On the Performance Comparison of Multi-objective Evolutionary UAV Path Planners

Eva Besada Portas[a], Luis de la Torre[b], Alejandro Moreno[b], José L. Risco-Martín[a]

[a] *Universidad Complutense Madrid. 28040 Madrid. Spain*
[b] *Universidad Nacional de Educación a Distancia. 28040 Madrid. Spain*

Abstract

The big number of evolutionary planners for Unmanned Aerial Vehicles (UAV) that have been developed demonstrates the good acceptance that the evolutionary techniques enjoy within the UAV community. However, the minor or nonexistent statistical characterization of the results obtained by the majority of the planners not only makes it difficult to assess their actual performance but also to justify the selection and/or parameterization of their supporting algorithms. To fill the gap, this paper proposes a method for comparing the planners performance by jointly employing several general and problem-specific quality indexes, which take into account the complexity and particularities of the problem. The generality of the performance metrics adopted, which are able to deal with any multi-objective dominance definition, makes them equally applicable to multi-objective planners with different relation operations (such as Pareto dominance, weighted objectives aggregation, and others). The specificity of the other indexes, which consider the types of solutions preferred by the problem experts, makes them

Email address: evabes@dacya.ucm.es (Eva Besada Portas)

especially attractive to characterize their planners' behavior. The paper also shows how to analyze the results of the quality indexes graphically in order to identify, for a particular UAV planning problem, the best planners within a set of 36 variants (based on Genetic Algorithms, Particle Swarm Optimization and Differential Evolution).

Keywords:
UAVs planning, Multiobjective Optimization, Evolutionary Heuristics, Performance Measures, Parameter Tuning

1. Introduction

Unmanned aerial vehicles (UAVs) are aircrafts without onboard pilots that can be remotely controlled or fly following preplanned routes [24]. They can also fly autonomously, if they are able to generate and adapt their trajectories to their mission and environment properties. Although the type of civil or military task of the UAVs determines the properties considered by the planner, the path generation/updating is often formulated as an optimization problem, where the feasibility of any route depends on the mission, environment and UAV physical constraints; while its optimality depends on the planning criteria (such as minimal path length or destruction risk).

The inherent NP-completeness [37] of the path generation problem benefits from the use of efficient heuristics capable of finding feasible suboptimal routes in an appropriated length of time. The versatility of many evolutionary heuristics such as Genetic Algorithms (GA, [10]), Particle Swarm Optimization (PSO, [15]) and Differential Evolution (DE, [29]), as well as their good performance in many real world optimization problems, have made them an attractive choice to tackle UAV planning problems [27, 22, 14, 46, 31, 43, 39, 38, 45, 25, 28, 33, 3, 12, 26, 1, 17, 36, 35, 18, 2].

The stochasticity of these randomized optimizers, a positive quality to avoid local optima, often makes them find different suboptimal solutions in consecutive runs. Moreover, the *relation* operations applied to compare two solutions (trajectories) defined by multiple objectives/criteria usually induce only a partial order in the search space, making the evolutionary algorithms return a set of equally suboptimal solutions instead of a single one. Hence, the performance of the algorithms should be based on a statistical comparison, in compliance with the *relation* operation used by the algorithm, of the results obtained in multiple runs [16].

Although the issue of performance assessment of randomized multi-objective optimizers is not new in the evolutionary heuristic literature [9, 34, 47], the results of the majority of the UAV planners are characterized by either the lack of a statistical analysis [27, 22, 46, 31, 39, 38, 45, 25, 28, 33, 3, 36, 26, 17, 35, 18] or by only a minor one [14, 43, 12, 2]. This situation makes difficult to determine the actual performance of these planners and justify the parameterization of their evolutionary algorithms. This study makes two contributions to the UAV research: 1) it demonstrates the use of statistical analysis of the results obtained by multi-objective randomized planners and 2) it shows how such analysis can be used to perform algorithm tuning.

The performance comparison approach presented in this paper makes use of:

- *Multiple performance metrics*, belonging to two complementary groups of quality indexes that highlight general and specific properties of the planners. The *general metrics* group consists of two quality indexes based on the statistical Pareto front ranking procedure presented in [16], because it is in compliance with any dominance definition. Hence, this type of analysis can be applied to many planners (as well as to other types of problems) incorporating the *relation* operation used in the evolutionary algorithm to compare two solutions within the metrics of this first group. The *specific metrics* group takes into account the fact that although multiple trajectories can be equally good according to the evolutionary heuristic, the final path followed by the UAV has to be selected according to the expert preferences. Therefore, the two metrics within this second group are quality indexes directly related to the problem and expert preferences.

- *Multiple graphical representations* of the results of the performance analysis carried out over the results obtained for different evolutionary planners, such as those based on GA, PSO and DE techniques. These graphical representations have been carefully designed to be able to efficiently compare the results of many variants of the same planner and visually determine, at a glance, the best planners within a group.

This paper also demonstrates the applicability of our approach in the performance comparison of 36 variants of the planner that tackles the offline single UAV problem presented in [1]. The feasibility and optimality of any route in this complex multi-objective problem depends on the UAV maneuverability and on the positions of the UAV obligatory passing points, prohibited flying zones (Non Flying Zones, NFZ), and Air Defense Units (ADU). The formal properties and positions of those elements constitute a scenario and their variations change the landscape of the values of the constraints and optimization indexes. Therefore, the selected problem under different scenarios is itself a good test bench to compare the performance of different evolutionary algorithms.

Finally, we want to underline that our selection of metrics does not intend to exclude the complementary insight that other statistical analysis could bring to the study of the heuristics performance. Nevertheless, we believe that it is a well-balanced choice between the types of metrics that the evolutionary community and domain expert prefer, which let us graphically compare the performance of many planner variants simultaneously, as illustrated by the analysis of 36 variants of the planner in 4 scenarios over the selected problem.

This paper is organized as follows. Section 2 summarizes the types of analysis presented in different UAV evolutionary planners. Section 3 presents the different performance metrics and graphical representations that we use to compare different path planner variants. Section 4 describes the selected problem and GA, PSO and DE based planner variants used to demonstrate the applicability of our comparison methodology, and analyze the results of the comparison. Finally, the conclusions are drawn in Section 5.

2. Related Work

The development of many UAV planners based on GA [27, 22, 14, 46, 31, 43, 39, 38, 45, 25, 28, 33, 3, 12, 26, 1], PSO [17, 36, 35, 18] and DE [2] shows the versatility of the evolutionary techniques for these types of problems. However, the assessment or comparison of the actual performance of these planners is not straightforward due to several factors. On one hand, they differ in their heuristic properties (parameters to tune the algorithm behavior, operators to evolve the solutions or the multi-objective relation applied to compare pairs of solutions), the representation of the trajectories, and the constraint/objective functions used to define the UAV mission as a multi-objective optimization problem. On the other hand, they usually converge on the limitations of the statistical analysis of their results, which often ignores the stochasticity and/or optimality diversity that their multi-objective evolutionary algorithms introduce.

Regarding the performance analysis limitation issue, the planners in the following papers are inadequately characterized, as they only present the results obtained in a *single run*: [27, 31, 39, 38, 45, 25, 28, 3, 26, 17, 36, 35, 18] only show the evolution of the best objective value or the best solution, [22, 33] go slightly further complementing the previous analysis with a representation of the best Pareto front [41], and [46] complements the representation of the solution evolution with the generation numbers required to obtain feasible and best solutions.

The situation improves in the *multi-run* statistical analysis of the following planners, which use the weighted sum of the objective values as the *relation* operation to compare two solutions in the planner: [14] shows the best value at the end of the algorithm in multiple runs, [43] represents the mean of the optimal value of multiple runs for different iteration number and parameterizations of the algorithm, [12] shows the best and mean values of multiple runs for different generations as well as how the best values are affected by the difficulty of the scenario, and [2] represents the best and worst values obtained over multiple runs for different iteration numbers. That is, the analysis of these planners is based on the best, mean and worst values, which are common strategies to characterize the behavior of evolutionary algorithms for mono-objective (and optionally also constrained) problems [13, 23, 40, 42, 44]. However, the analysis of all these planners ([14, 43, 12, 2]) lack their usually accompanying statistical tests (such as t-test, Wilconxon, Man-Whitney or t-student) that determine if there is a significant difference in the results obtained by several planner parameterizations. Moreover, the best, mean and worst values of the aggregated values of the objectives are not valid for planners with other relation operations, and these quality indexes cannot maintain the partial order imposed by the weighted sum relation.

To overcome these limitations, as well as to determine the best parameterization of a GA based planner and the iteration number (specific generation of the GA) up to the planner was able to improve the solutions, the study in [1] combines one quality index based on the statistical Pareto front ranking procedure presented in [16] with one quality indicator related to the final expert preferences. This article details and extends the analytical approach followed in [1], by 1) explaining how the statistical analysis in [16] can be carried out for any relation operation, 2) introducing a new quality indicator based on the statistical Pareto front ranking procedure presented in [16] to complement the one that was already being used, and 3) including two improved final expert preference quality indexes. Besides, this article also illustrates how a massive comparison of 36 variants of a planner (based on GA, PSO and DE) can be 1) easily interpreted with a carefully designed graphical representation and 2) used to infer some relevant characteristic of the problem/planner.

