MOGAMR: A Multi-Objective Genetic Algorithm for Real-Time Mission Replanning

Cristian Ramirez-Atencia
Departamento de Ingeniería Informática,
Universidad Autónoma de Madrid,
C/Francisco Tomás y Valiente 11, 28049 Madrid, Spain
Email: cristian.ramirez@inv.uam.es

Gema Bello-Orgaz
Departamento de Ingeniería Informática,
Universidad Autónoma de Madrid,
C/Francisco Tomás y Valiente 11, 28049 Madrid, Spain
Email: gema.bello@uam.es

Maria D. R-Moreno
Departamento de Automática,
Universidad de Alcalá,
Carretera Madrid Barcelona, km 33 600, 28871 Madrid, Spain
Email: mdolores@aut.uah.es

David Camacho
Departamento de Ingeniería Informática,
Universidad Autónoma de Madrid,
C/Francisco Tomás y Valiente 11, 28049 Madrid, Spain
Email: david.camacho@uam.es

***Abstract*—From the last few years the interest and repercussion on Unmanned Aerial Vehicle (UAV) technologies have been extended from pure military applications to industrial and societal applications. One of the basic tasks to any UAV problems is related to the Mission Planning. This problem is particularly complex when a set of UAVs is considered. In the field of Multi-UAV Mission Planning, some approaches have been carried out in the last years. However, there are few works related to real-time Mission Replanning, which is the focus of this work. In Mission Replanning, some changes in the mission, such as the arrival of new tasks, require to update the preplanned solution as fast as possible. In this paper a Multi-Objective Genetic Algorithm for Mission Replanning (MOGAMR) is proposed to handle this problem. This approach uses a set of previous plans (or solutions), generated using an oa liffline planning process, in order to initialize the population of the algorithm, then acts as a complete regeneration method. In order to simulate a real-time system we have fixed a time limit of 2 minutes. This has been considered as an appropriate time for a human operator to take a decision. Using this time restriction, a set of experiments adding from 1 to 5 new tasks in the Replanning Problems has been carried out. The experiments show that the algorithm works well with this few number of new tasks during the replanning process generating a set of feasible solutions under the time restriction considered.**

I. Introduction

The current interest on Unmanned Aerial Vehicle (UAV) capabilities has opened up new commercial applications for the industry. These unmanned vehicles can be used in many domains such as surveillance [1], flight training [2] or disaster and crisis management, since they avoid risking human lives while their manageability permits to reach areas of hard access. Mission Planning for a team of UAVs involves generating tactical goals, commanding structure, coordination, and timing. Nowadays, UAVs are controlled remotely by human operators from Ground Control Stations (GCSs), using rudimentary planning systems, such as following preplanned or manually provided plans.

The Mission Planning Problem (MPP) handles several variables, such as the assignments of tasks to the vehicles in charge of performing them, the sensors used by the vehicles, the flight profiles employed in every path or the orders of these assignments, the assignments of UAVs to the GCSs controlling them. In addition, several constraints must be also considered, such as the fuel available by the vehicles, the feasible paths or the correct orders and times. Besides, the problem can be considered as a Multi-Objective optimization, where the fuel consumption of every UAV, the makespan of the mission, the total cost, the risk of the mission and other objectives must be minimized.

In addition, some events that require a new plan of the current mission could happen during its execution, such as sensor failure or UAV malfunction, a new task arrival or cancellation, or some changes in tasks priorities. This process of Mission Replanning implies real-time rescheduling and re-routing of each UAV involved. Besides, this process requires to be fast enough so the status of the mission being executed does not change before the re-routing of the vehicles.

In this approach, a Multi-Objective Genetic Algorithm (MOGA) based on a previous approach for Mission Planning [3] is reformulated to solve the Mission Replanning Problem (MRP). It considers the Mission Plan being executed and one or more new tasks that must be performed, so the initial population of the Genetic Algorithm (GA) will be generated fixing the tasks already set in the previous plan and randomly setting the new tasks for each individual. On the other hand, in the experimental phase, a time limit of 2 minutes will be set for the algorithm execution, and a study of the optimality of the solutions as the number of new tasks to be resolved increases is carried out.

