Available online at www.sciencedirect.com

ScienceDirect

Computer Communications 30 (2007) 685–697

computer communications

www.elsevier.com/locate/comcom

A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan MANETs

E. Alba [a,*], B. Dorronsoro [a], F. Luna [a], A.J. Nebro [a], P. Bouvry [b], L. Hogie [b]

[a] Department of Computer Science, E.T.S. Ingeniería Informática, University of Málaga, Spain
[b] Faculty of Sciences, Technology and Communication, University of Luxembourg, Luxembourg

Available online 18 September 2006

Abstract

Mobile Ad Hoc Networks (MANETs) are composed of a set of communicating devices which are able to spontaneously interconnect without any pre-existing infrastructure. In such kind of networks, broadcasting becomes an operation of capital importance for the own existence and operation of the network. Optimizing a broadcasting strategy in MANETs is a multi-objective problem targeting three goals: reaching as many devices as possible, minimizing the network utilization, and reducing the duration time of the broadcasting process. In this paper, we study the fine-tuning of broadcasting strategies by using a cellular multi-objective genetic algorithm (cMOGA) which computes a Pareto front of solutions to empower a human designer with the ability of choosing the preferred configuration for the network. We define two formulations of the problem, one with three objectives and another one with two objectives plus a constraint. For our tests, a benchmark of three realistic environments for metropolitan MANETs has been defined. Our experiments using a complex and realistic MANET simulator reveal that cMOGA is a promising approach to solve the optimum broadcasting problem.
© 2006 Elsevier B.V. All rights reserved.

Keywords: Mobile ad hoc networks; Broadcasting; Multi-objective optimization; Cellular genetic algorithm

1. Introduction

Mobile Ad Hoc Networks (MANETs) are fluctuating networks populated by a set of communicating devices called *nodes* (or *devices*) which can spontaneously interconnect each other without any pre-existing infrastructure. This means that no organization is present in such networks as it is usual in communication networks. The most popular wireless networking technologies available nowadays for building MANETs are Bluetooth and IEEE802.11 (WiFi). This implies that (a) devices communicate within a limited range, and (b) devices may move while communicating. A consequence of mobility is that the topology of such networks may change quickly and in unpredictable ways. This dynamical behavior constitutes one of the main obstacles for performing efficient communications.

In this paper, we are considering the problem of broadcasting on a particular subclass of MANETs called *Metropolitan* MANETs, which have some specific properties: the density in the network is heterogeneous and dynamic (particularly, high density regions do not remain active full time). The broadcasting strategy we are considering in this work is the so called *Delayed Flooding with Cumulative Neighborhood* protocol (DCFN) [1]. Three real world examples of such networks, a shopping mall, a metropolitan area, and a highway environment, have been taken into account so that, instead of providing a multi-purpose protocol, the originality of our proposal lies in tuning the broadcasting service for each particular network. Optimizing a broadcasting strategy implies multiple goals to be satisfied at the same time, such as maximizing the number of devices reached (coverage), minimizing the network use (bandwidth),

* Corresponding author.
E-mail addresses: eat@lcc.uma.es (E. Alba), bernabe@lcc.uma.es (B. Dorronsoro), flv@lcc.uma.es (F. Luna), antonio@lcc.uma.es (A.J. Nebro), pascal.bouvry@uni.lu (P. Bouvry), luc.hogie@uni.lu (L. Hogie).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2006.08.033

and/or minimizing the duration of the process. Thus, what we are facing is known as a multi-objective optimization problem [2,3].

The main feature of multi-objective optimization is that it is not restricted to find a unique solution as in single-objective optimization but a set of solutions known as the *Pareto optimal set*. The reason is that, taking as an example the problem we are dealing with, one solution can represent the best result concerning the number of reached devices, while another solution could be the best concerning the duration of the broadcasting process. These solutions are said to be *non-dominated*. The result provided by a multi-objective optimization algorithm is therefore a set of non-dominated solutions (the *Pareto optimal set*) which, when plotted in the objective space, is collectively known as the *Pareto* front. The mission of the decision maker is to choose the most adequate solution from the Pareto front. Here, we study the use of a cellular multi-objective evolutionary algorithm (cMOGA) for solving the multi-objective problem of tuning a particular broadcasting strategy for metropolitan MANETs.

Many popular algorithms for solving multi-objective optimization problems are evolutionary algorithms (EAs) [2,3]. However, few works use genetic algorithms based on cellular models [4–6], even though cellular genetic algorithms (cGAs) have proved in the past very high efficiency and accuracy in single-objective optimization [7–10]; the algorithm we propose, cMOGA, is a contribution to this field. Furthermore, to the best of our knowledge, our work is the first attempt to solve the broadcasting problem on MANETs using a multi-objective EA.

The rest of the paper is organized as follows. In the next section, we present a brief survey on multi-objective optimization. Section 3 describes the problem we address, the proposed benchmark, and the broadcasting strategy we intend to optimize. We detail the proposed approach based on a cellular multi-objective genetic algorithm in Section 4. Section 5 presents a set of experiments and analyzes the results. The paper ends with some conclusions and future research lines.

2. Multi-objective optimization background

In this section, we include some background on multi-objective optimization. In concrete, we define the concepts of multi-objective optimization problem, Pareto optimality, Pareto dominance, Pareto optimal set, and Pareto front.

A general multi-objective optimization problem (MOP) can be formally defined as follows (we assume the minimization of all the objectives without loss of generality):

Definition 1. (MOP) Find a vector $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]$ which satisfies the m inequality constraints $g_i(\vec{x}) \geqslant 0, i = 1, 2, \ldots, m$, the p equality constraints $h_i(\vec{x}) = 0, i = 1, 2, \ldots, p$, and minimizes the vector function $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]^T$, where $\vec{x} = [x_1, x_2, \ldots, x_n]^T$ is the vector of decision variables.

The set of all values satisfying the constraints defines the *feasible region* Ω and any point $\vec{x} \in \Omega$ is a *feasible solution*. As mentioned before, we seek for the *Pareto optima*. Its formal definition is provided next:

Definition 2 (*Pareto optimality*). A point $\vec{x}^* \in \Omega$ is Pareto Optimal if for every $\vec{x} \in \Omega$ and $I = \{1, 2, \ldots, k\}$ either $\forall_{i \in I}(f_i(\vec{x}) = f_i(\vec{x}^*))$ or there is at least one $i \in I$ such that $f_i(\vec{x}) > f_i(\vec{x}^*)$.

This definition states that $\vec{x}^*$ is Pareto optimal if no feasible vector $\vec{x}$ exists which would decrease some criterion without causing a simultaneous increment in at least one other criterion. Other important definitions associated with Pareto optimality are the following:

Definition 3 (*Pareto dominance*). A vector $\vec{u} = (u_1, \ldots, u_k)$ is said to dominate $\vec{v} = (v_1, \ldots, v_k)$ (denoted by $\vec{u} \preccurlyeq \vec{v}$) if and only if $\vec{u}$ is partially less than $\vec{v}$, i.e., $\forall i \in \{1, \ldots, k\}, u_i \leqslant v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i$.

Definition 4 (*Pareto optimal set*). For a given MOP $\vec{f}(\vec{x})$, the Pareto optimal set is defined as $\mathscr{P}^* = \{\vec{x} \in \Omega | \neg\exists \vec{x'} \in \Omega, \vec{f}(\vec{x'}) \preccurlyeq \vec{f}(\vec{x})\}$.

Definition 5 (*Pareto front*). For a given MOP $\vec{f}(\vec{x})$ and its Pareto optimal set $\mathscr{P}^*$, the Pareto front is defined as $\mathscr{PF}^* = \{\vec{f}(\vec{x}), \vec{x} \in \mathscr{P}^*\}$.

