Intelligent Distributed Computing IX: Proceedings of the 9th International Symposium on Intelligent Distributed Computing – IDC'2015, Guimarães, Portugal, October 2015. Studies in Computational Intelligence, Volumen 616. Springer, 2016. 167-176

DOI: http://dx.doi.org/10.1007/978-3-319-25017-5_16

GAMPP: Genetic Algorithm for UAV Mission Planning Problems

Gema Bello-Orgaz, Cristian Ramirez-Atencia, Jaime Fradera-Gil and David Camacho

Abstract Due to the rapid development of the UAVs capabilities, these are being incorporated into many fields to perform increasingly complex tasks. Some of these tasks are becoming very important because they involve a high risk to the vehicle driver, such as detecting forest fires or rescue tasks, while using UAVs avoids risking human lives. Recent researches on artificial intelligence techniques applied to these systems provide a new degree of high-level autonomy of them. Mission planning for teams of UAVs can be defined as the planning process of locations to visit (waypoints) and the vehicle actions to do (loading/dropping a load, taking videos/pictures, acquiring information), typically over a time period. Currently, UAVs are controlled remotely by human operators from ground control stations, or use rudimentary systems. This paper presents a new Genetic Algorithm for solving Mission Planning Problems (GAMPP) using a cooperative team of UAVs. The fitness function has been designed combining several measures to look for optimal solutions minimizing the fuel consumption and the mission time (or makespan). The algorithm has been experimentally tested through several missions where its complexity is incrementally modified to measure the scalability of the problem. Experimental results show that the new algorithm is able to obtain good solutions improving the runtime of a previous approach based on CSPs.

Gema Bello-Orgaz
Escuela Politecnica Superior, Universidad Autonoma de Madrid, C/Francisco Tomas y Valiente 11, 28049, Madrid, Spain e-mail: gema.bello@uam.es

Cristian Ramirez-Atencia
Escuela Politecnica Superior, Universidad Autonoma de Madrid, C/Francisco Tomas y Valiente 11, 28049, Madrid, Spain e-mail: cristian.ramirez@inv.uam.es

Jaime Fradera-Gil
Escuela Politecnica Superior, Universidad Autonoma de Madrid, C/Francisco Tomas y Valiente 11, 28049, Madrid, Spain

David Camacho
Escuela Politecnica Superior, Universidad Autonoma de Madrid, C/Francisco Tomas y Valiente 11, 28049, Madrid, Spain e-mail: david.camacho@uam.es

1 Introduction

The potential applications of Unmanned Aerial Vehicles (UAVs) are varied, including surveillance [8], disaster and crisis management [11], and agriculture or forestry [7], among others. Therefore, over the past 20 years, a large number of research works related to this field have been carried out [5]. Due to the rapid development of the UAVs capabilities, these are being incorporated into many areas to perform increasingly complex tasks. Some of these tasks are becoming very important because they involve a high risk to the vehicle driver, such as detecting forest fire or rescue tasks, while using UAVs avoids risking human lives.

A mission can be described as a set of goals that are achieved by performing some tasks with a group of resources over a period of time. Specifically, mission planning for UAVs can be defined as the planning process of locations to visit (waypoints) and the vehicle actions to do (loading/dropping a load, taking videos/pictures, acquiring information), typically over a time period. There are some attempts to implement mission planning systems for UAVs in the literature. Doherty et al. [3] presented an architectural framework for mission planning and execution monitoring and its integration into a fully deployed unmanned helicopter. Then planning and monitoring modules use Temporal Action Logic (TAL) for reasoning about actions and changes, and the knowledge gathered from the sensors during plan execution is used in the process. Other novel approach formulates the mission planning problem as a Constraint Satisfaction Problem or CSP, where the tactic mission is modelled and solved using constraint satisfaction techniques [9].

