STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LV**, Number 3, 2010

SOLVING OPTIMAL BROADCASTING STRATEGY IN METROPOLITAN MANETS USING MOCELL ALGORITHM

M. GHONAMY, A. BADR, AND ABD EL FATAH HEGAZY

Abstract. Mobile ad-hoc networks (MANETs) are a set of communicating devices that are able to spontaneously interconnect without any pre-existing infrastructure. In such a scenario, broadcasting becomes very important to the existence and the operation of this network. The process of optimizing the broadcast strategy of MANETs is a multi-objective problem with three objectives: (1) reaching as many stations as possible, (2) minimizing the network utilization and (3) reducing the broadcasting duration. The main contribution of this paper is that it tackles this problem by using multi-objective cellular genetic algorithm that is called MOCELL. MOCELL computes a Pareto front of solutions to empower a human designer with the ability to choose the preferred configuration for the network. Our results are compared with those obtained from the previous proposals used for solving the problem, a cellular multi-objective genetic algorithm which called cMOGA (the old version of MOCELL). We conclude that MOCELL outperforms cMOGA with respect to set coverage metric.

1. Introduction

Mobile ad-hoc Networks (MANETs) are composed of a set of communicating devices that are able to spontaneously interconnect without any preexistence or operation of the network. There is no such an organization responsible for this kind of networks. Bluetooth and wifi are the most popular wireless networking technologies available. In MANET, devices communicate in a short limit and they can move while communicating. One of the main obstacles for performing efficient communication is that the topology may change quickly and unpredictably.

Received by the editors: June 2, 2010.

2010 *Mathematics Subject Classification.* 78M50.

1998 *CR Categories and Descriptors.* G.1.6 [**Mathematics of Computing**]: Numerical Analysis – *Optimization*.

Key words and phrases. Cellular Genetic Algorithm, Multi-Objective Optimization, Mobile Ad-hoc networks, Broadcasting.

The considered problem in this paper is broadcasting on a Metropolitan MANETs. Metropolitan MANETs is a subclass of MANETs which have some specific properties: Their density is heterogeneous and it is also dynamic (high density regions don't remain active full time). The considered broadcasting strategy in this work is Delay Flooding with Cumulative Neighborhood protocol (DFCN) [19].The considered three real world examples of such a network are mall environment, Metropolitan area, and highway environment. We took the previous environments into account so, instead of providing a special-purpose protocol for each environment, our suggestion lies in tuning the broadcasting process to adapt with each environment. The optimization of broadcasting process needs multi-goals to be satisfied at the same time by: (1) maximizing the number of reached devices (coverage) (2) minimizing the network usage (bandwidth) and (3) minimizing the duration of the process. This means that we are facing multi-objective optimization [5] [16].

The intended result of multi-objective optimization is not a single solution as the single-objective optimization. Rather, the goal is a set of solutions called Pareto optimal set (section 2). The goal is a set of solutions because one solution can provide the best result in one objective but another solution can provide the better results in another objective i.e: in our MOP, one solution can provide the best result in term of coverage but other solution can provide the best result in term of duration. These solutions are called non-dominated solution (Pareto optimal). When Pareto optimal plotted in the objective space, it is called Pareto front. Then, the role of the decision maker comes by choosing the most suitable solution from the Pareto front. In this paper, we investigate solving the problem of tuning some broadcasting strategy for metropolitan MANETs by using multi-objective optimization evolutionary algorithm (MOCELL).

Many evolutionary algorithms are used to solve multi-objective optimization problems. Although cellular genetic algorithm (cGA) has proved high efficiency and accuracy in solving single-objective optimization problems, a few works used genetic algorithm based on cellular population structure [20] in solving multi-objective optimization problem. The algorithm we propose is MOCELL which is presented in [2] as a new version of cMOGA (cellular Multi-Objective Genetic Algorithm).Our contribution lies in modifying MOCELL to be adapted with the nature of our problem (multiple decision variables with different data types). To the best of our knowledge, this is the first attempt to solve the broadcasting problem on MANETs by using MOCELL and the second with structured multi-objective EVs.

In order to verify the obtained results of the aforementioned algorithm MOCELL, we compared MOCELL results against CMOGA (the previous proposal used for solving our problem). But we needed to re-implement CMOGA in order to avoid the influence of the differences between the programming techniques used in this and the previous study.

The rest of the paper is organized as follows: section 2 presents a brief survey on multi-objective optimization and in section 3 we describe our problem and how the broadcasting protocol works. We present the chosen algorithm MOCELL in detail in section 4 and in section 5, we present our experiment in terms of the simulator configuration, parameters used with MOCELL, and the obtained results. We present and analyze the obtained result of comparing MOCELL results against cMOGA results in section 6.Finally; we summarize and suggest some topics for future research.

2. Multi-objective optimization

In this section we will revise some multi-objective optimization background. The concepts of multi-objective optimization, feasible region, Pareto optimality, Pareto dominance, Pareto optimal set, Pareto front, Pareto set approximation and Pareto front approximation are defined in the following subsections.

The scenario considered in this section involves an arbitrary optimization problem with p number of constrains and m objective functions which are (without loss of generality) to be maximized.All the objectives have equal weight.

Multi-objective optimization problem (MOP):
MOP can be defined as finding the vector $\overrightarrow{X^*} = [x_1^*, x_2^*, ..., x_n^*]$ which maximizes the vector function $\overrightarrow{f}(\overrightarrow{x}) = [f_1(\overrightarrow{x}), f_2(\overrightarrow{x}), ..., f_m(\overrightarrow{x})]$ where $\overrightarrow{x} = [x_1, x_2, ..., x_n]$ is the vector of decision variables. It also must satisfy the p constrains $h_i(\overrightarrow{x}), i = 1, 2, ..., p$

Feasible region
Feasible region Ω can be defined as the set of all vectors which satisfy all the constrains. Any point that belongs to the feasible region $\overrightarrow{x} \in \Omega$ is called *feasible solution*

Pareto dominance:
An objective vector $\overrightarrow{a} = (a_1, a_2, ..., a_n)$ is said to dominate $\overrightarrow{b} = (b_1, b_2, ..., b_n)$ (denoted by $\overrightarrow{a} \succ \overrightarrow{b}$) if and only if $\overrightarrow{b}$ is partially less than $\overrightarrow{a}$ i.e, $\forall i \in \{1, ..., n\}, a_i \geq b_i \wedge \exists i \in \{1, ..., n\} : a_i > b_i$ .In another word an objective vector $\overrightarrow{a}$ dominate $\overrightarrow{b}$ if no component of $\overrightarrow{a}$ is smaller than the corresponding component of $\overrightarrow{b}$ and at least one component is greater.

