Optimal Broadcasting in Metropolitan MANETs Using Multiobjective Scatter Search*

F. Luna[1], A.J. Nebro[1], B. Dorronsoro[1], E. Alba[1], P. Bouvry[2], and L. Hogie[2]

[1] Department of Computer Science, University of Málaga, Spain
{flv, antonio, dorronsoro, eat}@lcc.uma.es
[2] Faculty of Sciences, Technology and Communications, University of Luxembourg
{pascal.bouvry, luc.hogie}@uni.lu

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 scenario, broadcasting becomes an operation of capital importance for the own existence and operation of the network. Optimizing a broadcasting strategy in MANETs is a multiobjective problem accounting for three goals: reaching as many stations as possible, minimizing the network utilization, and reducing the makespan. In this paper, we face this multiobjective problem with a state-of-the-art multiobjective scatter search algorithm called AbSS (Archive-based Scatter Search) that computes a Pareto front of solutions to empower a human designer with the ability of choosing the preferred configuration for the network. Results are compared against those obtained with the previous proposal used for solving the problem, a cellular multiobjective genetic algorithm (cMOGA). We conclude that AbSS outperforms cMOGA with respect to three different metrics.

1 Introduction

Mobile Ad-hoc Networks (MANETs) are fluctuating networks populated by a set of communicating devices called *stations* (they are also called *terminals*) which can spontaneously interconnect each other without a pre-existing infrastructure. This means that no carrier is present in such networks as it is usual in many other types of communication networks. Stations in MANETs are usually laptops, PDAs, or mobile phones, equipped with network cards featuring wireless technologies such as Bluetooth and/or IEEE802.11 (WiFi). In this scenario, a) stations communicate within a limited range, and b) stations can 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 on such networks.

Broadcasting is a common operation at the application level and is also widely used for solving many network layer problems being, for example, the basis

* This work has been partially funded by the Ministry of Science and Technology and FEDER under contract TIN2005-08818-C04-01 (the OPLINK project).

F. Rothlauf et al. (Eds.): EvoWorkshops 2006, LNCS 3907, pp. 255–266, 2006.

mechanism for many routing protocols. In a given MANET, due to host mobility, broadcasting is expected to be performed very frequently (e.g., for paging a particular host, sending an alarm signal, and/or finding a route to a given target terminal). Broadcasting may also serve as a last resort to provide multicast services in networks with such rapidly changing topologies and stems for the organization of terminals in groups. Hence, having a well-tuned broadcasting strategy will result in a major impact in network performance.

In this paper we are considering the problem of broadcasting on a particular sub-class of MANETs called *Metropolitan* MANETs, which cover from shopping malls to metropolitan areas. Instead of providing a generic protocol performing well on average situations, our proposal consists of optimally tuning the broadcasting service for a set of networks and for a particular category of broadcast messages. Optimizing a broadcasting strategy is a multiobjective problem where multiple functions have to be satisfied at the same time: maximizing the number of stations reached, minimizing the network use, and minimizing the makespan are three examples of the potential objectives. In this work, the broadcasting strategy considered for optimization is DFCN [1], and the target networks are metropolitan MANETs. Since manipulating such networks is difficult, we must rely on software simulators for evaluating the scenarios from the designer point-of-view.

Contrary to single objective optimization, multiobjective optimization is not restricted to find a unique solution of a given multiobjective problem, but a set of solutions known as the *Pareto optimal set*. For instance, taking as an example the problem we are dealing with, one solution can represent the best result concerning the number of reached stations, while another solution could be the best one concerning the makespan. These solutions are said to be *nondominated*. The result provided by a multiobjective optimization algorithm is then a set of nondominated solutions (the *Pareto optima*) which are collectively known as the *Pareto* front when plotted in the objective space. The mission of the decision maker is to choose the most adequate solution from the Pareto front.

This multiobjective problem of broadcasting in MANETs, which has been previously addressed with a cellular genetic algorithm (cMOGA) in [2], is now tackled with a state-of-the-art multiobjective scatter search algorithm called AbSS (Archive-based Scatter Search) [3]. Scatter search [4, 5, 6] has been successfully applied to a wide variety of optimization problems [5], but it has not been extended to deal with MOPs until recently [3, 7, 8, 9]. This metaheuristic technique starts from an initial set of diverse solutions from which a subset, known as the *reference set* (*RefSet*), is built by including both high quality solutions and highly diverse solutions. Then, an iterative procedure systematically combines the solutions in RefSet somehow for generating new (hopefully better) solutions that may be used for updating the reference set and even the initial population. After that, an iterative procedure is used to locate an optimal solution.