These methods can be improved using stochastic search algorithms based on an objective function to be optimized, also known as Genetic Algorithms (GAs). There are many applications where GAs have been successful, from optimization [2] to Data Mining [6, 1]. GAs have demonstrated to be robust, able to find satisfactory solutions in highly multidimensional problems with complex relations between the variables. The Soliday et al.[10] approach developed a GA able to effectively solve UAV missions under complex constraints. The GA was constructed using a novel representation based on the nearest neighbour search, being each allele the N Nearest Neighbours. It uses a qualitative fitness function based on the number of mission objectives and the time allowed. Finally, other novel work has designed a graph based representation for mission planning of UAVs to carry out tasks in a flying space constrained with the presence of flight prohibited zones and radar sites [4].

This work aims to design and implement a new mission planning algorithm in order to improve the existing approaches using GAs. For this purpose a fitness function has been designed combining several measures to look for optimal solutions minimizing the fuel cost and the mission time (or makespan). These measures used in the fitness function and their weights can be changed in the algorithm settings. Then the algorithm is applied to real-world cases and a detailed analysis of the experimental results is carried out.

The rest of the paper is structured as follows. Section 2 describes the model designed for generating UAV missions. Section 3 presents the genetic algorithm, the encoding used and the fitness function implemented to solve UAV missions.

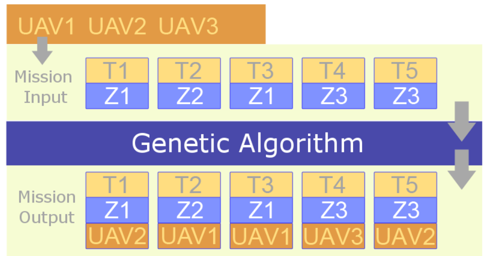

Fig. 1 Input/Output example of the genetic algorithm for mission planning

Section 4 provides a description of the dataset used, the experimental setup of the algorithm and a complete experimental evaluation of it. Finally, in Section 5, the conclusions and some future research lines of the work are presented.

2 UAV Mission Plan Description

This section details the proposed structure to generate missions that the genetic algorithm receives as input, and also the output obtained. A UAV mission can be defined as a number n of *tasks* to accomplish for a set of *UAVs*. A task could be exploring a specific area or search for an object in a zone. One or more *sensors or payloads* belonging to a particular UAV may be required to perform a task. Each task must be performed in a specific geographic *zone*, at a specific time interval.

As can be seen in Figure 1, the GA receives as input a list of tasks to be performed in specific zones ($[T_i, z_j]$), and a set of UAVs that can be used to perform these tasks. After the execution of the GA for an input mission, the output will be the possibles assignments of UAVs to tasks ($[T_i, z_j, UAV_k]$ where T_i is a task, z_j is the zone where the task is performed and UAV_k UAV is a vehicle from the set of available UAVs).

To perform a mission, there are m available UAVs, each one with specific characteristics such as fuel consumption rate, list of available payloads, an attribute indicating if the UAV is able to fly within restricted zones, and a position (geographical coordinates). A UAV can be equipped with one or more payloads that allows to perform different types of tasks:

- **Camera EO (Electro Optical)**: to take photos of large amplitude and long distance.
- **Camera IR (Infra-red)**: to take photos and videos at night or in conditions of very low luminosity. This sensor is also capable of performing thermal photos, especially used to forest areas analysis and fire detection and prevention.
- **Radar SAR (Synthetic Aperture Radar)**: to take images of an object allowing its representation in 2D and 3D. It can be used to track a zone.

3 GA for Mission Planning Problems

The Genetic Algorithm for Mission Planning Problems (GAMPP) is a genetic algorithm to solve mission planning problems using a team of UAVs. This section describes this algorithm, including the encoding, the fitness function, and the genetic operators applied. The algorithm is implemented according to the structure of a simple genetic algorithm as can be seen in Algorithm 1.