The rest of the paper is structured as follows. Section II describes the related work concerning mission planning and replanning, and other approaches in Multi-objective Optimization. Section III presents the details of the Mission Replanning Problem. Section IV presents the MOGA approach, the encoding designed and the fitness function implemented to solve the MRP. Section V provides a description of the dataset employed, the setup in MOGA and a complete experimental evaluation of it. Finally, in section VI the conclusions and some future research lines of the work are presented.

II. Related Work

Planning has been an area of research in Artificial Intelligence (AI) for over three decades. The Mission Planning Problem (MPP) can be summed up in finding the correct schedule of resource-task assignments that satisfies the proposed constraints. So a MPP can be formulated as a Constraint Satisfaction Problem (CSP), where the tactic mission is modelled and solved using constraint satisfaction techniques [4]. A CSP consists of a set of variables $V = v_1, ..., v_n$, each one with a finite set of possible values D_i (its domain), and a set of constraints C_i restricting the values that variables can simultaneously take. Moreover, the MPP must consider the time when the tasks in the mission start and end, so a particular class of CSP called Temporal Constraint Satisfaction Problem (TCSP) [5], where variables represent times (time points, time intervals or durations) and constraints represent sets of allowed temporal relations between them, must be considered.

An essential concept in UAV Mission Planning is Mission Replanning. Generally, the Mission Replanning process is event-driven. In this case, an executing mission plan becomes invalid due to some unexpected incidences. These replanning factors could be:

- UAV or sensor failure.
- Urgent (rush) task arrival.
- Task cancellation.
- Due date change (delay or advance).
- Change in task priorities or position of the targets.

On the other hand, there exist two possible strategies for Mission Replanning: *robust plan generation* or *planning repair*. With robust planning, a plan with some contingency rules is provided in preplanning phase, so when a replanning is required it can respond automatically to it. Planning repair consists in providing a new plan and re-routing each vehicle. This strategy could be performed with *partial repair methods* or with *complete regeneration methods*. The first ones just change the assignments involved in the replanning event, while the second ones could make a complete reassignment of the entire mission. Due to the requirements of the real-time problem, the algorithm must be very fast. Thus, the most common approaches are *repair methods*.

There exist few works in this field. Most of them focus on single-UAV missions, while this work focuses on Multi-UAV and Multi-GCS missions. Fukushima and Mita [6] proposed an algorithm for onboard mission replanning using the orthogonal array experiment design approach to solve the problem of repairing the original mission plan without human interactions. Chien et al. [7] studied the use of iterative repair search for spacecraft operations planning in the ASPEN System. A work from Pascarella et al. [8] uses an agent-based approach with a formal model to infer from real-time constraints, by which Mission Planning is dynamically scheduled. Finally, Chen et al. [9] developed a real-time planner for Multi-UCAV, where the initial plan is obtained by a GA and an utility function is used to check and reschedule the task assignments.

There are other approaches for Job-shop scheduling and other problems similar to Mission Planning that consider complete regeneration, using Branch & Bound [10], Tabu Search [11] or GA [12]. This last approach was the faster one, so our approach is also based on these techniques.

GAs have been traditionally used in a large number of different domains, mainly related to optimization problems [13]. These stochastic methods are inspired by natural evolution and genetics, and the complexity of the algorithm depends on the codification and the operations used to reproduce, cross, mutate and select the different individuals of the population.

Several criteria can be taken into account in MPPs for Multi-UAVs to measure the quality of a solution, such as the fuel consumption, the makespan or the cost of the mission, among others. Therefore, it can be interesting to optimize simultaneously different objectives in order to get the best solutions. This type of problems can be solved using Multi-Objective Genetic Algorithms (MOGAs) [14] [15] based on Pareto optimization techniques, which try to find the Pareto Optimal Frontier (POF). The most known approaches are SPEA2 [16] and NSGA-II [17].

Finally, there exist some metrics to evaluate the performance of the algorithm, such as the hypervolume [18] or the generational distance [19]. In this work, a special modification of the hypervolume metric is used. This new metric, applied to a set of solutions with *n* objective variables consists of the *n*-dimensional domain comprised between these solutions (the approximated POF) and the optimal POF of the problem. When the optimal POF is obtained, the volume comprised between the obtained solutions and the optimal POF is 0, and so is the hypervolume. Otherwise, the higher the hypervolume, the worse the approximated POF.

On the other hand, it is also necessary to decide when the algorithm has reached a good POF and stops its execution. There exist several stopping criteria [20] in the literature. One of the most used consists of a comparison function which will stop the execution if the POF remains changeless for a number of generations.

