

Bypassing or flying above the obstacles? A novel multi-objective UAV path planning problem

Mahmoud Golabi¹, Soheila Ghambari¹, Julien Lepagnot¹, Laetitia Jourdan²,
Mathieu Brévilliers¹, Lhassane Idoumghar¹

¹*University of Haute-Alsace, IRIMAS UR 7499, F-68100 Mulhouse, France*

¹{firstname.lastname}@uha.fr

²*University of Lille, CRISTAL, UMR 9189, CNRS, Centrale Lille, France*

²laetitia.jourdan@univ-lille.fr

Abstract—This study proposes a novel multi-objective integer programming model for a collision-free discrete drone path planning problem. Considering the possibility of bypassing obstacles or flying above them, this study aims to minimize the path length, energy consumption, and maximum path risk simultaneously. The static environment is represented as 3D grid cells. Due to the NP-hardness nature of the problem, several state-of-the-art evolutionary multi-objective optimization (EMO) algorithms with customized crossover and mutation operators are applied to find a set of non-dominated solutions. The results show the effectiveness of applied algorithms in solving several generated test cases.

Index Terms—Mathematical modelling, UAV, Offline path planning, Multi-objective optimization algorithms

I. INTRODUCTION

In recent years, the demand for Unmanned Aerial Vehicles (UAVs), which is also known as drone, in both military and civilian applications is increasing rapidly due to the unavailability of human workforces to perform tedious and extensive tasks. Operating UAVs in urban areas surrounded by a large number of buildings is a tremendous challenge to assuring safety, especially for surveillance missions or delivery. Optimizing different aspects of UAV-based operations such as locating the launch stations [1], locating the refuelling stations [2], determining the flying routes [3], and planning the flying path [4] has been scrutinized in several research studies.

The path planning problem has been one of the most essential aspects of autonomous UAVs navigation [5]. The main goal is to steer the UAV from the start position towards the destination via the shortest collision-free path. The path planning could be categorized into static and dynamic problems. While all the obstacles are considered to be fixed in static problems, moving obstacles transform the problem into a dynamic one [6]. When the environment is static and known beforehand, the flight path could be well designed offline [7]. Otherwise, the UAV needs to intelligently plan its path online [8].

As a matter of fact, the UAV is supposed to reach the goal with a minimum flying distance and power consumption as well as safety maximization. Hence, a suitable path planning strategy must be designed not only to improve the effectiveness

of the system but also to communicate with other elements in order to comply with the mission requirements. This purpose entails a deep analysis of various contributing techniques for implementing an effective method.

Previous studies have presented a series of techniques to tackle the aforementioned problem based on different necessities such as performance optimization, collision avoidance, real-time planning, and safety maximization. They have considered the static obstacles as the prohibited airspaces to guarantee the collision-freeness by bypassing the obstacles. Generally, we can categorize the existing works into classical techniques (i.e., graph-based search methods, sampling-based approaches, potential field), computational intelligence (CI) methods, and hybrid approaches. Graph-based searches (e.g., A* and Dijkstra) were developed to find the shortest path between two nodes of connected graphs using a greedy logic. One of the positive characteristics of these methods is their simplicity, which implies reduced computational time. They have deterministic nature and guarantee to find the optimal collision-free path, if it exists. On the other hand, sampling-based approaches, such as Probabilistic Roadmaps (PRM) [9] and Rapidly-exploring Random (RRT) [10] have proven to be an effective framework suitable for high-dimensional spaces to produce feasible solutions; nevertheless, they do not guarantee the optimality of the solution [11]. However, these algorithms are the most common and popular search techniques in the case of minimizing the path length as the only objective function. In recent years, CI methods including fuzzy systems, neural networks, and evolutionary algorithms (EAs) have received most of the research effort for solving the UAV path planning problem [12]. They attract the attention of researchers because of: a) their flexibility to solve large-scale complex problems, b) their ability to apply different learning strategies to perform an effective search towards the global optimum, and c) employing for both single and multiple UAVs using different objectives.

It is incontrovertible that in some cases, flying above the static obstacles results in obtaining shorter path lengths. As mentioned before, in the majority of studies in the literature, static obstacles are considered as prohibited airspace and therefore, shortening the path length by adjusting the altitude and flying above them is not possible. As a result, the majority

of similar studies have considered a fixed flying altitude to bypass the obstacles. This study aims to overcome this shortcoming by considering both possibilities of bypassing the altitudes or flying above them. To do so, a new multi-objective mathematical formulation that accounts for both mentioned possibilities is proposed to minimize the path length, energy consumption and path risk simultaneously. To calculate the energy required to travel from point i to point j at different altitude levels, the formulation proposed by [3] is used. To account for the required energy for increasing the altitude, the term $(Wg\Delta_{ij}^+)$ is added to the mentioned formulation:

$$\theta = W^{32} \sqrt{\frac{g^3}{2\rho\zeta n}} \frac{d_{ij}}{\nu} + Wg\Delta_{ij}^+ \quad (1)$$

Where W is the drone and battery weight (kg), g is the gravity (N), d_{ij} is the distance between point i to point j (m), Δ_{ij}^+ is the increased altitude (m), ν is the flying speed (m/s), ζ is the area of spinning blade disc in m^2 , and n is the number of rotors.

