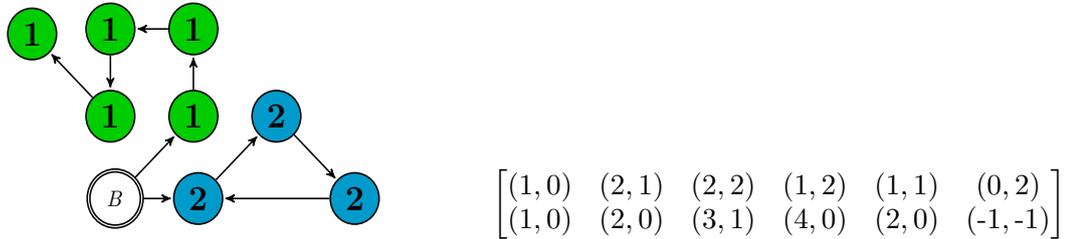
3.1. Algorithmic approach

The proposed solution for multiobjective UAV flight planning is based on the NSGA-II algorithm, a traditional MOEA for solving real-world problem in different application areas [8]. Specific modifications to the general skeleton of NSGA-II were included to solve the considered problem. The main implementation details are presented in the following subsections.

3.2. Solution encoding

Solutions to the multiobjective UAV flight planning problem are represented by a matrix-based encoding. A graph is built considering the discretization of the area to explore and the zones visited by each UAV. The graph connects the center of each zone, for each UAV route, in a Cartesian coordinate system. In the matrix encoding, each row encodes a route, i.e., the sequence of coordinates of the center of each zone for each route. Each element (i, j) in the matrix is a pair (x_{ij}, y_{ij}) that represents the position of UAV i at timestep j . The null value $(-1, -1)$ is used to fill those unused elements in the matrix, since some routes may be larger than others. The dimension of the matrix encoding is $|U| \times s$.

Figure 1 presents an example of encoding for a simple solution where two routes are defined for two UAVs (represented by the green nodes and the blue nodes in the graph in Figure 1(a)). Both UAVs depart from the base located at $(x_B, y_B) = (1, 0)$.



(a) A sample solution with two routes (b) Matrix encoding (first row: green route; second row: blue route)

Figure 1. Sample solution encoding for a flight planning in an scenario with two UAVs (routes for each UAV are marked by green and blue nodes, respectively).

3.3. Fitness assignment and stopping criterion

The traditional dominance rank method applied in NSGA-II is used to define the fitness of solutions, considering the problem objectives. Then, Pareto ranking is applied over normalized values of both objectives for solutions in the population. A stagnation stopping criterion is applied: the search stops when no new non-dominated solutions are computed in a generation.

3.4. Evolutionary operators

Specific evolutionary operators are applied to account for the problem objectives (exploration and alerts from PolS) [9]. The proposed evolutionary operators are as follows:

- (i) Initialization: Two methods were applied for population initialization: i) random path initialization and ii) seeded initialization, where the route of each UAV is generating by starting at (x_B, y_B) and moving in a direction determined by angle $i \times 2\pi \times 1/u + j \times 2\pi \times 1/(u \times Z)$, where $i \in [0, u - 1]$ is the UAV identifier, $j \in [0, Z]$ is the ordinal of the candidate solution, and Z is the population size of the MOEA.
- (ii) Selection: The classic tournament selection operator in NSGA-II was replaced by Stochastic Universal Sampling with sigma escalation (SUS+ σ) in the proposed MOEA. This decision was motivated by three useful properties of SUS+ σ : i) it is less biased to high quality solutions; ii) helps avoiding premature convergence caused by dominance of a group of solutions; and iii) helps amplifying small differences in advanced stages of the search.
- (iii) Recombination: The Single Point Crossover operator was applied for recombination, as it provided an appropriate search pattern in preliminary configuration experiments. The operator is directly applied over the proposed solution encoding, but it can generate infeasible solutions (e.g., long jumps in UAV routes). In this case, the correction operator described in item (v) is applied.
- (iv) Mutation: A specific mutation operator, based on modifying information on the encoded routes, was designed for the problem. The number of positions to be modified is selected with a uniform distribution in the interval $([1, u \times \frac{t}{s}])$. After that, a new direction is defined for the movement of the corresponding UAV, according to a uniform distribution in the seven different possible directions (excluding the one already traveled in the original solution). The speed of the UAV in this new direction is also selected according to a uniform distribution between $(0, v_D)$. The proposed mutation operator may generate infeasible solutions. In this case, the correction operator described in item (v) is applied.
- (v) Correction of infeasible solutions: Infeasible solutions can be generated by the evolutionary operators in two cases. In the case of the path of a UAV has two consecutive positions that are more distant than it can travel in a time interval, the correction operator modifies the position from which it is not possible to reach the nearest one when the UAV flies at maximum speed. A second type of infeasible solutions are those that do not allow the UAV to return to the base for recharging after finishing a route. In this situation, the correction operator truncates the route in the last movement where it is still possible to return to the base with the available battery charge.