Obtaining the Pareto front of a given MOP is the main goal of multi-objective optimization. However, given that a Pareto front can contain a large number of points, a good solution must contain a limited number of them, which should be as close as possible to the Pareto optimal set, as well as they should be uniformly spread. Otherwise, the decision maker could miss important information.

3. The problem

The problem we study in this paper consists of, given an input metropolitan MANET network, determining the most adequate parameters for a broadcasting strategy. We first describe in Section 3.1 the target networks we use. Section 3.2 is devoted to the presentation of DFCN [1], the broadcasting strategy to be tuned. Finally, the MOPs we define for this work are presented in Section 3.3.

3.1. Metropolitan mobile ad hoc networks

Metropolitan mobile ad hoc networks are MANETs with some particular properties. First, they have one or more areas where the node density is higher than the average. They are called *high density areas*, and they can be statistically detected. A high density area may be, for example, a supermarket, an airport, or an office. Second, high density areas do not remain active full time, i.e., they can appear and disappear from the network (e.g., a supermarket is open, roughly, from 9 a.m. to 10 p.m., and outside this period of time, the node density within the corresponding

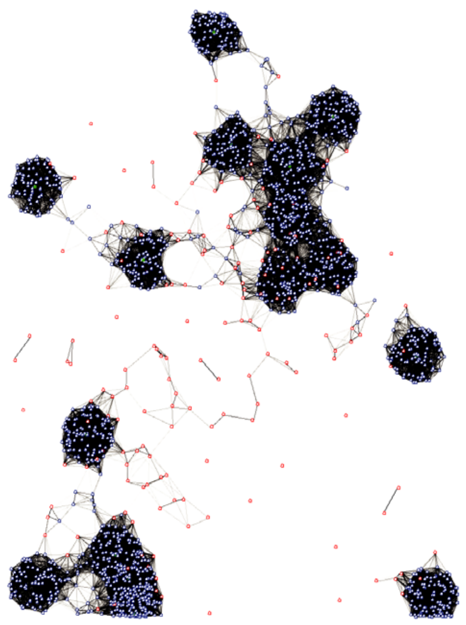


Fig. 1. An example metropolitan MANET.

area is close to zero). An example of 4 km^2, 2000 devices metropolitan network is displayed in Fig. 1.

To deal with such kind of networks, there is no solution other than resorting to software simulations. In this work, we have used *Madhoc* [11], a metropolitan MANET simulator.[1] It aims at providing a tool for simulating different level services based on distinct technologies on MANETs for different environments.

In the context of metropolitan MANETs, various topological configurations are very usually found. Some examples are networks built on-the-fly by people moving in concert places, market places, train stations, shopping centers, and city centers. All these scenarios have a number of different characteristics, such as the mobility and density of devices, the size of the area, and the presence or not of walls (which are obstacles for mobility and attenuate the signal strength), among others. Hence, three very different realistic scenarios, implemented by Madhoc, are used in this paper. These scenarios correspond to real world environments, and they aim at modelling a shopping mall, a metropolitan area, and a highway scenario:

- *Mall environment.* The mall environment is proposed for emulating MANETs in commercial shopping centers, in which stores are usually located together one each other in corridors. People go from one store to another through these corridors, occasionally stopping for looking some shopwindows. These shopping centers are usually very crowded (high density of devices), and people behave differently (in terms of mobility) when they are in or out of the stores. Additionally, a high density of shops can be found in this kind of scenario. Finally, in the mall environment both the mobility of devices and their signal propagation are restricted by the walls of the building.
- *Metropolitan environment.* The second realistic scenario we propose is the metropolitan environment. We have modelled it as an attempt for simulating the behavior of a MANET in a metropolitan area. For that, we locate a set of spots (crossroads) in the simulated surface, and link them with streets. In this case, both pedestrians and vehicles are modelled, and they are continuously moving from one crossroad to another. Like in the real world, devices must reduce their speed when they approach a crossroad. In this scenario, the obstruction of the walls in the signal strength will be stronger than in the case of the mall environment.
- *Highway environment.* We use this environment for simulating the behavior of MANETs out of cities. As an example, think on a large surface with roads, and people travelling by car. In this context, there should be a very low density of devices per square kilometer (all the devices are located on the roads), moving all of them in a fast manner. Additionally, there should not exist obstacles that attenuate the signal strength in communications.

In order to make our studies more realistic, an *observation window* has been included in the simulations, so that only the devices located into this window are taken into account for measurements. This makes possible the simulation of nodes exiting and joining the network by entering or leaving the observation window, respectively. Therefore, we are allowing the existence of a changing number of devices in the network, as it is the case in real MANETs. In all our tests in this work, this observation window covers the 70% of the whole area. As an example, in Fig. 2 we can see a MANET simulating a mall environment (left), and the observation window we study (right); supposing the 70% of the whole network. The circles represent the shops, while the points stand for the devices (those outside the observation window are grey colored, meaning that they are not considered for measurements).

3.2. *Delayed flooding with cumulative neighborhood*

Williams and Camp [12] and, more recently, Stojmenovic and Wu [13] proposed two of the most frequently referenced analysis of broadcasting protocols. In their proposal, Stojmenovic and Wu [13] state that protocols can be classified according to their algorithmic nature – determinism (no use of randomness), reliability (guarantee of full coverage) – or the information required by their execution (network information, ‘‘hello’’ messages content, broadcast messages content). Similarly, Wu and Lou [14] categorized protocols as *centralized* [15] and *localized* ones. Centralized protocols require a global or quasi-global knowledge of the network. They are hence not scalable. Localized protocols

[1] The MANET simulator is freely available at http://www-lih.univ-lehavre.fr/~hogie/madhoc/

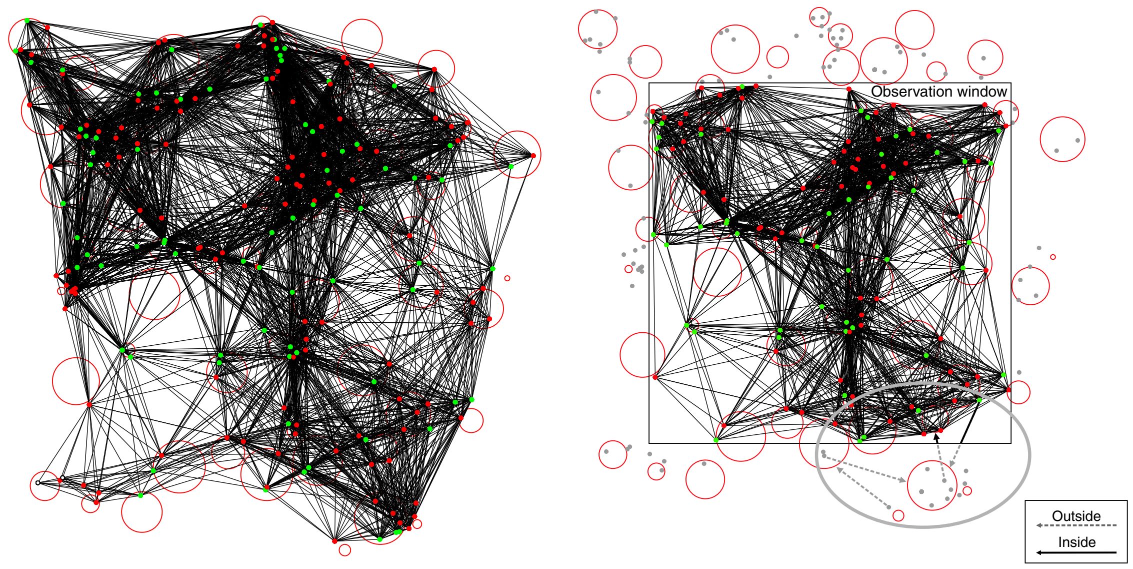


Fig. 2. The effects of introducing an observation window on the studied MANET.

are those which require some knowledge of the network at only 1- or 2-hops.