The effect of altitude level on the gravity could be neglected as the gravitational force would have decreased just by 1.2% if an object flies at an altitude of 40 km. Since the flying altitude of drones is considerably lower than what mentioned here, the effect of altitude on the gravity could be ignored. The other factor that changes by altitude is fluid density of air. As described by [13], the fluid density of air in different altitudes could be calculated according to Eq. 2 in which H is the altitude in *meters*.

$$\rho = (1 - 2.2558 \cdot 10^{-5} H)^{4.2577} \quad (2)$$

Furthermore, this study proposes a new solution representation, along with customized search operators in order to solve the problem in a flexible manner. To do so, several evolutionary multi-objective optimization algorithms including Non-dominated Sorted Genetic Algorithm (NSGA-II), Reference-point based Non-dominated Sorting Genetic Algorithm (NSGA-III), and Strength Pareto Evolutionary Algorithm (SPEA-II) are applied on five generated test cases. The obtained results demonstrate how new modeling with customized operations can efficiently help the algorithms to find the Pareto-regions, especially in cluttered spaces.

The rest of this paper is organized as follows. The problem definition is given in section II. Section IV starts with basic concepts of several multi-objective algorithms which are used as base line of this empirical study in subsection IV-A. Next, in subsections IV-B the details of implementation including environment modeling, solution representation, and search operators (initialization, crossover, and mutation) are provided. The simulation results and discussion are presented in Section V. Finally, the paper is concluded in Section VI.

II. PROBLEM DEFINITION

The aim of this study is to find the drones best path while traveling from point A to point N in an environment that is replete with different obstacles of different heights such that the total traveling distance, energy consumed and the

maximum path risk are minimized simultaneously. In order to have a comprehensive knowledge of the environment and identify the obstacles, the ground surface is decomposed into several small cells and the flying space is decomposed into several altitude levels. The cells containing the obstacles and the concomitant altitude level of the obstacle are identified. The drone starts the mission from point A and aims to reach point N in a manner that the mentioned objective functions are minimized. For each cell of the ground surface, the set of succeeding cells could be determined by allowing the drone to fly in five directions: to the north, to the south, to the east, to the north-east, and to the south-east. The distance between each cell and its succeeding cells is calculated in advance. While arriving at each cell, if there exists a succeeding cell containing an obstacle, the drone has the options of choosing other succeeding neighbour cells to take a detour and avoid passing through the obstacle, or setting the altitude according to the height of the obstacle and pass above it. It is obvious that if all the succeeding cells contain obstacles, the drone must adjust the altitude and pass above one of the neighbour cells. It is noteworthy to mention that the adjusted altitude while passing the cells could also be restricted. While passing across some areas, the authorization may restrict the altitude due to different reasons such as the proximity to military bases, airports, or traditional flying corridors. To find better solutions in terms of the mentioned objective functions, in this study the drone is allowed to decrease its altitude from the previously adjusted levels, if applicable. The drones energy consumption depends on the weight, travelled distance, and the fluid density of air. Since the fluid density of air changes with altitude, this study accounts for the real energy consumption based on both flying distance and flying altitude. For each two consecutive visited cells, the average altitude is used to calculate the fluid density of air. Additionally, in order to reflect a more realistic image of energy consumption, this study considers the required energy to increase the altitude while passing the obstacles. The direct distance between the cells is calculated from the north-west corner of the cells. The altitude-change procedure (if applicable) starts from the north-west corner of the current cell towards the north-west corner of the selected succeeding cell. So, before arriving at any cell, the altitude of the drone has already been adjusted. Therefore, considering the change in the altitude and the direct distance, the real travel distance between the cells is calculated using the Pythagoras theorem. It is assumed that the drones flying speed is fixed for the entire route. Based on the data of the other organizations drone routes and the recorded air traffic of the environment, a risk factor is assigned to all the consecutive cells that account for the collision probability at different altitude levels. For each visiting cell, the maximum risk factor of traversed altitude levels while passing to the successive cell, is calculated. The accumulated maximum path risk is considered as the third objective function.

III. MATHEMATICAL FORMULATION

In order to develop the mathematical model, the following sets, scalars, parameters, and variables are used:

Sets:

I : the set of ground cells (i, j, g, A (the starting cell), N (the final destination) $\in I$)

K : the set of altitude levels ($k, k' \in K$)

δ_i^+ : the sets of cell (i s succeeding cells)

δ_i^- : the set of cell (i s preceding cells)

Scalars:

M : a large positive number

v : the drones flying speed

θ : a scalar used in calculation of consumed energy ($\theta = W^{32} \sqrt{g^3/2\zeta n}$)

Parameters:

a_j : the length of obstacle located in cell j , if exists

U_j : the maximum allowed flying altitude over cell j

r_{ij}^k : the risk factor concomitant to flying from cell i to cell j at altitude k

h^A : the drones starting altitude

h^k : the altitude of level k

ρ^k : the fluid density of air while flying at altitude level k

ρ^A : the fluid density of air while starting flying at the first cell

a_{ij} : the direct distance between the neighbour cells i and j

Decision variables:

X_{ij}^k : 1 if drone enters cell j from cell i at an altitude of level k ; 0 otherwise

Δ_{ij} : the real altitude change while flying from cell i to cell j (Δ_{ij} free in sign)

Δ_{ij}^+ : the height ascended while flying from cell i to cell j ($\Delta_{ij}^+ \geq 0$)

Δ_{ij}^- : the height descended while flying from cell i to cell j ($\Delta_{ij}^- \geq 0$)

