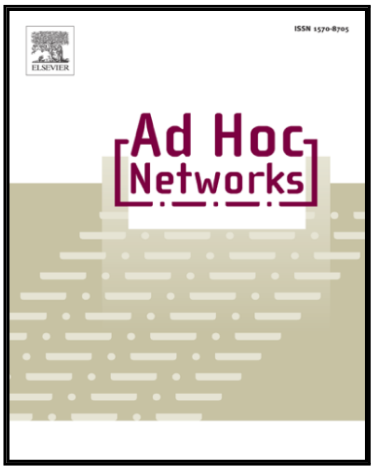
Accepted Manuscript



Multi-subpopulation Evolutionary Algorithms for Coverage Deployment of UAV-networks

D.G. Reina , H. Tawfik , S.L. Toral

PII: S1570-8705(17)30171-3
DOI: 10.1016/j.adhoc.2017.09.005
Reference: ADHOC 1586

To appear in: *Ad Hoc Networks*

Received date: 11 April 2017
Revised date: 26 September 2017
Accepted date: 28 September 2017

Please cite this article as: D.G. Reina , H. Tawfik , S.L. Toral , Multi-subpopulation Evolutionary Algorithms for Coverage Deployment of UAV-networks, *Ad Hoc Networks* (2017), doi: 10.1016/j.adhoc.2017.09.005

Multi-subpopulation Evolutionary Algorithms for Coverage Deployment of UAV-networks

D. G. Reina*[1], H. Tawfik*[2], S. L. Toral*[1]

*[1] Electronic Engineering Department, University of Seville. Camino de los Descubrimientos S/N, 41092, Seville, Spain.

*[2]School of Computing, Creative Technology and Engineering, Leeds Beckett University, Leeds, UK

Abstract: The deployment of an unmanned aerial network (UAV-network) for the optimal coverage of ground nodes is an NP-hard problem. This work focuses on the application of a multi-layout multi-subpopulation genetic algorithm (MLMPGA) to solve multi-objective coverage problems of UAV-networks. The multi-objective deployment is based on a weighted fitness function that takes into account coverage, fault-tolerance, and redundancy as relevant factors to optimally place the UAVs. The proposed approach takes advantage of different subpopulations evolving with different layouts. This feature is aimed at reflecting the evolutionary concept of different species adapting to the search space conditions of the multi-objective coverage problem better than single-population genetic algorithms. The proposed multi-subpopulation genetic algorithm is evaluated and compared against single-population genetic algorithm configurations and other well-known meta-heuristic optimization algorithms, such as particle swarm optimization and hill climbing algorithm, under different numbers of ground nodes. The proposed MLMPGA achieves significantly better performance results than the other meta-heuristic algorithms, such as classical genetic algorithms, hill climbing algorithm, and particle swarm optimization, in the vast majority of our simulation scenarios.

__Keywords:__ Unmanned Aerial Vehicles, UAV-network, Coverage, Evolutionary Algorithms, multiple objectives, optimization

1. Introduction

The variety of applications based on Unmanned Aerial Vehicles (UAVs) has enormously increased in the last decade [1][2]. The main reason is the enormous capabilities of such aerial vehicles in terms of mobility, autonomy, communication and processing power at a relatively low cost. Therefore, cooperative applications, where several UAVs forming an UAV-network cooperate with each other to accomplish a common objective [3], are feasible from both technical and economic point of views. Furthermore, the great developments in wireless multi-hop paradigms such as ad hoc networks [4], mesh networks [5], and Vehicular Ad Hoc Networks (VANETs) [6]; and wireless communications technologies such as IEEE 802.11 family [7] and IEEE 802.15.4, enable wireless communications among these flying objects in a multi-hop style [8]. Such multi-hop communications are crucial to perform mobile networks tasks such as self-organization and self-deployment [9][10].

Among the possible applications of UAV-networks, civil applications are envisioned to play an important role [1]. UAV-networks can be deployed in critical scenarios such as disaster situations to provide communication services to ground nodes, which can be people carrying their portable devices, such as smartphones and tables, and needing help [9][11]. It is well-known that during the aftermath of a natural or human made disaster scenario, the cell telecommunication infrastructure is likely to be destroyed or malfunctioning [12], leaving many people isolated. In the same line, UAV-networks can be used to alleviate congestion in dense scenarios, where many devices try to access the cell telecommunication infrastructure at the same time [13]; for instance, sport events, festivals, and concerts, among others. In addition to that, isolated areas in countryside can also be monitored by placing UAV-networks at targets locations in case of necessity for search and tracking operations of rescue personnel. Furthermore, UAVs equipped with 5G systems have been proposed for providing communication services in isolated areas and low-income regions [14].

Regarding such critical civil applications, the design of an UAV-network is challenging and should be done carefully. We model the UAV-network deployment problem as a Wireless Network Design (WND) [13][15], where the positions of the UAVs forming the network should be optimized to provide communication services to clients on the ground. Although the ideal topology of the UAV-network depend on the target communication services, there are some main requirements that should be satisfied to guarantee reliable and robust communication services such as: coverage, fault-tolerance, and redundancy [15]. Regarding coverage and using the available definitions [16][17], a wireless client is said to be covered or served by an UAV-network if it receives the service within a minimum level of quality. Therefore, the UAV-network should be able to cover all ground nodes or at least the maximum possible. Fault tolerance is the ability of a network to continue to perform a required function in the presence of faults or errors [18][19], in our case, UAVs' failures. This feature is important in UAV-networks since these networks can be affected by several factors such as rapid power battery depletion and air turbulences [11], among others. Furthermore, the wireless communications among UAVs are also prone to suffer from the unreliability of the wireless medium [20]. Finally, redundancy refers to the accessibility of the UAV-network. A client should be able to connect to the UAV-network by any UAV covering the given client. Therefore, an optimal deployment should maximize the redundancy coverage between UAVs and clients to guarantee a high accessibility of the UAV-network.

In this paper, the UAV-network deployment problem has been modeled as a weighted multi-objective coverage problem, where each of the aforementioned network requirements (coverage, fault-tolerance, and redundancy) contributes to the quality of the WND. A suitable weight selection of the three objectives is relevant for the performance of the UAV-network. An important question is how to select the weights for the three objectives in order to provide communication services like the Internet. Coverage is of paramount importance to develop any communication service in the UAV-network. Consequently, we assume that coverage has higher importance than fault tolerance and redundancy. The reason is that without coverage, the UAV-network will not be able to provide any communication service. Therefore, coverage is the first priority of an optimal UAV-network deployment. Once this objective is accomplished, the second priority is to maintain this coverage in case of failures of some nodes in order to deploy a robust network. Thus, fault tolerance should have a higher weight than redundancy. Finally, and once the coverage and fault tolerance is guaranteed, the third objective is providing redundancy on nodes' coverage so the accessibility of the network can be enhanced. More details about the selection of the weights are given in Section 3.

The deployment of wireless networks has been proven to be a combinatorial NP-hard problem [21][22]. On this front, Evolutionary Algorithms (EAs) [23][24] have successfully performed in NP-hard problems in multi-hop networks [25][26]. These metaheuristic algorithms are suitable for combinatorial optimization problems such as the one presented in this work. The reason behind it is that they provide quasi-optimal solutions in a reasonable time. Another reason is that, UAV-networks such as Wireless Sensor Network (WSNs) [23] and Mobile Ad Hoc Networks (MANETs) [4] are complex networks that involve many nodes. Therefore, it is very difficult to model their behavior mathematically with a numerical model to be optimized using traditional gradient-based methods. Furthermore, since nowadays most computer networks are evaluated by using network simulators, we can easily approximate the performance of the network without the necessity of a mathematical model. This property is suitable for metaheuristic optimization techniques [23]. In order to solve the proposed multi-objective coverage problem, we propose and evaluate MLMPGA (Multi-Layout Multi-Population Genetic Algorithm) against different configurations of single-population Genetic Algorithms (GAs). MLMPGA uses different subpopulations with independent genetic strategies [27] and the same weighted multi-objective fitness function. The idea is to use a multi-strategy EA that may adapt to the search requirement of the optimization problem in all the phase of the evolutionary algorithm [28]. Therefore, the goal is to effectively handle the configuration of the EA parameters, which is a scientific challenge itself that deserves further attention [29], in addition to the optimization problem to be solved. We compare the performance of the proposed MLMPGA with the traditional single-population GA technique. These GA configurations use different cooperative strategies for solving the proposed weighted multi-objective coverage problem.

The contributions of this paper are, i) modelling the coverage problem of UAV-networks as a multi-objective problem and solving it by the definition of a weighted multi-objective fitness function, ii) proposing a novel meta-heuristic optimization algorithm (MLMPGA) that makes advantages of different evolutionary strategies in a multi-population genetic algorithm, and iii) the evaluation of the proposed metaheuristic in different scenarios in terms of UAVs and ground nodes, and finally its comparison with other well-known meta-heuristic optimization algorithms, such as Hill Climbing Algorithm (HCA) and Particle Swarm Optimization (PSO). The proposed MLMPGA clearly outperforms the other metaheuristics.

The rest of this paper continues as follows, Section 2 presents related work to the present study, which is mainly focused on UAV deployment based on evolutionary algorithms. Section 3 describes the multi-objective coverage problem and the proposed weighted multi-objective fitness function to solve the coverage problem. Section 4 focuses on the proposed MLMPGA and its main differences with respect to classical GAs. Section 5 contains the simulation study carried out for evaluating the proposed MLMPG. It includes a comparison of the proposed MLMPGA against other metaheuristics, a discussion of the obtained results, and future work directions. Finally, Section 6 concludes this work.

2. Related work

The deployment of UAVs in complex scenarios is an emerging research topic [29][30][31]. Coverage problems have been widely studied in WSNs and mesh networks [32]. Recently, many works have focused on enhancing the capacity and robustness of future 5G networks through the deployment of UAVs in emergency situations and zones [33][34] [35][36][37][38].

Regarding the use of GAs for network coverage and fault-tolerance, the authors use a single-population GA to place a number of base stations in [29]. The objective of the GA used in [29] is to find the minimum number of nodes so as to cover all sensor nodes with at least m base stations. Our approach is different since the UAVs should form a network (the number of UAVs is a constraint) and also we do not force m-connectivity since it cannot be possible in many cases. We seek fault-tolerance as a desirable feature of the network (where possible) rather than a mandatory requirement. In [30], a single-population and multi-objective GA is used to address coverage and lifetime problems in WSNs. The authors divide the target area into different grids to reduce the search space. They also employ the k-connectivity metric to improve the coverage of sensor nodes. In [31], the authors do force the connectivity requirement among the deployed nodes. They impose a communication path among each deployed node and the sink node of a WSN. However, there are also some differences with respect to our proposed approach. First, the main objective of coverage problem in [31] is to find the minimum number of nodes that guarantees the coverage. In addition to that, they divide the target application area into a grid of possible positions in order to reduce the search space.

GA technique is also used in [32] to solve a coverage problem in WSNs. Similarly to the propose work, the authors encode the solutions as a list of nodes' coordinates and use a Boolean disk coverage model. As a main contribution, they propose a normalization method for improving the results of the encoding technique. Regarding our work, there are important differences with respect to the proposed approach. In [32], the authors do not require the connectivity among nodes as a mandatory requirement. Also, the fitness function used in [32] only takes into account the coverage capacity of the deployment without determining other aspects such as fault-tolerance and redundancy. Another important difference is that in [32] the nodes can have different transmission ranges.