III. The Mission Planning and Replanning Problems

In this section, the MPP will be explained in detail. Then, a CSP modelization of this problem will be presented. Finally, the MRP will be presented as an extension of the previous Mission Planning Problem.

A. Mission Planning Problem

The Multi-UAV Cooperative Mission Planning Problem (MCMPP) [4] can be defined as a number n of ***tasks***, $T = \{t_1, t_2, ...t_n\}$, performed by a team of m ***UAVs***, $U = \{u_1, u_2, ...u_m\}$, at a specific time interval. Each mission should be performed in a specific geographic zone. In addition, there is a number l of ***GCSs***, $G = \{g_1, g_2, ..., g_l\}$, controlling these UAVs. A solution for a mission planning problem should be the assignment of each task to a specific UAV, and each UAV to a specific GCS, ensuring that the mission can be successfully performed.

There are different kind of tasks (e.g. photographing or escorting a target, monitoring a zone, etc.). Some of them can be performed by several UAVs (Multi-UAV), reducing the time needed to perform the task (e.g. mapping an area, or Step & Stare). Each task must be performed in a specific geographic *area* and in a specific *time interval*. In addition, tasks can be carried out using the sensors available (i.e. Electro-optical or Infra-red (EO/IR) cameras, Synthetic Aperture Radar (SAR), Maritime Patrol Radar (MPR), etc.) by the UAVs in the mission. If a UAV has available more than one sensor to accomplish a task assigned to it, the **sensor** to perform the task **should be chosen** for the mission planning.

Figure 1 presents a Mission Scenario with 5 tasks (a surveillance task, a monitoring task and a step & stare task represented in green, a photographing task represented with a camera image, and a tracking task represented with an eye image), 5 UAVs and 2 GCSs. As shown in this figure, the zone of the mission could contain some No Flight Zones (NFZs) represented in red. These zones must be avoided in the trajectories of the UAVs during the mission.

Additionally, the vehicles performing the mission have some features that must be taken into account in order to check if a mission plan is correct: its **initial position**, its **initial fuel**, its **autonomy** or maximum flight time, its **range** or maximum flight distance, its **cost** per hour of usage, its **available sensors**, and one or more **flight profiles**. A vehicle's flight profile specifies at each moment its **speed**, its **fuel consumption ratio** and its **altitude**.

When a task is assigned to a vehicle, it is necessary to compute the **duration of the path** between the zone of the UAV's departure and the zone of the task. If a task is the last one assigned to a vehicle, in addition, the **duration of the return** from this last task to the base must be calculated. In order to compute these durations, it is necessary to know which of the UAV's flight profiles will be used, providing the fuel consumption ratio, speed and altitude as previously mentioned. For this reason, in these cases, the **flight profiles** used must also be **assigned** to solve the mission.

Finally, a mission could have some time and vehicle dependencies between different tasks. **Vehicle dependencies** consider if two tasks must be assigned to the **same UAV** or **different UAVs**. Moreover, we consider **time dependencies** given by the **Allen's Interval Algebra**[21].

B. CSP Modelling of the Mission Planning Problem

As previously mentioned, one way to model the MCMPPs is using CSPs, since we need to find the correct schedule of resource-task assignments which satisfies the proposed constraints.

The modelization shortly presented here was explained in more detail in a previous work [3]. Basically, we need to provide the variables and the constraints. The variables of the MCMPP are included into several sets of variables:

- **Assignments of tasks to UAVs**. As some tasks could be Multi-UAV, these variables are represented as a binary array of size $n \times m$.
- **Orders**. Define the sequence in which each UAV performs the tasks assigned to it. These variables are necessary when start and end times of tasks are not fixed.
- **Assignments of UAVs to GCSs**. We need to assign the control of each UAV to each GCS for monitoring the mission.
- **Path Flight Profiles**. Set the flight profile that the vehicle must take for the path performance.
- **Return Flight Profiles**. It is similar to the previous set of variables but for the return path of each UAV.
- **Sensor used in the task performance**. These variables set the sensor of the vehicle that will be used during the task performance. It will be necessary to consider these variables just in the case that the vehicle performing the task has several sensors that could perform that task.