The first set of constraints are called the network flow constraints:

$$\sum_{j \in \delta_A^+} \sum_{k \in K} X_{ij}^k = 1 \quad (3)$$

$$\sum_{j \in \delta_N^-} \sum_{k \in K} X_{iN}^k = 1 \quad (4)$$

$$\sum_{j \in \delta_i^+} \sum_{k \in K} X_{ij}^k = \sum_{j \in \delta_i^-} \sum_{k \in K} X_{ji}^k \quad (5)$$

$$\forall i \in I | \{i \neq A \text{ and } i \neq N\}$$

$$\sum_{j \in \delta_N^+} \sum_{k \in K} X_{Nj}^k = 0 \quad (6)$$

Eq. 3 and Eq. 4 indicate that the drone must leave the starting cell and arrive at the final destination. Eq. 5 assures that except for the starting cell and final destination, the

drone should leave the entered cells. Eq. 6 guarantees that after arriving at the final destination the path is finished. The following set of constraints accounts for the designated altitude level while arriving at each cell. Using the designated altitude level, the change in altitude for all visited pairs of cells could be calculated.

$$\sum_{k \in K} h^k X_{ij}^k \geq a_j - M(1 - \sum_{k \in K} X_{ij}^k) \quad \forall j \in I | \{j \neq A\}, \forall i \in \delta_j^- \quad (7)$$

$$\sum_{k \in K} h^k X_{ij}^k \leq U_j \quad \forall j \in I | \{j \neq A\}, \forall i \in \delta_j^- \quad (8)$$

Eq. 7 assures that in the case of traveling from cell i to cell j , the adjusted altitude level while arriving at cell j is higher than the located obstacle (if any). Eq. 8 defines an upper bound for the adjusted altitude level while arriving at cell j .

The change in altitude while traveling from cell i to cell j is calculated as follows:

$$\Delta_{Aj} = \sum_{k \in K} (h^k - h^A) X_{Aj}^k \quad \forall j \in \delta_A^+ \quad (9)$$

$$\Delta_{ij} = \sum_{g \in \delta_i^-} \sum_{k \in K} \sum_{k' \in K} (h^k - h^{k'}) X_{ij}^k X_{gi}^{k'} \quad (10)$$

$$\forall i \in I | \{i \neq A\}, \forall j \in \delta_i^+, \forall g \in \delta_i^-$$

Due to the multiplication of decision variables, Eq. 10 is non-linear. In order to linearize this formulation, assuming $X_{ij}^k X_{gi}^{k'} = U_{gi}^{kk'}$, Eq. 10 is replaced by the following constraints:

$$\Delta_{ij} = \sum_{g \in \delta_i^-} \sum_{k \in K} \sum_{k' \in K} (h^k - h^{k'}) U_{gi}^{kk'} \quad (11)$$

$$\forall i \in I | \{i \neq A\}, \forall j \in \delta_i^+, \forall g \in \delta_i^-$$

$$U_{gi}^{kk'} \geq X_{ij}^k + X_{gi}^{k'} - 1 \quad (12)$$

$$\forall i \in I | \{i \neq A\}, \forall j \in \delta_i^+, \forall g \in \delta_i^-$$

$$2U_{gi}^{kk'} \leq X_{ij}^k + X_{gi}^{k'} \quad (13)$$

$$\forall i \in I | \{i \neq A\}, \forall j \in \delta_i^+, \forall g \in \delta_i^-$$

Using the real change in altitude obtained from Eq. 11 to Eq. 13, the ascended and descended altitude while traveling from cell i to cell j is calculated according to the following set of constraints in which y_{ij} and y'_{ij} are the binary variables:

$$y_{ij} \Delta_{ij}^+ - y'_{ij} \Delta_{ij}^- = \Delta_{ij} \quad \forall i \in I, \forall j \in \delta_i^+ \quad (14)$$

$$\Delta_{ij}^+ - \Delta_{ij}^- = \Delta_{ij} \quad \forall i \in I, \forall j \in \delta_i^+ \quad (15)$$

$$y_{ij} + y'_{ij} = 1 \quad \forall i \in I, \forall j \in \delta_i^+ \quad (16)$$

Again, due to the multiplication of decision variables, Eq. 14 is non-linear. Therefore, considering $\wp_{ij}^+ = y_{ij}\Delta_{ij}^+$ and $\wp_{ij}^- = y'_{ij}\Delta_{ij}^-$, Eq. 14 is linearized by replacing to the following set of constraints:

$$\wp_{ij}^+ - \wp_{ij}^- = \Delta_{ij} \quad \forall i \in I, \forall j \in \delta_i^+ \quad (17)$$

$$\wp_{ij}^+ \geq \Delta_{ij}^+ - M(1 - y_{ij}) \quad \forall i \in I, \forall j \in \delta_i^+ \quad (18)$$

$$\wp_{ij}^+ \leq \Delta_{ij}^+ + M(1 - y_{ij}) \quad \forall i \in I, \forall j \in \delta_i^+ \quad (19)$$

$$\wp_{ij}^+ \leq My_{ij} \quad \forall i \in I, \forall j \in \delta_i^+ \quad (20)$$

$$\wp_{ij}^- \geq \Delta_{ij}^- - M(1 - y'_{ij}) \quad \forall i \in I, \forall j \in \delta_i^+ \quad (21)$$

$$\wp_{ij}^- \leq \Delta_{ij}^- + M(1 - y'_{ij}) \quad \forall i \in I, \forall j \in \delta_i^+ \quad (22)$$

$$\wp_{ij}^+ \geq My'_{ij} \quad \forall i \in I, \forall j \in \delta_i^+ \quad (23)$$