3. Comparison Performance Metrics

The stochasticity associated with multi-objective evolutionary meta-heuristics creates planners that do not necessarily obtain the same final set of solutions in all their executions. Hence, a systematic comparison of the performance of the planners with different parameterization or heuristics requires applying several statistical procedures to the results obtained in N_r runs of each planner variant. Additionally, when the performance analysis includes a big set of variants, it is useful to employ a graphical representation that facilitates the interpretation of the comparison results at a glance.

This section presents the comparison metrics and the graphical representation we propose to achieve both objectives. The metrics are organized in two groups: the first (Section 3.1) is associated with the general quality indicators that can also be used in other problems, and the second (Section 3.2) is associated with the particularities of the planning problem under test and directly related to the final expert preferences. The three types of graphical representations used to summarize the results of the all the metrics are explained in Section 3.3.

3.1. General Metrics

Our first group of metrics is based on the dominance ranking procedure presented in [16] that compares two *sets* of solutions exploiting the relation

(rel) operation[1] applied by the evolutionary algorithm for comparing two possible solutions a and b, using the values $f_k(a)$ and $f_k(b)$ of their violation constraint indicators and objective functions. In other words, the selected general metrics employ a procedure (stated in the Appendix) that compares two sets A and B of solutions (to determine if a set dominates the other) using the same relation rel operation that is applied by the algorithms that generate A and B to compare the solutions $a_i \in A$ against the solutions $b_j \in B$. This procedure makes the two general quality indexes in this section applicable to the performance comparison of any pair of meta-heuristics that implement the same relation operation because they are based on a multi-run comparison of the solutions of the best sets of solutions returned by each meta-heuristic.

The first metric is the Statistical Front-Dominance Ranking Procedure (SFDRP) presented in [16], which measures the results of algorithm A and B to see if they are statistically different by comparing the best solution sets A_l and B_m obtained by each algorithm in N_r executions. The method counts the number ($\diamondsuit_{A_l}^{B_{1:N_r}}$) of times that each of the N_r best Pareto fronts obtained by algorithm A is dominated by each of the N_r best Pareto fronts obtained by algorithm B and vice versa ($\diamondsuit_{B_l}^{A_{1:N_r}}$). That is, the method makes $\diamondsuit_{A_l}^{B_{1:N_r}} = \sum_{m=1:N_r} I_c\left(A_l \text{ is dominated by } B_m\right)$ and $\diamondsuit_{B_l}^{A_{1:N_r}} = \sum_{m=1:N_r} I_c\left(B_l \text{ is dominated by } A_m\right)$,

[1]The relation operation used in the variants of the planner selected to demonstrate the applicability of our approach (see section 4.1.3) is the non-standard dominance evaluation function based on goals, priorities and Pareto sets [8]. In other planners or problems, it is the comparison of the weighted sum of the objectives with a penalization term associated with the constraints violation, the basic Pareto dominance definition, or other.

where $I_c(\cdot)$ is the indicator function that returns 1 if the input condition is true and 0 otherwise. Then, it applies the non-parametric Mann-Whitney rank test [19] to the vectors $[\diamondsuit_{A_l}^{B_{1:N_r}}+1, \diamondsuit_{A_2}^{B_{1:N_r}}+1, ..., \diamondsuit_{A_{N_r}}^{B_{1:N_r}}+1]$ and $[\diamondsuit_{B_l}^{A_{1:N_r}}+1, \diamondsuit_{B_2}^{A_{1:N_r}}+1, ..., \diamondsuit_{B_{N_r}}^{A_{1:N_r}}+1]$, and when the test finds a statistical significant difference in the medians of those vectors, we can use the median to infer which of the two algorithms is usually less dominated by the other. This procedure is extended to groups of variants of evolutionary algorithms by comparing all pairs of variants in the group.

The second metric, one of the contributions of this article, calculates if the results obtained by algorithm A are usually better than the results obtained by algorithm B when both algorithms are initialized with the same population and immigrants (i.e. new possible solutions randomly generated every generation). To perform this new comparison, which we call Individual Ranking Procedure (IRP) hereafter, we generate N_r different initial populations and immigrant sets, run all the planners once for each of them, and count the number $(\star_{B_{1:N_r}}^{A_{1:N_r}})$ of times that the best front of planner A for the i-th initial population dominates the best front of planner B for the same i-th initial population and vice versa $(\star_{A_{1:N_r}}^{B_{1:N_r}})$. That is, the method makes $\star_{B_{1:N_r}}^{A_{1:N_r}} = \sum_{l=1:N_r} I_c\,(B_l \text{ is dominated by } A_l)$ and $\star_{A_{1:N_r}}^{B_{1:N_r}} = \sum_{l=1:N_r} I_c\,(A_l \text{ is dominated by } B_l)$. This procedure is extended to groups of variants of evolutionary algorithms, comparing all possible pairs of variants in the group.

In order to apply these general quality indexes to the results of any pair of evolutionary algorithms, we only need to include the relation *rel* operation used by the heuristic to compare *pairs* of solutions in the process outlined in the Appendix to compare *sets* of solutions. Therefore, we can directly apply them to the results of the planner variants under comparison in Section 4, whose relation operator is the non-standard dominance evaluation function based on goals, priorities and Pareto sets [8]. However, as SFDRP and IRP do not provide information about the goodness of the results of the algorithms, because they only inform us about the relations among their outcomes, we believe that it is useful to complement this information with the obtained from other statistical comparisons.

Some quality indicators for multi-objective problems that provide information about the goodness of the solution are attainment functions [9], hypervolumes [47], and R indicators ($I_{R\#}$, [11, 30]). However, as they are designed for specific relation operations (attainment functions and hypervolumes for the basic Pareto dominance definition [41] and I_{R2} for objective weighted sum approaches), they cannot be applied to other relation operations[2] before being adapted. That is, these performance metrics are not as general as SFDRP and IRP. Besides, the options that construct the optimal solution inversely to compare the solutions of the planner against it (such as the index that measures the probability of convergence to the optimum in [20]) can be difficult to carry out in complex planning problems[3]. There-

[2]For instance, to apply them to the relation operation used in the variants of the planner in Section 4, the different priority levels presented in Table 1 should be taken into account in the attainment, hypervolume of $I_{R\#}$ calculations.

[3]For example, finding the optimal solution of the problem under test in Section 4 is not an easy task due to the UAV physical constraints and the complex landscapes of the UAV detection and destruction probabilities, which depend on the terrain, radar cross section and missiles properties.

fore, we have decided to complement the SRDRP and IRP results with some quality indicators that take into account the characteristics of the problem and the experts' preferences.

Finally, it is important to highlight that the interpretation of SFDRP and IRP is different by definition. In short, SFDRP lets us identify the planners that, when initialized with any population, obtain at least as good results as the others, while IRP let us detect the planners that usually improve further a given initial population and immigrant set.

3.2. Problem Dependent Metrics (PDM)

To complement the metrics presented in the previous section, we use some meaningful quality indicators for the planner variants under test that take into account the experts' preferences. Such preferences allow the selection of the final solution eventually used by the UAV. Hence, this makes the quality indicators within this group dependent on the problem solved by the variants of the planner. However, these Problem Dependent Metrics (PDM) can be easily adapted to other planners (and problems), considering their own final experts' preferences.

From the PDM perspective, the solutions of the planning problem used to illustrate the applicability of our metrics in Section 4 have to:

1. Fulfill multiple constraints.
2. Simultaneously minimize the path length and the probability of destruction of the UAVs.
3. If they are equally valid regarding the path length and probability of destruction, simultaneously minimize the probability of detection and flight altitude.

That is, the planner variants under comparison are facing a constrained multi-objective multi-level priority optimization problem.

Taking into account the priority levels of the problem, our third metric is related to the solution feasibility, because constraints, considered as first level priorities objectives, need to be fulfilled by the optimal solution. Hence, we measure the number of best sets obtained by the N_r executions of each planner variant that fulfill the constraints at a given generation. Note that this quality index can be applied to other constrained problems too. However, we place it in the PDM section, because not all the problems have constraints and the results can be easily represented along the results associated with the following PDM.

The fourth metric is related to the experts' preferences associated with the second priority level, that contains both the path length and the destruction probability. In dynamic environments, it is equally important to minimize both: a shorter trajectory has a lower chance of finding an unexpected threat and the already known unsafe regions have to be avoided. Besides, addressing them at the same level using the relation operator presented in Section 4.1.3 reduces the number of discontinuities in the search space and allows the algorithm temporarily improve one of the objectives in spite of worsening the other. However, the path with the lowest destruction probability is selected by the experts as the final solution, because it is the safest path regarding the available information about the ADUs. As the path length and destruction probability are placed at the same priority level and the relation operator makes them compete in a Pareto fashion, the solution with the lowest destruction probability is the one with the highest (worst) path length. Therefore, incorporating the final experts' preferences, this metric calculates the mean of the worst[4] path lengths of the solutions in the best front of the N_r executions algorithm, excluding those best fronts that do not fulfill the constraints.