The CSP considers several constraints related to the different complexity issues explained in the previous section:

- **Sensor constraints**. Check if a UAV has the sensors needed to perform its assigned tasks.
- **Order constraints**. Assure that the values of the order variables are less than the number of tasks assigned to the UAV performing that task.
- **GCS constraints**. Assure that the GCSs assignments are correct, UAVs are assigned to GCSs able to control them, and they are located within the GCS coverage area.
- **Temporal constraints**. Assure the consistency of all the times and durations involved in the mission planning.
- **Dependency Constraints**. These constraints are related to the time and vehicle dependencies mentioned in the previous section.
- **Autonomy constraints**. Assure that the total flight time for each vehicle is less than its vehicle autonomy time.
- **Distance constraints**. Assure that the distance traversed by each vehicle is less than its range.
- **Fuel constraints**. Assure that the fuel consumed by each vehicle is less than its initial fuel.

There are several constraints implicit inside these constraints, such as avoiding NFZs, which are considered when computing temporal constraints. In addition, it is necessary to consider the route plan for every path performed by a UAV when going to the task zone.

C. Mission Replanning Problem

Now that the MPP has been described and modelled, the MRP can be extended from it. The MRP considers the same constraints as the Mission Planning. The main difference lies in the status of the tasks:

- Some tasks would have finished when replanning is called, so they will be erased from the problem.

Fig. 1: Mission with 5 tasks (1 of them Multi-UAV), 5 UAVs and 2 GCSs.

- Some tasks would have not started yet.
- Some tasks would have started but still not ended.

In this approach, we consider that these tasks in execution will remain fixed in the assignments.

Besides, we must also consider the position of the UAV when replanning. In this case, we will take the future position of the UAVs, and the status of the mission in general, after the call to the planner.

Due to the complexity of the problem, a partial repair of the mission plan may not be the optimal option in many cases. For example, a new task may appear next to a UAV that has many other tasks assigned. Then, this task could not be assigned to this UAV due to fuel or time constraints unless this UAV deallocates some of its previously assigned tasks. For this reason, a complete regeneration method is employed, as we will explain in the following section.

IV. Multi-Objective Genetic Algorithm for Mission Replanning

In this approach, a previous hybrid version based on MOGA and CSP [3] for the MCMPP is extended to solve the MRP. The CSP is computed inside the fitness function of the MOGA, checking that solutions fulfil all the constraints. This algorithm is also based on the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) for the Multi-Objective consideration.

The Mission Replanner will input a new problem considering just the tasks that have not been performed yet, the position of every vehicle (2 minutes in advance as explained in previous section) and the start time of use of these vehicles. This start time will be 0 if the vehicle is not performing any task at the replanning time point, or the end time of the task that it is performing at the replanning time point.

In addition, the algorithm will also receive the previous plan, which will be used to create the initial population of the MOGA. This population will fix the genes values with the ones provided by this plan, and randomly initialize the other gene values (the ones related to the new tasks).

The following subsections will present the encoding of this approach, the fitness function used, the initializer of the population and the algorithm itself.

A. Encoding

The encoding of this new approach consists of six different alleles representing the features described in the previous section. Figure 2 shows an example of this representation for a mission with 5 tasks, 3 UAVs and 2 GCSs.

1) **UAVs assigned to each task**. If the T_i task is Multi-UAV, then this cell contains a vector representing the different UAVs assigned to this task, as shown in Figure 2 for tasks T_1, T_3 and T_4.
2) **Permutation of the task orders**. These values indicate the absolute order of the tasks. It is only used if there are several tasks assigned to the same UAV (e.g. in Figure 2, UAV 1 performs tasks 1, 4 and 5 in this order).
3) **GCSs controlling each UAV**. What GCSs is assigned to monitor what UAV.
4) **Flight profiles used for each UAV to perform an assigned task**. As in the first allele, some of the cells could contain a vector if the corresponding task is performed by several UAVs.
5) **Sensors used for the task performance** by each UAV.
6) **Flight Profiles used by each** UAV to return to the base.