3.1 Encoding

A possible solution for a mission planning problem consists of the assignment of each task to a specific UAV. If the mission contains a number *N* of tasks, the genotypes (chromosomes) will be represented as a integer vector of size *N*. Each allele represents a UAV assigned to a task. Therefore, if *M* is the number of UAVs to solve the mission, the value of each allele is between 0 and *M-1* (see Fig. 2). In this example, two UAVs (UAV0 and UAV1) perform most of the tasks, meanwhile the rest of tasks have been assigned to other UAVs, therefore they can be performed in parallel.

Fig. 2 Chromosome representing a solution for a mission planning problem. Each allele represents a particular assignment of a UAV to a task of the mission. This example is the solution for a mission that contains 9 tasks and 6 UAVs for accomplishing these tasks.

3.2 Genetic Algorithm

In the new approach used to solve mission plans, the population evolves using a standard GA as it is shown in Algorithm 1. The algorithm performs an Elitism selection method, where the n-best chromosomes of the population are copied to the new population (line 8 in Algorithm 1). This prevents losing the n-best found solutions. Finally the genetic operators work as follows:

- ***Crossover***: One-point crossover operator is applied. Firstly, two individual are selected by tournament as parents, and a randomly point from the genome is chosen (see lines 11 and 12 in Algorithm 1). Then the information of both parents is swapped from this point to create two new offspring.
- ***Mutation***: Uniform mutation is applied. A value of the genome is randomly chosen, and this value (with a predefined mutation probability) changes from 1 to M, where M is the number of available UAVs. See lines 13 and 14 in Algorithm 1.

Input: A mission $M = (T,U)$ where T is a set of tasks to perform denoted by $\{t_1,\ldots,t_n\}$ and U is a set of UAVs denoted by $\{u_1,\ldots,u_m\}$ representing the available vehicles to perform the tasks. And positive numbers $generations$, $population$, μ, λ and $mutprobability$
Output: The chromosome $S_i = \{a_1,a_2,\ldots,a_n\}$ such that $Fitness(S_i)$ is maximized
$S \leftarrow$ randomly generated set of $population$ of p chromosomes of size n, and the value of each allele is from 1 to m
$i \leftarrow 1$
$convergence \leftarrow 0$
while $i \leq generations \wedge convergence = 0$ **do**
    $F \leftarrow \emptyset$
    for $j \leftarrow 1$ ***to*** p **do**
        $F \leftarrow Fitness(S_j)$
    end
    $Sbest \leftarrow SelectNBest(\lambda,F)$
    $S \leftarrow Sbest$
    for $j \leftarrow$ *1* ***to*** λ **do**
        $p1,p2 \leftarrow TournamentSelection(Sbest)$
        $i1,i2 \leftarrow OnePointCrossover(p1,p2)$
        $i1 \leftarrow Mutation(i1,mutprobability)$
        $i2 \leftarrow Mutation(i2,mutprobability)$
        $S \leftarrow I \cup \{i1,i2\}$
    end
    $i \leftarrow i+1$
    $convergence \leftarrow CheckConvergence(Sbest)$
end
return *SelectBest(S,F)*

Algorithm 1: Genetic Algorithm for Mission Planning Problems (GAMPP)

3.3 Fitness Function

The fitness function implemented consists of two distinct phases to evaluate the generated individuals. Firstly, the criteria ensuring that the mission can be resolved successfully are evaluated. Afterwards, the quality of the mission is measured. For this purpose, the fitness function has been designed combining various measures to find an assignment of UAVs to the mission tasks minimizing the *fuel cost* and the *makespan*. To look for optimal solutions, a weighted function based on these criteria is used. The weights can be changed in the algorithm settings, and the fitness function is calculated as follows:

$$F(i) = (Mak(i) \cdot w_{mak} + Fuel(i) \cdot w_{fuel}) \cdot \alpha \quad (1)$$

Where $w_{dur} \in [0,1]$, $w_{fuel} \in [0,1]$, $w_{dur} + w_{fuel} = 1$ and α is defined as:

$$\alpha = \prod_{i=0}^{n} checkPayloads(i) \cdot checkDur(i) \cdot isResZone(i) \tag{2}$$

