A Weighted Penalty Fitness for a Hybrid MOGA-CSP to solve Mission Planning Problems

Cristian Ramirez-Atencia[1], Gema Bello-Orgaz[1],
Maria D. R-Moreno[2], and David Camacho[1]

[1] Departamento de Ingeniería Informática, Universidad Autónoma de Madrid,
C/Francisco Tomás y Valiente 11, 28049 Madrid, Spain
cristian.ramirez@inv.uam.es, {gema.bello,david.camacho}@uam.es
aida.ii.uam.es
[2] Departamento de Automática, Universidad de Alcalá,
Carretera Madrid Barcelona, km 33 600, 28871 Madrid, Spain
mdolores@aut.uah.es

Abstract. Unmanned Aerial Vehicles (UAVs) are currently booming due to their high number of potential applications. In Mission Planning problems, several tasks must be performed by a team of UAVs, under the supervision of one or more Ground Control Stations (GCSs). In our approach, we have modelled the problem as a Constraint Satisfaction Problem (CSP), and solved it using a Multi-Objective Genetic Algorithm (MOGA). The algorithm has been designed to minimize several variables of the mission such as the fuel consumption or the makespan. In addition, the fitness function takes a new consideration when solutions are not valid. It uses the number of constraints fulfilled for each solution as a weighted penalty function. In this way, the number of constraints fulfilled is maximized in the elitism phase of the MOGA. Results show that the approach outperforms the convergence with respect to previous results.

Keywords: Unmanned Aerial Vehicles, Mission Planning, Constraint Satisfaction Problems, Multi-Objective Genetic Algorithm

1 Introduction

The current boom of Unmanned Aerial Vehicles (UAVs) capabilities has opened up new commercial applications for the industry. These vehicles can be used in many domains such as surveillance [2], flight training [10] 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.

Multi-UAV Cooperative Mission Planning Problems (MCMPPs) have a lot of requirements that need to be considered in order to coordinate all the UAVs.

Actas de la XVII Conferencia de la Asociación Española para la Inteligencia Artificial, pp. 305-314

These requirements generate search graphs that need huge process capabilities to find a solution. Nevertheless, there exists some works on UAV Mission Planning using Temporal Action Logic (TAL) for reasoning about actions and changes [4], Markov Decision Process (MDP) and dynamic programming algorithms [5], or hybrid Partial-Order Forward-Chaining (POFC) [6], among others.

Other modern approaches formulate the tactic mission planning problem as a Constraint Satisfaction Problem (CSP) [1]. A CSP consists of a set of variables (each one with its own domain) and a set of constraints restricting the values that variables can simultaneously take. So MCMPPs could be modelled as a CSP guided to find a correct schedule of UAV-task assignments.

In addition, Multi-UAV missions usually require the use of several GCSs for controlling all the UAVs involved. This generates a new Multi-GCS approach that makes the problem even more complex. Besides there are several parameters which can be used to define the quality of a solution, so an option to solve this type of problems could be using Multi-Objective Evolutionary Algorithms (MOEAs). In this work, we have extended a previous work [8] in order to design and implement a Multi-Objective Genetic Algorithm (MOGA) to solve the MCMPP. In this sense, the fitness function has been extended as follows. If all the constraints are fulfilled in the solution plan, then a Pareto-based function is used to optimize different quality parameters. Otherwise, the number of constraints fulfilled is used as a weighted penalty fitness. This heuristic intends to decrease the number of generations needed to converge to an optimal Pareto Optimal Frontier (POF) and, therefore, reduce the runtime of the algorithm as the experiments show.

The rest of the paper is structured as follows. Section 2 describes how a UAV Mission is defined. Section 3 presents the hybrid MOGA-CSP algorithm developed with the new fitness function considering the number of constraints. In Section 4 the new approach is tested against several mission datasets and compared with a previous approach. Finally, the last section presents the final analysis and conclusions of this work.