In Figure 2 example, according to the task assignments and the order permutation, we can see that UAV 1 performs tasks

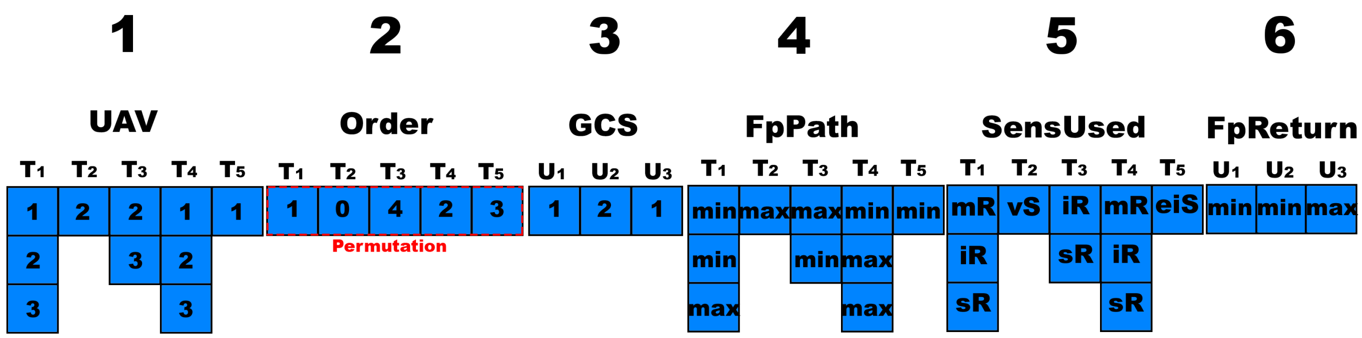

Fig. 2: Example of an individual that represents a possible solution for a problem with 5 tasks, 3 UAVs and 2 GCSs.

1, 4 and 5 in this order; UAV 2 performs tasks 2, 1, 4 and 3; and UAV 3 performs tasks 1, 4 and 3. On the other hand, according to GCSs assignments, UAVs 1 and 3 are controlled by GCS 1, while UAV 2 is controlled by GCS 2. Furthermore, looking at the Flight profiles per task, UAV 1 uses minimum consumption flight profile for all its assigned tasks; UAV 2 uses minimum consumption profile for task 1, and maximum speed profile for the rest of tasks, and UAV 3 uses minimum consumption profile for task 3, while maximum speed profile for the rest of tasks. Regarding the sensors used, it can be seen that task 1 is performed by UAV 1 using MPR radar (mR) sensor, while UAV 2 uses an ISAR radar (iR), and UAV 3 uses a SAR radar (sR); task 2 is performed using EO/IR sensor (eiS), etc. Finally, the last allele represents that UAVs 1 and 2 use minimum consumption profile for their return path, while UAV 3 uses maximum speed profile.

B. Fitness function

The fitness function checks that all constraints are fulfilled for a given solution. Then, the fitness works as a multi-objective function minimizing the objectives of the problem:

- The **number of UAVs** used in the mission.
- The **total fuel consumption**.
- The **total cost** of the mission.
- The end time of the mission or **makespan**.
- The **risk** of the mission, which is computed as an average percentage indicating how risky the mission is. We consider three risk factors: UAVs that finish the mission with low fuel, UAVs that fly near to the ground and UAVs that fly close between them.

C. Preplanned Population Initializer

The initializer of the population used in this approach receives the previous plan, which consists of an individual with the same encoding explained in the previous subsection, but with some empty assignments. The initializer will clone this individual as many times as the size of the population. Then, for each individual, the empty genes are assigned random values. In the case of the permutation, the new tasks are inserted in a random position of the permutation.

D. Algorithm

The Multi-Objective Genetic Algorithm for Mission Replanning (MOGAMR) is presented in Algorithm 1. In this new approach, firstly a **preplanned initialization** is performed in order to obtain the initial population (Line 1). Then this initial population evolves using a MOGA. The evaluation of the individuals is performed by a fitness function (Lines 7-11), and the NSGA-II approach [16] (Line 16) is used for the Multi-Objective elitism selection. This selection is based first on non-dominated rank (according to the POF partition), and secondly, for solutions having the same rank, on the crowding distance. Then, a tournament selection (Line 19) is used to provide the individuals that will be chosen to apply the genetic operators. The crossover operator (line 20) consists of an extension of the 2-point crossover and the Partially-Matched Crossover (PMX) independently applied to each allele of the chromosome. The mutation operator (lines 21-22) is also an extension of the uniform and the insert mutation applied to each allele.

Finally, the stopping criteria designed for this algorithm (Lines 12-15) compares the non dominated solutions obtained so far at each generation with the solutions from the previous one. If the solutions remains unchangeable for a number of generations, then the algorithm will stop and return these solutions as an approximation of the POF.