Finally, it is worth noting that although these two PDM are similar to some of the quality indexes used in the multi-run statistical analysis of some of the planners analyzed in Section 2, we do not use the information that they provide as the only source to determine which sets of solutions or heuristic parameterization are the best. Instead, we treat their results as complementary information that helps us, along with the results given by the generic metrics, to carry out the comparisons. Besides, we do not use the sum of the values of the different objectives to calculate these metrics. Instead, we develop expert preference compliant PDM.

3.3. Graphical Representation

In order to facilitate the interpretation of the performance comparison of many variants of a planner over different scenarios, we summarize the results of the 4 metrics applied to different subsets of planner variants using the 3 types of graphics presented in Figures 3-6. This section explains how to interpret these graphics correctly, as they can be slightly difficult to understand initially due to the different types of information they combine by

[4] In some cases, the best feasible Pareto front contains a unique solution that simultaneously has the minimal kill probability and path length. When that happens, the solution with the worst and best path length of the best front is the same. Therefore, calculating always the mean of the worst path length works well for the case of multiple solutions in the best front as well as for this particular case.

means of color and intensity scales. We will demonstrate its high utility in the exhaustive comparison of 36 planner variants presented in section 4.4.

The graphics in the first row of Figures 3-6 summarize the SFDPR results for different subsets of algorithms and scenarios. In each cell of each graphic we represent when the algorithm in the Y axis is better (less dominated, in white), equivalent (no statistically different, in gray) or worst (more dominated, in black) than the algorithm in the X axis. Hence, the best algorithms within each compared set (graphic) are those represented with lighter rows, as they are less dominated and/or not significantly different to the others.

The graphics in the second row of Figures 3-6 summarize the IRP results for different subsets of algorithms and scenarios. Each cell in each graphic represents the number of times that the algorithm in the Y axis dominates the algorithm in the X axis ($\star_{X_{1:N_r}}^{Y_{1:N_r}}$) by means of a double-colored scale that also shows if the number of times that algorithm Y dominates algorithm X is bigger or smaller than the number of times that algorithm X dominates algorithm Y, using warm colors for the first case ($\star_{X_{1:N_r}}^{Y_{1:N_r}} \geq \star_{Y_{1:N_r}}^{X_{1:N_r}}$) and cold colors for the second one ($\star_{X_{1:N_r}}^{Y_{1:N_r}} < \star_{Y_{1:N_r}}^{X_{1:N_r}}$). That is, the best algorithms within each compared set (graphic) are those represented with warm colored rows, as they will dominate the others more often when initialized with the same population. Besides, darker warm colored rows identify better algorithms than lighter warm colored rows, as the number of times that the first ones dominate the rest is bigger.

The graphics in the third row of Figures 3-6 summarize the PDM results for different subsets of algorithms and scenarios. The generations of each planner where at least one of the N_r best fronts does not fulfill the constraints are represented in dark red[5], while those that fulfill them are represented by a blue scale, associated with the mean path length, that employs light blue shades for its lower (better) values and dark blue shades for the higher (worse) ones. Hence, within each graph, redder rows identify the planners whose constraints are usually fulfilled later, and lighter blue lines show those planners that, in average, achieve better final preferred solutions sooner.

4. A Case Study

In order to demonstrate the utility of our approach, we selected the offline single UAV planner presented in [1], implemented 35 new versions of it, and used them over 4 different scenarios to obtain the best sets of solutions for the planner variants in multiple runs. Then we applied the comparison performance indexes to these best sets of solutions, and interpreted the graphic representation of the results of the comparison.

The following sections present relevant aspects of the problem and planners, the properties of the algorithms supporting the planner variants under test, the scenarios used to obtain the data, and the comparative results.

4.1. The problem

In short, the selected military UAV route planning problem is formulated as an optimization problem where the 3-D routes, compactly codified as cubic spline curves [7], are evaluated according to 10 objective functions. These

[5] We only show when the number of constrained best sets is smaller than N_r (dark red) because in the majority of the cases the third metric was equal to N_r and this way of proceeding lets us include the information of the third and fourth metric in a single graph.

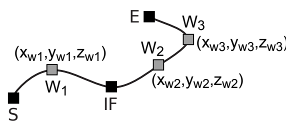


Figure 1: 3D cubic spline path codification

functions are related to the UAV physical constraints; the overflown terrain; the disposition and characteristics of the existing NFZs and ADUs; and the priority levels and goals associated with the objective functions within each mission. In this section we describe their main characteristics (see [1] for further details, including the objective function equations) and the relation operator (or dominance evaluation procedure) used to compare two possible solutions in the planners and to obtain the results of the performance indexes in the comparison.

4.1.1. Path Codification

The trajectories optimized by the UAV are represented as lists of 3-D floating points, coded in the 3-D absolute Cartesian space (x, y, z), that define the cubic splines to be followed by the UAV. This list contains some fixed points that the UAV is obliged to bypass and some undetermined points, called waypoints hereafter, whose feasible and optimal values are calculated by the planner for the selected scenario, mission, and objective functions. The number of waypoints between two fixed points is determined by the planner based on that fixed point distance and kept constant during the optimization. As an example, Figure 1 shows the cubic spline curve for the list $(S, W_1, IF, W_2, W_3, E)$, where S, IF and E stand for the fixed start, intermediate and end points; and W_i for the i-th waypoint with undetermined

(x_{Wi}, y_{Wi}, z_{Wi}) values that have to be optimized by the planner. Note how the trajectory connects all the points of the list, a property associated with cubic spline curves.

4.1.2. Objective Functions

To guide the optimization process, the planners evaluate the objective functions over the 3-D continuous-spline curves, discretized in as many 3-D Cartesian points as necessary to ensure that the distance between two consecutive discretized points is smaller than a given threshold. Besides the 3-D points of the discretized curve, the objective evaluation process takes into account the properties of the elements that appear in the optimization scenarios: the terrain, NFZs, ADUs and UAV. The feasible trajectories are constrained by some properties of the UAV (turning radius, maximum climbing and diving slopes, and initial fuel payload), the map limits, the map altitude and the NFZs regions. Their violation is measured, either as the number of times that the constraints are not fulfilled or as the distance to the fulfillment, by the corresponding constraint violation indexes that appear at the top of Table 1. The optimal trajectories minimize the path length ratio (path length normalized by the length of the straight trajectory), the flight altitude, and the probabilities of destruction and detection accumulated by the UAV trajectory when it bypasses areas with ADUs. These minimization indexes are presented at the bottom of Table 1.

The equations of the constraints and objective index can be found in [1], which also presents the expression of a coordination constraint not included in Table 1. This is so because the performance analysis of this article is only carried out for the offline single UAV problem due to the fact that the

Table 1: Objective functions, and dominance evaluation function priorities and limits

Constraint Violation Conditions						
Name	Turning Radius	Slopes	Fuel (kg)	Map Altitude	Map Limits	NFZs (m)
Level	1st	1st	1st	1st	1st	1st
Min	0	0	0	0	0	0
Max	∞	∞	∞	∞	∞	∞
Minimization Indexes						
Name	Path Length Ratio	Destruction Probability	Detection Probability		Flight Altitude (m)	
Level	2nd	2nd	3rd		3rd	
Min	1	0	0		25	
Max	∞	1	1		5000	

performance of the online and/or multi UAV planner versions highly depend on the performance of its offline single version.

4.1.3. Dominance Evaluation Function

The evolutionary heuristics require a relation operator or dominance evaluation function to rank/sort a set of solutions (trajectories) of a multi-objective constrained problem based on their objective functions. We maintain the method applied in [1]: the non-standard multi-objective Pareto-based dominance function proposed in [8], which supports the setting of goals for each objective function and their organization into priority levels. In short, it determines if a solution dominates/outperforms another in the higher possible level where the decision can be taken, using within each level the basic Pareto dominance definition over the values that fulfill the limits and over the distance to the limited region for the values that do not.

The selected levels and limits[6] are presented in Table 1. The constraint

[6]In our current setup, all the maximum limits except the one associated with the flight

violation conditions are placed in the first priority level. The minimizing objective indexes are located at lower priority levels: the second level contains the path length ratio and destruction probability, and the third level contains the detection probability and flight altitude. With this setup and dominance evaluation function: 1) feasible solutions are automatically better than unfeasible ones, 2) unfeasible solutions are basically compared in a Pareto fashion, and 3) feasible solutions also compete in a Pareto way, firstly according to their lower path length and destruction probability, and secondly according to their lower flight altitude and detection probability.