Using the classifications presented hereinbefore, DFCN [1] is a deterministic algorithm. It does not consist of a new approach for calculating dominating sets. Instead, it is a fully localized protocol which defines heuristics based on 1-hop information. This permits DFCN to achieve great scalability. The "hello" messages interchanged by the nodes do not carry any additional information. Only broadcast messages must embed the list of node's neighbors.

For being able to run the DFCN protocol, the following assumptions must be met:

- Like many other neighbor-knowledge-based broadcasting protocols (FWSP, SBA, etc.) [16,17], DFCN requires the knowledge of 1-hop neighborhood. This is obtained by the use of "hello" packets at a lower network layer. The set of neighbors of the device s is named $N(s)$.
- Each message m carries – embedded in its header – the set of IDs of the 1-hop neighbors of its most recent sender.
- Each device maintains local information about all messages received. Each instance of this local information consists of the following items:
  - the ID of the message received;
  - the set of IDs of the devices that are known to have received the message;
  - the decision of whether the message should be forwarded or not.
- DFCN requires the use of a random delay before possibly re-emitting a broadcast message m. This delay, called Random Assessment Delay (RAD), is intended to prevent collisions. More precisely, when a device s emits a message m, all the devices in $N(s)$ receive it at the same time. It is then likely that all of them forward m simultaneously, and this simultaneity entails network collisions. The RAD aims at randomly delaying the retransmission of m. As every device in $N(s)$ waits for the expiration of a different RAD before forwarding m, the risk of collisions is hugely reduced.

DFCN is an event driven algorithm which can be divided into three main parts: the two first ones deal with the handling of outcoming events, which are (1) new message reception and (2) detection of a new neighbor. The third part (3) consists of the decision making for emission as a follow-up of one of the two previous events. The behavior resulting from a message reception is referred to as *reactive* behavior; when a new neighbor is discovered, the behavior is referred as *proactive* behavior.

Let s_1 and s_2 be two devices in the neighborhood of one another. When s_1 sends a packet to s_2, it attaches to the packet the set $N(s_1)$. At reception, s_2 hence knows that each device in $N(s_1)$ has received the packet. The set of devices which has *potentially* not yet received the packet is then $N(s_2) - N(s_1)$. If s_2 re-emits the packet, the *effective* number of devices newly reached is maximized by the heuristic function: $h(s_2,s_1) = |N(s_2) - N(s_1)|$.

In order to minimize the network use caused by a possible packet re-emission, a message is forwarded only if the number of newly reached devices is greater than a given threshold. This threshold is a function of the number of devices in the neighborhood (the local network density) of the recipient device s_2. It is written "threshold($|N(s)|$)". The decision made by s_2 to re-emit the packet received from s_1 is defined by the boolean function:

$$B(s_2, s_1) = \begin{cases} \text{true} & h(s_2, s_1) \geqslant \text{threshold}(|N(s_2)|) \\ \text{false} & \text{otherwise} \end{cases}$$

If the threshold is exceeded, the recipient device s_2 becomes an emitter in turn. The message is effectively sent when the random delay (defined by the RAD) expires. The threshold function, which allows DFCN to facilitate the message re-broadcasting when the connectivity is low, depends on the size of the neighborhood n, as given by:

$$\text{threshold}(n) = \begin{cases} 1 & n \leqslant \text{safeDensity} \\ \text{minGain} * n & \text{otherwise}, \end{cases}$$

where *safeDensity* is the maximum safe density below DFCN always rebroadcasts and *minGain* is a parameter of DFCN used for computing the minimum threshold for forwarding a message, i.e., the ratio between the number of neighbors which have not received the message and the total number of neighbors.

Each time a device s discovers a new neighbor, the RAD for all messages is set to zero and, therefore, the messages are immediately candidate to emission. If $N(s)$ is greater than a given threshold, which we have called *proD*, this behavior is disabled, so no action is undertaken on new neighbor discovery.

3.3. MOPs definition

From the description of DFCN in the previous section, the following parameters are to be tuned:

- *minGain* is the minimum gain for rebroadcasting. This is the most important parameter for tuning DFCN, since minimizing the bandwidth should be highly dependent on the network density. It ranges from 0.0 to 1.0.
- *lowerBoundRAD,upperBoundRAD* define the RAD value (random delay for rebroadcasting in milliseconds). The two parameters take values in the interval [0.0,10.0] ms.
- *proD* is the maximal density ($proD \in [0,100]$) for which it is still needed using proactive behavior (i.e., reacting on new neighbors) for complementing the reactive behavior.
- *safeDensity* defines a maximum safe density of the threshold which ranges from 0 to 100 devices.

These five parameters, i.e., five decision variables which correspond to a DFCN configuration, characterize the search space. We have set wide enough intervals for the values of these parameters in order to include all the reasonable possibilities we can find in a real scenario. The objectives to optimize are: minimizing the duration of the broadcasting process, maximizing the network coverage, and minimizing the number of transmissions. Thus, we have defined a triple objective MOP, which is called DFCNT (which stands for DFCN Tuning). As we stated before, this problem is defined by a given target network in which the DFCN broadcasting strategy is used. Since three different real world metropolitan MANETs have been considered, three instances of DFCNT are to be solved: *DFCNT.Mall*, *DFCNT.Metropolitan*, and *DFCNT.Highway*.

Additionally, it is also interesting to consider the network coverage as a constraint instead of a goal for practical purposes. This way, we can define that the coverage must be, for example, over 90%, and then to find the best tradeoff between bandwidth and duration. The resulting problem, which is named cDFCNT (*constrained* DFCNT), is a bi-objective MOP with one constraint. Analogously, three different target networks here bring three different instances of cDFCNT: *cDFCNT.Mall*, *cDFCNT.Metropolitan*, and *DFCNT.Highway*.

4. The algorithm

The application of evolutionary algorithms (EAs) to complex optimization problems has been very intense during the last decade [18]. These algorithms work on a set (*population*) of potential solutions (*individuals*) by applying some stochastic operators to them in order to search for the best solutions. Most EAs use a single population (panmixia) of individuals and apply operators to them as a whole (see Fig. 3a). In contrast, there exists also some tradition in using structured EAs, where the population is decentralized somehow. Among the many types of structured EAs, *distributed* and *cellular* algorithms are two popular optimization variants [19,20] (see Figs. 3b and c). In many cases

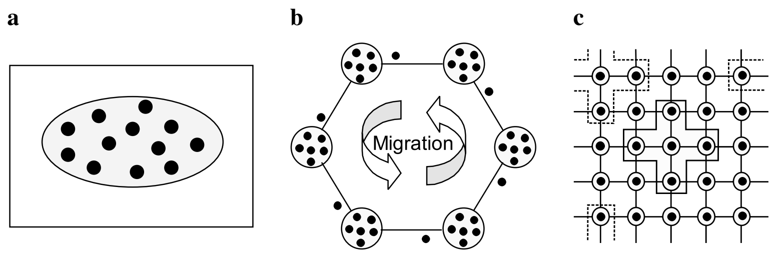


Fig. 3. Panmictic (a), distributed (b), and cellular (c) EAs.

[21], these decentralized algorithms provide a better sampling of the search space resulting in improved numerical behavior with respect to an equivalent algorithm in panmixia.