V. Experimentation

In this section, the experimental setup and the problem used in the experimental phase is presented. Then, using a previously planned mission, a series of replanning problems are proposed considering for each problem a higher number of new tasks to be replanned. For each one of these problems, the Mission Replanner is used to find an approximated set of solutions within the 2 minutes limit. In order to compare the results obtained, the Mission Replanner is used a second time with no time limit in order to find the optimal POF, and the hypervolume between these two set of solutions is computed.

A. Experimental Setup

For the experiments below, the mission from Figure 1 is used as the initial mission. This mission has been planned using the MOGA approach from a previous work [3]. Then, one

Algorithm 1: Multi-Objective Genetic Algorithm for Mission Replanning

Input: A mission $M = (T, U, G)$ where T is a set of tasks to perform denoted by $\{t_1, \ldots, t_n\}$, U is a set of UAVs denoted by $\{u_1, \ldots, u_m\}$ and G is a set of GCSs denoted by $\{g_1, \ldots, g_l\}$. The set of objectives O and their upper bounds $\overline{M} = \{M_i >> avg(o_i)\}$. A mission plan $P = (TA, OA, GA, PFpA, SA, RFpA)$, where TA are the task assignments of the plan, OA are the orders (permutation) assignments, GA are the GCS assignments, $PFpA$ are the path flight profile assignments, SA are the sensor assignments and $RFpA$ are the return flight profile assignments. And positive numbers time limit $maxTime$, elitism μ, population size λ, $mutprobability$ and $stopGeneration$.

Output: POF obtained with best solutions

$S \leftarrow$ PreplannedPopulationInitializer(P,λ)
$initTime \leftarrow now()$
$convergence \leftarrow 0$
$pof \leftarrow \emptyset$
while $now() - initTime \leq maxTime \land convergence < stopGenerations$ **do**
    $F \leftarrow \emptyset$
    for $j \leftarrow 1$ ***to*** λ **do**
        if $CSP_{Check}(S_j)$ **then**
            $F \leftarrow MultiObjectiveFitness(S_j)$
        else
            $F \leftarrow \overline{M}$
    $newpof \leftarrow createPOF(S)$
    if $newpof = pof$ **then**
        $convergence \leftarrow convergence + 1$
    $pof = newpof$
    $Sbest \leftarrow SelectNSGA2Best(\mu, F)$
    $newS \leftarrow Sbest$
    for $j \leftarrow \mu$ ***to*** λ **do**
        $p1, p2 \leftarrow TournamentSelection(Sbest)$
        $i1, i2 \leftarrow Crossover(p1, p2)$
        $i1 \leftarrow Mutation(i1, mutprobability)$
        $i2 \leftarrow Mutation(i2, mutprobability)$
        $newS \leftarrow newS \cup \{i1, i2\}$
    $S \leftarrow newS$
return pof

of the solution plans obtained has been selected. This plan is represented in Figure 3. In this assignment, MALE is assigned tasks Surveillance, Monitoring and Step & Stare Searching, in this order; URAV 1 is assigned the Identification/Tracking task 1; URAV 2 is assigned Step & Stare Searching; URAV 3 is assigned Identification 2, and URAV 4 is assigned Step & Stare Searching. As can be seen, Step & Stare Searching (which is a Multi-UAV task) is divided and executed by 3 UAVs. On the other hand, GCS controls MALE, URAV 1 and URAV 3, while GCS2 controls URAV 2 and URAV 4.