In [33], the authors proposed a heuristic algorithm called *Spiral* to determine the positions of a set of UAVs acting as Mobile Base Stations (MBSs). The idea is that they use the perimeter of the ground nodes to be covered to place the set of UAVs. The UAVs form a spiral within the target area towards its center. The results in [33] are good; however, they are significantly different from the work presented in this paper on more than one front. First, in [33] the authors do not consider that the UAVs should form a connected network. Second, the Spiral algorithm is suitable for coverage, but does not take into account other objectives handled in this work such as fault tolerance and redundancy. Similarly in [35], the authors make use of robotic navigation algorithms to deploy an aerial–ground cooperative vehicular networking architecture optimizing the power consumption during the UAVs flight. This work is mainly focused on the communication side, and conducts a comparison of link latencies for ZigBee and IEEE 802.11a. The authors propose path planning techniques to adaptively deploy network in interest areas.

Game theory is used in [35] to improve coverage in 5G networks by placing UAVs acting mobile base stations. They propose two solutions: one in which the UAVs cooperate with static MBs, and a second where the UAVs cooperate themselves to find their optimal positions. In both cases, the idea is to place the UAVs according to the demands of the users on the ground and the entropy of the network. Finally, the authors in [35] apply the Network Bargaining Problem (NBP) to accurately map the UAVs to target areas. Game theory is also used in [37] to deploy an UAV-network; however, the objective is to maximize the lifetime of the network.

In [38], the authors propose the use of UAVs to deploy 5G dynamic cellular base stations in the aftermath of a disaster scenario. They use a brute force algorithm to find the most optimal positions for the UAVs. The brute force algorithm is an exhaustive search algorithm, which tests all possible candidate solutions for an optimization problem. With simulations, they demonstrate that throughput coverage and the 5th percentile capacity of the network can be maximized by optimally placing the UAVs. However, brute force algorithms are only suitable for relatively simple scenarios (e.g. involving a small number of UAVs), unlike the case of the coverage problem presented in this work. In addition to the brute force algorithm in [38], the authors use other metaheuristic optimization techniques such as the Particle Swarm Optimization (PSO) [23] to place relay nodes in a multi-hop network in order to maximize the connectivity of the network [39]. They define connectivity as a combination of inter-node reachability and network throughput. The objective of the approach proposed in [38] is to optimally place a number of relay nodes to enhance the connectivity of the network from the users' point of view. In [40], the authors propose a neighborhood search optimization algorithm to design a robust and efficient deployment of nodes in a wireless body area network. This robust deployment considers the uncertainty of traffic among nodes in the network.

Other related works [41][42][43] are focused on placing Wi-Fi routers to form multi-hop mesh networks. Our work differs from the previous works [41][42][43] in several points. First, the previous works do not consider that the Wi-Fi routers have to form a connected mesh network. Second, they assume that the Wi-Fi routers have to be placed in a grid so that the discrete search space is reduced in comparison to the continuous search space considered in this work. Although these works also use genetic algorithms, they are mainly focused on one objective and apply single-population GAs. To the best of the authors' knowledge this is one of the first works that uses multi-population GAs for multi-objective covering problems in UAV-networks in order to tackle the complexity of deploying UAV-networks in communication and monitoring applications.

3. Multi-objective coverage deployment problem

This section is divided into two parts. First, we detail the statement problem. For this purpose, we define mathematically up to three network objectives for the UAV-network deployment such as coverage, fault tolerance, and redundancy. Then, we present the proposed weighted coverage optimization fitness function that takes into account these three network objectives simultaneously. Moreover, we provide some guidelines for the selection of the weights based on the importance of each objective for the performance of the UAV-network.

3.1. Problem statement

We consider a set of UAVs $U = \{u_1, u_2, ..., u_n\}$, where *n* is the number of UAVs, and a set of ground nodes $G = \{g_1, g_2, ..., g_k\}$, where *k* is the number of ground nodes, in a two dimensional (2D) area of interest *A*. Each UAV $u_i, i = 1..n$ is located at coordinates (x_i, y_i) within the boundaries of *A*. Similarly, each ground node $g_j, j = 1..k$ is located at coordinates (x_j, y_j) within the same area *A*. For the correct performance of the UAV-network, we require that the set *U* has to form a connected network.

DEFINITION 1. A network is said to be connected when there is path between every pair of nodes. In a connected network, there are no unreachable nodes.

Furthermore, the UAVs have a wireless transmission range *r*, and can cover ground nodes that are within its wireless transmission range. Also, communications among UAVs are possible as long as the UAVs can cover each other.

DEFINITION 2. A node is said to be covered when it is located at a Euclidean distance *d* shorter than or equal to *r* from an UAV.

Therefore, the disk model or Boolean disk coverage model is used [17][32]. This model considers as a Boolean variable the connectivity between an UAV and a ground node, and between UAVs. This model is given by:

$$f(d(u_i, g_j)) = \begin{cases} 1 & if \quad\quad d(u_i, g_j) < r \\ 0 & otherwise \end{cases} \quad (1)$$

It is important to recall that the Boolean disk model is widely used in coverage problems in wireless multi-hop networks [17][32]. Notice that the simplification of using a 2D problem only affects the calculation of the Euclidean distance *d*, but it does not undermine the validity of the model. Using the Boolean disk model (1), we define the set covered $SC_i \subseteq G$ of a UAV $i = 1..n$ as the set formed by including the ground nodes that return true (1 value) using (1). Thus, $g_j \in SC_i$ iff (if and only if) $f(d(u_i, g_i) = 1$. Next, we define the total coverage of the UAV-network *CO* as

$$CO = \bigcup_{i=1}^{n} SC_i \quad (2)$$

Therefore, the coverage of the UAV-network is given by the union of the *SCs* of every UAV in the network. The primary objective of the UAV-network should be to find the coordinates (x_i, y_i) to maximize CO:

$$\max CO = \bigcup_{i=1}^{n} SC_i \quad \text{s.t. U is connected} \quad (3)$$

Figure 1 illustrates graphically the coverage definition of an UAV-network. In Figure 1, the blue ground nodes are covered by at least one UAV. However, the red ground nodes are not covered because they are not within the transmission range of any UAV.

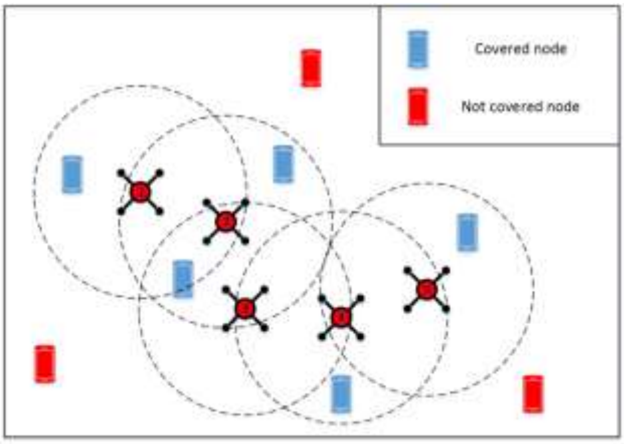


Figure 1. Example of ground nodes covered and not covered by an UAV-network

Next, we focus on fault tolerance of UAV-networks. As mentioned, the Fault tolerance is the ability of a network to continue to perform a required function in the presence of faults or errors [18][19]. We consider that an UAV-network will not function properly if the network is divided into different subnetworks. The reason is that we assume that there is one UAV that has the capability of connecting to the Internet via a long distance link, for instance, using satellite technology, and sharing such Internet connection with the other UAVs. We consider that this Internet connection is relevant for the communication services offered by an UAV-network to the ground nodes.

DEFINITION 3: we say that an UAV-network *U* has a fault tolerance TFO of *m*, if we remove whatever *m* elements from *U* and it remains a connected network.

Thus, the secondary objective of an UAV-network is to find the coordinates (x_i, y_i) that maximizes FTO.

$$\begin{aligned} &\max FTO \\ &\text{s.t. U is connected} \end{aligned} \tag{4}$$

Notice that the aforementioned definition of fault tolerance is not related the coverage provided by an UAV-network to the ground nodes. Therefore, the CO level of the UAV-network can be affected by the failures of some UAVs. To enhance the robustness of the deployment provided by the UAV-network, we also take into account, as the third objective, the redundancy among UAVs and ground nodes coverage. The idea is that a ground node can be covered by two or more UAVs simultaneously. This feature can also been seen as the accessibility of the UAV-network from the ground nodes.

DEFINITION 4: we say that a ground node *j* has a redundancy or accessibility *of t,* if it can be covered simultaneously from *t* different UAVs.

Using this definition of redundancy and the *SC* definition, we can calculate the total redundancy or accessibility of the network denoted by RO as

$$RO = \sum_{i=1}^{n} |SC_i| \tag{5}$$

Therefore, the redundancy objective of an UAV-network deployment is to find the coordinates (x_i, y_i) to maximize RO:

$$\max RO = \sum_{i=1}^{n} |SC_i| \\ \text{s.t. U is connected} \tag{6}$$

At this point, it is important to highlight the difference between FTO and RO. FTO seeks to maintain *U* connected in order to be still operative in the case of UAVs' failures, while RO is aimed at increasing the accessibility of the UAV-network from the ground nodes' point of view. Therefore, RO seeks to maintain the CO level in case of UAVs' failures. Notice that although they are related to each other, they are intended for different purposes.

Regarding the feasibility of the presented optimization problems, such as (3), (4), and (6), they are feasible problems since there is a possible solution for the three problems. However, finding the most optimal solution of these problems is not an easy task due to the high number of feasible solutions. Notice that these problems are unbounded since the coordinates of the UAVs are continuous variables. Furthermore, these problems have been demonstrated to be NP-complete problems [21][22]. For that reason, a suitable heuristic algorithm like the one presented in this work is necessary to tackle large-scale instances of the problem.

3.2. Proposed weighted optimization fitness function

The UAV-network deployment needs to be designed to optimize simultaneously the three defined objectives given by CO (3), FTO (4), and RO (6) respectively. However, these three objectives are counterbalanced in that increasing CO will require spreading out the UAVs throughout the area A in order cover as much area as possible. Conversely, the FTO objective will try to concentrate the coordinates of the UAVs in a short area so as to increase the coverage among them, and consequently, achieve better FTO values. Finally, RO will force the UAVs positions towards the area with higher concentration of ground nodes. However, if ground nodes are grouped into separated clusters, RO will tend to disconnect the UAV-network.

In order to address the 'counterbalance' issue among the defined objectives, we propose a weighted function F (7), which allows us to set different levels of importance for each objective (CO, FTO, and RO).

$$F = w_1 CO + w_2 FTO + w_3 RO \tag{7}$$

Where, w_1, w_2, and w_3, are the weights that set the contribution of each objective to the quality of the deployment. According to the priority order that we have established in previous section 3.1 for the three objectives, we consider that the following conditions should be met:

$$\begin{aligned} & \boldsymbol{w}_1 >> \boldsymbol{w}_2 \\ & \boldsymbol{w}_1 >> \boldsymbol{w}_3 \end{aligned} \tag{8}$$

The reasons to select these conditions are:

- By selecting $w_1 >> w_2$ and $w_1 >> w_3$, we ensure that *CO* will have higher weight than FTO and RO in (7) irrespective of the distribution of ground nodes in the area of interest *A*. If $w_1 = w_2 = \mathrm{w}_3$ or $w_1 > w_2, w_3$, there can be two possible issues. First, the FTO objective can force the deployment to a trivial solution, which is concentrating all UAVs around the same point. Second, RO can tend to concentrate the UAVs in dense areas of ground nodes, leaving uncover ground nodes located in other sparser areas. Furthermore, we assume that the major requirement of the UAV-network to be operative is to form a connected network. Therefore, FTO is a desirable quality but it is not mandatory for the operation of the network.
- Similarly, if $w_2 = w_3$ or $w_2 > w_3$, the RO term will dominate FTO in its contribution to *F* in (8). However, it is more desirable to have a higher level of fault tolerance than redundancy. Notice that by the definitions of coverage, fault tolerance, and redundancy, the values of RO will be always higher than CO and FTO. Notice that $RO \geq CO$ and in the worst case $RO = CO$, where each UAV covers a different ground node.