By means of the use of the selected relation operator we can easily reorganize the priority levels of the objective functions according to different mission setups. Nevertheless, any reorganization changes the behavior of the planner. For instance, in the selected setup the flight altitude and destruction probability directly compete, facilitating the search of a solution with good values in both objectives. However, by prioritizing NFZ fulfillment over flight altitude, the heuristics can be captured in local regions separated by the NFZ regions. Finally, it is worth highlighting that the developed PDM are associated with the selected priority setup presented in Table 1.

4.2. Evolutionary Heuristics

In this section we show the optimization planners under test that we use to illustrate the performance comparison. They are developed using GA,

---

altitude have been released because during the initial experiments carried out for our new PSO and DE based planners we realize that our previous setup introduced discontinuities in the search space, making the transitions through the different regions harder.

PSO and DE to evolve a population of routes coded as spline trajectories. They modify the values of the 3-D waypoints of the list accordingly with the selected 10 objective values, the dominance evaluation method in [8], and some operators characteristic of the randomized heuristic they are rooted in. Hence, they basically differ in the way they evolve their populations. Therefore, the performance comparison that we carry out over the solutions of the different planners can help us to determine if some operators (or their parameterization) are better for the selected problem than others.

The characteristics of all the planners, obtained as variants of GA, PSO, and DE by modifying some of the parameters/operators of each algorithm, are summarized in Table 2. It is worth highlighting that the selection of the modifiable parameters under study was originally bigger[7] but as the changes of the values of a few parameters did not introduce any significant difference in the results of the comparison, we eliminated them to facilitate the graphic visualization[8]. Besides, we do not claim that the selected variable parameters are the only that need to be analyzed. However, as our performance indexes can handle as many evolutionary planner/parameter variants as desired, we just show how our comparison methodology works by applying it to this big set of 36 planner variants.

---

[7]In the PSO case, we also analyzed the influence of maintaining the values of the optimizing waypoints between the limits of the map when the PSO moved any of them outbound by means of an elastic or an inelastic reflection of their values [21]. In the DE case, we also analyzed the influence of the immigrants existence and the possibility of doing dithering instead of jittering in the mutation step [29].

[8]Note that as the number of planner grows, the sizes of the cells of the graphs and the letters of their axis have to be reduced.

Table 2: Analyzed bio-inspired heuristic variants

GA-#: NSGA-II variants			PSO-#: OMOPSO variants					DE-#				
#	Immig.	SBX Xover & Polin. Mut.	#	Stand. Select.	Stand. Param.	Immig.	Stand. Mutat.	#	Mutation Selec.	Mutation Jitter	Bin. Xover	High. Param.
1		✓	1	✓	✓	✓		1	All		✓	✓
2			2	✓	✓	✓	✓	2	All		✓	
3	✓	✓	3	✓	✓			3	All			✓
4	✓		4	✓	✓		✓	4	All			
			5	✓		✓		5	All	✓	✓	✓
			6	✓		✓	✓	6	All	✓	✓	
			7	✓				7	All	✓		✓
			8	✓			✓	8	All	✓		
			9		✓	✓		9	Best		✓	✓
			10		✓	✓	✓	10	Best		✓	
			11		✓			11	Best			✓
			12		✓		✓	12	Best			
			13			✓		13	Best	✓	✓	✓
			14			✓	✓	14	Best	✓	✓	
			15					15	Best	✓		✓
			16				✓	16	Best	✓		

4.2.1. Genetic Algorithms (GA)

We consider that the 4 GA based planners are NSGA-II [5] variants, because they use its tournament selection and recombination method (which ranks the solutions by Pareto fronts and crowding distance). They all differ from NSGA-II, in replacing the basic Pareto dominance definition with the dominance evaluation function of [8].

The 4 GA variants are characterized by changing the values of two parameters. The first parameter is related to the inclusion or not inclusion of immigrants. The second parameter makes the GA planner use the SBX crossover [4] with $\eta_c = 0.9$ and Polinomial mutation [6] with $\eta_m = 0.9$, or the 1-point crossover and 2 level incremental mutation presented in [1]. All the possibilities are presented in the first group of columns of Table 2: the first column (#) shows the labels within the group of the GA variants, and the second (*Immig.*) and third (*SBX Xover & Polin. Mut.*) column respectively mark with ✓ those variants that include immigrants, or use the SBX crossover and Polinomial mutation.

Finally, note that our performance comparison confronts the GA planner presented in [1] (labeled as GA-4 in this paper) with a more standard GA version closer to NSGA-II (labeled as GA-1) and two intermediate configurations. Besides, we only compare 4 GA variants, because we believe GA-4 was already well tuned[9] for the problem in [1] using SRDRP and a PDM, and it is considered in this paper as the baseline to compare against with the other GA planners.

4.2.2. Particle Swarm Optimization (PSO)

The selected 16 PSO based planners are developed around OMOPSO [32]. Again, the use of the goal prioritized dominance evaluation function [8] makes the analyzed variants distinct from OMOPSO.

The 16 PSO variants are characterized by the changing values of 4 parameters, related to the method that selects the global best solutions, the values of their parameters, the inclusion/exclusion of immigrants, and the mutation applied. That is, in some variants the standard OMOPSO tournament (which selects the global best from the global best archive using a ranking of the solutions based on Pareto fronts and crowding distance) is

[9]The comparison in [1] analyzed the influence of the population sizes, number of children, and probabilities of crossover and mutation in the planner performance.

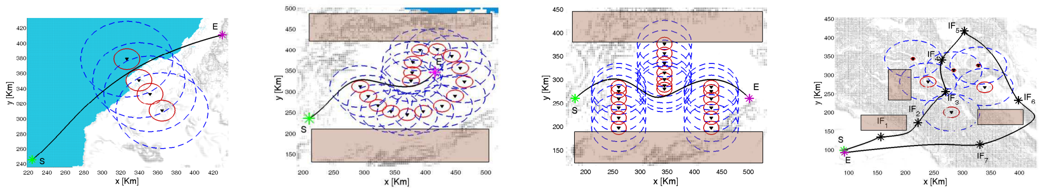
substituted by a random selection of any solution only placed in the best Pareto front of the global best archive. In others, the limits of the standard PSO parameters ($w_{min} = 0.4, w_{max} = 0.4, c_{i,min} = 2.0, c_{i,max} = 2.0$) are reduced to $w_{min} = 0.1, w_{max} = 0.3, c_{i,min} = 0.1, c_{i,max} = 1.0$. We also try variants that include (or do not) immigrants in every generation of the algorithm. Finally, the OMOPSO standard mutation (which maintains 1/3 of the swarm, uniformly mutates another 1/3, and non-uniformly mutates the remaining) is substituted by the 2 level mutation in [1]. All these possibilities are shown in the second group of columns of Table 2: the first column (#) shows the labels within the group of the PSO variants, and the second (*Stand. Selec.*), third (*Stand. Param*), fourth (*Immig.*) and fifth (*Stand. Mut.*) column respectively mark with ✓ those variants that use the standard OMOPSO selection, employ standard PSO parameters, include immigrants, and call the standard mutation method.

Finally, note that PSO-4 is the closer to OMOPSO as it is the variant that uses all the standard methods and standard parameters without immigrants.

4.2.3. Differential Evolution (DE)

The selected 16 DE based planners are developed around the classic DE structure (mutation + crossover) presented in [29]. Again, in all of them we include the goal prioritized dominance evaluation function [8].

The 16 DE variants are characterized by the changing values of 3 parameters in the mutation and 1 parameter in the crossover. The first variation is related to the selection mechanism of the base mutation vector: it can be randomly chosen either among all the solutions or among the solutions of the best Pareto front. The second variation is related to the selection of the values



(a) Scenario 1 (S1) (b) Scenario 2 (S2) (c) Scenario 3 (S3) (d) Scenario 4 (S4)

Figure 2: Scenarios

of the mutation parameter F: for each solution and variable, the values are set to a unique and fixed value (F_{fix}) or the obtained jitter, randomly generating their values using an uniform distribution between F_{min} and F_{max}. The third is related to the possible values of F_{fix}, or F_{min} and F_{max}: we parameterize the planner with either higher ($F_{fix} = 0.8, F_{min} = 0.1, F_{max} = 0.8$) or lower ($F_{fix} = 0.2, F_{min} = 0.1, F_{max} = 0.4$) values. Finally, the last variation depends on the crossover method: the planner uses either binary or 1-point crossover. All these possibilities are presented in the third group of columns of Table 2: the first column (#) shows the labels within the group of the DE variants; the second (*Mutation Selec.*) column respectively identifies with *All* and *Best*, the base mutation vector selection among all or the best; and the third (*Mutation Jitter*), forth (*Bin. Xover*) and fifth (*High. param*) column respectively mark with ✓ those variants that do jittering to select the values of F, use binary crossover, and use high parameterization values for F_{fix}, or F_{min} and F_{max}.