3.3.1 Validation Criteria

These criteria ensure that the mission can be solved successfully, avoiding invalid solutions. Invalid solutions are discarded giving them the lowest value of the fitness function, which is 0. To validate the solutions three types of constraints are checked:

- **Payload Constraint**: checks whether each UAV carries the corresponding payload to perform the task assigned to it.
- **Temporal Constraint**: ensures that each UAV does not perform tasks at the same time.
- **Restricted Zone**: checks whether only UAVs with permissions to fly within restricted zones perform tasks developed in these restricted zones.

3.3.2 Optimization Criteria

Secondly, the quality of the solution is measured for valid individuals. A mission performed with a lower duration and fuel consumed, is usually better. For this purpose, the fitness function combines two different criteria:

- **Makespan**: Time required to perform the complete mission. The different UAVs can perform tasks simultaneously. Therefore, the mission duration is given by the time interval from the start time of the first task to the end time of the last task.
- **Fuel cost**: Sum of the fuel consumed by each UAV at performing its assigned tasks. The fuel cost for a UAV k performing a task i is $fuel_i = fuelConsume * distance_{k \to z}$, being $fuelConsume$ the fuel consumption rate per distance unit of a UAV, and $distance_{k \to z}$ the distance from position k to position z given in geographic coordinates (latitude, longitude and altitude). This distance is calculated using the positions of the UAVs and the zones where tasks are performed.

4 Experimental Results

4.1 Dataset Description

In this work, 15 missions have been designed, each one composed by an increasing number of tasks from 1 to 15. The first mission is composed of one task; the second, two tasks; and so on up to 15, which is the most complex mission to solve. In order to solve these missions, there are a set of UAVs with specific equipments. Each task needs a particular payload to be performed, and each UAV has different available types of payloads.

4.2 Experimental Setup

The GA parameters and the weights of the fitness function were obtained experimentally by performing several tests with different range of values. Table 1 shows the parameters used throughout the experimental phase where $\mu + \lambda$ is the selection criteria used, being λ the number of offspring (population size), and μ the number of the best parents that survive from the current generation to the next.

Mutation probability	0.1
Generations	300
Population size	1000
Selection criteria ($\mu + \lambda$)	$100 + 1000$
Fitness function (w_{fuel})	0.7
Fitness function (w_{dur})	0.3

Table 1 Experimental setup for the genetic algorithm.

4.3 Results

Firstly, an analysis of the optimal solutions found is carried out. This can be seen in Table 2. Considering the values obtained to the fitness function, they begin close to 1 (very close to the best possible value). However, these values decrease as the number of tasks increase. Analysing the task assignments, as can bee seen in Figure 3, the algorithm carries out an equitable distribution of tasks between different available UAVs. There are several tasks performing in parallel, and therefore the mission duration is lower. It means that as the complexity of the missions increases, the quality of the optimal solution found decreases. But the algorithm is able to find solutions performing the complete mission with enough quality.

Finally, to study the computational performance of the algorithm, the runtime spent is compared with other approach based on a CSP model using Branch & Bound (B&B) [9]. The same dataset and optimization function ($0.7 \cdot Fuel(i) + 0.3 \cdot Mak(i)$) is used in both approaches. The results obtained are represented in Figure 4. The time difference observed is very high, being the time needed to assign 10 tasks in the order of seconds (2,3 sec) to the genetic algorithm, whereas the CSP based approach is in a order of minutes (4 min). It can be appreciated that the runtime of the GA has a linear growth directly proportional to the number of tasks, whereas the runtime spent in the approach based on CSPs grows exponentially.