In section 5, we test the proposed weighted optimization function with a set of values for w_1, w_2, and w_3 according to the number of UAVs used in the simulation study and the considered scenarios.

4. Evolutionary Algorithms: Single-population GA and Multi-population MPGA

Genetic algorithms are the most classical and used EA to solve optimization problems. It was proposed by John Holland back in 1970s [45]. A GA is a metaheuristic and stochastic algorithm based on the Darwinian evolution theory. It is also well-known as a guided and population-based search optimization algorithm. The main idea is that a population (set) of candidate solutions, which are called individuals, evolves through a number of generations (iterations) providing better (fitness function) solutions to the target optimization problem. A candidate solution is encoded in a chromosome like structure called genotype. Each gene represents part of the solution; normally, the genes are the independent variables of the fitness function. Two basic genetic operators are used to create new individuals (offspring) based on the existing individuals which are crossover and mutation. On the one hand, the crossover operation consists of exchanging or swapping genetic information (genes) from two selected individuals (parents) so as to create two new individuals (children). The way used to exchange the genes among the parents determines the crossover schema. There are many schemes available in the literature such as one-point, two-point and uniform crossover [23]. On the other hand, mutation operation consists of modifying the genes (one or multiple genes) of a selected individual in order to create a new one. Many mutation schemes can found in the existing literature such as Gaussian and shuffle schemes [23]. These two genetic operations are probabilistic in that they are only employed under prefixed probabilistic values such as crossover probability p_c and mutation probability p_m.

The main difference of Multi-Population Genetic Algorithms (MPGAs) with respect to GAs is that several subpopulations iterate in parallel and share individuals, called migrants, among them in order to cooperatively improve the evolving solutions. MPGAs are also well-known as cooperative or coevolution-based EAs [23]. The idea is that the multiple subpopulations cooperate towards a joint objective. It has been demonstrated that suitable individuals for an optimization problem can emerge from different genetic strategies [47]. Therefore, joining such efforts from different subpopulations can produce important benefits in the global evolution of the ecosystem. Each subpopulation can be considered as different species.

Regarding the specific properties of MPGA, the migration scheme determines how the individuals of different subpopulations move from one subpopulation to another. Therefore, there are many MPGA implementations in the literature depending on the migration scheme used, such as ring, mesh, random, among others. The migration rate determines the number of individuals that move among the subpopulations. This tuning parameter will impact on the results obtained. Therefore, it has to be selected carefully.

In this paper, we present a Multi-Layout Multi-Population Genetic Algorithm (MLMPGA) in which several subpopulations use different evolutionary strategies and merge their efforts to achieve a common multi-objective. In MLMPGA, each subpopulation optimizes the fitness function in parallel. In this case, each subpopulation uses a different implementation of the GA such as crossover and mutation schemes or probabilities. With this strategy the idea is to overcome one problem related to EAs in general, that is choosing the optimal configuration of the EA parameters for a target optimization problem [47]. The solution to this problem is very difficult and not universal [48]; thus, usually through trial-and-error methods an optimal configuration may be approximated. The present study uses and compares the proposed MLMPGA against single-population GAs in order to arrive at an optimal multi-objective coverage deployment of an UAV-network in a target scenario.

4.1. Evolutionary algorithm configuration

Since we are seeking for UAV-deployment in 2D space, the chromosome structure should represent the 2D coordinates of a set of UAV deployed in the target scenario. Therefore, an individual is represented as a list of 2D coordinates as follows:

$$Individual = \left[(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\right]$$

Where, n is the number of UAV forming the deployed network. The coordinates are limited by the boundaries of the target scenario.

We use an (μ, λ) implementation for the GA layout. It means that μ parents are selected to create λ new individuals. Then, the created individuals replaced the parents. The only difference from the classical (μ, λ) is that we emphasis the use of elitism to guarantee that the fittest parents always survive through the generations. The values of μ and λ will depend on GA layout (crossover and mutation percentages), but we fix the condition $\mu = \lambda$. We define χ as the number of individual created by crossover operation and η as the number of individuals created by mutation. The condition $\lambda = \chi + \eta$ is met. Regarding elitism mechanism, the best ε individuals from μ pass directly to the next generation. Therefore, $\overline{P} = \lambda + \varepsilon$, being $\overline{P}$ the number of individuals forming the population. Figure 2 shows the pseudocode of the GA. First, an initial population should be created. Then, this initial population is evaluated. Next, the initial population evolves through a number of generations using the genetic operators such as selection, crossover, mutation, and elitism (preserving the best individuals at each generation). At the end of each generation of the evolution, the offspring replace the old population. Once the evolution is finished, the best individual within the population is selected as the solution of the optimization problem.

Algorithm: GA(n, ε, χ, η)
// **Initialize generation 0:**
k := 0;
P_k := a population of n randomly-generated individuals;
// **Evaluate** P_k:
Compute *fitness(i)* for each i ∈ P_k;
do
{ // **Create generation** k + 1:
    // **1. Copy:**
    Select ε × n members of P_k and insert into P_{k+1};
    // **2. Crossover:**
    Select χ × n members of P_k; pair them up; produce offspring; insert the offspring into P_{k+1};
    // **3. Mutate:**
    Select η × n members of P_{k+1}; invert a randomly-selected bit in each;
    // **Evaluate** P_{k+1}:
    Compute *fitness(i)* for each i ∈ P_k;
    // **Increment:**
    k := k + 1;
}
while k < maximum number of generations;
return the fittest individual from P_k;

Figure 2. Genetic algorithm pseudocode

The tournament schema has been chosen as the selection mechanism for both GA and MPGA. In this schema, individuals content with each other in a tournament to be selected as a parent. In tournament selection, *k* nodes are selected randomly among the population list. Among the *k* individuals, the one with the highest fitness is chosen as the parent. Therefore, for μ individuals forming the offspring, $2\chi + \eta$ tournaments have to be carried out. Among the possible *k* values, $k = 2$ and $k = 3$ seem to be suitable values according to the specific scientific literature [23].

Regarding mutation operation (see Figure 3), we use a shift mutation operation that consists of modifying the genes of a selected parent by adding or subtracting a given value, namely *shift* (Δ in Figure 3). We demonstrate that this mutation schema provides suitable results for the target coverage problem [49]. It outperformed other well-known mutation schemes such as shuffle indexes and swapping coordinates. Figure 3 shows an example of the shift mutation algorithm. As mentioned, mutation is also a probabilistic operation. Therefore, at each iteration a random number *rand* is generated within the interval $[0,1]$, if $rand > p_m$, the selected individual is mutated, being p_m the mutation probability. Otherwise, the selected parent is not modified. If the selected individual is mutated, the mutation can affect each UAV in the chromosome probabilistically according to p_m' (see Figure 3). Furthermore, this mutation scheme can affect, adding or subtracting the shift value, any of two coordinates (x or y coordinates) of the positions of the UAVs. Notice that the modification of the selected coordinate should remain within the boundaries of the scenarios. Otherwise, the opposite operation is applied to the individual. Finally, after modifying the individual, we should check that the network remain connected. If the new UAV-network is not connected, the mutation is not put into practice.

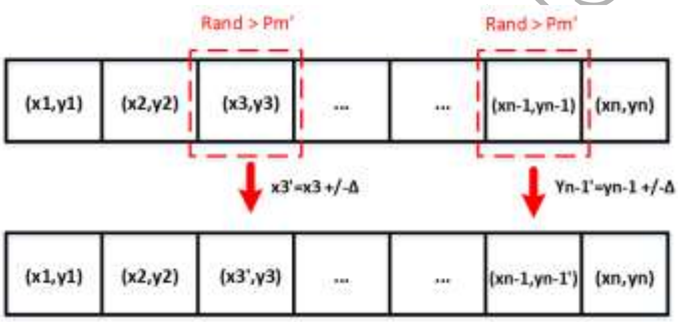


Figure 3. Shift mutation example

The fitness function is the expression that determines the quality of the individuals. In the coverage problem under study, the deployed UAV-network should maximize the aforementioned utility function (7). The only constraint is that the network must form a connected network. Therefore, the fitness function used to evaluate the individuals is

$$\begin{array}{ll} F = w_1 CL + w_2 FTL + w_3 RL & if \quad G(connected) \\ F = -1 & Otherwise \end{array} \tag{8}$$

In (8), *G* refers to the network formed by the UAVs. Notice that, we use the maximum (death) penalty for those individuals where the UAV-network is not connected. Consequently, the individuals penalized will not to take part of the genetic operations.

Finally, we consider as the stopping criterion a maximum number of generations. The criterion adopt is to guarantee that the convergence of the algorithm is reached before stopping the execution. It means that we can observe some plateau in the evolution of the best individuals throughout the generations.

4.2. Multi-Layer Multi-Population Genetic Algorithm (MLMPGA) configuration

Now, we describe the proposed MLMPGA, which is based on several subpopulation using different evolution strategies.

MLMPGA layout and evolution

The proposed MLMPGA technique has several subpopulations, each one having a different layout. We consider *N* subpopulations evolving with different genetic configurations. That is, we choose different χ and η for each subpopulation. In this MLMPGA layout, the *N* subpopulations have the same fitness function, which is given by (8). Each subpopulation represents different species, which uses different layout to generate the offspring. The GA layout presented in previous Section 4.1 is used for each subpopulation (see Figure 2). This layout is aimed at overcoming EA configuration since the different layout considered for each subpopulation can adapt to the requirements of the optimization problem. The evolution of each subpopulation of MLMPGA is the same presented in Figure 2. When the MLMPG finishes its evolution, the best individual from all the subpopulations is selected as the best individual.

MLMPGA migration schema

An important aspect in MPGAs is the migration schema used to share individuals among the different subpopulations. In fact, this is the main difference in comparison with single-population GA. Different migration schemes are adopted in the literature such as random, mesh, and ring schemes [27]. In our implementation, we use the migration scheme. In the ring scheme, the subpopulations share individuals following a ring connection configuration like the one shown in Figure 4.

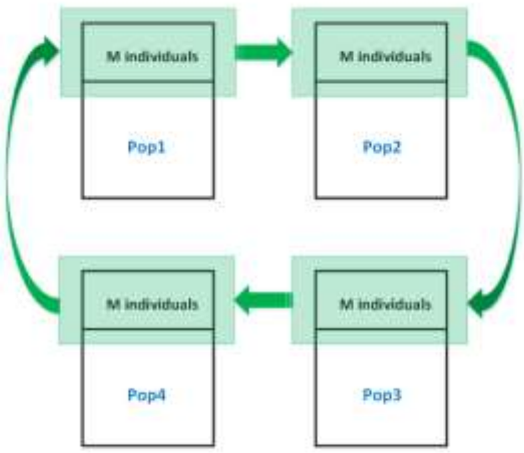


Figure 4. Example of ring migration schema with four subpopulations

M individuals of P_i replace *M* individuals in P_{i+1}, for $i \in [0, N-1]$ and *N* the number of subpopulations. For P_n, the adjacent population is P_0. The M individuals are selected using elitism, consequently, the *M* best individuals of each subpopulation are selected as migrants. This migration schema is suitable for MLMPGAs. In Figure 4, the rationale behind the ring migration schema can be graphically observed for 4 subpopulations, which is the number of subpopulations used in this work.

Another selection parameter is the migration frequency m_f, which indicates how often the individuals are exchanged among subpopulations. This value should not be too low to allow that the strategy used by each subpopulation takes place and affects on the cooperative evolution. Also, it should not be very high; otherwise, the multi-subpopulation algorithm may 'approximate' the behavior of a large single-population genetic algorithm since little information would be share among species (different subpopulations).