4. Experimental evaluation

This section reports the experimental evaluation of the proposed MOEA for UAV flight planning.

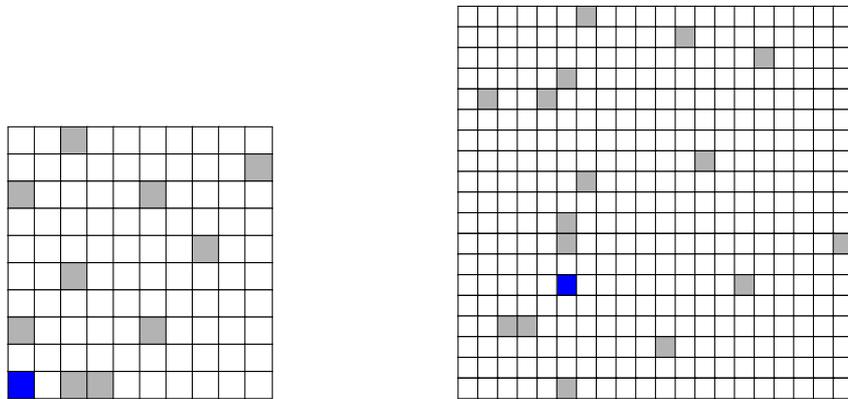
4.1. Scenarios and instances.

The proposed MOEA was evaluated in real and synthetic scenarios that include features of real facilities to perform surveillance. Five scenarios were considered: one small-sized scenario (#0) used for parameter calibration, one real scenario (#1) studied with low-cost Parrot Bebop 2 UAVs, two medium-size and one large-size scenario, whose details are presented in Table 1.

scenario	$H \times W$	(x_B, y_B)	$ U $	v_D	$ T $	s	$ O $	v_O	P	r_o
#0	50×50	(0, 0)	3	5	500	10	3	3	$\langle 1, 2, 3 \rangle$	5
#1	100×100	(30, 30)	3	5	500	10	3	3	$\langle 1, 2, 3 \rangle$	5
#2	1000×1000	(300, 300)	5	10	1000	10	4	5	$\langle 1, 2, 3, 4 \rangle$	5
#3	1000×1000	(700, 700)	10	10	1000	10	4	5	$\langle 2, 2, 8, 8 \rangle$	5
#4	10000×10000	(5000, 5000)	5	10	2000	20	5	0.1	$\langle 1, 1, 1, 1, 10 \rangle$	5

Table 1. Details of the problem instances used for evaluation of the proposed MOEA.

For each synthetic scenario, ten problem instances were created varying the locations of PoI and obstacles. Overall, 50 problem instances were used for the evaluation of the proposed MOEA. Figure 2 presents two sample scenarios and their discretizations. The base is marked with a blue square and obstacles are marked with gray squares. PoIs are located at random location and motion is generated by applying Rapidly-exploring Random Tree, an efficient strategy for multi-dimensional space searching, biased towards unexplored sections of the search space.



(a) Scenario #0 (dimension 50×50) (b) Scenario #1 (dimension 100×100)

Figure 2. Two sample scenarios for the experimental evaluation.

Scenarios #2 to #4 were studied using a distributed simulation approach, implemented over Sphinx, the official simulator for Parrot UAVs. Experiments were executed on Xeon Gold 6138 processors with 128 GB of RAM memory from National Supercomputing Center, Uruguay [10]. Parametric configuration experiments were performed over scenario #0.

4.2. Numerical results

Table 2 reports the results computed by the best compromise solution (i.e., the nearest solution to the ideal vector) of the proposed MOEA. Results are compared with the single-objective Evolutionary Algorithm (EA) for the problem using a linear aggregation of objectives [11]. The best improvements over the reference EA are marked in bold: up to 26.7% for the exploration objective and up to 22.3% for the surveillance objective. The best results were computed for the largest problem instance, suggesting that the proposed MOEA scales properly to large scenarios.

Table 2. Results of the proposed MOEA and improvement over linear aggregation approach [11].

#I	MOEA		Linear aggregation EA [11]		Improvements	
	PoI monitoring (δ)	exploration (ϕ)	PoI monitoring (δ)	exploration (ϕ)	$\Delta\delta$	$\Delta\phi$
0	1338	1996	1106	1755	17.3%	18.0%
1	1754	2907	1377	2495	21.5%	14.2%
2	2101	4221	1583	3669	24.7%	13.1%
3	14549	18921	12021	15021	17.4%	20.6%
4	12175	17416	8921	13533	26.7%	22.3%

Figure 3 presents a Pareto analysis for instances #2 and #4, which are representative of the results obtained in the other instances. Values of the baseline EA are computed by assigning different weights to the problem objectives in the linear aggregation. Results indicate that solutions computed by the MOEA clearly dominate the baseline EA solutions.