2 UAV Mission Planning

The MCMPP [9] can be defined as a number n of ***tasks***, $T = \{t_0, t_1, ...t_n\}$, performed by a team of m ***UAVs***, $U = \{u_0, u_1, ...u_m\}$, at a specific time interval. Each mission should be performed in a specific geographic zone. In addition, in this approach, there exist a number l of ***GCSs***, $G = \{g_0, g_1, ..., g_l\}$, 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 exists different kind of tasks (e.g. photographing or escorting a target, monitoring a zone, etc.). Some of them could be Multi-UAV, i.e. they are performed by several UAVs (e.g. mapping an area, also called Step & Stare) reducing the time needed to perform the task. Tasks can be carried out thanks to the sensors available (i.e. Electro-optical or Infra-red (EO/IR) cameras, Syn-

thetic Aperture Radar (SAR), Maritime Patrol Radar (MPR), etc.) by the UAVs performing the mission. In addition, each task must be performed in a specific geographic *area* and in a specific *time interval*.

In Figure 1, a Mission Scenario with 8 tasks (represented in green), 5 UAVs and 3 GCSs is presented. As can be seen 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.

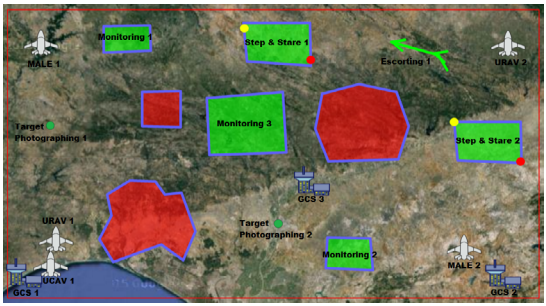

Fig. 1: Mission with 8 tasks (2 of them Multi-UAV), 5 UAVs and 3 GCSs.

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 start. 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 the **same UAV** or **different UAVs**. Moreover, we consider **time dependencies** given by the **Allen's Interval Algebra**.

3 Proposed MOGA-CSP Algorithm

To deal with the huge amount of constraints and the big search space of the problem, a hybrid approach based on MOGA and CSP is proposed for the MCMPP. There exists several approaches for hybridizing MOGAs and CSPs [7].

In our approach, the CSP is considered inside the fitness function of the MOGA, checking that the solutions fulfil all the constraints. When a solution does not fulfil some constraint, the number of constraints checked until fail in the propagation phase of the CSP is used as a weighted penalty function in order to reproduce the solutions fulfilling the highest number of constraints. This methodology will help the algorithm to converge faster when the space of solutions is very small compared to the search space, causing the randomly generated initial population to maybe not reach any valid solution.

3.1 Encoding

The encoding of this new approach consists of six different alleles representing the features described in the previous section. These alleles are divided into three groups according to their applicability of operators:

1. The first group of alleles is composed of three alleles of size n each one:
   - **UAVs assigned to each task**.
   - **Flight profiles used for each UAV to each assigned task**.
   - **Sensors used for the task performance** by each UAV.

   If the T_i task is Multi-UAV, then the corresponding cell of each allele contains a vector representing the different UAVs, flight profiles or sensors assigned to this task. In Figure 2, an example of this group for a mission with 5 tasks and 3 UAVs is presented. In the reproduction phase of the algorithm, this group of alleles is applied a 2-point crossover and an uniform mutation.

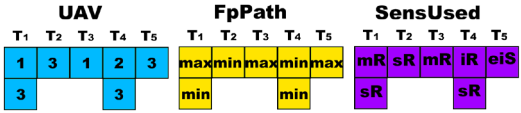

Fig. 2: Example of the first group of alleles from an individual representing a possible solution for a problem with 5 tasks and 3 UAVs. min refers to Minimum Consumption Profile, max to Max Speed Profile, mR to MPR, sR to SAR, iR to Inverse Synthetic Aperture Radar (ISAR) and eiS to EO/IR Sensor.

2. The second group is composed of an allele representing the **permutation of the task orders**. As described in a previous approach [8], these values are used to indicate the order in which each UAV performs the tasks assigned

to it. It is only used if there are several tasks assigned to the same UAV. In Figure 3 an example of this allele is presented, which together with the first allele (assignments) of Figure 2 shows that UAV 1 performs tasks 1 and 3 in this order; UAV 2 performs task 4; and UAV 3 performs tasks 5, 2, 1 and 4. In the reproduction phase, this allele is applied a Partially-Matched Crossover (PMX) and an Insert Mutation.

Fig. 3: Example of the second group of alleles from an individual representing a possible solution for a problem with 5 tasks and 3 UAVs. The permutation constraints the values of genes to be different and in the range 0 to $n-1$.