The first objective function is to minimize the path length. Considering a triangle in which the base is the direct distance between the visited consecutive cells and the height is the absolute value of altitude change, the drones travelled distance is calculated using the Pythagoras theorem. So, the first objective function that minimizes the total travelled distance is:

$$\begin{aligned} \text{Min } Z_1 = & \sum_{j \in \delta_A^+} \sum_{k \in K} ((h^k - h^A)^2 + d_{Aj}^2)^{1/2} X_{Aj}^k \\ & + \sum_{i \in I | i \neq A} \sum_{j \in \delta_i^+} \sum_{g \in \delta_i^-} \sum_{k \in K} \sum_{k' \in K} \\ & ((h^k - h^{k'})^2 + d_{ij}^2)^{1/2} U_{gij}^{kk'} \end{aligned} \quad (24)$$

The energy consumption while traveling from cell i to cell j depends on the related fluid density of air and the travelled distance. The fluid density of air is a function of flying altitude. In this research, the fluid density between cells i and j is considered as the average of fluid density of air while arriving at cells i and j . So, the second objective function could be written as:

$$\begin{aligned} \text{Min } Z_2 = & \sum_{j \in \delta_A^+} \sum_{k \in K} \frac{\theta}{\nu} \left(\frac{(h^k - h^{k'})^2 + d_{Aj}^2}{\rho^A + \rho^k} \right)^{1/2} X_{Aj}^k \\ & + \sum_{i \in I | i \neq A} \sum_{j \in \delta_i^+} \sum_{g \in \delta_i^-} \sum_{k \in K} \sum_{k' \in K} \frac{\theta}{\nu} \\ & \left(\frac{(h^k - h^{k'})^2 + d_{ij}^2}{\rho^{k'} + \rho^k} \right)^{1/2} U_{gij}^k k' + \sum_{i \in I} \sum_{j \in \delta_i^+} W g \Delta_{ij}^+ \end{aligned} \quad (25)$$

Finally, the third objective function that considers the maximum accumulated path risk is calculated as:

$$\begin{aligned} \text{Min } Z_3 = & \sum_{j \in \delta_A^+} \sum_{k \in K} \max_{\Phi \in [(\min(k, A)), (\max(k, A))]} \\ & \{r_A^\Phi\} X_{Aj}^k + \sum_{i \in I | i \neq A} \sum_{j \in \delta_i^+} \sum_{g \in \delta_i^-} \sum_{k \in K} \sum_{k' \in K} \\ & \left(\max_{\Phi \in [(\min(k, k')), (\max(k, k'))]} \{r_i^\Phi\} U_{gij}^{kk'} \right) \end{aligned} \quad (26)$$

IV. METHODOLOGY

The path planning problem for autonomous mobile robots is NP-hard [14]. Therefore, several intelligent optimization methods have been successfully used for solving this problem, especially while dealing with more than one objective function. This section explains how the proposed problem can be solved using evolutionary multi-objective optimization algorithms. For this purpose, several common algorithms including NSGA-II, NSGA-III, and SPEA-II are selected. These algorithms are similar in search operations. First, these algorithms are briefly described. Thereafter, specific operations are elaborated in more detail.

A. Preliminary concepts of algorithms

The strength pareto evolutionary algorithm (SPEA) is proposed by Zitzler et al. [15] in order to approximate the non-dominated solutions for multi-objective optimization problems. SPEA-II [16] is an improved elitist version of SPEA, which incorporates in contrast to its predecessor a fine-grained fitness assignment strategy, and techniques for both archive truncation and density-based selection.

NSGA-II is one of the most popular multi-objective optimization algorithms presented by Deb et al. [17] which uses an elitist principle. It has three special characteristics, fast non-dominated sorting approach, an explicit diversity preserving mechanism (crowded distance estimation procedure) and simple crowded comparison operator to emphasize the non-dominated solutions. It is claimed that this technique outperformed SPEA in terms of finding a diverse set of solutions.

Deb and Jain [18] developed the well-known NSGA-III in early 2014, to deal with both unconstrained and constrained many objective optimization problems. It is based on an external guidance mechanism to maintain diversity among its solutions. The main difference between the two NSGA-II and NSGA-III is that NSGA-III employed a set of reference points to keep the exploration of the Pareto points during the search. It yields better results, even distribution of Pareto points across

the objective space, even when the number of objectives is large.

B. Implementation details

The developed algorithms initiate with a randomly generated population of solutions of size N . Each solution, which is called a chromosome, has a variable length and could be represented as a matrix of two rows. The first-row alleles determine the index of visiting cells while the corresponding second-row alleles indicate the adjusted altitude just before entering that cell. To generate the first-row alleles, for each cell of the ground space, a set of neighbourhood cells that directs the mentioned cell to the destination is generated. The first-row starts with the index of the origin. Then, the next gene is filled with a randomly selected cell index in the neighbourhood set of the origin. Up to reaching the destination, the next genes are filled by the not-used randomly selected neighbour cells of the previous gene. If all the neighbour cells are used to fill the previous genes, the destination is not reached and the solution will be infeasible. For each first-row gene, the corresponding second-row gene is filled by a randomly selected altitude level that is in the range of the selected cells obstacle height (if any) and its maximum allowed flying altitude. To generate the reasonable altitude levels, adjusting the altitude to the height of entering cells obstacle height (if any), keeping the previous cells altitude level (if it is greater than the height of the obstacle existed on the entering cell, if any), or selecting an altitude level in the mentioned range in a random manner have the same probabilities of being applied.