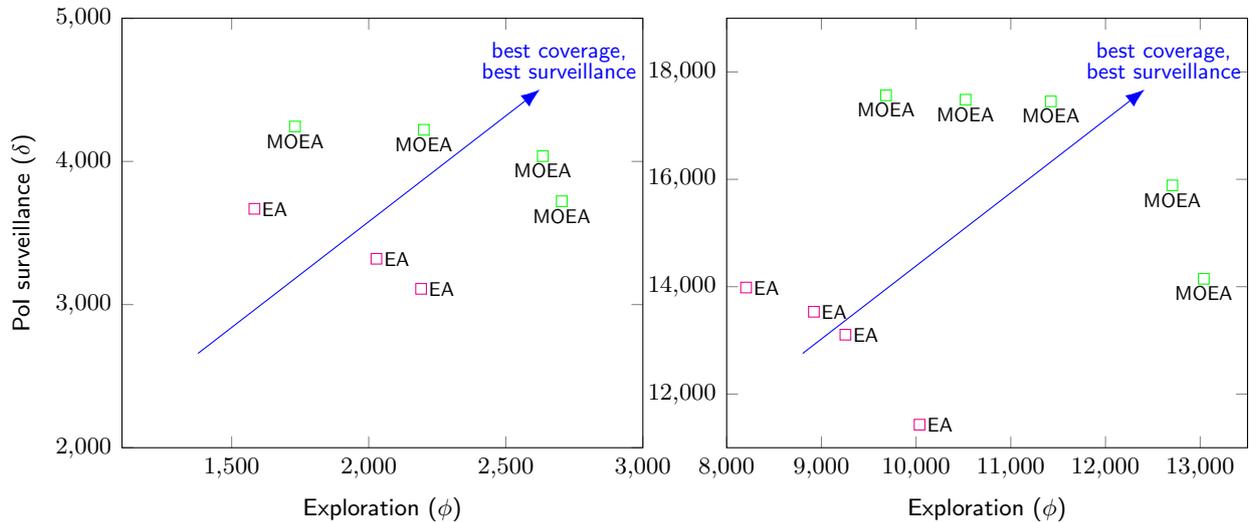


Figure 3. Pareto analysis for representative instances (left, instance #2; right, instance #4).

5. Conclusions and future work

This article presented a multiobjective evolutionary approach for UAV flight planning to optimize exploration and surveillance of a predefined set of PoI. A custom NSGA-II including specific routing-based evolutionary operators was proposed for the problem.

The experimental evaluation was performed on real and synthetic scenarios, modeling the surveillance of real facilities. Results demonstrate that the proposed MOEA improved over a previous EA approach in up to up to 26.7% (exploration) and up to 22.3% (surveillance).

The main lines for future work are related to extending the experimental evaluation and integrating more sophisticated control methods in the UAV hardware.

References

- [1] Al-Turjman F (ed) 2020 *Unmanned Aerial Vehicles in Smart Cities* (Springer International Publishing)
- [2] Zeng Y, Zhang R and Lim T 2016 *Communications Magazine* **54** 36–42
- [3] Garate B, Díaz S, Nasmachnow S, Iturriaga S, Shepelev V and Tchernykh A 2020 *Frontiers Robotics and AI*
- [4] Shi H, Sun X, Sun C, Chen D and An Y 2006 Research of the path planning complexity for autonomous mobile robot under dynamic environments *6th IEEE Intelligent Systems Design and Applications* pp 216–219
- [5] Valavanis K 2007 *Advances in Unmanned Aerial Vehicles* (Springer Netherlands)
- [6] Nasmachnow S 2014 *International Journal of Metaheuristics* **3** 320
- [7] Wietfeld C and Daniel K 2014 Cognitive networking for UAV swarms *Handbook of Unmanned Aerial Vehicles* (Springer Netherlands) pp 749–780
- [8] Deb K, Pratap A, Agarwal S and Meyarivan T 2002 *IEEE Transactions on Evolutionary Computation* **6** 182–197
- [9] Behak S, Rondán G, Zanetti M, Iturriaga S and Nasmachnow S 2020 Distributed greedy approach for autonomous surveillance using unmanned aerial vehicles *High Performance Computing Communications in Computer and Information Science* (Springer International Publishing) pp 1–15
- [10] Nasmachnow S and Iturriaga S 2019 Cluster-UY: Collaborative Scientific High Performance Computing in Uruguay *10th Int. Conf. on Supercomputing in Mexico* (Springer) pp 188–202
- [11] Gaudin A, Madruga G, Rodríguez C, Iturriaga S, Nasmachnow S, Paz C, Danoy G and Bouvry P 2020 Autonomous flight of unmanned aerial vehicles using evolutionary algorithms *High Performance Computing Communications in Computer and Information Science* (Springer International Publishing) pp 337–352