3. The third group of alleles is composed of two alleles of size m each one:
   - **GCSs controlling each UAV**.
   - **Flight Profiles used by each UAV to return** to the base.

   In Figure 4, an example of this group for a mission with 3 UAVs and 2 GCSs is presented. In the reproduction phase of the algorithm, this group of alleles is applied a 2-point crossover and an uniform mutation.

Fig. 4: Example of the third group of alleles from an individual representing a possible solution for a problem with 3 UAVs and 2 GCSs. min refers to Minimum Consumption Profile and max to Max Speed Profile.

3.2 Fitness Function

The fitness function of the algorithm considers a CSP modelling of the problem in order to check that all constraints are fulfilled for a given solution. This CSP model considers as variables the alleles of the encoding previously presented. In addition, it is necessary to define some extra variables that are computed in the propagation phase of the CSP solver. These variables are the time points ($departure$, $start$ and end) and durations ($durPath$, $durTask$, $durReturn$) of the mission, as well as fuel consumptions ($fuelPath$, $fuelTask$, $fuelReturn$).

On the other hand, the CSP considers several constraints related to the complexity issues explained in the previous section, checked in a specific order:

1. **Sensor constraints**: they check if a UAV has the sensor needed to perform its assigned tasks.
2. **Order constraints**: they assure that the values of the order variables are less than the number of tasks assigned to the UAV performing that task.
3. **Dependency Constraints**: these constraints are related to the time and vehicle dependencies mentioned in the previous section. Path consistency is checked with these dependencies and the orders of the tasks.
4. **GCS constraints**: they 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.
5. **Path constraints**: they assure for every assignment that the UAVs are able to reach the task zone avoiding the NFZs of the scenario and maintaining the line of sight with the GCSs controlling them.
6. **Temporal constraints**: they assure the consistency of all the time variables.
7. **Fuel constraints**: they assure that the fuel consumed by each vehicle is less than its initial fuel.
8. **Autonomy constraints**: they assure that the total flight time for each vehicle is less than its vehicle autonomy time.
9. **Distance constraints**: they assure that the distance traversed by each vehicle is less than its range.
10. **Distance between UAVs constraints**: they assure that UAVs keep a safe distance during the entire mission.

When a solution is invalid, the number of constraints fulfilled before the failure of some invalid constraint is returned. In this case, the fitness of the solution is assigned for each objective its upper bound minus the number of constraints fulfilled, in order to maximize them. This way, if no solution is valid, then the solutions satisfying the highest number of constraints are considered for next generation. We have denominated this fitness as Weighted Penalty Fitness (WPF), while previous approach uses a Binary Penalty Fitness (BPF).

On the other hand, if all constraints are fulfilled, the fitness works as a multi-objective function minimizing the objectives of the problem:

- The **number of UAVs** used in the mission
- The **total flight time**
- The **total fuel consumption**
- The **total distance** traversed
- The **total cost** of the mission
- The end time of the mission, or **makespan**

3.3 Algorithm

Algorithm 1 shows this new approach presented so far. Lines 8-13 show the implementation of the fitness function previously explained. After evaluation, a

μ elitist selection based on NSGA-II approach [3] is performed (Line 18). Then, a roulette wheel selection over these N individuals (Line 21) selects those that will be applied the genetic operators.

Algorithm 1: Hybrid MOGA-CSP algorithm for MCMPPs

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 $\overline{M} = \{M_i >> avg(o_i)\}$. And positive numbers *generations*, *population*, μ, λ, *mutprobability* and *stopGeneration*