Algorithm 1 Pseudocode for a Canonical cGA

```
 1: proc Steps_p(cga) //Algorithm parameters in 'cga'
 2: while not Termination_Condition() do
 3:       for individual ← 1 to cga.popSize do
 4:          n_list ← Get_Neighborhood(cga,position(individual));
 5:          parents ← Selection(n_list);
 6:          offspring ← Recombination(cga.Pc,parents);
 7:          offspring ← Mutation(cga.Pm,offspring);
 8:       Evaluate_Fitness(offspring);
 9:       Insert(position(individual),offspring,cga,aux_pop);
10:    end for
11:    cga.pop ← aux_pop;
12: end while
13: end_proc Steps_Up;
```

In this work, we focus on Cellular Genetic Algorithms (cGAs). In cGAs, the concept of (small) *neighborhood* is intensively used; this means that an individual may only interact with its nearby neighbors in the breeding loop [22]. The overlapped small neighborhoods of cGAs help in exploring the search space because the induced slow diffusion of solutions through the population provides a kind of exploration (diversification), while exploitation (intensification) takes place inside each neighborhood by genetic operations. These cGAs were initially designed for working in massively parallel machines, although the model itself has been adopted recently also for mono-processor machines, with no relation to parallelism at all. Besides, the neighborhood is defined among tentative solutions in the algorithm, with no relation to the geographical neighborhood definition in the problem space.

4.1. Cellular genetic algorithms

In this section, a detailed description of a canonical cGA is presented (we include a pseudo-code in Algorithm 1). In this basic cGA, the population is usually structured in a regular grid of d dimensions ($d = 1$, 2, 3), and a neighborhood is defined on it. The algorithm iteratively considers as current each individual in the grid (line 3). An individual may only interact with individuals belonging to its neighborhood (line 4), so the parents are chosen among its neighbors (line 5) with a given criterion. Crossover and mutation genetic operators are applied to the individuals in lines 6 and 7, with probabilities P_c and P_m, respectively. After applying these operators, the algorithm computes the fitness value of the new offspring individual (or individuals) (line 8), and inserts it (or one of them) into the equivalent place of the current individual in the new (auxiliary) population (line 9) following a given replacement policy.

Algorithm 2 Pseudocode of cMOGA

```
 1: proc Steps_Up(cmoga) //Algorithm parameters in 'cmoga'
 2: Pareto_front = Create_Front() //Creates an empty Pareto front
 3: while ! TerminationCondition() do
 4:    for individual ← 1 to cmoga.popSize do
 5:       n_list ← Get_Neighborhood(cmoga,position(individual));
 6:       parents ← Selection(n_list);
 7:       offspring ← Recombination(cmoga.Pc,parents);
 8:       offspring ← Mutation(cmoga.Pm,offspring);
 9:       Evaluate_Fitness(offspring);
10:       if Non-Dominated(offspring, position(individual)) then
11:          Insert(position(individual),offspring,cmoga,aux_pop);
12:          Insert_Pareto_Front(individual);
13:       end if
14:    end for
15:    cmoga.pop ← aux_pop;
16: end while
17: end_proc Steps_Up;
```

After applying this reproductive cycle to all the individuals in the population, the newly auxiliary population is assumed to be the new population for the next generation (line 11). This loop is repeated until a termination condition is met (line 2). The most usual termination conditions are to reach the optimal value, to perform a maximum number of fitness function evaluations, or a combination of they two.

4.2. A multi-objective cGA: cMOGA

In this section, we present a multi-objective algorithm based on a cGA model. Although other cellular-like genetic approaches exist in the literature, to the best of our knowledge, none of them follows the canonical cGA model (Algorithm 1). In [4], a multi-objective evolution strategy following a predator-prey model is presented. This is a model similar to the cGA, because solutions (preys) are placed on the vertices of an undirected connected graph, thus defining neighborhoods, where they are 'caught' by predators. Murata and Gen presented in [5] a cellular algorithm in which, for an n-objective MOP, the population is structured in an n-dimensional weight space, and the location of individuals (called cells) depends on their weight vector. Thus, the information given by the weight vector assigned to individuals is used for guiding the search. Finally, in [6] a metapopulation evolutionary algorithm (called MEA) is presented. This algorithm is a cellular model with the peculiarity that disasters can occasionally happen in the population, thus dying all the individuals located in the disaster area (extinction). Additionally, these empty areas can also be occupied by individuals (colonization). Thus, this model allows a flexible population size, combining the ideas of cellular and spatially distributed populations.

A pseudo-code of our cMOGA is given in Algorithm 2. We can observe that Algorithms 1 and 2 are very similar. One of the main differences between the two algorithms is the existence of a *Pareto front* (Definition 5) in the multi-objective case. The Pareto front is just an additional population composed of the non-dominated solutions found so far, which has a maximum size. In order to manage the insertion of solutions in the Pareto front with the goal of obtaining a diverse set, a crowding procedure has been used.

cMOGA starts by creating an empty Pareto front (line 2 in Algorithm 2). Individuals are arranged in a two-dimensional toroidal grid, and the genetic operators are successively applied to them (lines 7 and 8) until the termination condition is met (line 3). Hence, for each individual, the algorithm consists of selecting two parents from its neighborhood, recombining them in order to obtain an offspring, mutating it, evaluating the resulting individual, and inserting it if it is not dominated by the current individual in both the auxiliary population and the Pareto front (following a crowding procedure) – lines 10 to 13. Finally, after each generation, the old population is replaced by the auxiliary one.

4.3. Dealing with constraints

To deal with constrained MOPs, cMOGA uses a simple approach also encountered in other multi-objective evolutionary algorithms, like NSGA-II [23] and microGA [24]. Whenever two individuals are compared, their constraints are checked. If both are feasible, a Pareto dominance test (Definition 3) is directly applied. If one is feasible and the other is infeasible, the former dominates. In other case, if the two individuals are infeasible, then the one with the lowest amount of constraint violation dominates the other.

5. Experiments

In this section, we first describe the parameterization used by cMOGA. Next, the configurations of the network simulator for the three defined environments are described. Finally, we analyze the obtained results for both *DFCNT* and *cDFCNT* and compare them.

cMOGA has been implemented in Java and tested on a PC with a 2.8 GHz Pentium IV processor with 512 MB of RAM memory, and running SuSE Linux 8.1 (kernel 2.4.19-4 GB). The Java version used is 1.5.0_05.

5.1. cMOGA Parameterization

In Table 1 we show the parameters used by cMOGA. A square toroidal grid of 100 individuals has been chosen for structuring the population. The neighborhood used is composed of 5 individuals: the considered one plus those located at its North, East, West, and South (called NEWS, Linear5 or Von Neumann neighborhood). Due to the stochastic nature of the Madhoc simulator, five simulations per function evaluation are performed and the fitness values of the functions are computed as the average of the values obtained in each of these simulations. This important detail has a considerable influence in the running time to solve the problem, and explains why reporting 30 independent runs of the algorithm in our tests represents a high effort in studying this problem, since we want to ensure statistical confidence on the results.

Table 1
Parameterization used in cMOGA

Population size	100 Individuals (10×10)
Stopping condition	25,000 Function evaluations
Neighborhood	NEWS
Selection of parents	Current individual + Binary tournament
Recombination	Simulated binary, $p_c = 1.0$
Mutation	Polynomial, $p_m = 1.0/L$ (L = individual length)
Replacement	Rep_if_Non_Dominated
Archive size	100 Individuals
Crowding procedure	Adaptive grid

We use the simulated binary recombination operator (SBX) [25] with probability $P_c = 1.0$. As its name suggests, SBX simulates the working principle of the single-point crossover on binary genotypes. The mutation operator used is the so called *polynomial* [25], and it is applied to every allele of the individuals with probability $P_m = 1.0/L$. The resulting offspring replaces the individual at the current position if the latter does not dominate the former. For inserting the individuals in the Pareto front, an *adaptive grid algorithm* [26] is used. It consists of dividing up the objective space in hypercubes with the goal of balancing the density of non-dominated solutions in the hypercubes. Then, when inserting a non-dominated solution in the Pareto front, its grid location in the solution space is determined. If the Pareto front is already full and the grid location of the new solution does not match with the most crowded hypercube, a solution belonging to that most crowded hypercube is removed before inserting the new one.