4.3. Scenarios

The scenarios used to analyze the performance of the evolutionary planners are presented in Figure 2. The big dashed blue circles mark each ADU maximum distance of detection, while the small red solid circles represent their maximum shoot down risk distance. Hence, each circle represents the areas that can increment the probabilities of detection and destruction, although they do not necessary do it because these probabilities also depend on other factors, such as the radar cross section and the presence of mountains. The brown rectangles show the NFZs regions. Labels S, E, IF identify the start, end and intermediate fixed points of the UAV. One of the final trajectory obtained by a planner is represented with solid black lines. Although no isometric views of the scenarios are shown in Figure 2, the planners search for 3-D routes.

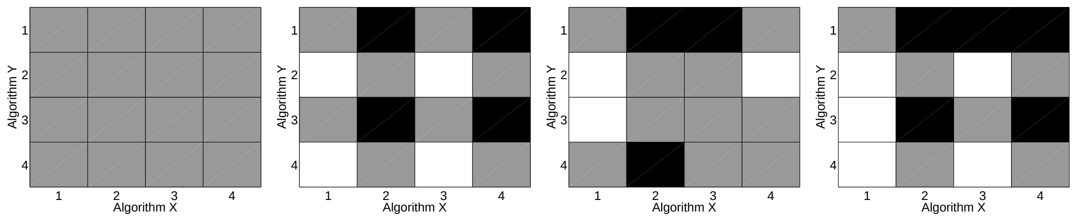
Finding the optimal path in these scenarios is a difficult task for the planners. In scenario S1 (Figure 2a), there is a little corridor (of 8 km) between the upper two ADUs, where the destruction probability is zero, that the planner has to find. In scenarios S2 and S3 (Figures 2b-2c) the planner has to determine the trajectories in a confined area with many ADUs where the destruction probability area has a complex shape. Finally, scenario S4 (Figure 2d) presents a more realistic and less academic setup with multiple NFZ and ADUs located around the intermediate points that the UAV has to visit. As each scenario imposes a different landscape in the constraint/objective functions, it allows us to analyze the utility of the quality indicators under different circumstances.

4.4. Planner Comparison Results

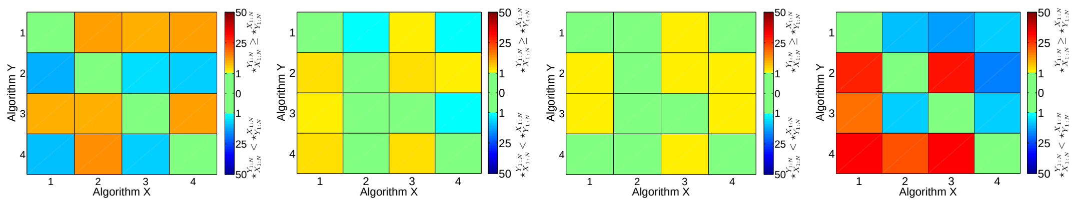
In order to compare the performance of all the planners in each of the scenarios shown in Figure 2, we first generated $N_r = 50$ pairs of initial populations (with 100 solutions each) and immigrant sets (with 250*5 solutions). Then, we obtained for each planner variant and scenario, the results of $N_r = 50$ optimizations, using a population size of 100 solutions, starting each optimization with one of the initial populations, and when required incorporating 5 immigrants of its accompanying immigrant set in each iteration. We ran each optimization during 250 iterations and stored the set of best solutions (best Pareto front) during all the iterations. In other words, for each scenario we used the same initial population and, when required, the same immigrant set (divided in 5 immigrants per iteration) in each of the $N_r = 50$ optimizations of the 36 planners[10].

Next, we calculated the quality index values at different iteration steps for 4 different subsets of planners: GA-based planners, PSO-based planners, DE based planner, and a subset of the best planners within each group. The results of each subset of planner comparison are presented in Figures 3-6, whose graphs are respectively sorted in columns and rows according to the scenarios (S1, S2, S3, S4) and the metrics (SFDRP, IRP, PDM). Due to a lack of space, SFDRP and IRP are only presented at iteration $I = 250$, except for the GA-based planners. SFDRP results were obtained with a 5% confidence. While the blue scale of PDM depends on the scenario, the color scales of the SFDRP and IRP graphs are always the same. Finally, in order to identify the planner variant within each graph, we only show their label (#) in Figures 3-5, and the initials of the type of evolutionary technique (G

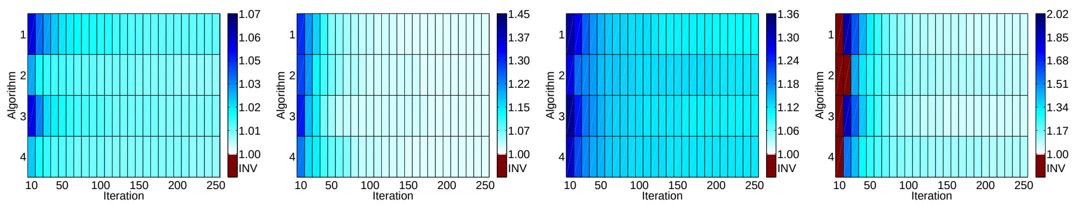
[10]Note that although we only ran 50 optimization with each planner variant, SFDRP does not really compare the solution obtained by two planners equally initialized, as it compares 1 front of A against 1 equally initialized front of B and 49 not equally initialized fronts of B.



(a) S1-SFDRP, $I = 250$ (b) S2-SFDRP, $I = 75$ (c) S3-SFDRP, $I = 75$ (d) S4-SFDRP, $I = 75$



(e) S1-IRP, $I = 250$ (f) S2-IRP, $I = 75$ (g) S3-IRP, $I = 75$ (h) S4-IRP, $I = 75$



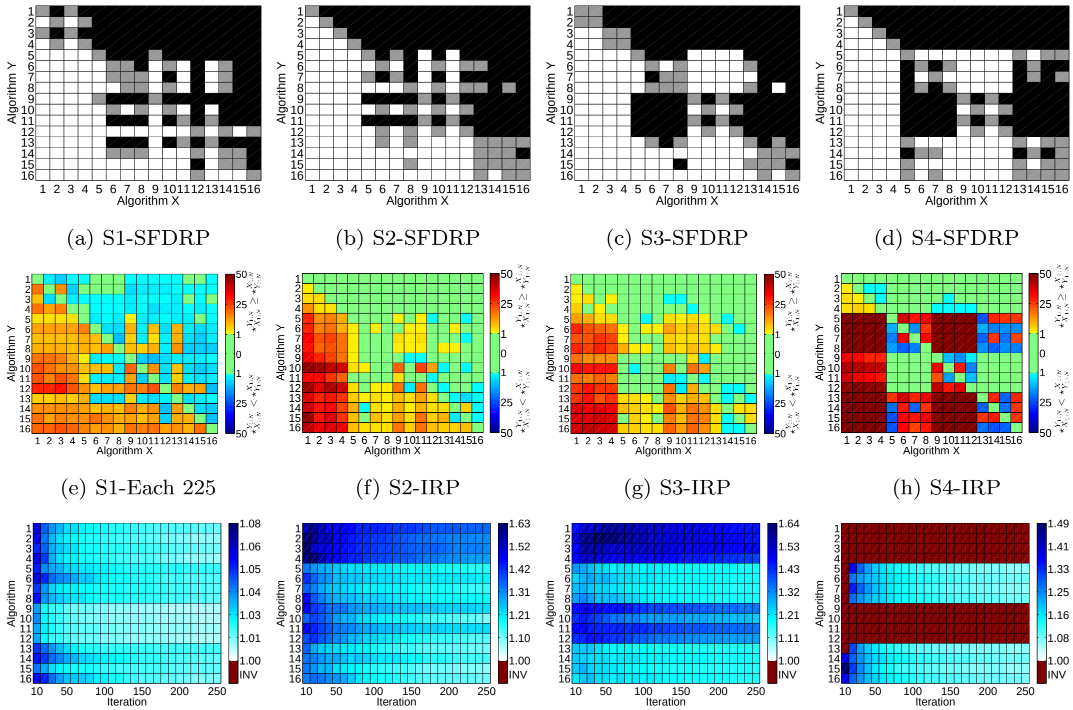
(i) S1-PDM, $I = 250$ (j) S2-PDM, $I = 75$ (k) S3-PDM, $I = 75$ (l) S4-PDM, $I = 75$

Figure 3: GA Comparison

for GA, P for PSO and D for DE) followed by the label (#) in Figure 6. In the following, we analyze the results of the comparisons for the 4 subsets of planners.