The contributions of this work are summarized in the following. Firstly, we solve the broadcasting problem on MANETs using a multiobjective scatter search, and compare the results with those obtained with cMOGA. Secondly, we are dealing in this work with a more realistic problem than the one faced in [2] because we are using an interesting real world scenario (a shopping mall) never tackled before.

The rest of the paper is structured as follows. In the next section, we detail the multiobjective problem of broadcasting in MANETs. Section 3 includes the description of the multiobjective scatter search algorithm. Metrics, parameterization, and results are presented in Sect. 4. Finally, conclusions and lines of future work are given in Sect. 5.

2 Problem Definition

The problem we study in this paper consists of, given an input MANET, determining the most adequate parameters for a broadcasting strategy in it. We first describe in Sect. 2.1 the target networks we have used. Section 2.2 is devoted to the presentation of DFCN, the broadcasting strategy to be tuned. Finally, the MOP we define for this work is presented in Sect. 2.3.

2.1 Metropolitan Mobile Ad-Hoc Networks

Metropolitan mobile ad-hoc networks are MANETs with some particular properties. Firstly, they have one or more areas where the node density is higher than the average. These points are called VHS, standing for *Virtual Hot Spots*, that can be statistically detected. A VHS may be, for example, a shopping center, an airport, or an office. Secondly, virtual hot spots do not remain active full time, i.e., they can appear and disappear from the network (e.g., supermarkets are open, roughly, from 9 a.m. to 9 p.m., and outside this period of time, the node density within the corresponding area is close to zero).

To deal with such kind of networks, we have to rely on software simulators. In this work we have used *Madhoc* [1], a metropolitan MANET simulator. It aims at providing a tool for simulating different level services based on different technologies on MANETs for different environments, ranging from open areas to metropolitan ones. In order to make more realistic the simulations, Madhoc has been endowed with an observation window such that only the devices located inside this window are taken into account for measurements. Hence, we allow the existence of a changing number of devices in the network as it happens in real MANETs. This recent feature of Madhoc is displayed in Fig. 1, where both an example of a metropolitan MANET (a) and the effects of introducing an observation window on it (b) are shown. We highlight as well a typical action of devices going in and leaving the window in the right part of the figure. In all the tests in this work, this observation window is 70% of the total simulation area. The main parameters of Madhoc used for defining the network characteristics are the following:

[1] http://www-lih.univ-lehavre.fr/∼hogie/madhoc/

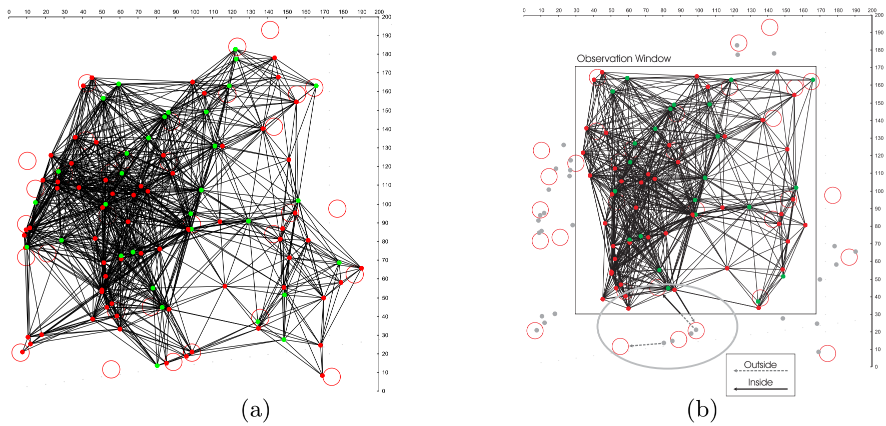

Fig. 1. (a) Metropolitan MANET, and (b) the effect of the observation window

***size*:** defines the network simulation area in terms of square meters.

***density*:** is the average density of nodes per square kilometer (i.e., the number of devices per square kilometer).

***environment*:** determines the mobility model for the stations and the radio wave propagation model. That is, this feature defines how the stations are moving as well as the area within which they are moving (open areas, buildings, streets, etc.), thus determining how radio waves are propagated.