Pareto Optimality:
A point $\overrightarrow{x}' \in \Omega$ is Pareto optimal only if $\neg\exists \overrightarrow{x} \in X, \overrightarrow{f}(\overrightarrow{x}') \prec \overrightarrow{f}(\overrightarrow{x})$, for all $\overrightarrow{x}$ which belong to the *decision space* X, such a $\overrightarrow{x}$ that dominates $\overrightarrow{x}'$ does not exist.

Pareto optimal set:
The Pareto optimal set for a given MOP $\vec{f}(\vec{x})$ can be defined as $P^* = \{\vec{x} \in \Omega | \neg\exists \vec{x}' \in \Omega, \vec{f}(\vec{x}') \succ \vec{f}(\vec{x})\}$. In another word, none of the elements of the Pareto optimal set is dominated by others which belongs to the feasible region.
Pareto front:
The Pareto front for a given MOP $\vec{f}(\vec{x})$and its Pareto optima set P^* can be defined as $PF^* = \{\vec{f}(\vec{x}), \vec{x} \in P^*\}$.
Pareto set approximation:
Most work in the area of evolutionary multi-objective optimization has focused on the approximation of the Pareto optimal set. So we consider the outcome of our algorithm as mutually nondominated solutions, or for short Pareto set approximation.
Pareto front approximation:
To sum up, we can describe Pareto front approximation as the front of Pareto set approximation.

3. The problem

The considered problem consists of finding the most adequate parameters for DFCN broadcasting algorithm. This section is arranged as follows; in section 3.1 we describe the considered network in our work. We describe in section 3.2 the target broadcasting algorithm DFCN which should be tuned and in section 3.3 we present the Multi-objective optimization problem of our work.

3.1. **Metropolitan mobile ad hoc networks (MANET).** Metropolitan mobile ad hoc network is MANET with the following properties. The first property is the high density areas where the nodes density is higher than the average i.e. school, airport, or supermarket. High density areas don't remain active all the time, they may appear or disappear from the system at any time i.e. school working hours from 8:00am to 5:00pm and the density of this school area outside this period is very low.

We needed a software simulator to represent such a network which allows us to tackle our problem. The chosen software simulator is Madhoc, a metropolitan MANET simulator [18]. Madhoc works as a tool to simulate different scenarios and environments based on some parameters.

There are a number of topological configurations such as people moving in a gallery place, airport place, and shopping center. The previous scenarios have different characteristics such as the size of the area, the mobility, the density of devices, the existence of walls (which has an effect on both the mobility and the signal strength), and other characteristics. We used three different scenarios implemented by Madhoc. The chosen scenarios are real world scenarios that model metropolitan area, shopping mall and a highway scenario.

- **Metropolitan environment** The metropolitan environment simulates MANETs in a metropolitan area. In this environment, we located a set of spots (crossroads) and connect them by streets. We model both human and vehicles, and they are continuously moving from one crossroad to another through streets. It is obvious that devices need to reduce their speed while attempting to cross a crossroad (like in the real world).
- **Mall environment** The mall environment is used to simulate MANETs in commercial shopping center. In this environment, the shops are located together in the corridors. The people move from one shop to another through corridors, and sometimes they stop to watch some shop window. These malls are very crowded (the density of devices is high).The behavior of people in shops is different from their behavior out of those shops (in term of mobility). There is a high density of shops in this environment. At the end, the walls of building restrict the mobility of devices and their signal propagation.
- **Highway environment** The highway environment simulates MANETs outside cities. This environment is characterized by the large surface with roads, and people travelling by car. Therefore, the density of this environment is very low since all devices are located in the roads moving in a high speed (in term of mobility). The obstacles that attenuate the signal strength and devices movement do not exist.

3.2. **Delayed flooding with cumulative neighborhood (DFCN).** The broadcasting protocols can be classified according to their algorithmic nature by the following criteria: determinism, reliability, or the information required by their execution such that the content of the hello messages. The deterministic algorithms do not use any randomness while the reliable algorithms guarantee the full coverage of the network [12]. In another work [14] the protocols are categorized as centralized and localized. Centralized protocols [1] need a global or semi-global knowledge of the network. So they are not scalable. On the other hand, the local protocols need some knowledge about one or two hops in the network.

According to the classification presented earlier, DFCN is a deterministic algorithm. It is a local protocol which works with 1-hop knowledge that permits DFCN to achieve great scalability. In DFCN, the "hello" messages do not carry any additional information but the broadcasting messages embed the list of node's neighbors.

Here is some additional information about DFCN.

- DFNC requires 1-hope neighborhood information like many other neighborhood knowledge based broadcasting protocols. DFNC obtains the required information through "hello" packets which work on network layer. The set of neighbors of device x is called N(x).
- The set of IDs of the 1-hop neighbors of every broadcasted message m is embedded in the header of m.
- Each device records local information about all the received messages. The single record of this local information consists of:
  - The received message ID.
  - The set of IDs of the devices that receive the message.
  - The decision of whether or not the message should be forwarded.
- Random Assessment Delay (RAD) is a random delay used by DFCN before re-forwarding a broadcast message m. It is used to prevent the collisions. In another word, while a device x forwards a message m, all the devices in N(x) receive it in the same time. Then all of them will re-forward the message m simultaneously and this causes network collisions. The goal of using RAD is delaying the process of re-forwarding the message m for each device in N(x) with a random value. Therefore, the risk of collisions is significantly reduced.

DFCN algorithm can be divided into three parts. The first two parts are responsible for dealing with outcoming events. The first part is responsible for dealing with new message reception, while the second is responsible for detecting a new neighbor. The third part is responsible for re-forwarding the received messages or detecting new neighbor during the follow-up of one of the previous parts. Reactive behavior is the behavior resulting from a message reception. Proactive behavior is the behavior resulting when a new neighbor is discovered.