5.2. Madhoc configuration

As we stated in Section 3.1, we have defined three different environments for MANETs modelling three possible scenarios that can be found in real world. Their main features are explained in Sections 5.2.1 to 5.2.3, and they are summarized in Table 2. We show in Fig. 4 example MANETs for each of the studied scenarios. These example networks were obtained by using the graphical interface of Madhoc with exactly the same parameterization suggested in our proposed benchmark. We consider that the broadcasting is completed when either the coverage reaches 100% or it does not vary in a reasonable period of time (set to 1.5 s after some preliminary experimentation). This point is important since an improper termination condition will lead us to bad results or slow simulations.

Table 2
Main features of the proposed environments

		Mall	Metropolitan	Highway
Surface (m^2)		40,000	160,000	1,000,000
Density of spots		800 (shops/km^2)	50 (crossroads/km^2)	3 (joints/km^2)
	Speed out of spots (m/s)	0.3–1	1–25	30–50
Devices	Speed in spots (m/s)	0.3–0.8	0.3–10	20–30
	Density (dev./km^2)	2000	500	50
Wall obstruction (%)		70	90	0

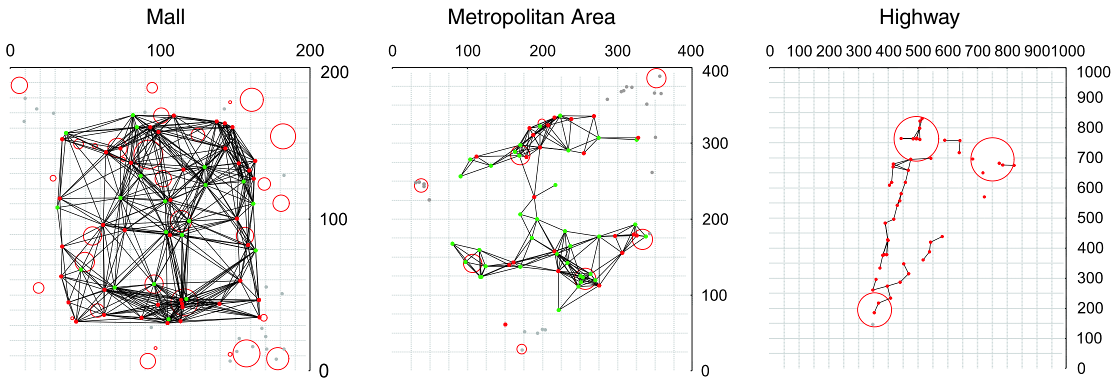


Fig. 4. The three studied scenarios for MANETs.

5.2.1. *The mall environment*

In this section, we proceed to explain the parameterization we used for modelling the mall environment. In malls, the density of both shops (spots) and devices is usually very high. Additionally, there exist walls which attenuate the signal and limit the movements of devices, and these movements are usually very slow, since we are modelling people walking. We have defined for this work a shopping center of 200×200 square meters of surface with densities of 800 stores and 2000 devices per square kilometer. Stores are circles of radius between 1 and 10 m, and the obstruction of the walls is computed with a penalty of the 70% in the signal strength. Finally, devices travel with a speed ranging between 0.3 and 1 m/s in the corridors and between 0.3 and 0.8 m/s when they are inside the stores.

Regarding the mall environment, it is worth noting that the resulting connection graph is quite dense (see Fig. 4). The reason for that is that the coverage of mobile devices ranges between 40 and 80 m (randomly selected value), and the simulation area is small. Hence, the problems *DFCNT.Mall* and *cDFCNT.Mall* are more difficult to solve due to the broadcast storm problem [27]. This problem consists of severe network congestions provoked by packet re-emissions due to frequent packet collisions.

5.2.2. *The metropolitan environment*

In this second environment, we study the behavior of DFCN in a metropolitan MANET. For modelling this environment we set a surface of 400×400 square meters, with a density of 50 spots (standing for crossroads) per square kilometer having a circle surface of radius between 3 and 15 m. The wall obstruction is in this case higher than for the mall environment (up to 90%), and the density of devices is 500 elements per square kilometer. When setting the speed of devices, the cases when people move on foot or by car must be taken into account, so its value ranges between 0.3 and 10 m/s when they are in a crossroad, and between 1 and 25 m/s in other case (streets).

In Fig. 4, it can be seen that the resulting network in a metropolitan area is not as dense as that of the mall environment. Generally speaking, this kind of network is composed of a few number of subnetworks, which are usually connected one each other by only a few links, typically one or two, or even zero (unconnected subnetworks). Additionally, some devices could not be part of any subnetwork (isolated nodes). The topology of this network can change in a fast manner, since some of the devices are travelling in vehicles at high speeds. All these features difficult the broadcasting task, and that makes interesting the study of this kind of networks for us.

5.2.3. *The highway environment*

As we previously commented in Section 3.1, this environment is composed of a few number of devices, travelling at high speeds. Thus, we used the *human environment* scenario provided by Madhoc for modelling the network with the peculiarity of setting the wall obstruction to 0%. The simulated surface was set to 1000×1000 square meters with a density of only 50 devices per square kilometer. These devices travel at random speeds between 30 and 50 m/s. We define the roads as the straight lines connecting two spots, and we establish a density of only 3 spots (highway entrances and/or exits) for modelling the scenario. The speed of devices in the spots must be reduced to the range between 20 and 30 m/s. The size of each spot (length of the entrance/exit) is set to a random value between 50 and 200 m (spots radius ∈[25,100] m).

It can be seen in Fig. 4 how the resulting network using this parameterization is composed of a set of multiple (usually disconnected) sub-networks involving a small number of devices (even only one). The existence of these small and unconnected networks supposes a hard obstacle for the task of the broadcasting protocol. Additionally, the topology of the network changes very fast due to the high speed on the devices movement. Hence, as a consequence of these high speeds, new subnetworks are continuously being made and disappearing, what hinders the broadcasting process even more.

5.3. DFCNT results

We now turn to present and analyze the obtained results for the *DFCNT* problem (Section 3.3) with the three environments. Let us remind that this problem is composed of five decision variables and three objective functions. All the values presented are the average over 30 independent runs of cMOGA.

In Table 3 we show the mean and the standard deviation of the execution time (in hours) and the number of non-dominated solutions found by cMOGA for the three *DFCNT* instances: *DFCNT.Mall*, *DFCNT.Metropolitan* and *DFCNT.Highway*. As it can be seen, the execution time of a single run is very long, since it ranges from 1.98 days for the highway scenario, up to 4.51 days in the case of *DFCNT.Metropolitan*. The reason is the high cost of computing the fitness function, since we launch five simulations per evaluation, and each run of the simulator requires between 1 and 4 s. Regarding the number of solutions found, the obtained results are highly satisfactory in the three *DFCNT* instances, since the number of different solutions found is 97.77, 93.40, and 52.27 on average (the maximum is 100) for *DFCNT.Mall*, *DFCNT.Metropolitan* and *DFCNT.Highway*, respectively. Thus, we allow the decision maker to choose from a wide range of possibilities. Notice that the number of solutions composing the Pareto front decreases when decreasing the device density of the network (i.e., increasing the surface and decreasing the number of devices). This is because we stick to just one single criterion for considering that the broadcasting process is done in three very different environments. As a future work, we plan to customize this criterion for each environment, what hopefully will lead us to obtain still better results.