2.2 Delayed Flooding with Cumulative Neighborhood

Broadcasting strategies in MANETs can be classified into four categories: simple flooding, probability-based methods, area-based methods, and neighbor-knowledge-based methods (a survey can be found in [10]). This categorization is based on the way that protocols select re-broadcasting stations.

Broadcasting protocols can also be classified depending on whether they deal with mobility or not. The vast majority of present protocols do not consider any active management of station mobility. The Delayed Flooding with Cumulative Neighborhood (DFCN) protocol belongs to the neighbor-knowledge-based class, and it features an active management of station mobility so it is able to make new broadcasting decisions on new neighbor discovery. 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.), DFCN requires the knowledge of 1-hop neighborhood, which can be obtained by using "hello" packets at a lower network layer. The set of neighbors of station 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 station maintains local information about all the 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 stations 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 preventing collisions. More precisely, when a station s emits a message m, all the stations 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 station 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 station 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 of the station for emission as a follow-up of one of the two previous events. The behavior resulting from 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 stations in the neighborhood of one another. When s_1 sends a packet to s_2, it attaches the set $N(s_1)$ to the packet. At reception, s_2 hence knows that each station in $N(s_1)$ has received the packet. The set of stations which have *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 stations 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 overload caused by a possible packet re-emission, this re-emission occurs only if the number of newly reached stations is greater than a given threshold. This threshold is a function of the number of stations in the neighborhood (the local network density) of the recipient station 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:

$$\text{Re-emit}\,(s_2,\ s_1) = \begin{cases} \text{true} & \text{h}\,(s_2,\ s_1)\ \geq\ \text{threshold}\,(|N\,(s_2)|) \\ \\ \text{false} & \text{otherwise}\ . \end{cases} \tag{1}$$

If the threshold is exceeded, the recipient station s_2 becomes an emitter after a random delay defined by RAD. 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:

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

where $safeDensity$ is the maximum safe density below which DFCN always rebroadcasts and *minGain* is the minimum gain for rebroadcasting, i.e., the ratio between the number of neighbors which have not received the message and the total number of neighbors.

Each time a station s gets a new neighbor, the RAD for all messages is set to zero and, therefore, 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. *proD* is used for avoiding massive packet rebroadcasting when a new station appears in highly dense areas, that is, avoiding network congestions on the proactive behavior.

2.3 MOP Definition: DFCNT

From the description of the previous section, the following DFCN 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] defines the RAD value (random delay for rebroadcasting in milliseconds). Both parameters take values in the interval $[0.0, 10.0]$ milliseconds.

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 parameters, i.e., a DFCN configuration, characterize the search space. Here, the objectives to be optimized are: minimizing the makespan (in seconds), maximizing the network coverage (percentage of devices having received the broadcasting message), and minimizing the bandwidth used (in number of transmissions). Thus, we have defined a triple objective MOP, which has been called DFCNT (standing for DFCN Tuning). For obtaining the values of these objective functions we have used Madhoc because it implements the DFCN broadcasting protocol. Then, our goal is to obtain the Pareto front of DFCNT (and the corresponding DFCN configurations) in terms of these three objectives.

3 Multiobjective Scatter Search

In this section, we first give a brief overview of the scatter search technique and, second, we describe the modifications on this standard scatter search for dealing with MOPs to explain our proposed AbSS.

3.1 Scatter Search

Most implementations of scatter search use the template proposed by Glover in [4]. As depicted in Fig. 2, this metaheuristic consists of five methods: diversification generation, improvement, reference set update, subset generation, and solution combination.