Let x_1 , x_2 are two neighbor devices. When x_1 sends a message m to x_2 , the list of $N(x_1)$ are embedded in the sent message m. After x_2 receives the message m, it knows the set of all the message m recipients $N(x_1)$. Therefore, $N(x_2) - N(x_1)$ are the set of devices that have not received the message m yet. If x_2 re-forwards the message m, the number of devices that receives m for the first time is maximized through the following equation: $h(x_2, x_1) = |N(x_2) - N(x_1)|$.

The received message m is re-forwarded only if the number of neighbors who have not received the message m yet is greater than a given threshold to reduce the usage of network bandwidth. The threshold is a function of the neighbor devices for the receptor x_2 and it is written as threshold $(|N(x_2)|)$.

The device x_2 uses a Boolean function $B(x_2, x_1)$ to decide whether to re-forward the message m or not. The Boolean function $B(x_2, x_1)$ is defined as:

$$B(x_1, x_2) = \begin{cases} true, & h(x_1,x_2) \geq threshold(|N(x_2)|) \\ false, & otherwise \end{cases} \tag{1}$$

The recipient devise x_2 re-forwards the message m only if the threshold is exceeded. After the random delay defined by RAD is finished, the message m is re-forwarded. The threshold function allows DFCN to facilitate the message re-forward when the connectivity is low. It takes the recipient device x_2 neighbors number as a parameter and it is defined as:

$$threshold(n) = \begin{cases} 1, & n \leq safeDensity \\ minGain * n, & otherwise \end{cases} \tag{2}$$

DFCN always re-forwards while the density is below the maximum safe density called safe Density. DFCN uses minGain parameter to compute the minimum threshold for forwarding a message.

When the device x discovers a new neighbor, it forwards this discovery if $N(x)$ is lower than the given threshold called $proID$, otherwise this behavior is disabled, which means that there is no action taken in case of the new neighbor discovery.

3.3. **DFCNT (DFCN Tuning) as MOP.** In this subsection, we present the Tuning of DFCN as a multi-objective optimization problem that we call DFCNT. The following are the five parameters that must be tuned with the role and range of each parameter in the DFCN.

- *minGain* is the minimum gain from the re-broadcasting process. Since minimizing the bandwidth should be highly dependent on the network density, minGain is the most important parameter for tuning DFCN. It ranges from 0.0 to 1.0
- *lowerBoundRAD* parameter is used for defining the lower bound of RAD value (random delay in re-broadcasting in milliseconds). This parameter takes values in the interval [0.0, 10.0] ms.
- *upperBoundRAD* parameter is used for defining the upper bound of RAD value. The parameter takes values in the interval [0.0, 10.0]ms.
- *proD* parameter is used for setting the maximum density to enable the proactive behavior (reacting to new neighbor). The parameter takes values in the interval [0, 100].
- $safeDensity$ parameter is used for defining a maximum safe density of the threshold. This parameter takes values in the interval [0, 100].

The previous five parameters are considered as decision variables that characterized the search space. The chosen intervals are wide enough to include all the reasonable values that can be found in real scenarios. The three objective functions are defined as follows: the first objective function is minimizing the duration of the broadcasting process, the second is maximizing the network coverage and the third is minimizing the number of transmission (reduce bandwidth usage).Since we have three different real world Metropolitan MANETs scenarios, three instances of DFCNT have to be solved: DFCNT, Meropolitan, DFCNT, Mall and DFCNT, Highway.

4. The algorithm

Using EAs (Evolutionary Algorithms) in solving optimization problem has been very intense during the last decade [22]. It is possible to find this kind of algorithms tackling complex problems like constrained optimization task. These algorithms work on a set (population) of solution (individuals) by applying some stochastic operator on them to search for the best solution. Most EAs use a single population of individuals. They also apply their stochastic operator on the whole population as illustrated in figure [1]. On the other hand, there are other EVs that use structured population. In that case, the population is somehow decentralized. Structured EVs most suited to parallel implementation. The EAs that use decentralized population provide a sampling of the search space which improves both numerical behavior and execution time better than those that use single population. Distributed and cellular EVs are the most popular among many types of structured EAs as illustrated in figure [1] [4][6][7][11].We focus in this work on Cellular Genetic Algorithms (CGAs). CGAs use a small neighborhood concept, which mean that individual can only interact with his neighbors [4]. The overlapped small neighborhoods of CGAs help with exploring the space because the induced slow diffusion of solutions through the population provides a kind of exploration while exploitation takes place inside each neighborhood by genetic operations. Although CGAs were initially designed to parallel processors machines, they were adapted to suit mono-processor machines and accomplish good results. The neighborhood definition (during the CGA execution) did not depend on the graphical neighborhood definition in the problem space.

4.1. **Cellular genetic algorithm.** In this subsection, we present the canonical CGA in detail as published on [9]. CGA pseudo-code is presented in Algorithm 1. Since CGA is a structured EA, its population is structured as follows: it is usually structured in a regular grid of d dimensions with the neighborhood defined on it. The algorithm works on each individual in the population according to its place orderly (Algorithm1 line 5). The current individual can only interact with his or her neighbors (Algorithm 1line 6). The current individual parents are chosen from the neighbors by using some selection technique (Algorithm 1 line 7). In line 8 and 9, crossover and mutation operators are applied to the current individual with probabilities P_c, P_m respectively. After that, the algorithm computes the fitness values of the offsprings (line 10) then, inserts them or one of them instead of the current individual either in the current population or in a new one according to the chosen replacement policy (line 11).

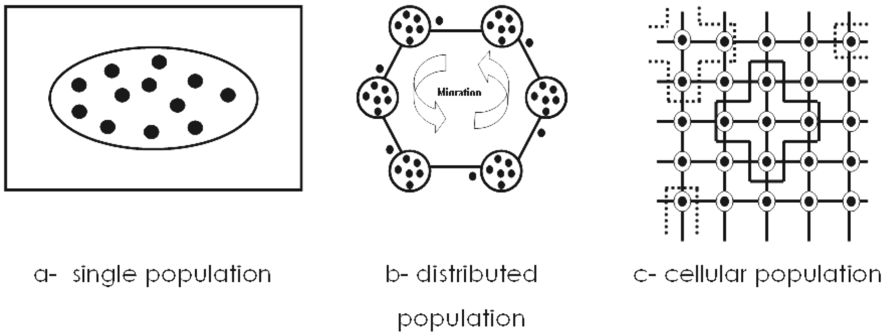

FIGURE 1. single (a), distributed (b), and cellular (c) EAs

After finishing the previous cycle for all individuals, we get a new population for the next generation (line 13). The loop continues until termination condition is met (line 4). The termination condition is met either by finding the optimum solution or exceeding the maximum number of calling the evaluation function or composed of both.