The one-point crossover operation applied in this study reproduces two offspring chromosomes by mating two randomly selected parental chromosomes. The crossover point is selected randomly along the length of the mating chromosomes and decomposes them into two segments. Suppose that C is the position of the crossover point. Considering the first-row genes, if the $C+1^{th}$ allele of the second parent is among the neighbourhood set of the C^{th} allele of the first parent, the first child is generated by merging all the first segment of the first parent with all the second segment of the second parent. Otherwise, up to reaching a point by which merging the parents is possible, crossover point will be iteratively shifted to the right-hand side for the selected parent (see child 1 in Fig. 1.b) or for both of the parents (see child 2 in Fig. 1.b)

Since any further change in the first-row genes requires a corresponding search and repairing procedure that was explained in crossover section, the mutation operation is applied to the second-row genes. Based on the mutation probability, a random number of second-row genes are selected and using the probabilistic procedure previously mentioned about filling the second-row genes, their alleles are replaced with randomly generated numbers.

To show the utility of the modeling and customized operators, a simple 10×10 environment is considered and the concomitant path obtained from one of the members of the best Pareto-front of NSGA-II is illustrated in Fig. 2. The dashes show the altitude of each cell's obstacle. As shown in this

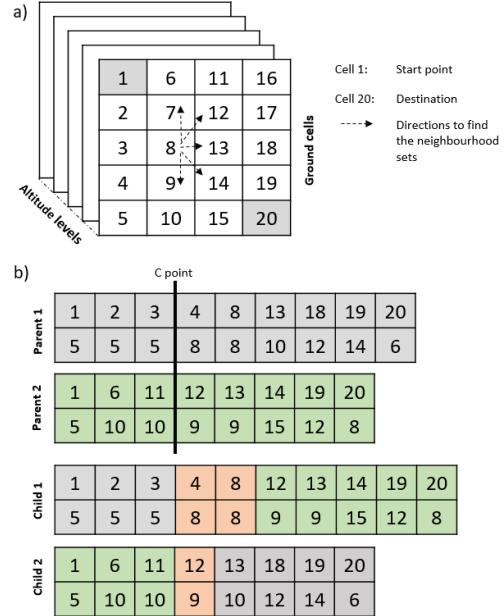


Fig. 1: a) The flying environment; b) The crossover operation

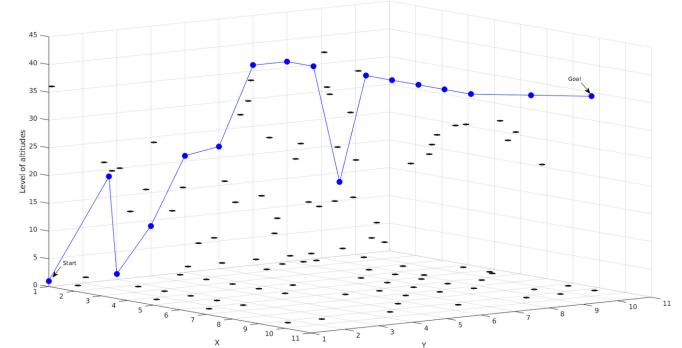


Fig. 2: The obtained path from pareto-front set with NSGA-II

figure, in some cases, the drone has bypassed the obstacles by selecting alternative cells, while in some other cases, the drone has flown above the obstacles by adjusting the altitude.

V. EXPERIMENT RESULTS

A. Experimental protocol

The simulations are executed to analyze and prove the utility and validity of the presented mathematical modeling in this study. To do so, several EMO algorithms including NSGA-II, NSGA-III, and SPEA-II are applied on five randomly generated 3D test cases in order to accomplish a series of experiments. There are five test cases which are different in environment size. Each test case has four different versions that differ in the number of obstacles, starting point, and destination. The obstacles are randomly distributed in the area. Test cases are available in the GitHub library¹.

The applied algorithm parameters are the population size, crossover probability (CR), mutation probability (MP), and

¹(<https://github.com/Sghambari/3D-dataset-for-mathematical-modelling.git>)

mutation rate (mu) with their permissible range values. Since the parameters influence the efficiency and effectiveness of applied algorithms, it is necessary to adjust them in advance to implement the algorithms. Hence, instead of using a trial-and-error approach to identify good values for them, *MAC* method [19] is utilized as an automated algorithm configuration tool for multi-objective methods in order to obtain the best parameter values. In *MAC*, the maximum budget of 100 function evaluations or reaching 48 hours of computation is used for each run.

This section first investigates the correlation between path length and energy consumption objective functions. Indeed, it is explained that the weighted sum approach is utilized to combine these objective functions. Thereafter, the results of applying the tuned and non-tuned algorithms on the generated test cases are reported.

B. weighted sum approach for path length and energy consumption objectives

The experimental results obtained from solving different test cases using the EMO algorithms show that there is a correlation between path length and energy consumption. This correlation is illustrated in figure 3, as an examples of one test case from both 2D and 3D view.

The weighted sum approach is one of the classical methods for solving multiple non-conflicting objective functions. As it is formulated in Eq. (27), this approach assigns a weight w_i to each objective function $z_i(x)$ and converts the problem to a single objective one.

$$\text{Min}Z = w_1 z_1(x) + w_2 z_2(x) + \dots + w_n z_n(x) \quad (27)$$

where $\sum w_i = 1$. The main challenge of applying this method is the selection of a weight vector for each run. To address this issue, Hajela and Lin [20] proposed a multi-criterion optimal design for multi-objective optimization in order to automate this procedure. Inspired by the mentioned design, in this study, each solution x_i in the population utilizes a different weight vector in the calculation of the new objective function which is the combination of path length and energy consumption. The weights vector of w_i is embedded within the chromosome of solution x_i . Therefore, multiple solutions can be simultaneously searched in a single run. Furthermore, weights vector can be adjusted to promote the diversity of the population.