5. Simulation Study

This section presents the simulation experiments design, results and performance evaluation of the EA techniques described in this paper. The objective of this simulation study is to demonstrate the validity of the proposed deployment approach by investigating how the proposed MLPGA adapts to the search space of the coverage optimization problem under consideration, compared to classical GA configurations. For this purpose, a target scenario with a variable number of ground nodes non-uniformly distributed over a 1000 x 1000 m^2 region was considered. Such scenarios have been used to solve the network coverage problem. For instance, a non-uniform deployment of nodes aggregated in several clusters was used in [50] for coverage preservation in wireless sensor networks. Sparsely and non-uniformly deployed networks have been shown to be more challenging for the network coverage problem than densely or randomly deployed networks [50][51]. The implementation details, the selection of the parameters of the evolutionary techniques and the discussion of the simulations results are also detailed in the subsequent subsections.

5.1. Simulator

A simulator was entirely developed by the authors in Python 2.7. The Python package DEAP is used to develop the EAs [52] and the Particle Swarm Optimization used for comparison. The simulator is available online in [54]. Furthermore, the networkx library has been used for some graph theory algorithms. For instance, to check that the set of UAV form a connected network, the is_connected function is used. The algorithm used to determine the fault-tolerance is Algorithm 11 in [45], which is implemented in the Python library Networkx [55] as minimum_node_cut function. The simulator can be easily extended to add new features, such as optimization algorithms, fitness functions, and positions of ground nodes thanks to its modularized structured. In general, it consists of several Python modules and currently allows the users to configure simulation parameters, such as number of simulation, optimization algorithm used, and numbers of UAVs that form the UAV/network, among others.

5.2. Simulation scenario

We consider a simulation scenario formed by a number k of ground nodes and different number of UAVs (10, 14, and 18 UAVs). Our simulations use the *k* values, which are within the list 50, 75, 100 and 125 (Figure 5).

As shown in Figure 5, the ground nodes in our scenario are spread over two, three or four main clusters. Such distribution is challenging for the coverage problem since it is difficult to cover the four corners of the scenario while maintaining at the same time the UAV-network connected as a multi hop network at the same time. Notice that some UAVs need to be deployed in the central area of the scenario, where there are no ground nodes. Therefore, there are no incentives for the EAs to place nodes in those positions. Examples of such distribution can be found in real life. For instance, people normally tend to concentrate in groups during public events, such as concerts and festivals. Another possible scenario is disaster scenarios, where groups of people may remain isolated without a communication infrastructure to communicate with other groups or emergency services.

Regarding the components of the fitness function (8), it is expected to have higher *FTO* and *RO* values for low number of ground nodes, and vice-versa. As for the *CO*, it is more challenging as the number of ground nodes increases. Therefore, the four scenarios are considered suitable for evaluating the EA in different performance conditions.

The main features of the scenarios are included in Table 1. We consider that the ground nodes are static. Although in a real scenario the ground nodes can be mobile, the adaptation of the UAV positions to the mobile conditions is out of the scope of this paper, which is more focused on addressing the problem from the coverage point of view. The UAV transmission range is 250 m, which is in line with many Wi-Fi studies [56][57]. We have considered 10 and 14 UAVs.

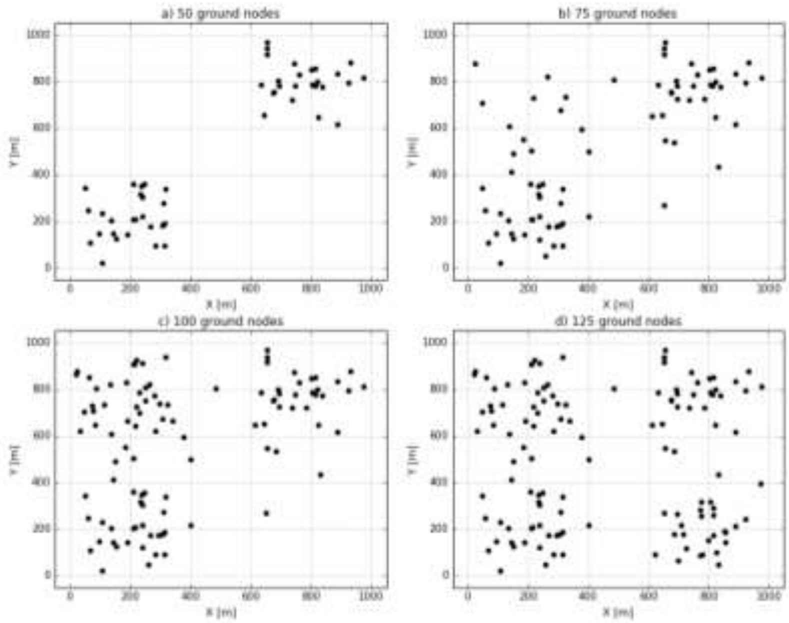


Figure 5. Scenarios considered in the simulations: a) 50 ground nodes, b) 75 ground nodes, c) 100 ground nodes, and d) 125 ground nodes

Simulation parameter	Value
Area dimensions	1000 m x 1000 m
Total Nº ground nodes	[50, 75, 100, 125]
Mobility of ground nodes	Static
Total Nº UAV	[10, 14, 18]
UAV's transmission radio range	250 m

Table 1. Simulation parameters for the target scenario.

5.3. GA and MPGA settings

We focus on detailing the tuning parameters used for both evolutionary techniques (Table 2). Some of the parameters contained in Table 2 refer to both the GA and MLMPGA implementation while other parameters, such as the number of subpopulations, the migration type and the migration frequency, are only applicable in the case of the MPGA (Section 4.3).

We test a wide range of crossover percentages for both the GA and the MLMPGA. In order to make a fair comparison among the evolutionary techniques, the GA technique uses different configurations and combinations in terms of χ and η values. Furthermore, as the MLMPGA demands more computational resources in terms of fitness functions evaluations, we run 4 times more trials of the GA technique than the MLMPGA. Consequently, the GA is executed for 120 independent trials, and the MLMPGA is run for 30 independent trials.

In the case of MLMPGA implementation, each subpopulation uses different χ and η values and all subpopulations use the same p_c and p_m. Each population contains 60 individuals, and after 5 generations they conduct ring migration. To this goal, 10 individuals replace 10 individuals of an adjacent subpopulation. The four subpopulations employ expression (8) as fitness function. As for the p_m, values higher than 0.5 are normally used to guarantee a suitable ratio of crossover.

Regarding the elitism percentage ε, we choose a value of 0.1 since the elitism should be selected as a low value [47]. Higher values may impact negatively on the diversity of the populations [24]. Furthermore, tournament selection is used. We have also tested roulette wheel section; however, the GA-based approaches require higher time for convergence, and in general, the obtained results are worse.

Tuning parameter	**Value**
Crossover probability (p_c)	0.6
Crossover percentage (χ)	[0.5, 0.6, 0.7, 0.8]
Mutation probability (p_m)	0.05
Mutation percentage (η)	[0.4, 0.3, 0.2, 0.1]
Elitism percentage (ε)	0.1
Nº Individuals (I)	60
Nº Generations (stopping criterion)	150
Selection type	Tournament (k= 3)
Crossover type	Two-point
Mutation type	Shift (Δ=5m)
Nº subpopulations (N)	4
Migration type	Ring
Nº migrants (M)	10
Migration frequency (m_f)	5
Fitness function	Equation (8) $w_1 = 1000, w_2 = 100, w_3 = 1$

Table 2. Tuning parameters used to conduct the simulations for GA and MPGAs

5.4. Simulation results

This section presents the simulation results obtained for the EAs under different numbers of ground nodes [50, 75, 100, 125] and three numbers of UAVs [10, 14, 18] in order to produce scenarios with a ranging degree of demand in terms complexity and search for optimization. The proposed MLMPGA may be evaluated under other ranges in terms of ground nodes and UAVs, bearing in mind that the computational time required obtaining a good solution increases with the number of UAVs and the number of ground nodes. The simulation results have been obtained using a workstation Intel ® Xeon® CPU ES-2603 v3 @1.60 GHz (2 processors) and Windows Server 2012 R2. The time required by the proposed MLMPGA is less than 20 minutes for the more complex scenario tested (18 UAVs and 125 ground nodes).

In the following tables (Table 3-Table 6), the headers represent the next metrics: UAV is the number of UAVs used, CO is the coverage objective in (8), FTO is the fault-tolerance objective in (8), RO is the redundancy objective in (8). F (8) max. is the maximum fitness function value achieved through all the trials (30 for MLMPGA, and 120 for GAs), the F(8) mean field is calculated by averaging out all the trials, and the F(8) std. represents the standard deviation of the simulation results. In addition, we highlight the best EA with green color, and the runner-up EA with brown color for the sake of clarity. Also some remarkable results obtained by other configurations are highlighted with purple color. Notice that the value of ε is not included in Table 3-Table 6) because it is always set as 0.1.

Table 3 contains the obtained results for 125 ground nodes. For the case of 10 UAVs, the maximum fitness is obtained for MLMPGA, which achieves the best CO, FTO, and RO levels. These results are reflected in F(8)max and F(8)mean. With 10 UAVs, MLMPGA is closed to cover all ground nodes. The runner-up EA is GA with $(\chi = 0.8, \eta = 0.1)$ as layout, which covers 120 ground nodes. Regarding the results for 14 UAVs, the best algorithms are again MLMPGA and GA with $(\chi = 0.8, \eta = 0.1)$ respectively. With this number of UAVs, both techniques are able to cover all ground nodes and even MLPGA achieves FTO=2, which is a remarkable result. When 18 UAVs are used, the proposed MLMPGA outperforms the other meta-heuristics. The best single-population GA is $(\chi = 0.7, \eta = 0.2)$ configuration.

Table 4 contains the obtained results for 100 ground nodes. Again, the best EA is MLMPGA, which achieves the maximum values for the three components of the multi-objective fitness function such as *CO*, *FTO*, and *RO*. For 10 UAVs, MLMPGA is the only configuration that achieves to cover all ground nodes. The best GA layout is again $(\chi = 0.8, \eta = 0.1)$, which covers 99 ground nodes. For 14 UAVs, several approaches obtained FTO values higher than 1. Therefore, it is the RO metric that determines the best technique in terms of F(8)max and F(8)mean. Looking at Table 4, we observe that there are other configurations of GA such as $(\chi = 0.7, \eta = 0.2)$ and $(\chi = 0.6, \eta = 0.3)$, which also achieve significant values in terms of RO. If 18 UAVs are used, the proposed MLMPGA is able to reach a FTO of 3. In this case, the best singe-population configuration is $(\chi = 0.6, \eta = 0.3)$ since it achieves significant results in terms of RO.

UAVs	**EA**	χ	η	**CO**	**FTO**	**RO**	**F (8) max.**	**F (8) mean.**	**F (8) std.**
10	GA	0.5	0.4	119	1	255	119355	102626.6	7507.3
	GA	0.6	0.3	116	1	243	116343	105923.8	6093.0
	GA	0.7	0.2	115	1	233	115333	105294.5	6561.6
	GA	0.8	0.1	**120**	**1**	**256**	**120346**	**108228.9**	**6345.7**
	MLMPGA	n/a	n/a	**123**	**1**	**262**	**123362**	**118775.1**	**2516.63**
14	GA	0.5	0.4	**125**	1	310	125410	116817.8	4797.1
	GA	0.6	0.3	123	1	319	123469	116322.9	4942.4

	GA	0.7	0.2	124	1	324	124425	115614.6	4967.5
	GA	0.8	0.1	**125**	**1**	**327**	**125427**	**118633.5**	**4656.8**
	MLMPGA	n/a	n/a	**125**	**2**	**360**	**125560**	**124021.2**	**1461.5**
18	GA	0.5	0.4	125	1	449	125549	120113.9	4336.2
	GA	0.6	0.3	125	1	450	125550	121258.3	3074.4
	GA	0.7	0.2	**125**	**2**	**437**	**125637**	**122148.6**	**3309.9**
	GA	0.8	0.1	125	2	419	125619	122389.7	2605.6
	MLMPGA	n/a	n/a	**125**	**2**	**475**	**125675**	**125242.7**	**779.7**

Table 3. Simulation results for 125 ground nodes, and 10, 14, and 18 UAVs.