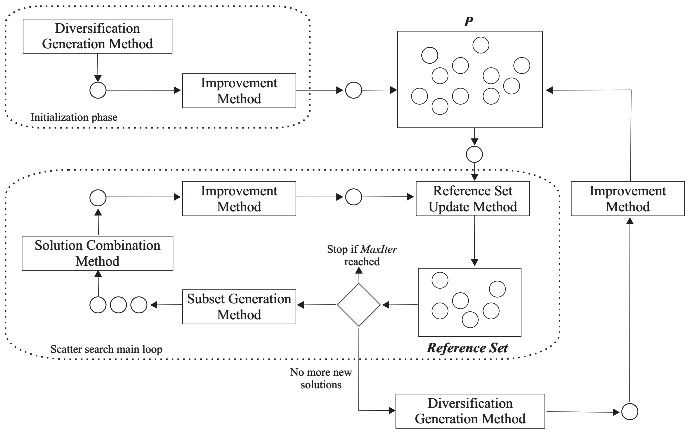

Fig. 2. Outline of the standard scatter search algorithm

The scatter search technique starts by creating an initial set of diverse individuals in the initialization phase. This phase consists of iteratively generating new solutions by invoking the diversification generation method; each solution is passed to the improvement method, which usually applies a local search procedure in an iterative manner, and the resulting individual is included into the initial set P. After the initial phase, the scatter search main loop starts.

The main loop begins building the reference set from the initial set by invoking the reference set update method. The reference set is a collection of both high quality solutions and diverse solutions that are used for generating new individuals. Solutions in this set are systematically grouped into subsets of two or more individuals by means of the subset generation method. In the next step, solutions in each subset are combined to create a new individual, according to the solution combination method. Then, the improvement method is applied to every new individual. The final step consists of deciding whether the resulting solution is inserted into the reference set or not. This loop is executed until a termination condition is met (for example, a given number of iterations has been performed, or the subset generation method does not produce new subsets).

Optionally, there is a re-start process invoked when the subset generation method does not produce new subsets of solutions. The idea is to obtain a new initial set, which will now include the current individuals in the reference set. The rest of individuals is generated by using the diversification generation and improvement methods, as in the initial phase.

3.2 AbSS

AbSS (Archive-based Scatter Search) [3] is based on the aforementioned scatter search template and its application to solve bounded continuous single objective optimization problems [6]. It uses an external archive for storing nondominated solutions and combines ideas of three state-of-the-art evolutionary algorithms for solving MOPs. In concrete, the archive management follows the scheme of

PAES [11], but using the crowding distance of NSGA-II [12] as a niching measure instead of the PAES adaptive grid; additionally, the density estimation found in SPEA2 [13] is adopted for selecting the solutions from the initial set that will build the reference set. Once described the overall view of the technique, we now detail the five methods to engineer AbSS:

- **Diversification Generation Method:** Its goal is to generate an initial set P of diverse solutions. The method consists of dividing, for every new solution, the range of each variable into a number of subranges of equal size; then, each solution is created in two steps. Firstly, a subrange is randomly chosen, with the probability of selecting a subrange being inversely proportional to its frequency count (the number of times the subrange has been previously selected); secondly, a value is uniformly randomly generated within the selected range.
- **Improvement Method:** It is a local search method based on a mutation operator (Polynomial mutation [14]) and a Pareto dominance test. It operates by iteratively mutating an individual with the aim of improving it. Since we are dealing with MOPs, it may occur that the newly generated individual and the current one are nondominated each other (Pareto dominance test). In this case, the original individual is inserted into the external archive and the mutated individual becomes the new current one.
- **Reference Set Update Method:** A similar issue rises when building the RefSet in this method, i.e., how to pick up the best among a set of nondominated solutions. RefSet is composed of two subsets, $RefSet_1$ and $RefSet_2$ so that the first one contains the best quality solutions in the initial set of solutions, while the second subset should be filled with solutions promoting diversity. While $RefSet_2$ is constructed by choosing those individuals whose minimum Euclidean distance to the reference set is the highest, $RefSet_1$ is built by using the concepts of strength raw fitness and a density estimation of SPEA2 [13] when choosing the best individuals.
- **Subset Generation Method:** It generates all pairwise combinations of solutions in RefSet_1 and, separately, in RefSet_2.
- **Solution Combination Method:** The simulated binary crossover (SBX) [14] is used for combining solutions in AbSS.

4 Experiments

This section is devoted to presenting the experiments performed for this work. We first describe the metrics used for measuring the performance of the resulting Pareto fronts. Next, the parameterization of AbSS and Madhoc is detailed. Finally, we show the results for DFCNT and compare them against cMOGA [2].