Output: POF obtained with best solutions

1 $S \leftarrow$ randomly generated set of *population* of p chromosomes
2 $i \leftarrow 1$
3 $convergence \leftarrow 0$
4 $pof \leftarrow \emptyset$
5 **while** $i \leq generations \wedge convergence < stopGenerations$ **do**
6 $\quad newS \leftarrow \emptyset$
7 $\quad F \leftarrow \emptyset$
8 $\quad$ **for** $j \leftarrow 1$ ***to*** p **do**
9 $\quad\quad [valid, numValidConstraints] = CSP_{Check}(S_j)$
10 $\quad\quad$ **if** $valid$ **then**
11 $\quad\quad\quad F \leftarrow F \cup MultiObjectiveFitness(S_j)$
12 $\quad\quad$ **else**
13 $\quad\quad\quad F \leftarrow F \cup (\overline{M} - numValidConstraints)$
14 $\quad newpof \leftarrow createPOF(S)$
15 $\quad$ **if** $newpof = pof$ **then**
16 $\quad\quad convergence \leftarrow convergence + 1$
17 $\quad pof = newpof$
18 $\quad Sbest \leftarrow SelectNSGA2Best(\mu, F)$
19 $\quad newS \leftarrow Sbest$
20 $\quad$ **for** $j \leftarrow \mu$ ***to*** λ **do**
21 $\quad\quad p1, p2 \leftarrow RouleWheelSelection(Sbest)$
22 $\quad\quad i1, i2 \leftarrow Crossover(p1, p2)$
23 $\quad\quad i1 \leftarrow Mutation(i1, mutprobability)$
24 $\quad\quad i2 \leftarrow Mutation(i2, mutprobability)$
25 $\quad\quad newS \leftarrow newS \cup \{i1, i2\}$
26 $\quad S \leftarrow newS$
27 $\quad i \leftarrow i + 1$
28 **return** pof

Next, we use a proper crossover operator (Line 22), consisting of a specific crossover operation for each group of the alleles of the representation, as explained in section 3.1 Then, a mutation operator (Lines 23-24) will be applied

depending on a probability P_m. This mutation operator, as well as the crossover, consists of a specific mutation for each group of alleles of the encoding.

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

4 Experimental Results

In order to test this new approach, four Mission Scenarios have been designed (see Table 1). In each scenario, tasks, UAVs, GCSs and NFZs are scattered throughout the map. Each UAV has different sensors and each task can be performed by different sets of sensors, so there are several possible solutions. On the other hand, each UAV has been set with a different amount of initial fuel. The increasing complexity of these datasets allows to compare how this new approach behaves according to the complexity of the problem.

Table 1: Features of the different datasets designed.

Dataset Id.	Tasks	Multi-UAV Tasks	UAVs	GCSs	NFZs	Time Dependencies
1	6	0	5	2	2	1
2	8	2	5	3	3	2
3	12	3	6	3	3	3
4	16	4	8	3	4	4

The new MOGA-CSP approach, with its WPF is compared against a previous approach [8] where the CSP is used as a penalty function (BPF). To setup the MOGA of both approaches, the selection criteria $(\mu + \lambda)$ used was $100 + 1000$, where λ is the number of offspring (population size), and μ the elitism size, i.e. the number of the best parents that survive from current generation to the next. The mutation probability is 10%, and the number of generations used in the stopping criteria is 10. Each problem is run 10 times, extracting for each one the number of generations needed to converge, and the hypervolume metric [11] of the solutions obtained. Results are shown in Table 2 and Table 3.

As shown in these tables, it is appreciable that the hypervolumes for both approaches are similar in the fourth datasets. On the other hand, the number of generations needed to converge decrease in the MOGA-CSP-WP approach respect to the MOGA-CSP-BP approach as the complexity of the problem grows. In Figure 5, the gap of improvement of the convergence between the two approaches as the complexity grows is clearly appreciated. With this, we can say that considering the number of constraints as a heuristic decreases the convergence time. This is due to the big search space of complex problems compared to its small solution space. For simpler problems, as dataset 1, it is appreciable

A Weighted Penalty Fitness for a Hybrid MOGA-CSP to solve MPPs

Table 2: Number of generations until convergence for the MOGA-CSP-BP and the new MOGA-CSP-WP approach.

Dataset Id.	MOGA-CSP-BP			MOGA-CSP-WP		
	Min.	Avg.	Max.	Min.	Avg.	Max.
1	47	55.2	72	49	56.4	67
2	122	142.6	163	97	123.9	142
3	594	629.8	647	472	489.4	507
4	1304	1321.4	1351	875	956.1	1012

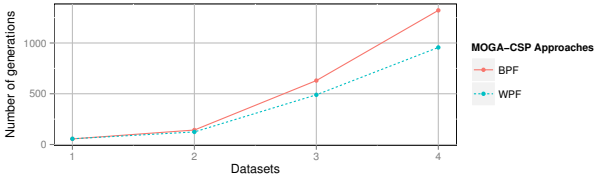

Fig. 5: Average number of generations until convergence for the MOGA-CSP-BP and the new MOGA-CSP-WP approach.