4.4.1. Comparison of GA-based Planners

The SFDRP and IRP graphs of the GA planner variants, collected in Figure 3, are related to different iteration values (stated as $I = \#$ in the label of each graph) because their usual behavior at $I = 250$ was the presented in scenario S1: the grey SFDRP graph shows no significant difference among any variant, the IRP graph with similar intensity in the warm and cold colors



(a) S1-SFDRP (b) S2-SFDRP (c) S3-SFDRP (d) S4-SFDRP

(e) S1-Each 225 (f) S2-IRP (g) S3-IRP (h) S4-IRP

(i) S1-PDM (j) S2-PDM (k) S3-PDM (l) S4-PDM

Figure 4: PSO Comparison

shows that the number of times that algorithm Y dominates algorithm X is similar to the number of times that algorithm X dominates algorithm Y. Hence, we analyzed the differences of S2, S3, and S4 at lower iterations ($I = 75$). The lighter rows in their SFDRP graphs for GA-2 and GA-4 in scenarios S2 and S4, and for GA-2 in scenario S3 mean that the selected algorithms are better than the remaining ones. The yellowish IRP rows in scenarios S2 and S3 associated with the best rows of SFDRP imply that although the number of times that the Y algorithm dominates the X algorithm is bigger than the other way around, the number of times that it happens is really small. The reddish IRP rows in scenario S4 imply that the number of times that GA-2 and GA-4 dominate the others is bigger than the case when the same variants dominate the rest in scenario S2.

The PDM graphs show that the constraints were satisfied at iteration 10 in all the scenarios but S4. Besides, the similarity within the rows of each PDM graph at the final iterations show really small differences among the mean normalized path length of the final selected solutions provided by each GA variant.

The overall conclusion extracted from Figure 3 is that all the GA versions have a similar performance at generation 250, although GA-2 and sometimes GA-4, which use the non-standard mutation and crossover of [1], are slightly better in early iterations. Therefore, the GA-based planners can initially benefit from the special genetic operators that we have developed, although they do not necessary benefit from the inclusion of immigrants.

4.4.2. Comparison of PSO-based Planners

The graphs of the PSO planner variants, collected in Figure 4, show some interesting patterns. The SFDRP behavior at iteration 250 changed dramatically every 4 rows and the behavior was roughly maintained every 4 rows (i.e. under the same selection and parameterization). The first 4 rows are usually dark, the second 4 lighter, the third 4 dark or light depending on the scenario, and the last 4 lighter again. The changes within the immigrants (every 2 rows) and mutation (every 1 row) parameterization have a smaller overall influence. Hence, and according to SFDRP, the PSO variants with standard selection and standard parameters (1-4) are the worst, some PSO variants with selection among the best and standard parameters (9-12) are good for some problems, and many PSO variants with lower parameters (5-8

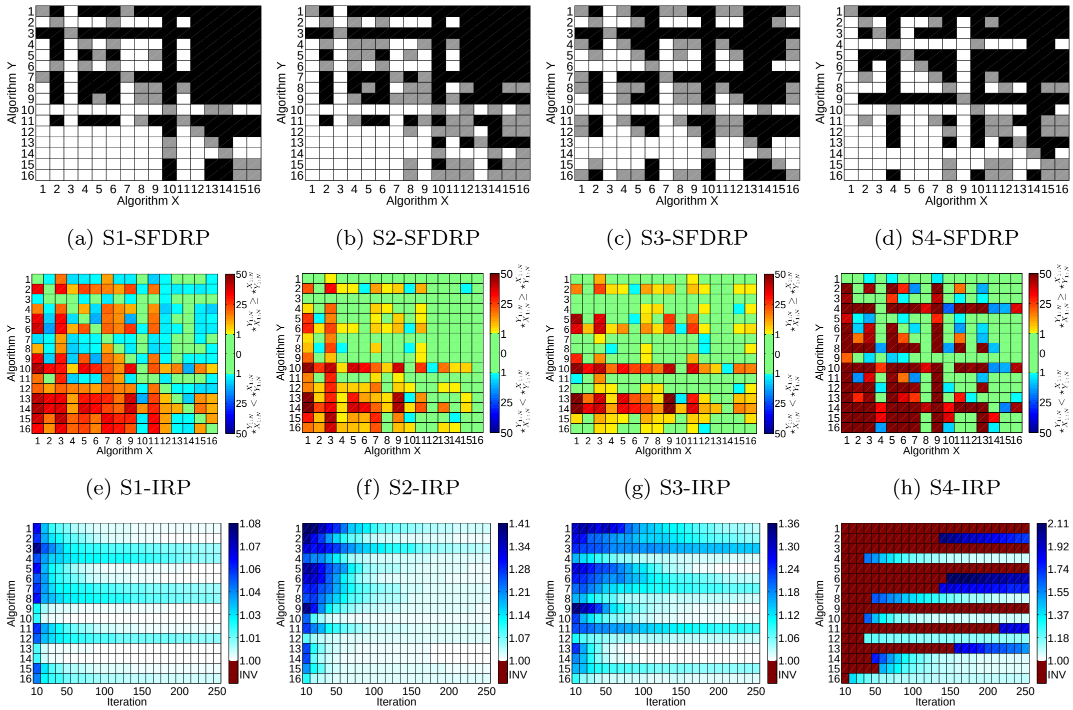
& 13-16) are in general, good. The IRP red columns of S2, S3 and S4 (which also appears lighter in S1) clearly show that the PSO variants with standard selection and standard parameter (1-4) are the worst. Besides, the warmer colors that usually appear in the last 4 rows show that some variants with selection among the best and low parameters (13-16) are at least as good as a few of the 5-12 variants.

The PDM graphs also show the 4 row pattern, which is especially significant in scenario S4, where unconstrained solutions are still found in the last generations for the standard parameter variants (1-4, 9-12).

The overall conclusion extracted from Figure 4 is that the versions with the standard selection and parameters of OMOPSO (1-4) are usually the worst, while some versions with selection among the best and low parameter values (13-16) usually belong to the best group of versions. This can be due to the fact that the standard selection and parameters variants (1-4) create many solutions that do not fulfill the constraints (and in some cases, not even finding any feasible solution in the last generation). On the contrary, using the selection among the best with low parameters (13-16) makes the algorithm slowly improve all the solutions towards feasible optimal regions. Finally, the set of best PSO planners for each scenario is different, although PSO-16 (with selection among the best, low parameter values, no immigrants, and OMOPSO standard mutation) always belongs to the set.

4.4.3. Comparison of DE-based Planners

Although the patterns presented in the graphs of the DE comparison, collected in Figure 5, are weaker than in the graphs of the PSO comparison, they do exist and allow us to identify good DE variants. The last 8 rows of



(a) S1-SFDRP (b) S2-SFDRP (c) S3-SFDRP (d) S4-SFDRP

(e) S1-IRP (f) S2-IRP (g) S3-IRP (h) S4-IRP

(i) S1-PDM (j) S2-PDM (k) S3-PDM (l) S4-PDM

Figure 5: DE Comparison

SFDRP/IRP/PDM are usually lighter/warmer/less red & lighter blue than their corresponding counterparts in the first 8 rows. Hence, selecting the base vector among the best instead of randomly among all the population lets the DE planners find better solutions. The same type of behavior is usually observed among each even variant and its previous (odd) one. This implies that planners with lower mutation parameter values are better than their corresponding counterparts with higher values. The influence of the other parametric variations (jittered/fixed values of F and crossover method) is not that clear.

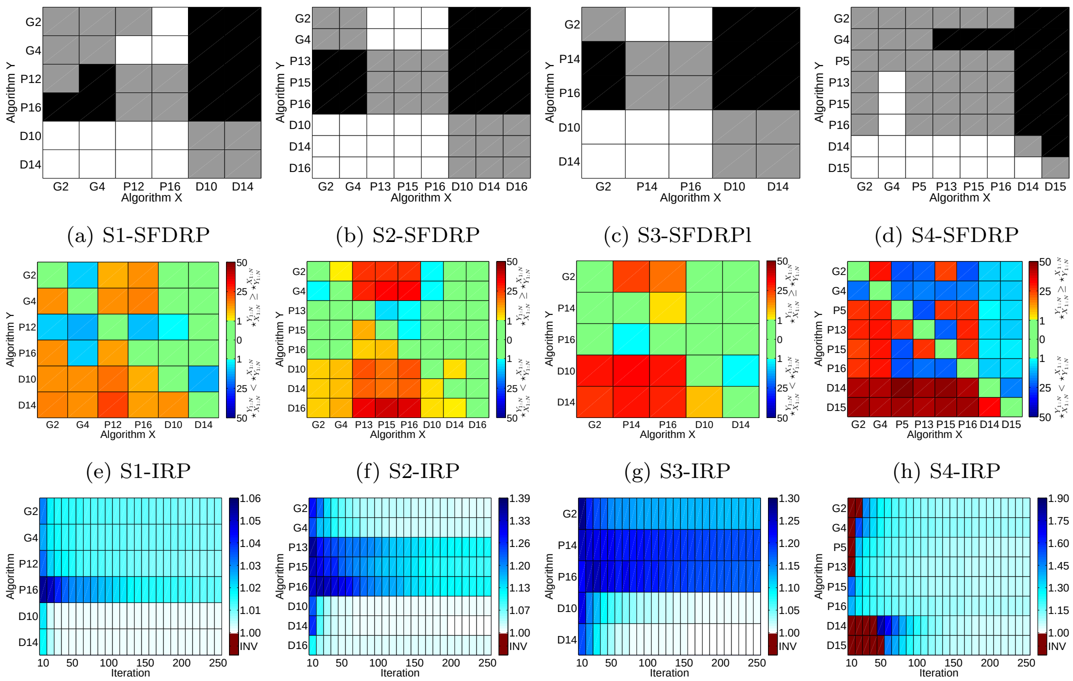
Hence, the overall conclusion is that the versions of DE with low mutation parameter values and selection among the best are usually better than their counterparts. Unsurprisingly, this behavior is somehow similar to the observed behavior in the PSO case, where the best planners are usually found among the PSO-versions with low parameter values and selection of the base vector of the global best among the best[11]. Finally, note that there are two DE variants that can be usually considered among the best ones in all the scenarios. They are DE-10 and DE-14, which select the base vector among the best, use binary crossover, have low mutation parameter values, and fixed or jittered mutation parameters.