4.1 Metrics

We have used three metrics for assessing the performance of both AbSS and cMOGA: the number of Pareto optima that the optimizers are able to find, Set

Coverage [15] which allows two algorithms to be compared in terms of Pareto dominance, and Hypervolume [16] which measures both convergence and diversity at the same time in the resulting Pareto fronts. They are defined as:

– **Number of Pareto optima:** Given that DFCNT is a difficult problem, finding a high number of nondominated solutions could be itself a hard challenge for any multiobjective optimizer. In this sense, the number of Pareto optima can be considered as a measure of the ability of the algorithm for exploring difficult search spaces defined by hard MOPs like DFCNT.
– **Set Coverage:** The set coverage metric $\mathcal{C}(A, B)$ calculates the proportion of solutions in B which are dominated by solutions of A: $\mathcal{C}(A, B) = \frac{|\{b \in B \mid \exists a \in A : a \preceq b\}|}{|B|}$.

  A metric value $\mathcal{C}(A, B) = 1$ means that all members of B are dominated by A, whereas $\mathcal{C}(A, B) = 0$ means that no member of B is dominated by A. This way, the larger the $\mathcal{C}(A, B)$, the better the Pareto front A with respect to B. Since the dominance operator is not symmetric, $\mathcal{C}(A, B)$ is not necessarily equal to $1 - \mathcal{C}(B, A)$, and both $\mathcal{C}(A, B)$ and $\mathcal{C}(B, A)$ have to be computed for understanding how many solutions of A are covered by B and vice versa.
– **Hypervolume:** This metric calculates the volume (in the objective space) covered by members of a nondominated set of solutions Q. Let v_i be the volume enclosed by solution $i \in Q$. Then, a union of all hypercubes is found and its hypervolume (HV) is calculated: $HV = volume\left(\bigcup_{i=1}^{|Q|} v_i\right)$.

  Algorithms with larger values of HV are desirable. Since this metric is not free from arbitrary scaling of objectives, we have evaluated the metric by using normalized objective function values.

4.2 Parameterization

As we stated in Sect. 2.1, the behavior of Madhoc has been defined based on three parameters mainly: the size of the simulation area, the density of mobile stations, and the type of environment. For our experiments, we have used a simulation area of 40,000 square meters, a density of 2,000 stations per square kilometer, and, from the available environments of Madhoc, the mall environment has been used. This environment is intended to model a commercial shopping center, in which stores are usually located together one each other in corridors. People go from one store to another by these corridors, occasionally stopping for looking at some shopwindows. Both the mobility of devices and their signal propagation are restricted due to the walls of the building. A metropolitan MANET with such a configuration has been shown in Fig. 1. Due to the stochastic nature of Madhoc, five simulations (i.e., five different network instances) per function evaluation have been performed so that the fitness values of the functions are computed as the average resulting values of these five different network instances.

The configuration used for cMOGA is the same as that used in [2]: a population of 100 individuals arranged in a 10×10 square toroidal grid, the neighborhood is NEWS, binary tournament selection, simulated binary crossover (SBX)

Table 1. Performance metrics for AbSS and cMOGA when solving DFCNT

Metric	AbSS		cMOGA		t-test
	average	std	average	std	
Number of Pareto Optima	98.7586	2.8119	98.1053	2.9000	–
Set Coverage	0.9865	0.0103	0.9793	0.0076	+
Hypervolume	0.8989	0.0695	0.8199	0.0854	+

with $p_c = 1.0$, polynomial mutation ($p_m = 1.0/L$, $L =$ individual length), archive size of 100 individuals, and the adaptive grid of PAES [11] has been used as crowding method (see [2] for further the details). Regarding AbSS, we have utilized the parameterization proposed in [3]: external archive maximum size of 100 nondominated solutions, the size of the initial set P is 20, the number of iterations in the improvement method is 5 (polynomial mutation with a distribution index equal to 10), SBX crossover (solution combination method) also with a distribution index equal to 10, and the size of $RefSet_1$ and $RefSet_2$ as well is 10. Both cMOGA and AbSS stop when 25,000 function evaluations have been computed. It is important to note that 25,000 evals $\times$ 5 simulations/eval means that DFCN has been optimized over 125,000 different network instances.

4.3 Results

Let us now begin with the analysis of the results, which are presented in Table 1. Since both AbSS and cMOGA are stochastic algorithms and we want to provide the results with statistical confidence, 30 independent runs of each multiobjective optimizer have been performed, as well as t-tests at 95% of significance level (last column of Table 1). The t-test assesses whether the means of two samples are statistically different from each other.