Table 3: Hypervolume results of the solutions obtained by the MOGA-CSP-BP and the new MOGA-CSP-WP approach.

Dataset Id.	MOGA-CSP-BP			MOGA-CSP-WP		
	Min.	Avg.	Max.	Min.	Avg.	Max.
1	0	0.04	0.1	0	0.05	0.12
2	0	1.18	2.03	0	0.97	1.57
3	32.68	59.24	87.35	44.73	61.83	80.15
4	112.34	118.56	132.15	111.98	120.01	134.79

that the results are pretty similar. This is due to the smaller search space, which could induce to find valid solutions in early generations of the algorithm.

5 Conclusions

In this paper, we have presented a MOGA-CSP approach using a weighted penalty fitness function to solve Multi-UAV Mission Planning Problems. The presented approach considers missions consisting of several tasks to be performed by several UAVs using a specific sensor. Each UAV is controlled by a GCS and use specific Flight Profiles. The problem has been modelled as a CSP considering the assignments of tasks to UAVs, the orders of tasks, the assignments of UAVs to GCSs, the sensors used for each task, the flight profiles used by each UAV in its path to each assigned task and the return flight profile used by each UAV as

variables; and several constraints involving these variables. On the other hand, the encoding of the MOGA considers the same variables as the CSP. A new fitness function has been designed for this approach: first the solution is check with the CSP model and if it is not valid, the number of constraints unfulfilled is used as weighted penalty function; otherwise, a multi-objective fitness function optimizing several objectives is considered.

The experiments performed over several missions show that this new approach outperforms the results obtained previously in terms of convergence time, specially for complex problems with huge search space and reduced solution space. Nevertheless, these results could be outperformed combining other new methodologies with this one, which is focused as future research.

Future works will 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.

References

1. Barták, R.: Constraint programming: In pursuit of the holy grail. In: Week of Doctoral Students. pp. 555–564 (1999)
2. Casbeer, D.W., Kingston, D.B., Beard, R.W., McLain, T.W.: Cooperative forest fire surveillance using a team of small unmanned air vehicles. International Journal of Systems Science 37(6), 351–360 (2006)
3. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation 6(2), 182–197 (2002)
4. Doherty, P., Kvarnström, J., Heintz, F.: A temporal logic-based planning and execution monitoring framework for Unmanned Aircraft Systems. Autonomous Agents and Multi-Agent Systems 19(3), 332–377 (December 2009)
5. Fabiani, P., Fuertes, V., Piquereau, A., Mampey, R., Teichteil-Konigsbuch, F.: Autonomous flight and navigation of VTOL UAVs: from autonomy demonstrations to out-of-sight flights. Aerospace Science and Technology 11(2-3), 183 – 193 (2007)
6. Kvarnström, J., Doherty, P.: Automated planning for collaborative UAV systems. In: Control Automation Robotics & Vision. pp. 1078–1085 (December 2010)
7. Michalewicz, Z., Schoenauer, M.: Evolutionary algorithms for constrained parameter optimization problems. Evolutionary computation 4(1), 1–32 (1996)
8. Ramirez-Atencia, C., Bello-Orgaz, G., R-Moreno, M.D., Camacho, D.: A Hybrid MOGA-CSP for Multi-UAV Mission Planning. In: Proceedings of the Companion Publication of the 2015 on Genetic and Evolutionary Computation Conference. pp. 1205–1208. ACM (2015)
9. Ramirez-Atencia, C., Bello-Orgaz, G., R-Moreno, M.D., Camacho, D.: Performance Evaluation of Multi-UAV Cooperative Mission Planning Models. In: Computational Collective Intelligence, pp. 203–212. Springer International Publishing (2015)
10. Rodríguez-Fernández, V., Menéndez, H.D., Camacho, D.: Automatic profile generation for uav operators using a simulation-based training environment. Progress in Artificial Intelligence 5, 37–46 (2015)
11. Zitzler, E., Brockhoff, D., Thiele, L.: The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In: Evolutionary multi-criterion optimization. pp. 862–876. Springer (2007)