C. Results and discussions

In this section, the computational results of implementing the EMO algorithms on the generated test cases are provided, followed by a general discussion of the utility of the mathematical modeling and the effect of automatic parameter configuration on non-dominated (Pareto sets of) solutions. There are various indicators in the literature to evaluate the quality of Pareto sets of solutions. In this study, unary hypervolume (*HV*) [21], as one of the broadly used performance indicator for EMO algorithms, is considered as a baseline of the comparison

for two-objective formulation (combination of path length and energy consumption with path risk objective).

Table I reports the percentage of improvement of *HV* for different algorithms with and without automatic parameter tuning. The default values of parameters for the selected algorithms are taken from their original papers [16]–[18]. As can be seen, the best results are obtained by employing *MAC*. Among the compared algorithms with *MAC*, for test case 1, SPEA-II outperforms the other competitors. For test cases 2 and 3, NSGA-II and NSGA-III have the same performance. However, NSGA-III had a better performance for test case 4. For test case 5, NSGA-II could win with three 100% of improvement in comparison to NSGA-III and SPEA-II. Also, for the reported results without parameter tuning, SPEA-II outperforms in test cases 1 and 2. For test case 3 NSGA-III and for test case 4, NSGA-II had better performances. Finally, for test case 5, both NSGA-II and NSGA-III have the same performance. Generally, it can be concluded that the fine-tuned NSGA-II algorithm outperforms the other competitors in most of the test cases.

Figure 4 displays the Pareto front sets of combined objectives vs path risk for NSGA-II, NSGA-III, and SPEA-II with *MAC* (blue points) and without it (red points), respectively. As can be seen, the obtained results show the superiority of *MAC* in finding the best results with a higher number of non-dominated solutions. Furthermore, Figure 5 illustrates the convergence rate of *HV* value for different test cases. These figures clarify the impact of applying *MAC* for improving the quality of the results.

VI. CONCLUSIONS AND FUTURE PERSPECTIVES

This study presents a new mathematical formulation for a multi-objective UAV path planning problem. For this purpose, the path length, energy consumption, and path risk are introduced and considered as objective functions. The novelty of this study is the possibility of bypassing obstacles or adjusting the altitude for flying above them. Besides, customized search operators are suggested for solving the problem. The experiments are accomplished using five randomly generated test cases by applying several EMO algorithms. The results are reported with and without parameter tuning. The obtained results show the effect of applying automatic configuration for parameter tuning and also the superiority of the NSGA-II algorithm.

For future research, based on the dynamics of the considered UAV, the problem could be studied in a dynamic environment comprised of moving and static obstacles especially for 3D real urban environment. Developing other solution algorithms to obtain better results in less computational time could be of great interest.

ACKNOWLEDGMENT

This work is part of a research project funded by the French *Agence Nationale de la Recherche* under grant number ANR-16-SEBM-0004. Also, the authors would like to acknowledge their indebtedness to *Hojjat Rakhshani* for his guidance and

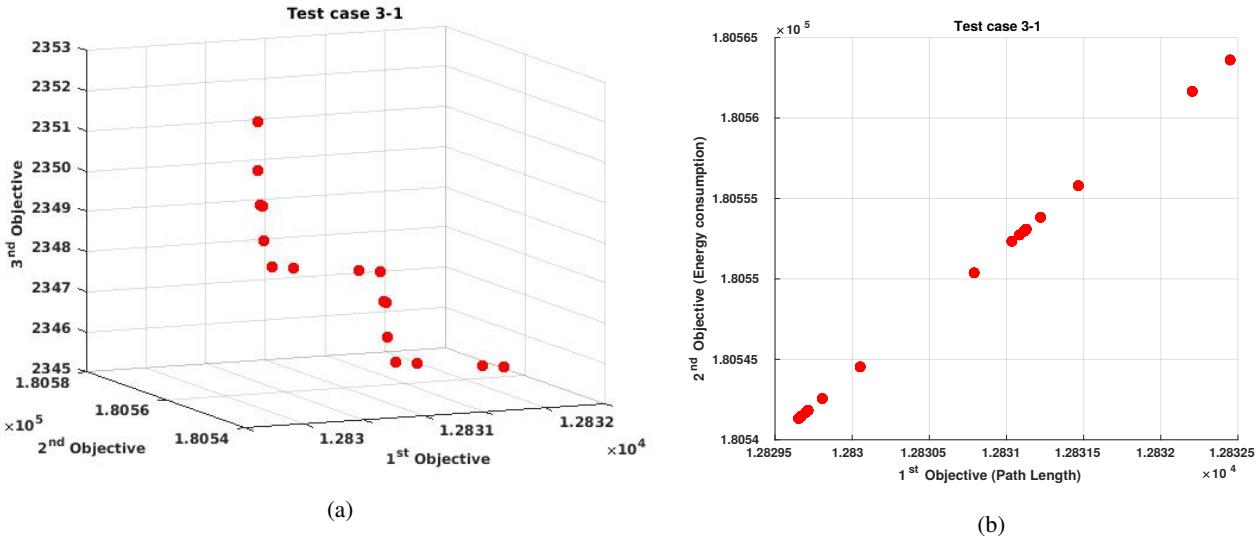


Fig. 3: The correlation between path length and energy consumption; the 3-D view of three objectives is shown in left-side figure and the right-side figure is for a 2-D view of path length and energy consumption

TABLE I: The percentage of improvement in Hypervolume indicator for different test cases.