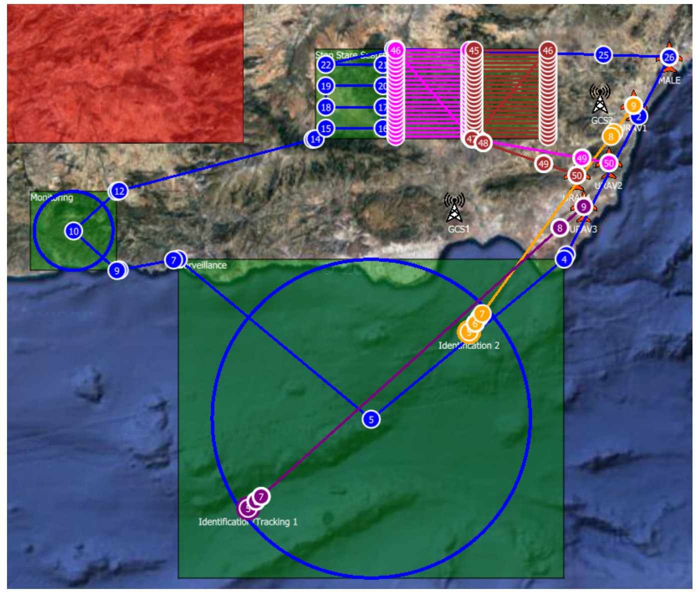

Fig. 3: Plan selected for Mission from Figure 1

Now, using this plan, 5 MRPs are proposed where the number of new tasks is gradually incremented from 1 to 5. These new tasks, which will consist of Photographing targets, are spread along the Mission Scenario.

On the other hand, the setup of the algorithm is presented in Table I. The size of the initial population will be based on the number of new tasks added to the mission (n_{new}), the number of UAVs (m) and the number of GCSs (l).

TABLE I: Experimental setup for the MOGAMR.

Mutation probability	0.1
Time Limit	2 min
Population size (λ)	$n_{new}^2 * m + m^2 * l$
Elitism size (μ)	0.1*λ
Stopping criteria generations	10

B. *Experimental Results*

Using the five problems designed, we run the MOGAMR with two approaches: 1) the presented in the experimental setup, with a time limit of 2 minutes, and 2) an unlimited time approach, which will be used to compare the results obtained by the 2 minutes approach. With the solutions obtained from each approach, we compute the Hypervolume comprised between these two sets of solutions. When the hypervolume is 0, it means the solutions are equal, so the 2 minutes approach obtains the optimal solutions despite the time limit. Otherwise, if the hypervolume is positive, it means some optimal solutions have not been found in the 2 minutes approach. Table II presents the results obtained for these experiments, including the hypervolume, the number of generations needed by each approach and the execution time needed by the unlimited

TABLE II: Results obtained using the MOGAMR with plan from Figure 3 and adding 1-5 new tasks. The hypervolume represents the Hypevolume comprised between the solutions obtained by the 2-time limit approach and the unlimited approach.

New tasks	Hypervolume	No. Generations 2-min approach	No. Generations Unlimited approach	Time Unlimited approach
1	0	15	16	0min 54s
2	0	43	43	1min 45s
3	0	52	67	3min 24s
4	1.58	43	64	5min 44s
5	6.43	32	79	8min 12s

approach. The experiments have been run in an Intel Core i5-6200 2,3 GHz with 8 cores and 16GB DDR3 RAM.

As can be observed in the table, for 1 and 2 new tasks, the algorithm converges in less than 2 minutes, so the results obtained by the two approaches are the same. For 3 new tasks, it can be seen that the algorithm does not converge in 2 minutes, but the results obtained in this time are not outperformed when the algorithm converges. On the other hand, it can be appreciated that for 4 and 5 tasks, better solutions have been obtained in the unlimited approach. So it can be concluded that the MOGAMR is appropriated with a time limit of 2 minutes for 3 or less new tasks.

VI. Conclusions and Future Work

In this paper, a previous approach of Multi-Objective Genetic Algorithm for Mission Planning has been extended to be used in Mission Replanning Problems, so our previous solution to the Mission Planning problem has been extended from a static perspective to a dynamic one. The new approach takes the executing plan in order to initialize the population of the Genetic Algorithm based on the previous assignments. The fitness function designed checks that all constraints are fulfilled, and if so, optimizes several objectives, including the makespan, the fuel consumption, the total cost or the risk of the mission.

The initial experiments, carried out using several problems with an increasing complexity in terms of number of tasks that need to be planned, show that the algorithm works well when only some few new tasks are added. However, the computation time grows very fast when the number of new tasks increases, as expected. This generates a serious problem to include this kind of approaches in real time systems, where a fast generation of solutions is needed. For this reason, a deep study on different methods and heuristics that could be applied to outperform the results obtained, and to improve the time computation (under a time limitation), will be done in the near future. Finally, we will also focus on developing a Decision Support System (DSS) for this problem, in order to select one solution among those obtained by the algorithm according to some quality metrics and the GCS operator profile.