UAVs	EA	χ	η	CO	FTO	RO	F (8) max.	F (8) mean.	F (8) std.
10	GA	0.5	0.4	98	1	240	98340	91211.0	4609.1
	GA	0.6	0.3	98	1	245	98345	91966.7	4196.7
	GA	0.7	0.2	98	1	**260**	98360	91943.7	3955.0
	GA	0.8	0.1	**99**	**1**	**245**	**99345**	**93979.7**	**3911.8**
	MLMPGA	n/a	n/a	**100**	**1**	**261**	**100361**	**98221.8**	**1404.5**
14	GA	0.5	0.4	100	**2**	320	100520	96720.1	3207.14
	GA	0.6	0.3	100	1	**345**	100445	97300.4	2428.5
	GA	0.7	0.2	100	**2**	316	100516	97162.8	2663.4
	GA	0.8	0.1	**100**	**2**	**335**	**100535**	**98311.2**	**2098.4**
	MLMPGA	n/a	n/a	**100**	**2**	**381**	**100581**	**100152.0**	**662.4**
18	GA	0.5	0.4	100	2	455	100655	99077.4	2127.6
	GA	0.6	0.3	**100**	**2**	**460**	**100660**	**98915.0**	**1891.4**
	GA	0.7	0.2	100	2	433	100633	99443.6	1188.9
	GA	0.8	0.1	100	2	436	100636	99045.1	1742.9
	MLMPGA	n/a	n/a	**100**	**3**	**475**	**100755**	**100551.2**	**396.4**

Table 4. Simulation results for 100 ground nodes, and 10, 14, and 18 UAVs.

Table 5 shows the obtained results for 75 ground nodes. For 10 UAVs, the best EA is MLMPGA algorithm, which covers all ground nodes. The best GA configuration is $(\chi = 0.7, \eta = 0.2)$, which almost covers all ground nodes (74 see Table 5). Also, this GA configuration achieves a RO level even better than MLMPGA. According to the results in Table 5, for 14 UAVs the RO is the component of the multi-objective fitness function (8) that determines the differences among the approaches since there are several approaches that achieved to cover all ground nodes and FTO=2. The RO obtained by MLMPGA is 285, which is considerably better than the RO values obtained by the rest of approaches. The layout $(\chi = 0.6, \eta = 0.3)$ seems to be the most suitable for 14 UAVs since it is able to achieve a FTO=2 and a remarkable RO value with respect to the other GA layouts. If 18 UAVs are used, the proposed MLMPGA achieves the best results for the three objectives and GA with $(\chi = 0.7, \eta = 0.2)$ configuration is the best option for single-population schemes.

Table 6 shows the obtained results for 50 ground nodes. For 10 UAVs, the best EA is MLMPGA which achieved the best results in terms of CO and FTO components. The best EA in terms of RO is GA with layout $(\chi = 0.7, \eta = 0.2)$, which reaches up to 184 possible connections among UAV and ground nodes. GA with $(\chi = 0.6, \eta = 0.3)$ is the runner-up algorithm, achieving

FTO=2 as the main remark. Considering 14 UVAs, the main difference is made by those EAs that achieve FTO=3, such as the proposed MLMPGA and GA with $(\chi = 0.8, \eta = 0.1)$. The latter is the best algorithm since it obtains the best results in terms of RO. Notice that this is the only case tested where a single-population GA outperforms the proposed MLMPGA, being the difference only 3 possible connections. Therefore, we can say that the proposed MLMPGA outperforms single-population configurations for the majority of cases evaluated. When 18 UAV are employed, the proposed MLMPGA achieves a FTO of 4, which is a remarkable result. Again MLMPGA outperforms the other meta-heuristic optimization algorithms.

UAVs	EA	χ	η	CO	FTO	RO	F (8) max.	F (8) mean.	F (8) std.
10	GA	0.5	0.4	73	1	194	73294	68821.9	3165.2
	GA	0.6	0.3	**74**	1	188	74288	70157.6	3172.7
	GA	0.7	0.2	**74**	**1**	**244**	**74314**	**70227.6**	**3444.5**
	GA	0.8	0.1	73	1	202	73302	69761.1	2960.9
	MLMPGA	n/a	n/a	**75**	**1**	**197**	**75297**	**73982.1**	**924.4**
14	GA	0.5	0.4	75	1	257	75357	72666.6	2302.7
	GA	0.6	0.3	**75**	**2**	**267**	**75467**	**72819.6**	**2258.4**
	GA	0.7	0.2	75	1	**274**	75374	73272.2	1587.4
	GA	0.8	0.1	75	**2**	240	75440	73649.8	1241.1
	MLMPGA	n/a	n/a	**75**	**2**	**285**	**75487**	**75228.4**	**381.73**
18	GA	0.5	0.4	75	2	335	75535	73737.2	1674.5
	GA	0.6	0.3	75	2	346	75546	74389.1	989.1
	GA	0.7	0.2	**75**	**3**	**357**	**75657**	**74738.9**	**1061.2**
	GA	0.8	0.1	75	**3**	305	75605	74598.5	685.6
	MLMPGA	n/a	n/a	**75**	**3**	**353**	**75653.0**	**75526.8**	**179.4**

Table 5. Simulation results for 75 ground nodes, and 10 and 14 UAVs.

UAVs	EA	χ	η	CO	FTO	RO	F (8) max.	F (8) mean.	F (8) std.
10	GA	0.5	0.4	50	1	**182**	50282	46320.3	4607.7
	GA	0.6	0.3	50	**2**	**140**	**50340**	**48757.8**	**1593.1**
	GA	0.7	0.2	50	1	**184**	50284	48119.4	2321.0
	GA	0.8	0.1	50	1	179	50279	48148.4	1710.4
	MLMPGA	n/a	n/a	**50**	**2**	**158**	**50358**	**50261.0**	**183.5**
14	GA	0.5	0.4	50	2	192	50398	49043.5	1975.14
	GA	0.6	0.3	50	2	**237**	50437	49381.4	1400.5
	GA	0.7	0.2	50	2	227	50427	49904.4	622.27
	GA	0.8	0.1	**50**	**3**	**203**	**50503**	**49718.9**	**955.26**
	MLMPGA	n/a	n/a	**50**	**3**	**200**	**50500**	**50414.3**	**38.6**
18	GA	0.5	0.4	50	2	**276**	50476	49796.6	1028.8
	GA	0.6	0.3	**50**	**3**	**257**	**50557**	**49896.7**	**939.3**
	GA	0.7	0.2	50	3	253	50553	50236.9	741.2
	GA	0.8	0.1	50	3	255	50555	50219.0	555.2
	MLMPGA	n/a	n/a	**50**	**4**	**266**	**50666**	**50543.2**	**58.5**

Table 6. Simulation results for 50 ground nodes, and 10 and 14 UAVs.

Although converge information has not been included in the shown results, the convergence of the proposed MLMPGA is very similar to the single-population GAs. All EAs evaluated converge on average within the interval [90-120] generations.

5.5. Comparison with other well-known meta-heuristic optimization algorithm

The objective of this subsection is to compare the proposed MLMPGA optimization algorithm against other optimization algorithms, which have been widely used in the literature to solve optimization problems, such as Particle Swarm Optimization (PSO) [58] and Hill Climbing Algorithm (HCA) [24]. These two meta-heuristic algorithms have been selected for the following reasons. Frist, the PSO is a widely used trajectory and population based meta-heuristic optimization algorithm [24], which is suitable for optimization problems with continuous variables as the one addressed in this study. Second, the HCA is a simple meta-heuristic algorithm, which tries to improve an initial random solution by testing small modifications or new trajectories of the recorded best solution [24]. Therefore, we benchmark the proposed MLMPGA against an optimization algorithm that exploits local search methods. Notice that it can be seen as a pure mutation-based optimization algorithm. Furthermore, the random algorithm is used as worst case scenario, where the positions of the UAVs are selected randomly with the only requirement to form a connected UAV-network.

Figure 6 shows the pseudocode of the HCA implemented. The number of iterations used is 5000 and the HCA has been run for 30 independent trials. Figure 7 illustrates the pseudocode of the PSO algorithm used. The number of iterations has been set to 150, the number of particles is 60, the v_{max}= 5, C_{lmax}= 2, and C_{gmax}= 2. The PSO has been run for 30 independent trials for each considered scenario.

Algorithm: HCA()
// **Initialize iteration 0:**
iter := 0;
S_{iter} := an initial random solution;
// **Evaluate** S_{iter}:
Best:= Compute *fitness(*S_{iter}*)*;
do
{
    // **1. Modify current solution:**
    S_{new}= *modify*(S_{iter});
    // **2. Evaluate new solution:**
    $F_{new:}$:= Compute *fitness(*S_{new}*)*;
    // **3. Check improvement:**
    If (F_{new} > Best); S_{iter}:= S_{new}; Best:= F_{new};
    // **Increment iteration:**
    iter := iter + 1;
}
while iter < maximum number of iteration;
return Best;

Figure 6. HCA pseudocode

Algorithm: PSO(v_{max}, C_{lmax}, C_{gmax})

// **Initialize iteration 0:**
k := 0;
P_k := a population of n randomly-generated particles;
// **Evaluate** P_k:
Compute *fitness(i)* for each i ∈ P_k;
Compute $Best_{locali}$ for each i ∈ P_k;
Compute $Best_{global}(P_k)$
do
{ // **Update particles, run step 1-8 for each i ∈ P_k** :
// **1. Distance from the local, best d_{local}:= $Best_{local}$-i**
Compute d_{local};
// **2. Distance from the global, best d_{global}:= $Best_{global}$-i**
Compute d_{global};
// **3. Update speed of particles v_i:**
Select C_{local} and C_{global} from [0, C_{lmax}] and [0, C_{gmax}];
v_i:= C_{locali} * d_{locali} + $C_{globali}$ * $d_{globali}$;
// **4. Check speed limits:**
If $v_i > v_{max}$; vi: = v_{max};
// **5. Update positions:**
i:= i + v_i;
// **6. Evaluate P_{k+1}:**
Compute *fitness(i)*;
// **7. Update best local:**
If $i_{fitness} > Best_{locali}$; $Best_{locali}$:= i;
// **8. Update best global:**
If $i_{fitness} > Best_{global}$; $Best_{global}$:= i;
// **Increment:**
k := k + 1;
}
while k < maximum number of iterations;
return $Best_{global}$;

Figure 7. PSO pseudocode

We have compared the meta-heuristic optimization algorithms for 125 and 50 ground nodes (maximum and minimum values), and using 10, 14, and 18 UAVs. The obtained results are shown in Table 7 and Table 8. As it can be seen, the proposed MLMPGA outperforms all other metaheuristics. The differences are even bigger as the number of UAVs used increases. The reason behind is that as the number of UAVs gets higher the complexity of the optimization problem in terms of feasible combinations increments, and under this complex scenario, the multi-strategy used by the MLMPGA makes the difference with respect to single evolutionary strategies and/or local search mechanisms.