Dataset	Algorithms + MAC			Algorithms		
	SPEA-II	NSGA-II	NSGA-III	SPEA-II	NSGA-II	NSGA-III
T1-1	100.00%	99.96%	100.00%	95.25%	94.67%	92.56%
T1-2	97.19%	100.00%	97.10%	94.34%	93.84%	92.60%
T1-3	100.00%	99.05%	97.98%	95.60%	95.23%	96.48%
T1-4	99.18%	99.88%	98.10%	95.77%	96.94%	100.00%
T2-1	98.67%	98.51%	100.00%	99.30%	94.64%	92.69%
T2-2	98.30%	100.00%	99.44%	95.40%	96.13%	93.26%
T2-3	99.56%	98.89%	100.00%	96.26%	94.00%	93.74%
T2-4	98.47%	100.00%	99.93%	96.83%	94.34%	94.64%
T3-1	99.16%	100.00%	99.55%	95.50%	94.63%	96.08%
T3-2	98.30%	100.00%	99.82%	95.57%	97.10%	94.88%
T3-3	98.97%	99.88%	100.00%	96.13%	95.31%	97.26%
T3-4	96.99%	98.82%	100.00%	95.14%	94.23%	95.88%
T4-1	100.00%	98.49%	99.57%	94.93%	95.02%	94.01%
T4-2	98.71%	99.20%	100.00%	95.10%	96.58%	95.31%
T4-3	99.53%	99.78%	100.00%	95.33%	95.86%	95.80%
T4-4	98.39%	100.00%	99.01%	97.60%	94.58%	94.65%
T5-1	97.54%	100.00%	99.02%	95.09%	95.73%	96.85%
T5-2	96.08%	97.86%	100.00%	94.14%	93.16%	94.42%
T5-3	98.57%	100.00%	98.53%	97.12%	94.50%	94.33%
T5-4	98.90%	100.00%	99.89%	95.27%	94.95%	94.65%

expert advice which have been invaluable throughout all stages of the work.

REFERENCES

- [1] M. Golabi, S. M. Shavarani, and G. Izbirak, “An edge-based stochastic facility location problem in uav-supported humanitarian relief logistics: a case study of tehran earthquake,” *Natural Hazards*, vol. 87, no. 3, pp. 1545–1565, 2017.
- [2] S. M. Shavarani, S. Mosallaeipour, M. Golabi, and G. Izbirak, “A congested capacitated multi-level fuzzy facility location problem: An efficient drone delivery system,” *Computers & Operations Research*, vol. 108, pp. 57–68, 2019.
- [3] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, “Vehicle routing problems for drone delivery,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 1, pp. 70–85, 2016.
- [4] S. Ghambari, J. Lepagnot, L. Jourdan, and L. Idoumghar, “A comparative study of meta-heuristic algorithms for solving uav path planning,” in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 174–181.
- [5] R. Samar and W. A. Kamal, “Optimal path computation for autonomous aerial vehicles,” *Cognitive Computation*, vol. 4, no. 4, pp. 515–525, 2012.
- [6] M. Bekhti, M. Abdennabi, N. Achir, and K. Boussetta, “Path planning of unmanned aerial vehicles with terrestrial wireless network tracking,” in *2016 Wireless Days (WD)*. IEEE, 2016, pp. 1–6.
- [7] I. K. Nikолос, K. P. Valavanis, N. C. Tsourveloudis, and A. N. Kostaras, “Evolutionary algorithm based offline/online path planner for uav navigation,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 6, pp. 898–912, 2003.
- [8] X. Peng and D. Xu, “Intelligent online path planning for uavs in adversarial environments,” *International Journal of Advanced Robotic Systems*, vol. 9, no. 1, p. 3, 2012.
- [9] L. Kavraki, P. Svestka, and M. H. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. Unknown Publisher, 1994, vol. 1994.
- [10] S. M. LaValle and J. J. Kuffner Jr, “Randomized kinodynamic planning,” *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.
- [11] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 174–181.