4.4.4. Comparison of Selected Planners

The results of the comparison of the best planners selected for each problem are presented in Figure 6. The order is as follows: first the GA-based planners (G#), next the PSO (P#) and finally the DE (D#).

We again observed some really strong patterns in the graphs in this section. The final white/warm rows associated with the DE based-planners in the SFDRP/IRP graphs imply that the selected DE planners usually obtain better solutions than the selected GA and PSO ones. Moreover, the clearly

[11] We can establish some indirect relationship although the mutation parameter values and selection of the base vector, and the low parameters values of PSO and selection of the global best among the best influence differently DE-mutation and PSO-velocity & position update. On one hand, selecting among the best solutions in both cases makes the population tend towards the better (and therefore feasible) solutions. On the other hand, using lower parameter values decrements the possible changes of the solutions. We believe that the combined behavior is helping the planners to reach feasible regions and stay within them.



(a) S1-SFDRP (b) S2-SFDRP (c) S3-SFDRPl (d) S4-SFDRP

(e) S1-IRP (f) S2-IRP (g) S3-IRP (h) S4-IRP

(i) S1-PDM (j) S2-PDM (k) S3-PDM (l) S4-PDM

Figure 6: Selected Planners Comparison

lighter rows of the DE planners in the PDM graph shows that the solutions are significantly better. These graphs also show that 1) in scenarios S1, S2 and S3 the selected GA variants are better than the selected PSO variants, and 2) in scenario S4, the selected PSO variants are better than GA-4 but similar to GA-2.

Therefore, and according to the applied quality indexes that we have used to compare the algorithms, the finally selected DE variants are the best ones, followed by the GA ones in the majority of the cases, and in a few cases by either one GA variant or some PSO ones. Nevertheless, this observation does not imply that PSO is necessary the worst algorithm for the problem or that DE is the best, because other variants/parameterization of any of them could present different behaviors. That is, the presented methodology let us determine the best variants in a given set of possibilities.

5. Conclusions

In order to compare the performance of multiple UAV planners, we propose the combined use of some general based metrics capable of dealing with different relation operation definitions, and some problem specific metrics that consider the final preferences by the experts. Besides, we complement the performance indexes with 3 graphic representation of their results that allow us to compare the performance of many UAV planners at a glance.

The applicability of our comparison approach is analyzed over the results obtained for different scenarios by a comprehensive set of 36 evolutionary based planner variants that tackle the problem in [1]. Its utility is demonstrated by the results of the comparison over 4 different scenarios, which help identify 1) the best usual variants of the whole set of planners for each scenario and 2) the set of parameters whose values improve/degrade the overall performance of each planner type.

Besides, the systematic analysis presented in this study can also be used to carry out performance evolution studies based on SRDRP & IRP graphs at different iterations. This type of study allows us to evaluate if the best variants are always the same during different generations of the algorithm, and when it is not the case, analyze if the application of different techniques during different phases of the algorithm lets the planner find the solution quicker. Furthermore, the results of this type of study can be used to develop new parallel versions of the planners that evolve, using a group of the best identified algorithms, different subpopulations of solutions. As an immediate future study, we are also planning to use fuzzy logic capable of automatically selecting the best algorithm/solution. Rules would be based in the metrics used in this study, which have proven to be useful for a decision maker.

Finally, it is worth highlighting that the generality of the SRDRP & IRP metrics and their graphs, supported by the inclusion of the relation operation used in the evolutionary algorithms in the step that compares sets of solutions of different variants of the algorithm, makes them directly applicable to compare the performance of other planners and/or path planning tasks, or even for other multi-objective optimization problems. Additionally, although the PDM have been designed for the selected problem, the comparison method proposed here can also be easily adapted to consider the experts' preferences for other types of planning tasks or problems.

Appendix A. Comparison of two Sets of Best Solutions

This appendix shows the procedure that compares two sets A and B of solutions returned by the multi-objective meta-heuristics by comparing all the solutions $a^i \in A$ and $b^j \in B$ using the relation (rel) operation applied by the evolutionary algorithm for comparing a^i and b^j by means of the values $f_k(a^i)$ and $f_k(b^j)$ of their violation constraint indicators and objective functions.

The comparison is based on the following facts:

1. In general, the results of comparing solutions a^i and b^j according to rel using $f_k(a^i)$ and $f_k(b^j)$ can be: a^i is better than (dominates) b^j, b^j is

better than (dominates) a^i, a^i and b^j are equal ($\forall k\ f_k(a^i) = f_k(b^j)$), or none is better than the other (i.e. they do not dominate each other). For instance, in the weighted sum function case: a^i is better than b^j when $\sum_k w_k f_k(a^i)$ is better than $\sum_k w_k f_k(b^j)$. None is better when $\exists r f_r(a^i) \neq f_r(b^j)$ but $\sum_k w_k f_k(a^i) = \sum_k w_k f_k(b^j)$. In the basic Pareto dominance definition case, a^i is better than b^j when either a^i strictly dominates b^j ($\forall_k\ f_k(a^i)$ is better than $f_k(b^j)$) or when a^i dominates b^j ($\exists r\ f_r(a^i)$ is better than $f_r(b^j)$ and $\forall k \neq r\ f_k(b^i)$ is not worse than $f_k(b^j)$). They are not comparable when $\exists r\ f_r(a^i)$ is better than $f_r(b^j)$ and $\exists k\ f_k(b^j)$ is better than $f_k(a^i)$.

2. The elements (trajectories) a^i of a set A of best solutions (best Pareto front) returned by the planner are 1) not worse than any other solution found by the algorithm and 2) not better than others in the returned set A.

The generalization of the comparison of the elements a^i and b^j in the sets A and B of best solutions (returned by two different runs of the planner or two different randomized planners) to the comparison of the sets themselves, creates a best set comparison procedure that returns three different possibilities:

1. *Set B is worse than (dominated by) set A.* This happens when $\forall b^j \in B$, 1) $\exists a^i \in A$ where either a^i is better or equal to b^j and 2) $\nexists a^i \in A$ where b^j is better than a^i.
2. *Set A is worse than (dominated by) set B.* When $\forall a^i \in A$, 1) $\exists b^j \in B$ where either b^j is better or equal to a^i and 2) $\nexists b^j \in B$ where a^i is better than b^j).

3. *None is worse than the other,* meaning that they neither dominate nor are dominated by the other set.

This procedure is the basis of the two general quality indexes that we have presented in Section 3.1 to perform a multi-run comparison of sets of best solutions returned by pairs of planners.

Acknowledgements

This work has been supported by the Spanish National grant DPI-2009-14552-C02. The authors want to thank the reviewers for their useful comments and Dr. S. Mittal for his help in preparing the article.

References

[1] E. Besada-Portas, L. de la Torre, J.M. de la Cruz, B. Andres-Toro, Evolutionary trajectory planner for multiple UAVs in realistic scenarios, IEEE Transactions on Robotics 26 (2010) 619–634.

[2] A. Brintaki, I. Nikolos, Coordinated UAV path planning using differential evolution, Operational Research 5 (2005) 487–502.

[3] J.M. de la Cruz, E. Besada-Portas, L. de la Torre, B. Andres-Toro, J.A. Lopez-Orozco, Evolutionary path planner for UAVs in realistic environments, in: Proceedings of Genetic Evolutionary Computation Conference, pp. 1447–1484.

[4] K. Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, Complex Systems 5 (1995) 115–148.

[5] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, in: The Parallel Problem Solving from Nature VI Conference, Springer, 2000, pp. 849–858.

[6] K. Deb, M. Goyal, A combined genetic adaptive search (geneAS) for engineering design, Computer Science and Informatics 26 (1996) 30–45.

[7] G. Farin, Curves and Surfaces for Computer Aided Geometric Design. A practical Guide, New York: Academic, 1988.

[8] C.M. Fonseca, P.J. Fleming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms- part I: A unified formulation, IEEE Transactions on Systems, Man and Cybernetics part A: Syst. Humans 18 (1988) 26–37.