Acknowledgment

This work has been supported by the next research projects: Airbus Defence & Space (FUAM-076915), EphemeCH (TIN2014-56494-C4-4-P) Spanish Ministry of Science and Education and Competitivity, CIBERDINE S2013/ICE-3095, both under the European Regional Development Fund FEDER, and RiskTrack (JUST-2015-JCOO-AG-723180). The authors would like to acknowledge the support obtained from Airbus Defence & Space, specially from Savier Open Innovation project members: José Insenser, César Castro and Gemma Blasco.

References

[1] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain, “Cooperative forest fire surveillance using a team of small Unmanned Air Vehicles,” *International Journal of Systems Science*, vol. 37, no. 6, pp. 351–360, 2006.

[2] V. Rodríguez-Fernández, H. D. Menéndez, and D. Camacho, “Automatic profile generation for uav operators using a simulation-based training environment,” *Progress in Artificial Intelligence*, vol. 5, pp. 37–46, 2015.

[3] C. Ramirez-Atencia, G. Bello-Orgaz, M. D. R-Moreno, and D. Camacho, “Solving Complex Multi-UAV Mission Planning Problems using Multi-objective Genetic Algorithms,” *Soft Computing*, pp. 1–18, 2016.

[4] ——, “Performance Evaluation of Multi-UAV Cooperative Mission Planning Models,” in *Computational Collective Intelligence*. Springer International Publishing, 2015, pp. 203–212.

[5] E. Schwalb and L. Vila, “Temporal constraints: A survey,” *Constraints*, vol. 3, no. 2-3, pp. 129–149, 1998.

[6] Y. Fukushima and M. Mita, “A new approach to autonomous onboard mission replanning using orthogonal array design,” in *Space Mission Challenges for Information Technology (SMC-IT), 2011 IEEE Fourth International Conference on*. IEEE, 2011, pp. 43–50.

[7] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, “Using iterative repair to increase the responsiveness of planning and scheduling for autonomous spacecraft,” NASA Technical Reports Server, Tech. Rep., 1998.

[8] D. Pascarella, S. Venticinque, and R. Aversa, “Autonomic agents for real time uav mission planning,” in *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*. IEEE, 2013, pp. 410–415.

[9] X. Chen, T. Tang, and L. Li, “Study on the real-time task assignment of Multi-UCAV in uncertain environments based on genetic algorithm,” in *2011 IEEE 5th International Conference on Robotics, Automation and Mechatronics (RAM)*. IEEE, 2011, pp. 73–76.

[10] H.-H. Wu and R.-K. Li, “A new rescheduling method for computer based scheduling systems,” *International journal of production research*, vol. 33, no. 8, pp. 2097–2110, 1995.

[11] F. Corman, A. DAriano, D. Pacciarelli, and M. Pranzo, “A tabu search algorithm for rerouting trains during rail operations,” *Transportation Research Part B: Methodological*, vol. 44, no. 1, pp. 175–192, 2010.

[12] C. Bierwirth and D. C. Mattfeld, “Production scheduling and rescheduling with Genetic Algorithms,” *Evolutionary computation*, vol. 7, no. 1, pp. 1–17, 1999.

[13] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Cambridge, MA, USA: MIT Press, 1992.

[14] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, “Multiobjective evolutionary algorithms: A survey of the state of the art,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[15] E. Zitzler, M. Laumanns, and S. Bleuler, “A tutorial on evolutionary multiobjective optimization,” in *Metaheuristics for multiobjective optimisation*. Springer, 2004, pp. 3–37.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist Multiobjective Genetic Algorithm: NSGA-II," *Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[17] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," in *Evolutionary Methods for Design, Optimisation, and Control*, 2002, pp. 95–100.

[18] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration," in *Evolutionary multi-criterion optimization*. Springer, 2007, pp. 862–876.

[19] D. Van Veldhuizen, G. B. Lamont *et al.*, "On measuring Multiobjective Evolutionary Algorithm performance," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1. IEEE, 2000, pp. 204–211.

[20] T. Wagner, H. Trautmann, and L. Martí, "A taxonomy of online stopping criteria for Multi-objective Evolutionary Algorithms," in *Evolutionary Multi-Criterion Optimization*. Springer, 2011, pp. 16–30.

[21] J. F. Allen, "Maintaining knowledge about temporal intervals," *Communications of the ACM*, vol. 26, no. 11, pp. 832–843, 1983.