Table 3
Results of cMOGA for the three *DFCNT* instances

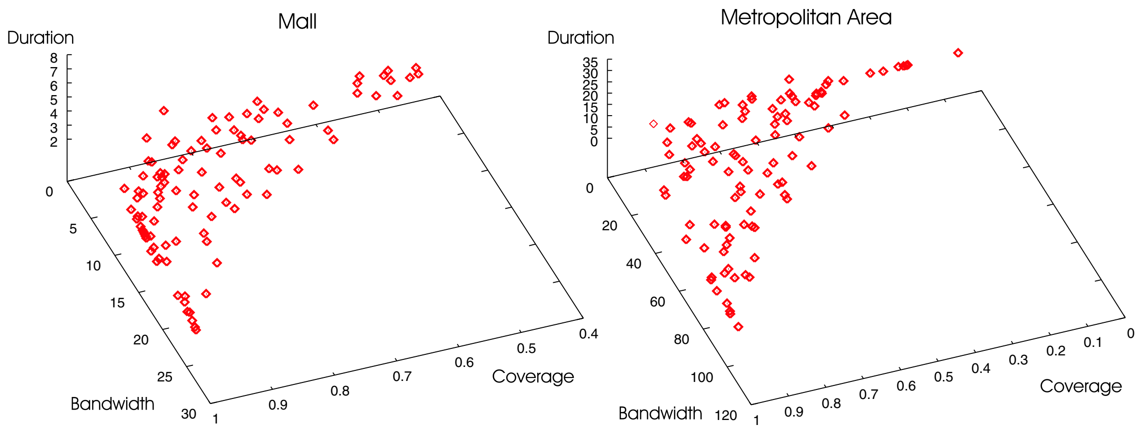
Environmment	Time (h)	Number of Pareto optima
DFCNT.Mall	66.12 ± 7.94	97.77 ± 3.20
DFCNT.Metropolitan	108.21 ± 8.41	93.40 ± 18.02
DFCNT.Highway	47.57 ± 0.42	52.27 ± 10.99

As an example of the diverse and wide set of solutions reported by the multi-objective optimizer, we plot in Fig. 5 an example Pareto front obtained with cMOGA for the three studied environments. Best solutions are those implying (i) high coverage, (ii) low bandwidth, and (iii) a short duration of the broadcasting process, (i) and (ii) being the most important parameters. In fact, Pareto optima in these fronts reaching a coverage over 95% need on average 3.77 seconds and 17.25 messages for mall, and 13.78 s and 75.71 messages (i.e., intense bandwidth usage) in the case of the metropolitan area. In contrast, in the case of the highway environment only 5 solutions of the Pareto front have a coverage over 95% (38 for *DFCNT.Mall* and 16 for *DFCNT.Metropolitan*), achieving an average of 74.88 messages sent in 13.37 s, which are values similar to those of the metropolitan area. Finally, notice that if coverage were not a hard constraint in our network, we could use very cheap solutions in terms of time and bandwidth for the two environments.

Comparing the three graphics, it is possible to observe that the broadcasting is more efficient in the mall environment than in the other two cases, since it takes less than 8 s (duration), transmitting always less than 23 messages (bandwidth), and reaching more than 40% of the devices (coverage). However, in the metropolitan and highway environments the broadcasting process is usually longer, with a larger number of transmitted messages, and the coverage is in some cases less than 10%. Finally, although the front obtained for the highway environment has similar bounding values as those observed in the case of the metropolitan area, the diversity is much lower in the highway scenario.

The difference on the coverage of solutions found in the three environments is a common sense result, since the probability of having isolated subnetworks (composed of one or more devices) grows when increasing the simulation surface and decreasing the number of devices. Hence, the difference in the quality of the solutions is a consequence of the different topology of the networks, since the high connectivity of the devices in the mall environment allows one message to reach many more devices than in the case of the other two studied environments. Conversely, this high connectivity increases the risk of a broadcast storm, making *DFCNT.Mall* very difficult to solve. From our results we can conclude that cMOGA has dealt with the problem successfully.

The Pareto fronts on Fig. 5 fulfill the design objectives of the DFCN protocol: most of the plots (in the center of the clouds) provide sets of parameters which make DFCN achieving a coverage rate close to 100%, keeping the network throughput very low. What makes the DFCNT problem particularly interesting from an applicative point of view is that it permits the decision maker to discard this default behavior by setting a *degree of coverage* for the broadcasting application. Indeed not all applications



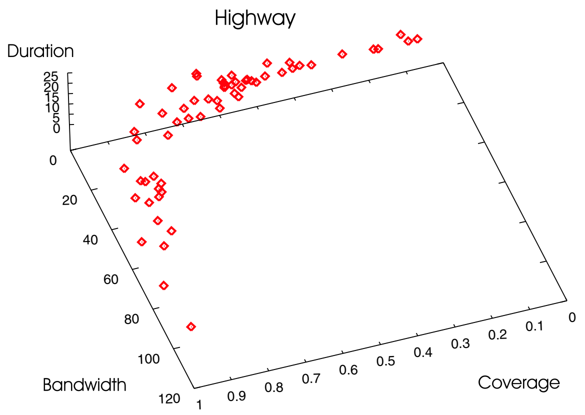


Fig. 5. Pareto fronts for the three environments and the *DFCNT* problem.

require the maximization of the coverage rate. For example, *local advertising* – which consists in spreading advertisement messages to devices a few hops away from the source – needs the broadcasting process to cease after a while. Sometimes high coverage is even to be avoided. For example, trying to achieve a high coverage on metropolitan MANETs (which may realistically be made of thousands devices) is harmful, since it is likely to lead to severe network congestions.

5.4. *cDFCNT* results

Now, we analyze the results of the *cDFCNT* problem, where the coverage has become a constraint: at least, 90% of the devices must receive the broadcasting message. This way, cMOGA has to find solutions with a tradeoff between bandwidth and duration of the broadcasting process. Just like for *DFCNT*, Table 4 presents the average time and the number of Pareto optima that cMOGA is able to find over 30 independent runs when solving *cDFCNT* for the three instances: *cDFCNT.Mall*, *cDFCNT.Metropolitan*, and *cDFCNT.Highway*.

Regarding the execution time, *cDFCNT* can be solved a bit faster than its unconstrained counterpart for the mall environment (56.53 against 66.12 h), although the difference is smaller (about 2%) in the cases of the metropolitan area (106.15 against 108.21 h) and the highway environment (46.99 versus 47.57 h). Two reasons can explain this behavior. First, dominance tests are less costly due to the lower number of objectives, thus reducing the time needed to check whether a new solution is dominated or not by the current front. Second, since a lower number of points is found for the constrained MOP, this previous test is still less costly, because of the lower number of comparisons. As it was stated before, if we analyze the non-dominated solutions found, results show that the number of points has been drastically reduced in the three studied environments with respect to the unconstrained instances. This indicates that the coverage constraint made the problem very difficult to solve (specially for the metropolitan and highway environments), maybe excessively hard if the designer is satisfied with the *DFCNT* scenario.