[9] V. Grunert da Fonseca, C. Fonseca, A. Hall, Inferential performance assessment of stochastic optimisers and the attainment function, in: First International Conference in Evolutionary Multi-Criterion Optimization, pp. 213–225.

[10] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning., Addison Wesley, 1989.

[11] M.P. Hansen, M.P. Hansen, A. Jaszkiewicz, A. Jaszkiewicz, Evaluating the Quality of Approximations to the Non-Dominated Set, Technical Report IMM-REP-98-7, Computer Engineering and Networks Laboratory, ETH Zurich, Technical University of Denmark, 1998.

[12] I. Hasircioglu, H.R. Topcuoglu, M. Ermis, 3-d path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms, in: Genetic Evolutionary Computation Conference, pp. 1499–1506.

[13] D. Jia, G. Zheng, M.K. Khan, An effective memetic differential evolution algorithm based on chaotic local search, Information Sciences 181 (2011) 3175 – 3187.

[14] D. Jian, J. Vagners, Parallel evolutionary algorithms for UAV path planning, in: AIAA 1st Intelligent Systems Technical Conference, pp. 1499–1506.

[15] J. Kennedy, R. Eberhart, Particle swarm optimization, in: IEEE International Conference on Neural Networks, pp. 1942–1948.

[16] J. Knowles, L. Thiele, E. Zitzler, A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers, Technical Report TIK-Report No. 214, Computer Engineering and Networks Laboratory, ETH Zurich, Zurich, Switzerland, 2006.

[17] S. Li, X. Sun, Y. Xu, Particle swarm optimization for route planning of unmanned aerial vehicles, in: 2006 IEEE International Conference on Information Acquisition, pp. 1213–1218.

[18] Q. Ma, X. Lei, Application of improved particle swarm optimization algorithm in UCAV path planning, in: International Conference on Artificial Intelligence and Computational Intelligence, pp. 206–214.

[19] H.B. Mann, D.R. Whitney, On a test of whether one of two random

variables is stochastically larger than the other, Annals of Mathematical Statistics 18 (1947) 50–60.

[20] E. Mezura-Montes, M.E. Miranda-Varela, R. del Carmen Gómez-Ramón, Differential evolution in constrained numerical optimization: An empirical study, Inf. Sci. 180 (2010) 4223–4262.

[21] S.M. Mikki, A.A. Kishk, Particle Swarm Optimization: A Physics-Based Approach, Morgan and Claypool, 2008.

[22] S. Mittal, K. Deb, Three-dimensional offline path planning for UAVs using multiobjective evolutionary algorithms, in: IEEE Congress on Evolutionary Computation, volume 7, pp. 3195–3202.

[23] E. Montero, M.C. Riff, On-the-fly calibrating strategies for evolutionary algorithms, Inf. Sci. 181 (2011) 552–566.

[24] L.R. Newcome, Unmanned Aviation: A Brief History of Unmanned Aerial Vehicles, Library of Flight Series, AIAA, Reston, 2004.

[25] I.K. Nikolos, N.C. Tsourveloudis, K.P. Valavanis, Advances in unmanned aerial vehicles, Advances in Unmanned Aerial Vehicles, Springer Verlag, 2007, pp. 309–340.

[26] I.K. Nikolos, N.C. Tsourvelouds, Path planning for cooperating unmanned vehicles over 3-d terrain, Informatics in Control, Automation and Robotics 24 (2009) 153–168.

[27] I.K. Nikolos, K.P. Valavanis, N.C. Tsourveloudis, A.N. Kostaras, Evolutionary algorithm based offline/online path planner for UAV navigation,

IEEE Transactions on Systems, Man and Cybernetics part B: Cybernetics 33 (2003) 898–912.

[28] Y.V. Pehlivanoglu, O. Baysal, A. Hacioglu, Vibrational genetic algorithm based path planner for autonomous UAV in spatial data based environments, in: Proceedings of 3rd International Conference on Recent Advances in Space Technologies, volume 7, pp. 573–578.

[29] K. Price, R. Storn, J. Lampinen, Differential Evolution. A Practical Approach to Global Optimization, Springer, 2005.

[30] B. Qu, P. Suganthan, Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection, Information Sciences 180 (2010) 3170 – 3181.

[31] Y. Qu, Q. Pan, J. Yan, Flight path planning of UAV based on heuristically search and genetic algorithms, in: Proceedings of 31st Annual Conference of IEEE Industrial Electronics Society 2005, pp. 1–5.

[32] M. Reyes, C. Coello, Improving PSO-based multiobjective optimization using crowding, mutation and edominance, in: Evolution Multi-Criterion Optimizaton (EMO 2005), pp. 505–519.

[33] G. Sanders, T. Ray, Optimal offline path planning of a fixed wing unmanned aerial vehicle (UAV) using an evolutionary algorithm, in: 2007 IEEE Congress on Evolutionary Computation, pp. 4410–4416.

[34] D. Shilane, J. Martikainen, S. Dudoit, S.J. Ovaska, A general framework for statistical performance comparison of evolutionary computation algorithms, Information Sciences 178 (2008) 2870–2879.

[35] P.B. Sujit, R. Beard, Multiple uav path planning using anytime algorithms, in: 2009 American Control Conference, pp. 2978–2983.

[36] T.Y. Sun, C.L. Huo, S.J. Tsai, C.C. Liu, Optimal uav flight path planning using skeletonization and pso, in: GECCO, pp. 1183–1188.

[37] R.J. Szczerba, Threat netting for real-time, intelligent route planners, in: Proceedings of the IEEE Symposium Inf., Decision and Control, pp. 377–382.

[38] J. Tian, L. Shen, Y. Zheng, Genetic algorithm based approach for multi-UAV cooperative reconnaissance mission planning problem, Lecture Notes in Computer Science 4203 (2006) 101–110.

[39] J. Tian, Y. Zheng, H. Zhu, L. Shen, A MPC and genetic algorithm based approach for multiple UAVs cooperative search, Lecture Notes in Computer Science 3801 (2005) 399–404.

[40] F. Valdez, P. Melin, O. Castillo, An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms, Applied Soft Computing 11 (2011) 2625 – 2632.

[41] D.A.V. Veldhuizen, G.B. Lamont, Evolutionary computation and convergence to a pareto front, in: Proceedings of the Genetic Programming Conference, pp. 221–228.

[42] Z. Yang, K. Tang, X. Yao, Large scale evolutionary optimization using cooperative coevolution, Inf. Sci. 178 (2008) 2985–2999.

[43] N. Yokoyama, S. Suzuki, Modified genetic algorithm for constrained trajectory optimization, Journal of Guidance, Control and Dynamics 28 (2005) 139–144.

[44] M. Zhang, W. Luo, X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization, Inf. Sci. 178 (2008) 3043–3074.

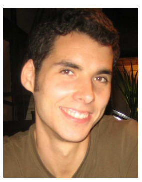
[45] R. Zhang, C. Zheng, P. Yan, Route planning for unmanned air vehicles with multiple missions using an evolutionary algorithm, in: IEEE Third Internatinal conference on Natural Computation, pp. 1499–1506.

[46] C. Zheng, L. Li, F. Xu, F. Sun, M. Ding, Evolutionary route planner for unmanned air vehicles, IEEE Transactions on Robotics 21 (2005) 609–620.

[47] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation 3 (1999) 257–201.

Eva Besada-Portas is an Assistant Professor of the Department of Computer Architecture and Automatic Control of the Complutense University of Spain, Madrid, since 2005. She has been also a post post-doctoral visiting researcher of the Department of Computer Science of the University of New Mexico, USA, from 2006 to 2010. Her research interests are in optimal control, evolutionary algorithms, UAVs, multisensor fusion systems, and machine learning techniques.

Luis de la Torre is a research fellow of the Department of Computer Sciences Automatic and Control of the UNED (Open University of Spain), Madrid, where he begun his Ph.D degree in Systems Engineering and Automatic Control in 2008. He previously joined the Department of Computers Architecture and Automatic Control of the UCM as a research fellow in 2007, while he was finishing his MSc. degree. His research interests are in evolutionary algorithms, UAVs, path planning, remote and virtual laboratories, and distance education.



Alejandro Moreno is a Ph.D. student at the Computer Science and Automation Department of the National University of Distance Learning (UNED), Spain. His research work comprehends the development and analysis of distributed, parallel and interoperable simulation frameworks of heterogeneous systems backed up by Discrete Event System Specification (DEVS) modelling and simulation formalism.

Jose L. Risco-Martin is Assistant Professor at the Computer Architecture and Automation Department of Complutense University of Madrid (UCM), Spain. His research interests focus on design methodologies for integrated systems and high-performance embedded systems, including new modeling frameworks to explore thermal management techniques for Multi-Processor System-on-Chip , computational theory of modeling and simulation with emphasis on Discrete Event Systems Specification (DEVS) and evolutionary computation.