4.2. **Multi-objective cellular GA: MOCELL.** In this subsection, we present MOCELL, a multi-objective optimization algorithm based on a cGA model as presented in [2][3]. But we needed to modify it in order to tackle our problem in terms of dealing with multiple non-heterogeneous decision variables .We observed that Algorithms 1 and 2 were very similar. One of the main differences between the two algorithms is the existence of a Pareto front (see section 2) in the MOCELL algorithm. The Pareto front is just an additional population (the external archive) composed of a number of the non-dominated solutions found since it 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 density estimator based on the crowding distance proposed for NSGA-II [17] has been used. This measure is also used to remove solutions from the archive when it is full.

MOCell starts by creating an empty Pareto front (line 2 in Algorithm 2). Individuals are arranged in a 2-dimensional grid and the genetic operators were successively applied on them (lines 9 and 10) until the termination condition was met (line 5). Hence, the algorithm for each individual consists of two parents from their neighborhood, recombining them in order to obtain an offspring, mutating it, evaluating the resulting individual and inserting it in both the auxiliary population (if it is not dominated by the current individual) and the Pareto front. Finally, after each generation, the auxiliary one replaces the old population and a feedback procedure is invoked to replace a fixed number of randomly chosen individuals of the population by solutions from the archive.

Algorithm 1 Pseudo-code of a canonical cGA

```
1:Proc Evolve(cga)
2:GenerateInitialPopulation(cga.pop);
3:Evaluation(cga.pop);
4:While ! StopCondition() do
5: for  individual = 1 to cga.popSize do
6:   neighbors =calculateNeighborhood(cga, position(individual));
7:   parents   =selection(neighbors);
8:   offspring =recombination(cga.Pc, parents);
9:   offspring =mutation(cga.Pm,offspring);
10:  evaluation(offspring);
11:  replacement(position(individual), auxiliary_pop,offspring);
12: End for
13: Cga.pop     =auxiliary_pop;
14:end while
15:end proc Evolve
```

5. Experiments

In this section, we first describe the configuration of the network simulator (MadHoc). Next, we present the parameterization used by MOCELL. Finally, we present the analysis of the obtained results for DFCNT.

MOCELL has been implemented in Java and tested on a PC with a 2.8 GHz (dual-core) processor with 2GB of RAM memory and running windows XP service back 3. The java version used is 1.7.0. Although cMOGA was used in previous research to tackle our problem, we re-implemented it in order to avoid the influence of the differences between the programming techniques used in this and the previous study.

Algorithm 2 Pseudo-code of MOCELL

```
Proc Evolve(mocell)
Pareto_front = createPFront();
GenerateInitialPopulation(mocell.pop);
Evaluation(mocell.pop);
while ! StopCondition() do
 for  individual = 1 to mocell.popSize do
   neighbors = getNeighborhood(mocell, position(individual));
   parents   = selection(neighbors);
   offspring = recombination(mocell.Pc, parents);
   offspring = mutation(mocell.Pm,offspring);
   evaluation(offspring);
   Insert(position(individual),offspring,mocell, auxiliary_pop);
   InsertInParetoFront(individual,Pareto_front);
 end for
mocell.pop = auxiliary_pop;
mocell.pop = Feedback(mocell,Pareto_Front);
end while
end proc Evolve
```

5.1. **Madhoc Configuration.** There are three different environments for MANETs that Model three possible real-world scenarios. The main features of these environments are explained in this chapter and they are summarized in table [1]. In figure [2], we show an example for each environment. The examples in figure [2] are obtained by using the graphical user interface of Madhoc simulator by using the proposed configurations summarized in table [1].The broadcasting process is considered to be completed when either the coverage is 100% or it does not vary for 1.5 second. The broadcasting process termination is truly important since improper termination condition can lead to bad results or slow simulation.

Table 1. Main features of Madhoc environment

		Metropolitan	**Mall**	**Highway**
Surface (m^2)		160,000	40,000	1,000,000
Density of spots		50 (crossroad/km^2)	800 (store/km^2)	3 (joints/km^2)
Spots radius (m)		3 - 15	1 - 10	50 - 20
	Speed out of spots (m/s)	1 - 25	0.3 - 1	30 - 50
Devices	Speed in spots (m/s)	0.3 - 10	0.3 - 0.8	20 - 30
	Density(dev./km^2)	500	2000	50
Wall obstruction (%)		90	70	0

5.1.1. *The Metropolitan Environment.* In this section, we study the behavior of DFCN in the Metropolitan environment. In this environment modulation, we set the surface as 400 * 400 square meters. The density of spots (crossroads) is 50 per square kilometer. Each spot has a circle surface of radius between 3 and 15 meters. In this scenario, the wall obstruction (penalty of signal) is up to 90%. The density of the devices is 500 elements per square kilometer. While setting the speed parameter, we should consider the cases when people or cars move, so the value of movement speed in crossroads area ranges between 0.3 and 10 m/s, and between 1 and 25 m/s in other cases (streets). This kind of environment consists of a few numbers of sub-networks that are connected to each other by few links, one or two or even zero in case of unconnected subnetworks. Isolated nodes are those devices that are not connected to any subnetworks as illustrated in figure [2]. The topology of this environment can vary in a very fast way since the devices can move through cars. All of the previous properties show us how hard is the broadcasting process through this network and this was what made this scenario challenging for us.