Table 4
Results of cMOGA for the three *cDFCNT* instances

Environment	Time (h)	Number of Pareto optima
Mall	56.53 ± 10.56	13.47 ± 2.70
Metropolitan	106.15 ± 9.11	5.57 ± 1.98
Highway	46.99 ± 4.32	3.40 ± 1.76

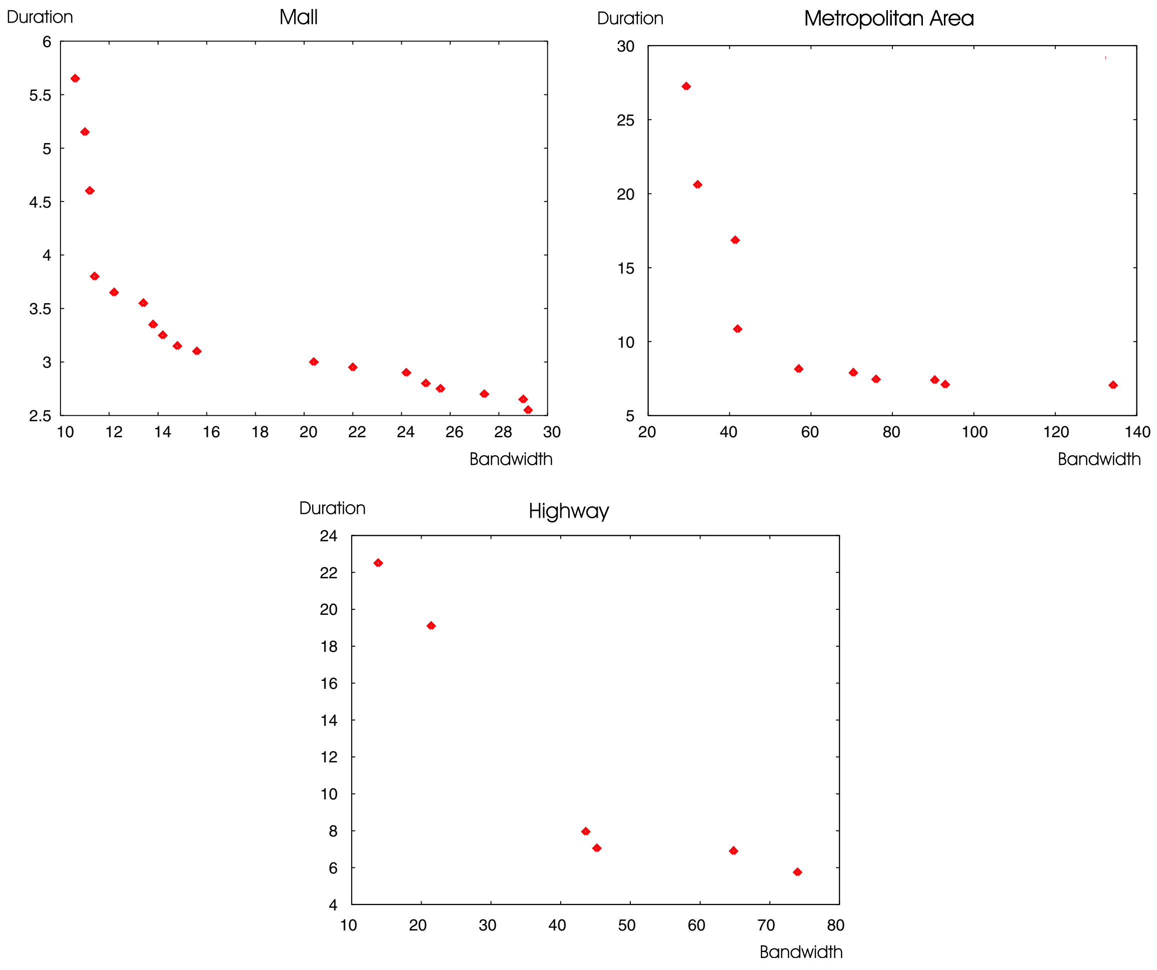


Fig. 6. Pareto fronts for the three environments and the *cDFCNT* problem.

Fig. 6 depicts three typical Pareto fronts corresponding to the three proposed instances of *cDFCNT*. The relationship between the two objective functions is clear in the three cases: if a message has to be rapidly broadcasted, it involves using a high bandwidth. Otherwise, cheap solutions in terms of bandwidth could be obtained by considering long duration times. Once more, one can see that the solutions for the metropolitan and highway environments are more expensive than in the case of the mall environment (in terms of time and bandwidth used). The low number of points found for the metropolitan and highway environments with respect to the mall environment can be explained by the lower final coverage found in the solutions due to the network topology, as we previously commented in the case of *DFCNT* problem.

As suggested before, all broadcasting protocols follow the next rule: the more opportunistic they are the faster they proceed (by not considering the impact of packet collisions), but the higher bandwidth they use. DFCN has been designed with this in mind: its behavior – when used with appropriate parameters – is an exception to this rule. However, we have a different objective in this study, since we are seeking for Pareto optimal set of parameters. Consequently, a diverse subset of the behavior exhibited by all broadcasting protocols is reported on the Pareto fronts. In these Pareto fronts we can see that achieving very short duration times entails high bandwidth and that very low bandwidth is only achievable by using slow forwarding policies. Aside to this asymptotic behaviors, the Pareto fronts also show that DFCN can be tuned in such a way that it permits to obtain a reasonably short duration of the broadcasting process while keeping the network throughput (e.g., the number of packet emission) low. Since good coverage is guaranteed, these settings are appropriate to most broadcasting applications.

5.5. Comparing DFCNT and cDFCNT

In this section, we compare the complexity of the proposed unconstrained problems versus their constrained versions. For that, a summary of the results reported in Tables 3 and 4 is given in Table 5. As we stated before, two facts can be deduced from these results: first, solving *DFCNT* instances are more costly than *cDFCNT* in terms of the computational effort (the difference becomes more important as the number of devices grows) and, second, cMOGA is able to find a higher number of Pareto optima for *DFCNT* than for *cDFCNT*.

Table 5
Comparison of the results for *DFCNT* and *cDFCNT*

		DFCNT	*cDFCNT*
Mall	Time (h)	66.12 ± 7.94	56.53 ± 10.56
	Pareto optima	97.77 ± 3.20	93.40 ± 18.02
Metropolitan	Time (h)	108.21 ± 8.41	106.15 ± 9.11
	Pareto optima	93.40 ± 18.02	5.57 ± 1.98
Highway	Time (h)	47.57 ± 0.42	46.99 ± 4.32
	Pareto optima	52.27 ± 10.99	3.40 ± 1.76

In order to provide confidence from the statistical point of view, we have performed several statistical tests (*t*-tests) at the 95% significance level. These *t*-tests have been performed after ensuring that data follow a normal distribution (by using the Kolmogorov–Smirnov test). These tests report statistically significant differences for all the results in Table 5, except when comparing the Metropolitan and Highway instances in terms of run time. Thus, if we pay attention to the average number of solutions found in the Pareto fronts, cMOGA finds more difficulties for solving *cDFCNT* than in the case of *DFCNT*, with statistical confidence in all the studied cases. Hence, we can conclude that *cDFCNT* is a more complex problem since, although the unconstrained problems (*DFCNT*) require higher computational effort (longer execution times) than the constrained ones, we only found statistically significant differences in the case of the mall environment.

6. Conclusions and further work

This paper presents a first approximation to the problem of optimally tuning DFCN, a broadcasting protocol for MANETs, using cMOGA, a new cellular multi-objective genetic algorithm. cMOGA has been used to solve two formulations of the problem; the first one, named *DFCNT*, is defined as a three-objective MOP, with the goals of minimizing the duration of the broadcasting process, maximizing network coverage, and minimizing the network usage; the second one, *cDFCNT*, is a two-objective MOP, considering the coverage objective as a constraint.

Three different realistic scenarios were used. We are then solving three different instances of each MOP. They correspond to a shopping center (*DFCNT.Mall* and *cDFCNT.Mall*), the streets in a city (*DFCNT.Metropolitan* and *cDFCNT.Metropolitan*), and a wide non-metropolitan area wherein several roads exist (*DFCNT.Highway* and *cDFCNT.Highway*). Our experiments reveal that solving *DFCNT* instances provides a populated Pareto front composed of more than 50 points for *DFCNT.Highway*, and more than 95 points (more than 95 different DFCN configurations) in the case of the other two environments, all this at the expense of a considerable amount of time. However, for a network designer, this time can be affordable depending on the target of the study. Conversely, if time is a critical issue, *cDFCNT* represents an interesting alternative that is a bit faster, ensuring coverage losses below 10%. The solved problem however admits a very reduced Pareto front for the three instances (i.e., a smaller number of design options), indicating that the coverage constraint makes the problem very hard.