UAVs	EA	CO	FTO	RO	F (8) max.	F (8) mean.	F (8) std.
10	Random	85	2	167	85367	n/a	n/a
	HCA	95	1	255	95355	55865	15493.7
	PSO	107	1	229	107329	97371.4	5401.9
	GA (0.8,0.1)	120	1	256	120346	108228.9	6345.7
	MLMPGA	123	1	262	123362	118775.1	2516.63

14	Random	78	1	303	78403	n/a	n/a
	HCA	101	1	313	101413	68123	17438.8
	PSO	116	1	304	116404	108987.0	3790.8
	GA (0.8,0.1)	125	1	327	125427	118633.5	4656.8
	MLMPGA	125	2	360	125560	124021.2	1461.5
18	Random	90	2	314	90514	n/a	n/a
	HCA	118	1	445	118545	70663.0	23712.8
	PSO	125	1	381	125484	115806.6	4579.6
	GA (0.7,0.2)	125	2	437	125637	122148.6	3309.9
	MLMPGA	125	2	475	125675	125242.7	779.7

Table 7. Comparison of meta-heuristic algorithms for 125 ground nodes, and 10, 14, and 18 UAVs.

UAVs	EA	CO	FTO	RO	F (8) max.	F (8) mean.	F (8) std.
10	Random	36	2	77	36277	n/a	n/a
	HCA	40	2	128	40328	22228.0	8587.3
	PSO	50	1	152	50252	44066.1	3375.8
	GA (0.6,0.3)	50	2	140	50340	48757.8	1593.1
	MLMPGA	50	2	158	50358	50261.0	183.5
14	Random	46	1	127	46227	n/a	n/a
	HCA	43	2	103	43303	24034	9316.9
	PSO	49	1	177	49279	46109.2	2230.2
	GA (0.8,0.1)	50	3	203	50503	49718.9	955.26
	MLMPGA	50	3	200	50500	50414.3	38.6
18	Random	48	2	171	48371	n/a	n/a
	HCA	49	1	294	49394	27715	9629.19
	PSO	50	1	216	50416	46246	1404.9
	GA (0.6,0.3)	50	3	257	50557	49896.7	939.3
	MLMPGA	50	4	266	50666	50543.2	58.5

Table 8. Comparison of meta-heuristic algorithms for 50 ground nodes, and 10, 14, and 18 UAVs.

It is interesting to indicate that although the HCA does not achieve the best results, it is the one that lend itself best to performing in a distributed way. In this case, the UAVs will try random movements from initial positions and test two conditions, such as i) the solution is valid (UAV-network is connected) and ii) the solution improves the previous one (better fitness function). If the two conditions are met, the conducted movements are accepted; otherwise the UAVs should go backwards to the previous positions Therefore, in cases where there is no global information, it can be a suitable algorithm.

5.6. Discussion of simulation results

The following summary highlights the main findings derived from our simulation results:

- Among the studied EAs in this work, MLMPGA is the one that in general achieved the best results for the multi-objective problem. It has been the best EA in terms of CO, FTO, and RO in the majority of cases evaluated such as 125, 100, and 75 ground nodes, and 10, 14, and 18 UAVs. Only in the case of 50 ground nodes and 14 UAVs, a single-population GA configuration outperforms the proposed MLMPGA.

- Different GA layouts are able to achieve remarkable results depending on the scenarios. However, it seems that there is not a layout configuration optimal for all the studied cases. For 125 and 100 ground nodes the best GA layout is $(\chi = 0.8, \eta = 0.1)$ for 10 and 14 UAVs. In the case of 18 UAVs, $(\chi = 0.7, \eta = 0.2)$ for 125 ground nodes. For 75 ground nodes, $(\chi = 0.7, \eta = 0.2)$, and $(\chi = 0.6, \eta = 0.3)$ are better than $(\chi = 0.8, \eta = 0.1)$, when using 10, 14, and 18 UAVs respectively. Similarly, for 50 ground nodes and 10 UAVs $(\chi = 0.5, \eta = 0.4)$ is the best GA layout, for 14 UAVs $(\chi = 0.8, \eta = 0.1)$ is the best configuration, and $(\chi = 0.6, \eta = 0.3)$ is the best for 18 UAV. This result is relevant since trial-and-error calibration methods are high time consuming. This issue is significantly solved by the MLMPGA.
- The optimal configuration for the GA depends on the dominant component of the multi-objective fitness function (8) as it has been stated in previous findings. This is due to the fact that depending on the landscape of the fitness function a GA needs different rates of exploration/exploitation and vice versa [47]. The proposed MLMPGA achieves a suitable balance of both features by evolving different subpopulations with different levels of crossover and mutations layouts.
- The components of same multi-objective fitness function (7) change its importance or dominance with small changes. That is, by changing the number of ground nodes (125, 100, 75, and 50) and UAVs used (10, 14, and 18). For 125 nodes and 10 UAVs, CO is the most dominant component. However, if 14 and 18 UAVs are used, FTO is more dominant that CO due to the fact that there are enough UAVs to cover all ground nodes. For 100 ground nodes and 10 UAVs, CO is still the dominant component, but when using 14 and 18 UAVs, all EAs achieve similar results in terms of CO and FTO, and RO is the component that makes the difference. For 75 ground nodes, the distribution of these ground nodes is complex (see Figure 5) since there are still ground nodes in the four clusters. However, there are 3 clusters with much more ground nodes. It can lead to local optima due to RO component. However, CO is dominant for 10 UAVs due to the weights given in (7). FTO and RO determine the fitness when using 14 and 18 UAVs. Finally, for 50 ground nodes, CO is not relevant for 10, 14, and 18 UAVs, and FTO makes the difference for the three numbers of UAVs. Therefore and considering the previous remark 3, it is more suitable to use adaptive EA such as the proposed MLMPGA than fix configuration GA like the traditional single-population GAs.
- The proposed MLMPGA also outperforms other widely used metaheuristics, such as HCA and PSO. The differences are even bigger as the complexity of the scenarios increases.

Figure 8, Figure 9, and Figure 10 represent the best positions obtained by the best EA for the different scenarios considered (125 , 100, 75, and 50 ground-nodes) for three numbers of UAVs such as 10 (see Figure 8), 14 UAVs (see Figure 9), and 18 UAVs (see Figure 10). In Figure 8 -Figure 10, GN stands for ground-nodes, UAV-links represent the connections among UAVs, and GN-links indicate the possible links or connections among UAVs and ground-nodes (Notice that in Figure 8-Figure 10, the links among UAVs have not been plotted on the right plots just for improving the understanding of the figures since the objective of these plots is to highlight the possible connections among UAVs and ground nodes). It is interesting to observe how the UAV-network achieves a better FTO level as the number of ground-nodes is reduced and how the UAV-network obtains better FTO levels as it is able to arrange them in a ring configuration (see Figure 9 and Figure 10).

6. Conclusion

A multi-subpopulation genetic algorithm namely MLMPGA has been proposed and evaluated against single-population GAs for solving multi-objective coverage problems of UAV-networks under different scenarios. We have demonstrated that the proposed MLMPGA adapts better to the landscape and dominant components of the used multi-objective fitness function such as coverage objective, fault-tolerance objective, and redundancy objective, by employing a good balance between exploration and exploitation in search for optimum UAV deployment solutions. The idea is to take advantage of different GA layouts in a multi-subpopulation approach, where each subpopulation evolves largely independently and exchanges its best individuals to other sub-populations based on a migration model. We have also demonstrated that a coverage multi-objective problem may vary a lot in terms of landscape of the search space of possible solutions by varying the number of ground nodes to be covered and the number of UAV used. Since it is extremely difficult to find a suitable GA configuration that performs properly in all possible scenarios, it is more appropriate to use the proposed MLMPGA to guarantee suitable results under various scenarios. The proposed MLMPGA has been compared against other metaheuristics, such as classical GAs, HCA and PSO, and it clearly outperforms the other algorithms in 11 out of 12 considered scenarios. As future work, the following research directions can be considered: i) modelling the weighted fitness function as a multi-objective problem based on the Pareto dominance. The idea is to study the Pareto front solutions for the three coverage problems such as coverage, fault-tolerance, and redundancy; ii) relaxing the connectivity requirement among UAVs in the UAV-network. On this line, the idea is to consider the UAV-network as a Delay Tolerant Network (DTN) and take advantage of the possible encounters among UAVs to share relevant information of the network; and iii) evaluating the proposed weighted fitness function and MLMPGAs in other scenarios that follows other distributions of nodes such random, small-world, and exponential among others.

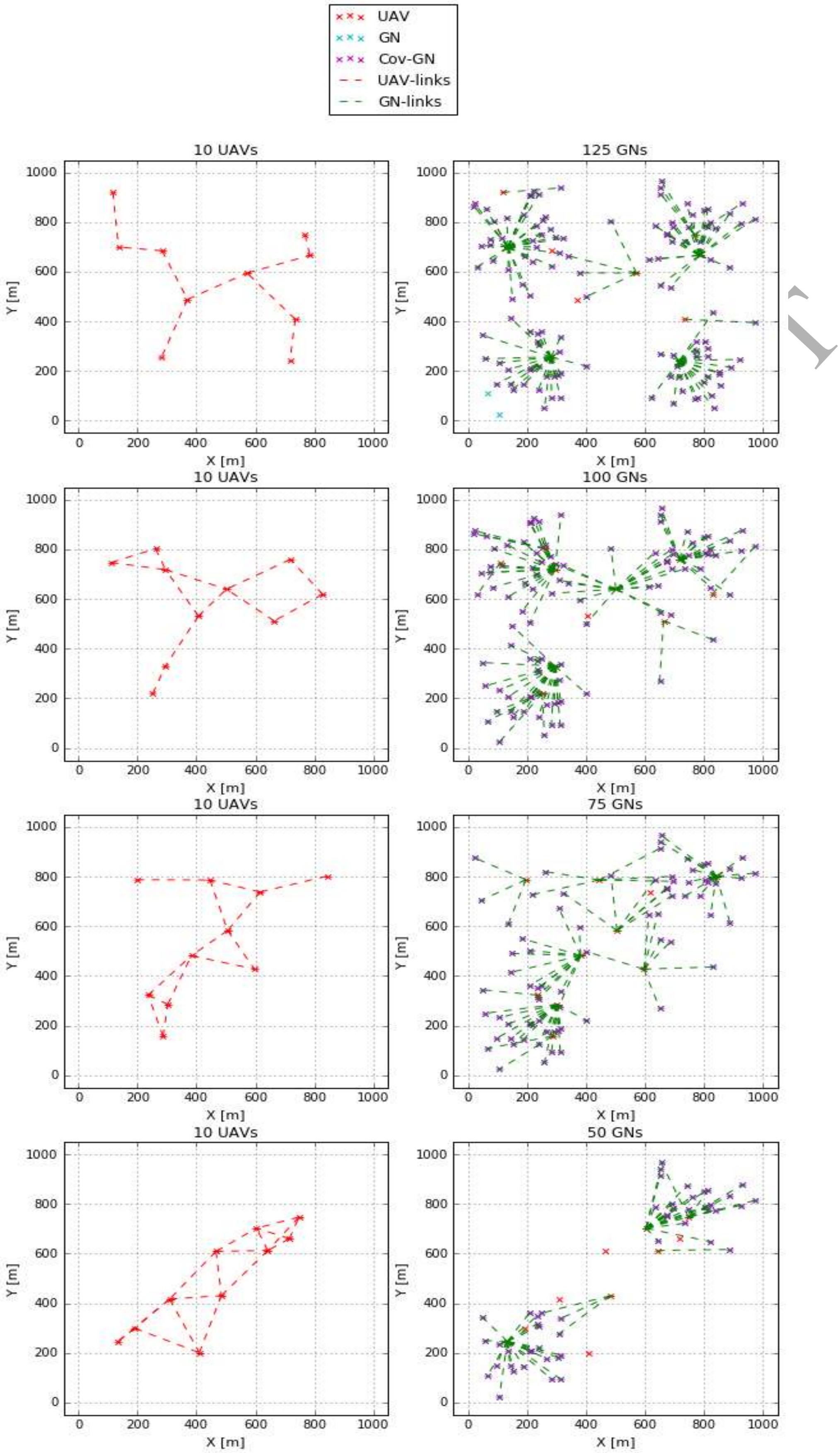


Figure 8. Best positions obtained by the proposed MLMPGA for 10 UAVs and different number of ground-nodes