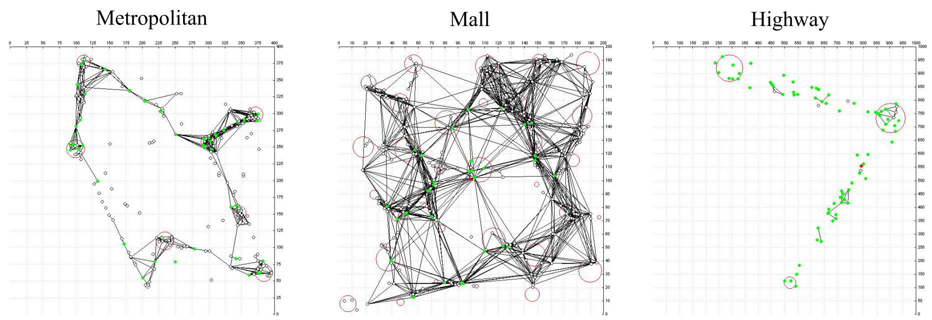

FIGURE 2. MANET scenario

5.1.2. *The Mall Environment.* In this section, we show the parameter of Madhoc configuration of the mall environment. In this scenario, the number of both shops (spots) and devices is very high (density). There are walls that have two roles, the first is to attenuate the signals and the second is to slow down the speed of devices that is already slow since we are modeling people walking. The surface of this environment is defined as 200 * 200 square meters. The number of devices per kilometer is 2000. The number of stores (spots) per kilometer is 800. Each store (spots) has a circle of radius ranging between 1 and 10 meters. The obstruction of the wall is measured by 70% attenuation of the signal strength. At the end, the speed of the devices range between 0.3

and 1 m/s inside the corridors (speed out of spots) and between 0.3 and 0.8 m/s inside stores (speed in spots).

In figure [2], we can notice that the mall environment diagram is a very condensed graph. The graph is condensed because the mobile devices coverage ranges between 40 and 80 meters. Therefore, the Mall environment problem is the hardest problem because of the broadcast storm [21].

5.1.3. *The Highway Environment.* The highway environment consists of a small number of devices moving in a high-speed manner. In this environment, there is no wall obstruction. The signal attenuation is set to 0%. The surface in this environment is 1000 * 1000. The number of devices is 50 devices per square kilometer. There are three spots (highway entrances or exits) in this scenario. The speed of devices outside the spots ranges from 30 to 50 m/s. The speed of the devices inside the spots ranges from 20 and 50 m/s. The radius of each spot ranges from 25 to 100 meter.

In figure [2], we can notice that the highway environment consists of a number of subnetworks usually unconnected. Each subnetwork is composed of a small number of devices. The main challenge in this scenario is how fast the topology changes because of the speed of the devices in the highway. The faster change the topology makes, the harder the broadcast process becomes.

5.2. **Parameterization of MOCELL.** In this section, we explain the parameter used by MOCELL in our experiment. The population consists of 100 individuals formed as square toroidal grid. We used C9 (compact nine) neighborhood composed of 9 individuals; the selected one and all adjacent individuals as illustrated in figure [3]. Per each evaluation function calling, we call the madhoc simulator five times because of the stochastic nature of the simulator. The objectives (time, coverage, and bandwidth) are calculated as an average of the five returned values through the five simulator calling. Calling the simulator five times per function has a great effect on our experiment time. The previous details show us why the number of algorithm is just 30 times.

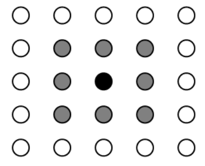

FIGURE 3. C9 neighborhood

Table 2. MOCELL Parameters

Parameter Name	**Parameter value**
Population size	100 individuals
Stop condition	25000 function evaluations
Neighborhood	C9
Parent selection	Binary Tournament + Binary Tournament
Recombination	Simulated Binary probability = 1.0
Mutation	Polynomial probability = 1.0/L
Replacement	Rep_If_Better
Archive size	100
Density estimator	Crowding distance
Feedback	20 individuals

Simulated Binary operator (SBX) [15] is used in the recombination phase with probability $p_c = 1.0$ since we deal with continuous decision variables. SBX simulates the behavior of the single point binary crossover on double individuals. We used polynomial operator [15] as a mutation operator with probability $p_m = 1.0/L$ for every allele (where L is the length of individual). We chose both parents by using Binary Tournament. The resulting offspring replaces the current individual if it dominates the current individual. We used adaptive grid algorithm to insert the individuals into the Pareto Front [13]. This algorithm divides the objective space into hypercubes that lead to the balance of the density of the non-dominated solutions in these cubes. In the case of inserting a new non-dominated solution into the Pareto Front, the grid location of the solution is determined. If the Pareto Front is already full and the new non-dominated solution does not belong to the most crowded hypercube then one of the solutions that belongs to that hypercube is removed to leave a space for the new non-dominated solution.

Using the try and error technique, we conclude that the previous MOCELL parameters are considered the best parameters for MOCELL in solving the aforementioned problem.

5.3. **Results for DFCNT.** In this section, we analyze the result of DFCNT in the three different environments. The DFCNT problem is composed of five decision variables and three objective functions. The experiment consists of 30 independent runs for each problem environment. The experiment execution time is almost 2 months.

We show the mean and standard deviation of both time (in hours) and number of Pareto optima obtained by MOCELL for the three different instances of the DFCNT problem (metropolitan, mall, and highway) in table

[3]. As we can see, the single execution run is 23 hours for Metropolitan and 16 hours for mall and 10 hours for highway. The complexity of the evaluation function, since we call the simulator five times, is the only reason for the long time of our experiment. The average of the number of Pareto optima obtained is 98.9 for Metropolitan, 99.6 for Mall, and 97.4 for highway where the maximum is 100 solutions per run. This result is very satisfying for the three instances of the problem since we provide the decision makers with a wide range of solutions.

In Figure [4], we show the diversity of MOCELL result for each of the three instances of the DFCNT problem. Best solutions are those that satisfy the following objective functions (maximize the coverage, minimize bandwidth and minimize the duration of the broadcasting process). From the obtained results, the solutions that cover over 95% in the broadcasting process need in average 720.8 ms and 69.91 messages (bandwidth usage) for the Metropolitan scenario. In addition, they need 163 ms and 22.45 messages for the mall scenario and 827.1 ms and 71.61 messages for the highway scenario. In fact, only 11% from the Pareto optima solutions reach 95% coverage for the metropolitan environment while 39% and 6% for mall and highway in consecutive. The previous results reflect the importance of the coverage and how hard it is to satisfy this objective function.

By looking to figure [4], we can note that in the case of the mall scenario, the duration is less than 250 ms, bandwidth usage is less than 30 messages and the coverage is always more than 0.4. Therefore, it is so obvious that the broadcasting process in the mall scenario is better than the other scenarios. In figure [4], the Duration axis (time in milliseconds) shows that the broadcasting process in both metropolitan and highway scenarios takes longer time than mall scenario. The bandwidth axis (number of sent messages) shows us that the broadcasting process in both metropolitan and highway scenarios take longer time than the mall scenario. The coverage axis (percentage of all devices in the network) shows us that there are some solutions with coverage less than 10% in both metropolitan and highway scenarios.