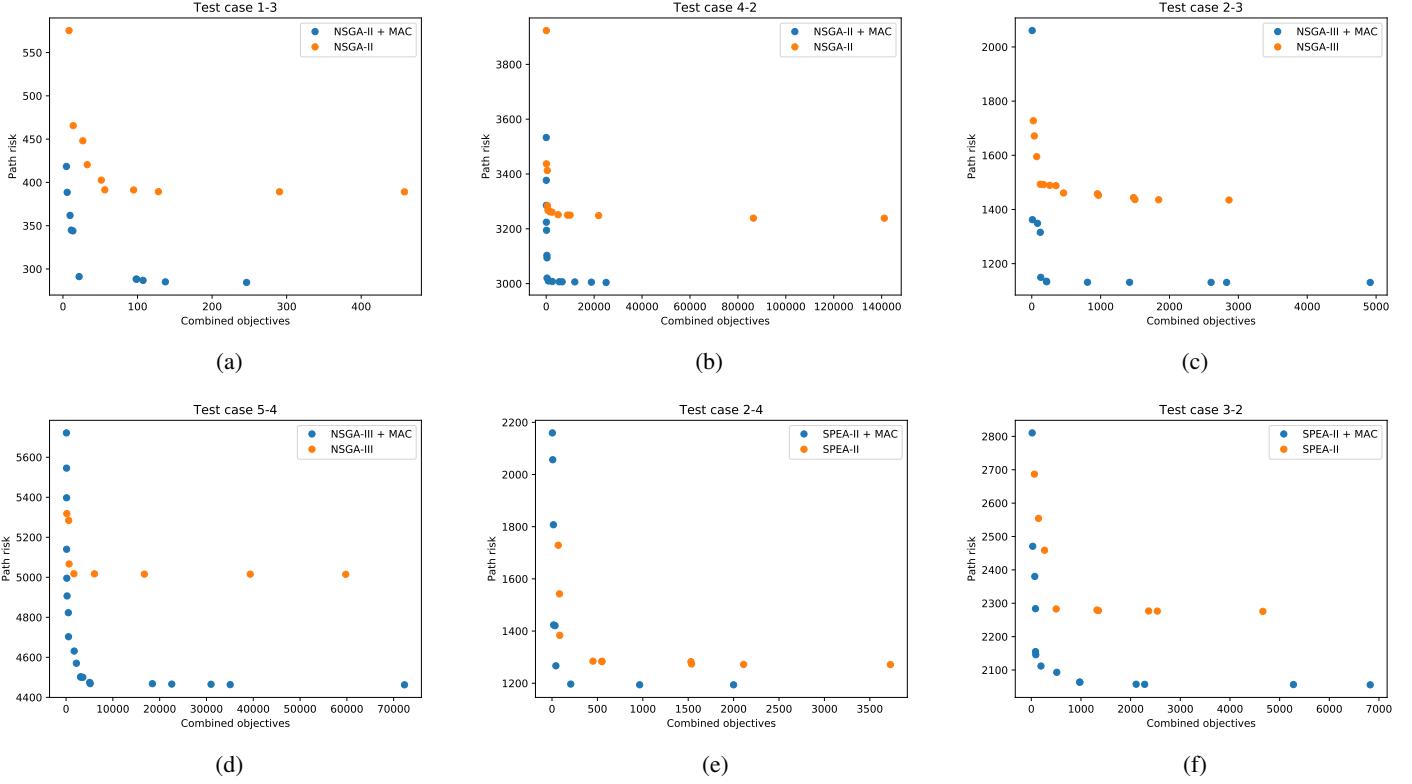


Fig. 4: The plot of the non-dominated solutions obtained using applied algorithms with (blue points) and without parameter tuning (red points).

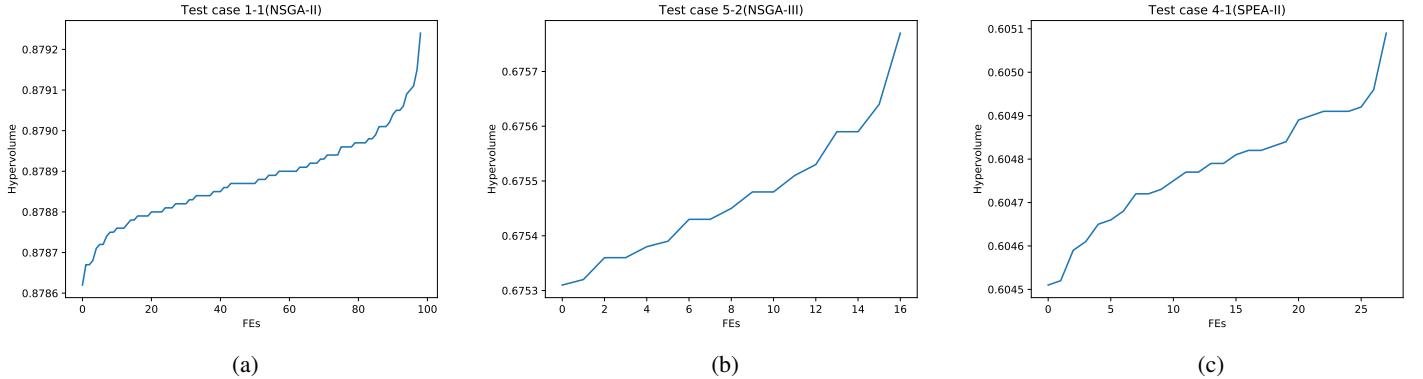


Fig. 5: The plot of convergence rate of HV for different test cases and algorithms

- Conference on Intelligent Robots and Systems. IEEE, 2014, pp. 2997–3004.
- [12] Y. Zhao, Z. Zheng, and Y. Liu, “Survey on computational-intelligence-based uav path planning,” *Knowledge-Based Systems*, vol. 158, pp. 54–64, 2018.
- [13] J. A. Paredes, C. Saito, M. Abarca, and F. Cuellar, “Study of effects of high-altitude environments on multicopter and fixed-wing uavs’ energy consumption and flight time,” in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*. IEEE, 2017, pp. 1645–1650.
- [14] G. Li, A. Yamashita, H. Asama, and Y. Tamura, “An efficient improved artificial potential field based regression search method for robot path planning,” in *2012 IEEE International Conference on Mechatronics and Automation*. IEEE, 2012, pp. 1227–1232.
- [15] E. Zitzler and L. Thiele, “An evolutionary algorithm for multiobjective optimization: The strength pareto approach,” *TIK-report*, vol. 43, 1998.
- [16] E. Zitzler, M. Laumanns, and L. Thiele, “Spea2: Improving the strength pareto evolutionary algorithm,” *TIK-report*, vol. 103, 2001.
- [17] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii,” in *International conference on parallel problem solving from nature*. Springer, 2000, pp. 849–858.
- [18] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.
- [19] H. Rakhsani, L. Idoumghar, J. Lepagnot, and M. Brévilliers, “Mac: Many-objective automatic algorithm configuration,” in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2019, pp. 241–253.
- [20] P. Hajela and C.-Y. Lin, “Genetic search strategies in multicriterior optimal design,” *Structural optimization*, vol. 4, no. 2, pp. 99–107, 1992.

- [21] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach,” *IEEE transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.