Task Number	F_{val}	Duration	Fuel Consumed
1	0.964	20 min	10.341 L
2	0.967	20 min	16.568 L
3	0.971	30 min	18.791 L
4	0.978	50 min	25.463 L
5	0.985	65 min	36.583 L
6	0.981	65 min	49.036 L
7	0.983	60 min	55.708 L
8	0.978	50 min	76.724 L
9	0.957	120 min	127.095 L
10	0.907	135 min	181.803 L
11	0.901	135 min	211.715 L
12	0.899	135 min	227.060 L
13	0.899	200 min	249.076 L
14	0.858	200 min	253.859 L
15	0.858	200 min	255.859 L

Table 2 Optimal solutions found for missions with 1 to 15 tasks.

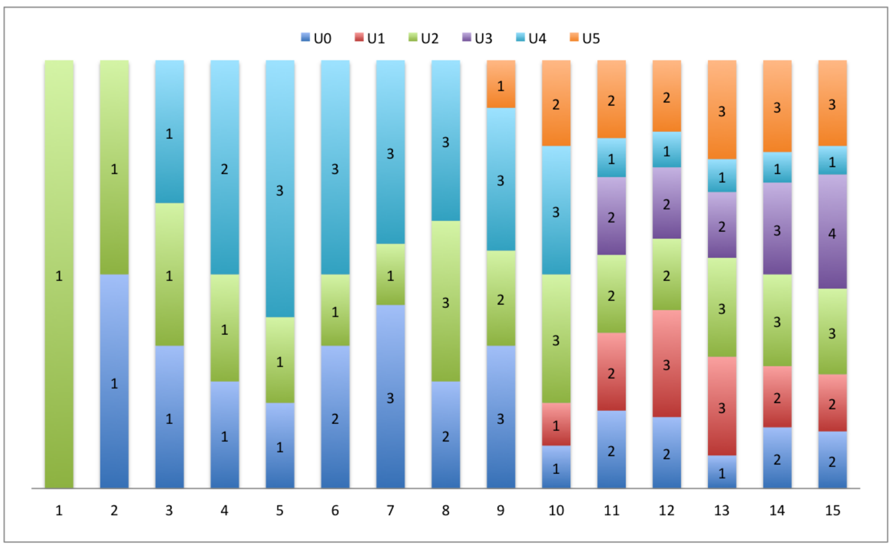

Fig. 3 Task assignments between the different available UAVs for solving the missions.

5 Conclusions and Future Work

In this work, a new mission planning algorithm for UAVs using GAs has been designed and implemented. For this purpose, a model to generate UAV missions is designed. Using this model, a mission is defined as a set of tasks to be performed in specific zones by several UAVs with some capabilities. In order to guide the search of possible solutions, a fitness function has been designed to look for optimal solutions minimizing the fuel cost and the makespan. The new algorithm has been

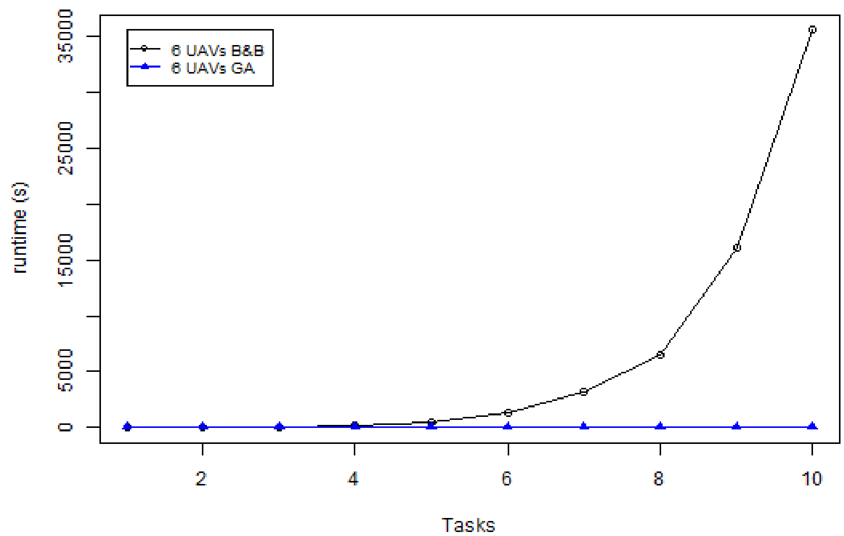

Fig. 4 Comparative assessment of runtime using other approach for the mission planning problem.