TABLE 3. experiment's time, and number of Pareto optimal for each problem

Environment	Time (h)	Number of Pareto optima
DFCNT.Metropolitan	$23.09\pm_{0.998}$	$98.9\pm_{1.45}$
DFCNT.Mall	$15.87\pm_{0.368}$	$99.6\pm_{0.966}$
DFCNT.Highway	$9.852\pm_{0.181}$	$97.4\pm_{5.235}$

The previous coverage results are expected because they depend on the difference between the scenarios. The probability of having isolated sub-networks (consists of one or two devices) increases with the decrease in devices density (increase simulation area and decrease devices number). Since the mall scenario has the highest connectivity (highest devices density), it has the best coverage results. However, the high density has its drawback because it increases the risk of broadcast storm which makes solving DFCNT.mall very hard. Based on these results, we note that MOCELL succeeded in dealing with this problem.

The Pareto fronts illustrated in figure 4 achieves the designs objectives of the DFCN protocol, since most of the plots are distributed on a wide range that provides a decision maker with a wide variety of solutions. Our results also have a set of solutions that allow DFCN to achieve a coverage rate close to 100%, while keeping the network throughput very low.

6. Comparing MOCELL against CMOGA

In this section, we compare our study with those that used CMOGA on DFCNT problem. The three instances of DFCNT problem (metropolitan, mall, and highway) are solved with CMOGA to make this case study. Although the DFCNT problem has previously been solved by CMOGA algorithm in [8], we re-implemented CMOGA algorithm in order to insure high accuracy in our comparative study by avoiding implementing differences effect.

6.1. Parameterization of CMOGA.

In this section, we show the CMOGA algorithm parameters. The algorithm population size is 100 individuals. It stops when 25000 evaluation functions have been made. We chose C9 as a neighborhood operator described in section 2 and illustrated in figure [3]. Both the parents are chosen by Binary tournament operator. In the Recombination step, we used simulated binary crossover [15] with probability = 1. In the Mutation step, we used polynomial [15] with probability $p_m = 1.0/L$ for every allele (where L is the length of individual). The offspring replaces the current individual only if the former dominates the latter. The maximum archive size is 100 individuals. We used adaptive grid algorithm to insert the individuals into the Pareto Front [13]. As you notice, we used almost the same parameters of MOCELL for CMOGA algorithm to insure the accuracy in our comparative study. For evaluating each individual, we had to call the simulator five times as in the case of MOCELL. Therefore, in each single run of CMOGA algorithm, we had called the simulator 125,000 times.

We used the same parameters previously used with MOCELL in order to insure the precision of our comparative study.

6.2. Evaluation of the Results.

As Cellular Genetic Algorithms belong to meta-heuristic algorithms, it is considered as a non- deterministic technique