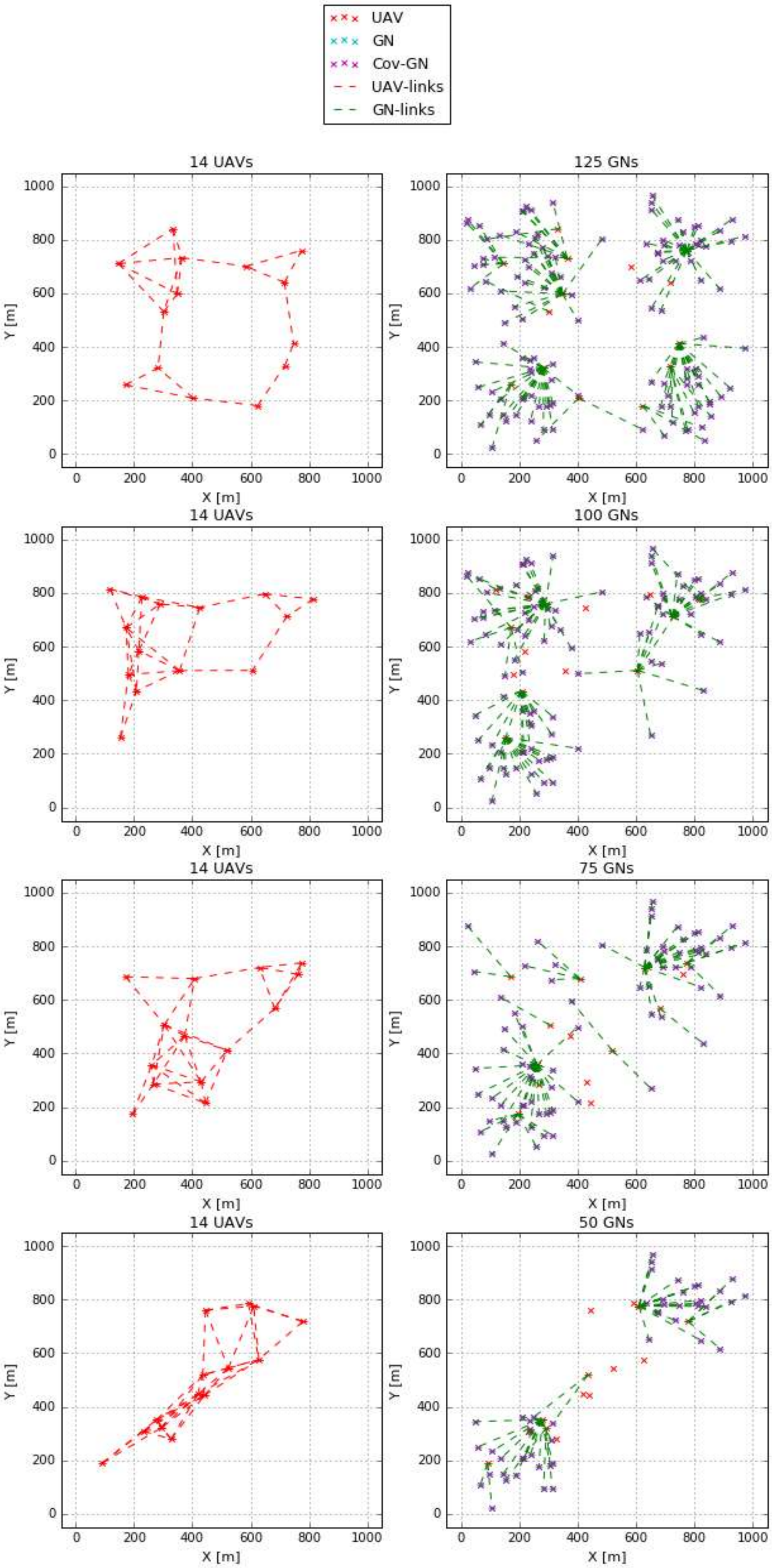


Figure 9. Best positions obtained by the proposed MLMPGA for 14 UAVs and different number of ground-nodes

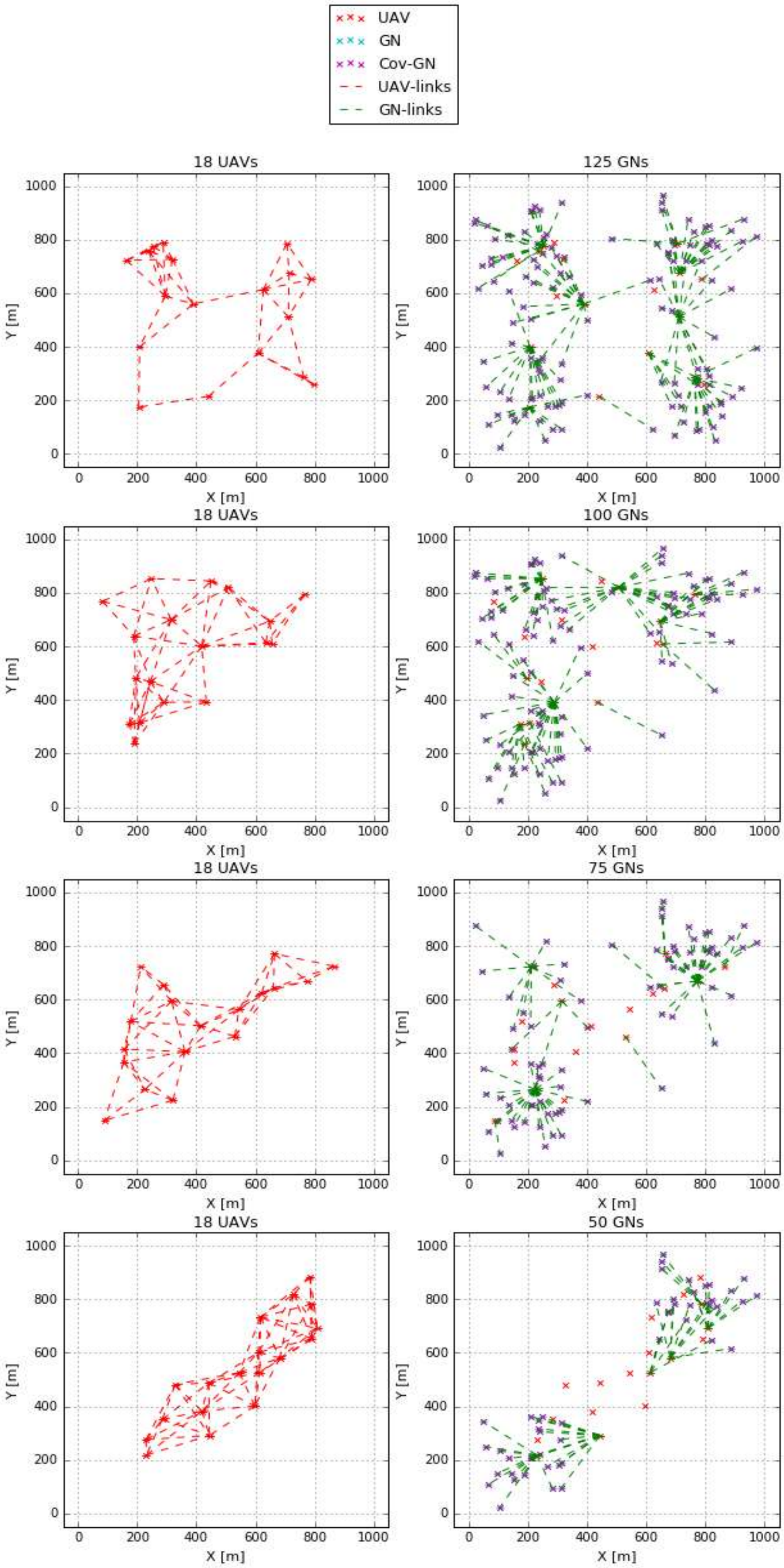


Figure 10. Best positions obtained by the proposed MLMPGA for 18 UAVs and different number of ground-nodes

Acknowledgments

The authors would like to thank Mónica Díaz López for her help in proofreading the manuscript.

References

[1] S. Hayat, E. Yanmaz and R. Muzaffar, "Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint", IEEE Communications Surveys & Tutorials. Vol. 18, pp. 2624-2661, 2016.

[2] I. Bekmezci, O. K. Sahingoz and Ş. Temel, "Flying Ad-Hoc Networks (FANETs): A survey", Ad Hoc Networks. Vol. 11, pp. 1254-1270, 2013.

[3] N. Kubota, Y. Toda and S. Suzuki, "Cooperative Formation of Multi-robot Based on Spring Model", in International Conference on Robot, Vision and Signal Processing, Kitakyushu, Japan, pp. 72 - 77. 2013.

[4] D. G. Reina, S. L. Toral, P. Johnson and F. Barrero, "A survey on probabilistic broadcast schemes for wireless Ad Hoc networks", Ad Hoc Networks. Vol. 25, pp. 263 - 292, 2015.

[5] M. Conti and S. Giordano, "Mobile Ad Hoc Networking: Milestones, Challenges, and New Research Directions", IEEE Communications Magazine. Vol. 52, pp. 85-96, 2014.

[6] J. M. García-Campos, J. Sánchez-García, D. G. Reina, S. L. Toral, F Barrero. "An evaluation methodology for reliable simulation based studies of routing protocols in VANETs". Simulation Modelling Practice and Theory. Vol. 66, pp. 139-165. 2016.

[7] G. R. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. Pérez-Costa, B. Walke. "The IEEE 802.11 Universe". IEEE Communications Magazine. Vol. 48, pp. 62-70. 2010.

[8] J. Sánchez-García, J. García-Campos, S. Toral, D. G. Reina and F. Barrero, "An Intelligent Strategy for Tactical Movements of UAVs in Disaster Scenarios", International Journal of Distributed Sensor Networks. Vol. 12, pp. 1-20, 2016.

[9] V. Sharma, D. G. Reina, R. Kumar. "HMADSO: a novel hill Myna and desert Sparrow optimization algorithm for cooperative rendezvous and task allocation in FANETs". Soft Computing. DOI 10.1007/s00500-017-2686-4.

[10] D. G. Reina, M. Askalani, S. L. Toral, F. Barrero, E. Asimakopoulou and N. Bessis, "A Survey on Multihop Ad Hoc Networks for Disaster Response Scenarios", International Journal of Distributed Sensor Networks. Vol. 11, pp. 1-16, 2015.

[11] M. Erdelj, E. Natalizio, K. R. Chowdhury, I. F. Akyildiz. "Help from the Sky: Leveraging UAVs for Disaster Management". IEEE Pervasive Computing. Vol. 16, pp. 24-32. 2017.

[12] S. F. Ochoa, A. Neyem, J. A. Pino, M. Borges. "Supporting group decision making and coordination in urban disasters relief efforts". Journal of Decision Systems. Vol. 16, pp. 143-172. 2007.

[13] Kennington, E. Olinick, D. Rajan. "Wireless Network Design: Optimization Models and Solution Procedures" Springer, 2010.