tested using several UAV missions, and the experimental results obtained are analysed. Regarding the quality of the solutions, the algorithm performs a task assignment where the mission tasks are equitably distributed between the different UAVs available. Additionally, a comparative assessment of runtime to solve the mission planning problem is carried out. The experimental results obtained show that the new approach reaches a better runtime than a previous approach based on CSPs.

Finally, some improvements can be made to the algorithm. It is important to remark that the results obtained are highly dependant on the mission designed and on the topology of the zones the missions are developed in. Therefore, further works should consider different mission scenarios and topologies. In addition, the tactical scenarios for the missions are on real-time and dynamic. Many changes can affect the pre-loaded planning during its execution (UAVs failures, weather conditions, new tasks, etc...). Therefore an on-line distributed variant of the algorithm could be very useful.

Acknowledgment

This work is supported by Comunidad Autónoma de Madrid under project CIBERDINE S2013/ICE-3095, Spanish Ministry of Science and Education under Project

Code TIN2014-56494-C4-4-P and Savier Project (Airbus Defence & Space, FUAM-076915). 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. Gema Bello-Orgaz and David Camacho. Evolutionary clustering algorithm for community detection using graph-based information. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 930–937. IEEE, 2014.
2. Xiao Bin, Wang Min, Liu Yanming, and Fang Yu. Improved genetic algorithm research for route optimization of logistic distribution. In *Proceedings of the 2010 International Conference on Computational and Information Sciences*, ICCIS '10, pages 1087–1090, Washington, DC, USA, 2010. IEEE Computer Society.
3. Patrick Doherty, Jonas Kvarnström, and Fredrik Heintz. A temporal logic-based planning and execution monitoring framework for unmanned aircraft systems. *Autonomous Agents and Multi-Agent Systems*, 19(3):332–377, 2009.
4. L Geng, YF Zhang, JJ Wang, Jerry YH Fuh, and SH Teo. Cooperative task planning for multiple autonomous uavs with graph representation and genetic algorithm. In *Control and Automation (ICCA), 2013 10th IEEE International Conference on*, pages 394–399. IEEE, 2013.
5. Farid Kendoul. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics*, 29(2):315–378, 2012.
6. Hector D Menendez, David F Barrero, and David Camacho. A co-evolutionary multi-objective approach for a k-adaptive graph-based clustering algorithm. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 2724–2731. IEEE, 2014.
7. Luis Merino, Fernando Caballero, J Ramiro Martínez-de Dios, Joaquin Ferruz, and Aníbal Ollero. A cooperative perception system for multiple uavs: Application to automatic detection of forest fires. *Journal of Field Robotics*, 23(3-4):165–184, 2006.
8. E Pereira, R Bencatel, J Correia, L Félix, G Gonçalves, J Morgado, and J Sousa. Unmanned air vehicles for coastal and environmental research. *Journal of Coastal Research*, pages 1557–1561, 2009.
9. Cristian Ramírez-Atencia, Gema Bello-Orgaz, Maria D R-Moreno, and David Camacho. Branching to find feasible solutions in unmanned air vehicle mission planning. In *Intelligent Data Engineering and Automated Learning–IDEAL 2014*, pages 286–294. Springer, 2014.
10. Stephen W Soliday et al. A genetic algorithm model for mission planning and dynamic resource allocation of airborne sensors. In *Proceedings, 1999 IRIS National Symposium on Sensor and Data Fusion*. Citeseer, 1999.
11. Jun Wu and Guoqing Zhou. High-resolution planimetric mapping from uav video for quick-response to natural disaster. In *Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on*, pages 3333–3336. IEEE, 2006.