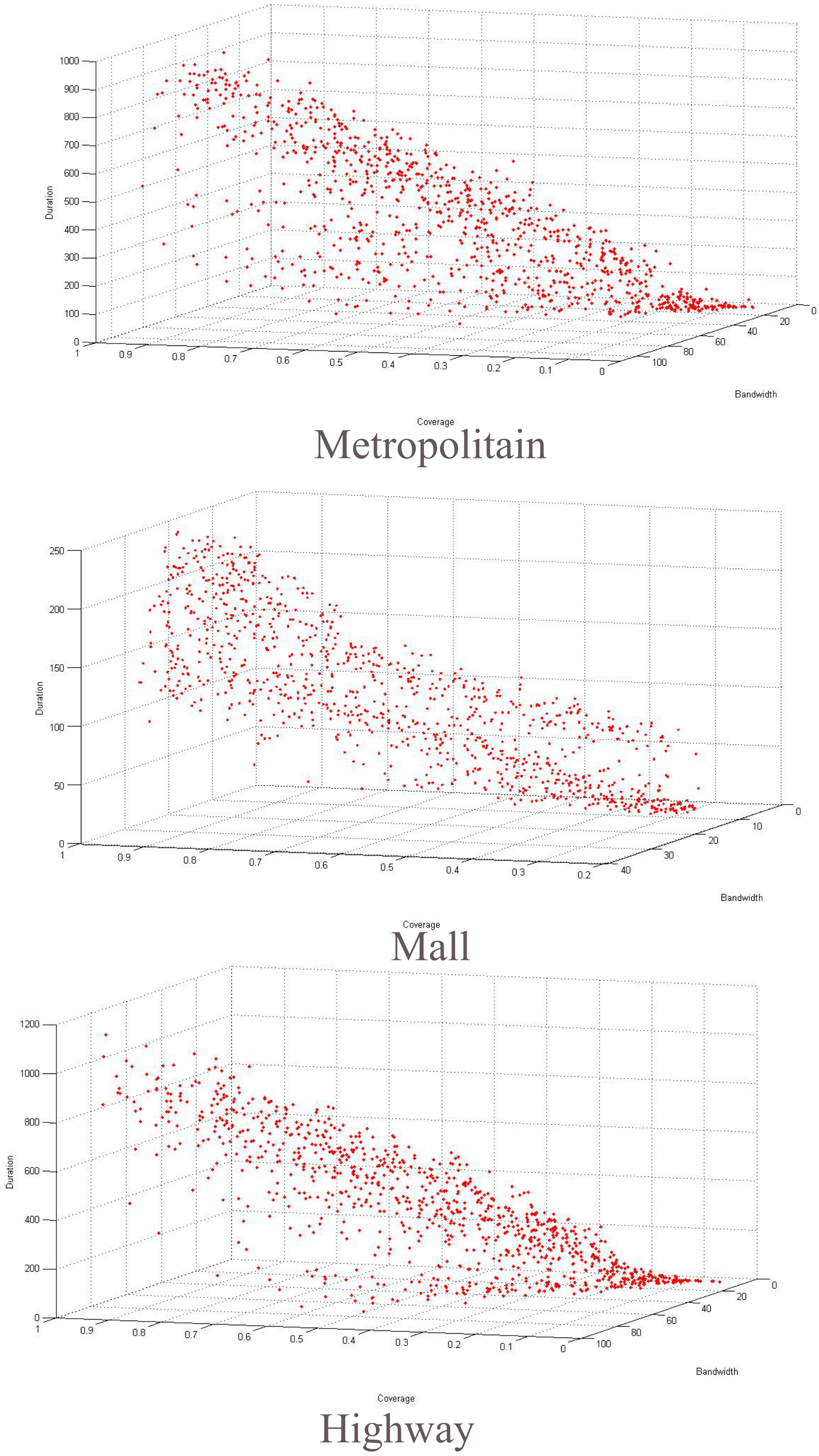

FIGURE 4. Pareto Fronts For the three environment

TABLE 4. CMOGA Parameters

Parameter Name	**Parameter value**
Population size	100 individuals
Stop condition	25000 function evaluations
Neighborhood	C9
Parent selection	Binary Tournament + Binary Tournament
Recombination	Simulated Binary probability = 1.0
Mutation	Polynomial probability = 1.0/L
Replacement	Rep_If_Better
Archive size	100
Density estimator	Crowding distance

and this means that different solutions can be reached by using the same algorithm twice on the same problem. The previous detail makes a serious problem for the researchers in evaluating their results and in comparing their algorithms results to existing algorithms.

The studied algorithms are applied to 3 scenarios of real-world problems to insure that the proposed algorithms are capable of tackling such problems.

In our case of multiobjective optimization algorithms, we have to use metrics to compare the quality of the obtained solutions. However, until now there is no single metric that proves its superiority to the other metrics. So, we need to use more than one metric to insure the accuracy in our comparative study. The chosen metrics are a number of Pareto Optima, hypervolume, and set coverage [10]. Once we apply any of the previous metrics on our obtained pareto front, we get a single value.

TABLE 5. MOCELL vs. CMOGA Number of Pareto optima

	Algorithm	X	Max	Min	Test
DFCNT.Metropolitan	MOCELL	$98.9 \pm_{1.45}$	100	96	-
	CMOGA	$99.7 \pm_{0.675}$	100	98	
DFCNT.Mall	MOCELL	$99.6 \pm_{0.966}$	100	97	-
	CMOGA	$99.9 \pm_{0.316}$	100	99	
DFCNT.Highway	MOCELL	$97.4 \pm_{5.23}$	100	84	-
	CMOGA	$94.7 \pm_{8.68}$	100	72	

Since the proposed algorithms are non-deterministic, the comparison of a single execution is inconsistent. So, the comparison must be applied on a large set of results obtained after a high number of independent executions (in our case 30 independent runs) for the algorithms on a given problem. We used a statistics function in order to make the comparisons between the obtained results. Our statistics reflect the significance of the obtained results and the comparisons as shown in the Test column.

We applied Kruskal-Wallis test on our results. Kruskal-Wallis function allows us to determine whether the effects observed in our results are significant or it appeared because of error in the collected samples. We chose this statistics function since we have non-normal data distribution. We used Kolmogorov-Smirnov test to check if our data distribution is gaussian or not. We considered a confidence level of 95% in our comparison study and this means that we can guarantee that the differences of the compared algorithms are significant or not with a probability of 95% or with the p-value less than 0.05.

Table 6. MOCELL vs. CMOGA for Hyper volume metric

	Algorithm	X	Max	Min	Test
DFCNT.Metropolitan	MOCELL	$0.9998 \pm_{5.32E-04}$	1	0.998	-
	CMOGA	$0.9996 \pm_{1.22E-03}$	1	0.996	
DFCNT.Mall	MOCELL	$0.9965 \pm_{3.77E-03}$	1	0.989	-
	CMOGA	$0.9964 \pm_{3.91E-03}$	1	0.99	
DFCNT.Highway	MOCELL	$0.9998 \pm_{6.71E-04}$	1	0.998	-
	CMOGA	$0.9999 \pm_{1.46E-04}$	1	0.999	

6.3. **Discussion.** In this section, we made the comparison between MOCELL and CMOGA algorithms. As previously mentioned, the results are obtained after making 30 independent runs of every experiment for each algorithm and the used metrics are number of non-dominated solutions found in the Pareto Front, Hypervolum, and Set Coverage.

The obtained results are shown in tables [5], [6], and [7]. The previous tables include x (the mean) and the standard-deviation of our results. They also include the maximum and minimum obtained values for each metric.

In table [5], although the obtained results are not statistically significant, we can notice that both algorithms MOCELL and CMOGA reached a high number of Pareto Optima since the maximum number of Pareto Optima is 100 solutions. In table [6], MOCELL improves CMOGA in metropolitan, and mall scenarios in terms of the hypervolume metric but the difference is not statistically significant. But CMOGA improves MOCELL in the mall scenario in terms of the hypervolume metric without statistical significance. We can notice that both of algorithms have reached high level of Hypervolume metric since the maximum value is 1. The result of the set coverage metric is shown in table [7]. The MOCELL outperforms CMOGA in two of the studied problems (metropolitan, and highway scenarios) with statistical significance in terms of the set coverage metric. In contrast to the previous scenarios, CMOGA outperforms MOCELL in the mall scenario with statistical significant in terms of the set coverage metric.

To sum up, there is no algorithms better than the others. But MOCELL seems to be better than CMOGA in terms of hypervolume and set coverage. On the other hand, CMOGA outperforms MOCELL in the case of number of pareto optima. The differences between the two algorithms are statistically significant for the set coverage metric. On the other hand, we did not find any important differences in the other two metrics (number of pareto optima, and hypervolume).

Table 7. MOCELL vs. CMOGA for Set Coverage metric

	C(A,B)					
	A	B	X	Max	Min	Test
DFCNT.	MOCELL	CMOGA	$0.3501 \pm_{9.42E-02}$	0.6	0.122449	+
Metropolitan	CMOGA	MOCELL	$0.3209 \pm_{8.14E-02}$	0.51	0.15625	
DFCNT.	MOCELL	CMOGA	$0.2841 \pm_{6.91E-02}$	0.4848	0.16	+
Mall	CMOGA	MOCELL	$0.3322 \pm_{7.56E-02}$	0.51	0.175258	
DFCNT.	MOCELL	CMOGA	$0.3704 \pm_{9.60E-02}$	0.6	0.180556	+
Highway	CMOGA	MOCELL	$0.3577 \pm_{9.88E-02}$	0.6071	0.113402	

7. Conclusions and future works

In this paper we present the problem of optimally tuning DFCN (broadcasting protocol) which works on MANET (Mobile Ad-hoc wireless Network), by using MOCELL (Multi-objective optimization algorithm). DFCNT is defined as a three objectives MOP, with the goals of minimizing the network usage, maximizing network coverage and minimizing the duration of broadcasting.