[14] L. Chiaraviglio, N. Blefari-Melazzi, W. Liu, J. A. Gutierrez, J. Beek, R. Birke, L. Chen, F. Idzikowski, D. Kilper, P. Monti, A. Bagula, J. Wu. "Bringing 5G in Rural and Low-Income Areas: Is it Feasible?". IEEE Communications Standards Magazine. Accepted in IEEE Communication Magazine.

[15]F. D'Andreagiovanni, C. Mannino, A. Sassano. "GUB Covers and Power-Indexed Formulations for Wireless Network Design". Management Science. Vol. 59, pp. 142-156. 2013.

[16]T. Bauschert, C. Büsing, F. D'Andreagiovanni, A. M. C. A. Koster, M. Kutschka, U. Steglich. "Network Planning under Demand Uncertainty with Robust Optimization". IEEE Communications Magazine. Vol. 52, pp. 178-185. 2014.

[17]C. Huang, Y. Tseng. "The Coverage Problem in a Wireless Sensor Network". Mobile Networks and Applications. Vol. 10, pp. 519-528. 2005.

[18]S. Chouikhi, Inès El Korbi, Y. Ghamri-Doudane, L. Azouz Saidane. "A survey on fault tolerance in small and large scale wireless sensor networks". Computer Communications. Vol. 69, pp. 22-37.

[19]A. Merwaday, I. Güvenç. "UAV assisted heterogeneous networks for public safety communications". IEEE Wireless Communications and Networking Conference Workshops (WCNCW), pp. 329-334. 2015.

[20]N. Wisitpongphan, O. K. Tonguz, J. S. Parikh, P. Mudalige, F. Bai, and V. Sadekar. Broadcast storm mitigation techniques in vehicular ad hoc networks. IEEE Wireless Communications. Vol. 14, pp. 84-94. 2007.

[21]Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo. "A Survey of Multi-Objective Optimization in Wireless Sensor Networks: Metrics, Algorithms and Open Problems". IEEE Communications Surveys & Tutorials. Vol. 19, pp. 550-586. 2017.

[22]F. Uriel. "A threshold of ln n for approximating set cover". Journal of the ACM. Vol. 45, pp. 634-652. 1998.

[23]K. Akkaya, M. Younis. "A survey on routing protocols for wireless sensor networks". Ad Hoc Networks. Vol. 3, pp. 325-349. 2005.

[24]S. Luke. "Essentials of Metaheuristics". A Set of Undergraduate Lectures. Online Version 2.2. 2015.

[25]D. G. Reina, S. L. Toral, N. Bessis, F. Barrero, E Asimakopoulou. "An evolutionary computation approach for optimizing connectivity in disaster response scenarios". Applied Soft Computing. Vol. 13, pp. 833-845. 2013.

[26]F. D'Andreagiovanni. "On Improving the Capacity of Solving Large-scale Wireless Network Design Problems by Genetic Algorithms" (Applications of Evolutionary Computation, LNCS 6625, 11-20, 2011, DOI: 10.1007/978-3-642-20520-0_2

[27]Erick Cantú-Paz. A Survey of Parallel Genetic Algorithms. Calculateurs Paralleles. Vol. 10. 1998.

[28]F. Lu, Y. Ge, and L. Gao. "A novel genetic algorithm with multiple sub-population parallel search mechanism." 2010 Sixth International Conference on Natural Computation (ICNC), Vol. 5. IEEE, 2010.

[29]S. Kumar-Gupta, P. Kuila, P. K. Jana. "Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks". Computers and Electrical Engineering. Vol. 56, pp. 544-556. 2016.

[30]A. Konstantinidis, K. Yang. "Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem specific constrained evolutionary algorithm based on decomposition". Computer Communications. Vol. 34, pp. 83-98. 2011.

[31]M. Rebai, M. Le Berre, H. Snoussi, F. Hnaien, L. Khoukhi. “Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks”. Computers & Operations Research. Vol. 59, pp. 11-21. 2015.

[32]Y. Yoon and Y. Kim. “An Efficient Genetic Algorithm for Maximum Coverage Deployment in Wireless Sensor Networks”. IEEE Transactions on Cybernetics. Vol. 43, pp. 1473-1483.

[33]A. Marotta, F. D’Andreagiovanni, A. Kassler, E. Zola. "On the energy cost of robustness for green virtual network function placement in 5G virtualized infrastructures". Computer Networks, DOI: doi.org/10.1016/j.comnet.2017.04.045.

[34]J. Lyu, Y. Zeng, R. Zhang, T. Joon. “Placement Optimization of UAV-Mounted Mobile Base Stations”. IEEE Communications Letters. Vol. 21, pp. 604-607. 2016.

[35]Y. Zhou, N. Cheng, N. Lu, X. S. Shen. (2015). “Multi-UAV-Aided networks: Aerial-ground cooperative vehicular networking architecture”. IEEE Vehicular Technology Magazine, Vol. 10, pp. 36-44. 2015.

[36]V. Sharma, K. Srinivasan, H. Chao, K. Hua. W. Cheng. “Intelligent deployment of UAVs in 5G heterogeneous communication environment for improved coverage”. Journal of Network and Computer Applications. DOI. http://dx.doi.org/10.1016/j.jnca.2016.12.012.

[37]S. Koulali, E. Sabir, T. Taleb, M. Azizi. (2016). A green strategic activity scheduling for UAV networks: a sub-modular game perspective. IEEE Communications Magazine. Vol. 54, pp. 58-64. 2016.

[38]R. Magán-Carrión, R. A. Rodríguez-Gómez, J. Camacho, P. García-Teodoro. “Optimal relay placement in multi-hop wireless networks”. Ad Hoc Networks. Vol. 46, pp. 23-36. 2016.

[39]E. Natalizio, V. Loscri, E. Viterbo, "Optimal placement of wireless nodes for maximizing path lifetime". IEEE Communication Letters. Vol. 12, 2008.

[40]F. D'Andreagiovanni and A. Nardin, "Towards the fast and robust optimal design of wireless body area networks". Applied Soft Computing, Vol. 37, pp. 971-982. 2015.

[41]L.E. Agustín-Blas, S. Salcedo-Sanz, P. Vidales, G. Urueta. J.A. Portilla-Figueras. “Near optimal citywide WiFi network deployment using a hybrid grouping genetic algorithm”. Expert Systems with Applications, Vol. 38, pp. 9543-9556. 2011.

[42]T. Oda, S. Sakamoto, E. Spaho, A. Barolli, L. Barolli, F. Xhafa. “Node placement in WMNs using WMN-GA system considering uniform and normal distribution of mesh clients”. Eighth International Conference on Complex, Intelligent and Software Intensive Systems, pp. 120-127. 2014.

[43]X. Chang, T. Oda, E. Spaho, M. Ikeda, L. Barolli, F. Xhafa. “Node placement in WMNs sensing WMN-HC system and different movement methods”. Eighth International Conference on Complex, Intelligent and Software Intensive Systems, pp. 1148-1153. 2014.

[44]P. A. Consoli, Y. Mei, and L. L. Minku, X. Yao. “Dynamic selection of evolutionary operators based on online learning and fitness landscape analysis”. Soft Computing. Vol. 20, pp. 3889-3914. 2016.

[45]A. Esfahanian. “Connectivity Algorithms”. 2012. Available at http://www.cse.msu.edu/~cse835/Papers/Graph_connectivity_revised.pdf

[46]J. H. Holland. “Adaptation in Natural and Artificial Systems”. University of Michigan Press. 1975.

[47]M. Črepinšek, S. Liu, M. Mernik. “Exploration and exploitation in evolutionary algorithms: A survey”. ACM Computing Surveys (CSUR). Vol. 45, pp. 1-33. 2013.

[48]F. J. Lobo, C. F. Lima, Z. Michalewicz. “Parameter Setting in Evolutionary Algorithms”. Springer. 2007.

[49]D. G. Reina, S. L. Toral, H. Tawfit. “UAVs deployment in Disaster Scenarios based on Global and Local Search Optimization Algorithms”. 9th International Conference on the Developments on eSystems Engineering (DESE’16), pp.1-6. 2016.

[50]S. Soro, and W. B. Heinzelman. "Cluster head election techniques for coverage preservation in wireless sensor networks." Ad Hoc Networks, Vol. 7, Iss. 5, pp. 955-972, 2009.

[51]J. Yu, Y. Qi, G. Wang, X. Gu "A cluster-based routing protocol for wireless sensor networks with nonuniform node distribution." AEU-International Journal of Electronics and Communications, Vol. 66, Iss. 1, pp. 54-61, 2012.

[52]F. Fortin, F. De Rainville, M. Gardner, M. Parizeau, C. Gagne. “DEAP: Evolutionary Algorithms Made Easy”. Journal of Machine Learning Research. Vol. 13, pp. 2171—2175. 2012.

[53]A. A. Hagberg, D. A. Schult, P- J. Swart, “Exploring network structure, dynamics, and function using NetworkX”. Proceedings of the 7th Python in Science Conference (SciPy2008), pp. 11–15, 2008.

[54]https://github.com/Dany503/DeploNet

[55]A. A. Hagberg, D. A. Schult, and P. J. Swart, “Exploring network structure, dynamics, and function using NetworkX”. Proceedings of the 7th Python in Science Conference (SciPy2008), Gäel Varoquaux, Travis Vaught, and Jarrod Millman (Eds), (Pasadena, CA USA), pp. 11–15, 2008.

[56]K. Xu, M. Gerla, and S. Bae. "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks." Ad hoc networks, Vol. 1, Iss. 1, pp. 107-123, 2003.

[57]F. Klemm, Z. Ye, S. V. Krishnamurthy, S. K. Tripathi "Improving TCP performance in ad hoc networks using signal strength based link management." Ad Hoc Networks, Vol. 3, Iss. 2, pp. 175-191, 2005.

[58]N. Aziz, A. W. Mohemmed, M. Y. Alias. “A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram”. International Conference on Networking, Sensing and Control (ICNSC '09). 2009.

Authors' Information

D. G. Reina was born in Seville, Spain, in 1983. He received the B.E. degree in electronic engineering and M.S. degree in electronics and telecommunications from the University of Seville, Seville, Spain, in 2009 and 2011 respectively. He obtained the Ph.D. degree in electronic engineering in 2015 by the University of Seville, Seville. His current research interests include wireless networks such as ad hoc networks, delay tolerant networks and flying ad hoc networks.

Sergio Toral was born in Rabat, Morocco, in 1972. He received the M.S. and Ph.D. degrees in electrical and electronic engineering from the University of Seville, Spain, in 1995 and 1999, respectively. He is currently a full Professor with the Department of Electronic Engineering, US. His main research interests include ad hoc networks and their routing protocols, deployment of wireless sensor networks, real-time and distributed systems, intelligent transportation systems, and embedded operating systems.

Hissam Tawfik is a Professor of Computer Science with research expertise and interests in the areas of Biologically Inspired Computing, Health Informatics and Technology Acceptance. Hissam has a research track record of more than 100 refereed journal and conference publications and he is a visiting Professor at the University of Seville. Hissam Tawfik holds a PhD in Computer Engineering from the University of Manchester (then UMIST) and has been actively publishing since 1997. His main research areas include Neural and Evolutionary Computing, ICT for Active Ageing, Technology Acceptance and Intelligent Systems and Simulation applications. Some of the projects that Hissam is currently working on include the use of various neural network paradigms for time-series prediction, designing E-Health solutions to support active ageing and people with Dementia, and investigating cultural factors that influence the adoption of E-Health technologies. Hissam has previously worked on a number of EU funded research projects applying Virtual Reality Technology for construction and urban planning and led a British Council funded project on user-centred Health Informatics. Hissam serves on various editorial boards and review committees for international journals and conferences and is a chair and organiser of the International Conference Series on Developments in eSystems Engineering (DESE).