As a future work, we plan to perform a more in-depth analysis on the cMOGA search model by itself. We also intend to parallelize cMOGA for reducing the execution times down to affordable values. Hence, we will hopefully be able to enlarge the simulation area to still larger mall or metropolitan networks. Finally, another interesting and immediate next step consists in the design and study of new different environments simulating other real world scenarios.

Acknowledgements

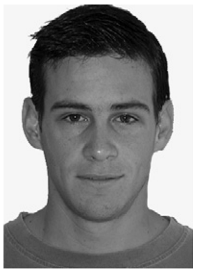
E. Alba, B. Dorronsoro, F. Luna, and A.J. Nebro acknowledge that this work has been partially funded by the Spanish MEC and FEDER under contract TIN2005-08818-C04-01 (the OPLINK project).

References

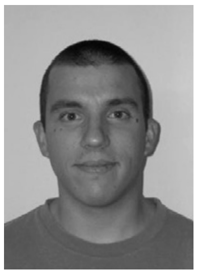
[1] L. Hogie, M. Seredynski, F. Guinand, P. Bouvry, A bandwidth-efficient broadcasting protocol for mobile multi-hop ad hoc networks, in: ICN'06, 5th International Conference on Networking (to appear), IEEE, 2006.
[2] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, London, 2001.
[3] C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, 2002.
[4] M. Laumanns, G. Rudolph, H.P. Schwefel, A spatial predator-prey approach to multi-objective optimization: a preliminary study, in: PPSN V, 1998, pp. 241–249.
[5] T. Murata, M. Gen, Cellular genetic algorithm for multi-objective optimization, in: Proceedings of the 4th Asian Fuzzy System Symposium, 2002, pp. 538–542.
[6] M. Kirley, MEA: A metapopulation evolutionary algorithm for multi-objective optimisation problems, in: Proceedings of the 2001 Congress on Evolutionary Computation CEC2001, IEEE Press, 2001, pp. 949–956.
[7] E. Alba, B. Dorronsoro, The exploration/exploitation tradeoff in dynamic cellular evolutionary algorithms, IEEE Trans. Evol. Comput. 9 (2) (2005) 126–142.
[8] E. Alba, B. Dorronsoro, M. Giacobini, M. Tomassini, Handbook of Bioinspired Algorithms and Applications, Chapman & Hall/CRC, 2006, Ch. 7: Decentralized Cellular Evolutionary Algorithms, pp. 103–120.
[9] E. Alba, M. Giacobini, M. Tomassini, Comparing synchronous and asynchronous cellular genetic algorithms, in: PPSN VII, LNCS 2439, 2002, pp. 601–610.
[10] M. Giacobini, E. Alba, M. Tomassini, Selection intensity in asynchronous cellular evolutionary algorithms, in: GECCO 2003, 2003, pp. 955–966.
[11] L. Hogie, F. Guinand, P. Bouvry, The Madhoc Metropolitan Adhoc Network Simulator, Université du Luxembourg and Université du Havre, France, available at http://www-lih.univ-lehavre.fr/~hogie/madhoc/.
[12] B. Williams, T. Camp, Comparison of broadcasting techniques for mobile ad hoc networks, in: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC), 2002, pp. 194–205.
[13] I. Stojmenovic, J. Wu, Broadcasting and activity scheduling in ad hoc networks, in: Mobile Ad Hoc Networking, IEEE/Wiley, 2004, pp. 205–229.

[14] J. Wu, W. Lou, Forward-node-set-based broadcast in clustered mobile ad hoc networks, Wirel. Commun. Mobile Comput. 3 (2) (2003) 155.
[15] A. Pelc, Handbook of Wireless Networks and Mobile Computing, Wiley, 2002, Ch. Broadcasting In Wireless Networks, pp. 509–528.
[16] H. Lim, C. Kim, Multicast tree construction and flooding in wireless ad hoc networks, in: ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM), 2000, pp. 61–68.
[17] W. Peng, X.-C. Lu, On the reduction of broadcast redundancy in mobile ad hoc networks, in: MobiHoc '00: Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing, IEEE Press, 2000, pp. 129–130.
[18] T. Bäck, D. Fogel, Z. Michalewicz (Eds.), Handbook of Evolutionary Computation, Oxford University Press, 1997.
[19] E. Alba, M. Tomassini, Parallelism and evolutionary algorithms, IEEE Trans. Evol. Comput. 6 (5) (2002) 443–462.
[20] E. Cantú-Paz, Efficient and Accurate Parallel Genetic Algorithms, Kluwer Academic Publishers, 2000.
[21] E. Alba, J.M. Troya, Improving flexibility and efficiency by adding parallelism to genetic algorithms, Stat. Comput. 12 (2) (2002) 91–114.
[22] B. Manderick, P. Spiessens, Fine-grained parallel genetic algorithm, in: Proc. of the Third Int. Conf. on Genetic Algorithms (ICGA), 1989, pp. 428–433.
[23] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[24] C.A. Coello, G. Toscano, Multiobjective optimization using a micro-genetic algorithm, in: GECCO 2001, 2001, pp. 274–282.
[25] K. Deb, R. Agrawal, Simulated binary crossover for continuous search space, Complex Syst. 9 (1995) 115–148.
[26] J. Knowles, D. Corne, Approximating the nondominated front using the Pareto archived evolution strategy, Evol. Comput. 8 (2) (2001) 149–172.
[27] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, J.-P. Sheu, The broadcast storm problem in a mobile ad hoc network, in: Proceedings of the 5th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking, 1999, pp. 151–162.

Enrique Alba is a Professor of Computer Science at the University of Málaga, Spain. He got his Ph.D. degree on designing and analyzing parallel and distributed genetic algorithms. His current research interests involve the design and application of evolutionary algorithms, neural networks, and other bio-inspired systems to real problems including telecommunications, combinatorial optimization, and bioinformatics. The main focus of all his work is on parallelism. Dr. Alba has published many scientific papers in international conferences and journals, as well as he holds national and international awards to his research results.



Bernabé Dorronsoro received the degree in Computer Science from the University of Málaga, Spain in 2002. At present he is a Ph.D. student working in the design of new evolutionary algorithms, specifically on structured algorithms, and their application to complex problems in the domains of logistics, telecommunications, combinatorial, and multi-objective optimization.



Francisco Luna received the Engineering degree in Computer Science from the University of Málaga, Málaga, Spain, in 2002. He is working towards the Ph.D. degree in the design of parallel heterogeneous metaheuristics and their application to solve complex problems in the domain of telecommunications and combinatorial optimization.

Antonio J. Nebro received his M.S. and Ph.D. degrees in Computer Science from the University of Malaga, Spain, in 1992 and 1999, respectively. He is currently an Associate Professor of Computer Science at the University of Malaga, Spain. He has coauthored several book chapters, and over 20 papers. His current research interests include the design and implementation of parallel evolutionary algorithms, multi-objective optimization, grid computing applied to meta-heuristic techniques, and applications to telecommunications and bioinformatics.

Pascal Bouvry earned his Ph.D. degree ('94) in computer science at the University of Grenoble, France. He is now Professor at the Faculty of Sciences, Technology and Communication of the University of Luxembourg and heading the Computer Science and Communication research unit (http://www.uni.lu). Pascal Bouvry is specialized in parallel and evolutionary computing. His current interest concerns the application of nature-inspired computing for solving security issues.

Luc Hogie earned both Research and Professional Master degrees at the University of Le Havre (FR) in 2001. He then worked for the CRS4 (IT) as a research engineer. After a short stay in the software development industry (SOGET, FR), he began in 2003 a Ph.D. of le Havre, advsived by Profs. Frédéric Guinand (Le Havre, FR) and Pascal Bouvry (Luxembourg). Meanwhile, he is working as a teaching assistant at Luxembourg University. His main concerns are simulation of large mobile ad hoc networks as well as network broadcasting.