We used three different realistic scenarios. Three different instances of MOP have been solved. They are city's streets (DFCNT.Metropolitan), mall center (DFCNT.mall) and Highway streets (DFCNT.Highway). we can conclude that solving DFCNT by MOCELL provides a Pareto front set that consists of more than 95 points in the case of the highway scenario and more than 99 points in the case of the other two scenarios.

In the second part of this paper, we compared our chosen algorithm MOCELL versus cMOGA (cellular Multi-Objective Genetic Algorithm) for the three proposed problems. Three different metrics were used in order to compare the algorithms: The number of Pareto optima, the hypervolume, and the set coverage metrics. We observed that MOCELL seemed to be better than CMOGA in terms of hypervolume and set coverage. On the other hand, CMOGA outperformed MOCELL in the case of number of pareto optima. Although the differences between the two algorithms in hypervolume and number of pareto optima metrics are not statically significant, both of them reach a high pareto optima result (since the maximum Pareto front is 100) and a high hypervolume results (since the maximum value is 1.0). Regardless the hypervolume and the number of Pareto optima metrics, MOCELL won. From these results, a clear conclusion can be drawn: MOCELL is a promising approach for solving DFCNT with advantage over the existing one.

Future research is needed to tackle the MOPs with MOCELL. In addition, research that parallels MOCELL to reduce the execution time is needed because reducing time will enable us to study other real-world scenarios that are larger and have bigger number of devices.

References

[1] A. Pelc, Handbook of Wireless Networks and Mobile Computing, Wiley, 2002, Ch. Broadcasting In Wireless Networks, pp. 509-528.

[2] A.J. Nebro, J.J. Durillo, F. Luna, B. Dorronsoro, and E. Alba. A cellular genetic algorithm for multiobjective optimization. In D.A. Pelta and N. Krasnogor, editors, Proceedings of the NICSO, pages 25-36, Granada, Spain, 2006.

[3] A.J. Nebro, J.J. Durillo, F. Luna, B. Dorronsoro, and E. Alba. MOCell: A cellular genetic algorithm for multiobjective optimization. International Journal of Intelligent Systems, 2007.

[4] B. Manderick and P. Spiessens. Fine-grained parallel genetic algorithm. In J.D. Schaffer, editor, Proc. of the Third International Conference on Genetic Algorithms (ICGA), pages 428-433. Morgan Kaufmann, 1989.

[5] C.A. Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer Academic Publishers, 2002.

[6] D. Whitley. Cellular genetic algorithms. In S. Forrest, editor, Proc. of the Fifth International Conference on Genetic Algorithms (ICGA), page 658, California, CA, USA, 1993. Morgan Kaufmann.

[7] E. Alba and M. Tomassini. Parallelism and evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 6(5):443-462, October 2002.

[8] E. Alba, B. Dorronsoro, F. Luna, A.J. Nebro, P. Bouvry, and L. Hogie. A cellular multi-objective genetic algorithm for optimal broadcasting strategy in metropolitan MANETs. Computer Communications, 30(4):685-697, 2007.

[9] E. Alba, B.Dorronsoro, Handbook of Cellular Genetic Algorithms, 2008, Ch. Introduction to Cellular Genetic Algorithms, pp. 3-20.

[10] E. Alba, B.Dorronsoro, Handbook of Cellular Genetic Algorithms, 2008, Ch. Algorithmic and Experimental Design, pp. 73-82.

[11] E. Cantu-Paz. Efficient and Accurate Parallel Genetic Algorithms, volume 1 of Book Series on Genetic Algorithms and Evolutionary Computation. Kluwer Academic Publishers, 2nd edition, 2000.

[12] I. Stojmenovic, J. Wu, Broadcasting and activity scheduling in ad hoc networks, in: Mobile Ad Hoc Networking, IEEE/Wiley, 2004, pp. 205-229.

[13] J. Knowles and D. Corne. Approximating the nondominated front using the Pareto archived evolution strategy. Evolutionary Computation, 8(2):149-172, 2001.

[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] K. Deb and R.B. Agrawal. Simulated binary crossover for continuous search space. Complex Systems, 9:115-148, 1995.

[16] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms,John Wiley & Sons, London, 2001.

[17] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elist Multi- objective Genetic Algorithm: NSGA-II. IEEE TEC, 6(2):182197, 2002.

[18] L. Hogie, F. Guinand, P. Bouvry, The Madhoc Metropolitan AdhocNetwork Simulator, Universite du Luxembourg and Universite du Havre, France, available at http://www-lih.univ-lehavre.fr/∼hogie/madhoc/.

[19] L. Hogie, M. Seredynski, F. Guinand, P. Bouvry, A bandwidthefficient broadcasting protocol for mobile multi-hop ad hoc networks, in: ICN'06, 5th International Conference on Networking (to appear), IEEE, 2006.

[20] M. Laumanns, G. Rudolph, H.P. Schwefel, A spatial predator-prey approach to multiobjective optimization: a preliminary study, in:PPSN V, 1998, pp. 241-249.

[21] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu. The broadcast storm problem in a mobile ad hoc network. In Proc. of the Annual ACM/IEEE International Conference on Mobile Computing and Networking, pages 151-162, 1999.

[22] T. Back, D.B. Fogel, and Z. Michalewicz, editors. Handbook of Evolutionary Computation. Oxford University Press, 1997.

Misr University for Science and Technology, Faculty of Information Technology, Al-Motamayez District, 6th of October City, Egypt
E-mail address: `MS_Ghonamy@hotmail.com`

Cairo University, Faculty of Computers and Information, 5 Dr. Ahmed Zewail Street, Giza city, Egypt
E-mail address: `a.badr@fci-cu.edu.eg`

Arab Academy for Science and Technology, College of Computing & information Technology, El Moshir Ismail Street, Misr El Gedida, Cairo, Egypt
E-mail address: `abdheg@yahoo.com`