If we consider that the two algorithms are configured for obtaining 100 nondominated solutions at most (maximum archive size), values shown in Table 1 point out that most executions of the optimizers fill up the whole archive. Though AbSS returns a slightly higher number of Pareto optima on average than cMOGA does, the difference is negligible and no statistical confidence exists ("–" symbol in t-test column), thus showing that both optimizers have a similar ability for exploring the search space of DFCNT.

As regards to the Set Coverage metric, we want to clarify that results shown in column "AbSS" correspond to $\mathcal{C}(AbSS, cMOGA)$ whereas those presented in column "cMOGA" are $\mathcal{C}(cMOGA, AbSS)$. As it can be seen in Table 1, AbSS gets larger values for this metric than cMOGA and there exists statistical confidence for this claim (see "+" symbol in the last column). This fact points out that AbSS can find solutions that dominate more solutions of cMOGA than vice versa. However, Set Coverage values are similar in both the cases, what indicates that each algorithm computes high quality solutions that dominate most solutions of the other, but those high quality solutions are in turn nondominated.

Last row in Table 1 presents the results of the Hypervolume metric. They clearly show now that AbSS overcomes cMOGA when considering at the same

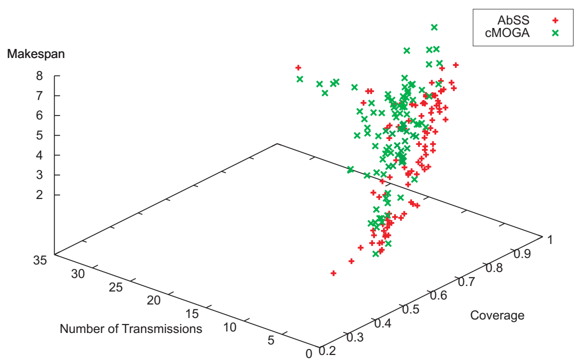

Fig. 3. Two DFCNT fronts from both AbSS and cMOGA

time both convergence and diversity in the resulting Pareto fronts (all this supported with statistical confidence). Since the Set Coverage metric showed that both optimizers were similar in terms of convergence, we can conclude that AbSS is reaching this Hypervolume value because of the diversity in the found Pareto front. That is, the set of nondominated solutions computed by AbSS covers a larger region of the objective space, what is an important feature for actual designs of MANETs. We show an example Pareto front that capture the previous claims in Fig. 3. Regarding coverage, the AbSS front ("+" symbols) is behind (on the right) cMOGA solutions ("×" symbols). With respect to diversity, it also can be seen that there are nondominated solutions from AbSS that reach DFCN configurations where message coverage is around 40% of the stations while cMOGA is not able to get solutions in this region of the objective space. Therefore, using AbSS provides the network designer (decision maker) with a wider set of DFCN parameter settings which ranges from configurations that get a high coverage in a short makespan but using a high bandwidth to those cheap solutions in terms of time and bandwidth being suitable if coverage is not a hard constraint in the network.

5 Conclusions and Future Work

This paper investigated the usage of AbSS, a multiobjective scatter search method, for optimally tuning the DFCN broadcasting strategy for MANETs. The multiobjective problem to be solved is called DFCNT and has three goals: minimizing makespan, maximizing network coverage, and minimizing the network usage. DFCNT has been previously tackled with a cellular multiobjective genetic algorithm called cMOGA.

Three metrics have been used for comparing the optimizers: Number of Pareto optima, Set Coverage, and Hypervolume. Regarding the number of nondominated solutions found, AbSS got a slightly higher number of configurations for DFCN on average than cMOGA, but differences are negligible. Regarding Set Coverage and Hypervolume, resulting values from the metrics claim that solutions from the scatter search approach dominated those obtained with cMOGA (convergence) as well as covered a larger region of the objective space (diversity). From these results, a clear conclusion can be drawn: AbSS is a promising approach for solving DFCNT with advantages over the existing one.

As a future work, we plan to perform more in depth analysis on using AbSS for solving real world MOPs. On the one hand, we also intend to use different scenarios where DFCN has to be tuned and, on the other hand, enlarge the simulation area to a still larger metropolitan network for large cities.

References

1. Hogie, L., Guinand, F., Bouvry, P.: A Heuristic for Efficient Broadcasting in the Metropolitan Ad Hoc Network. In: 8th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems. (2004) 727–733
2. Alba, E., Dorronsoro, B., Luna, F., Nebro, A., Bouvry, P.: A Cellular Multi-Objective Genetic Algorithm for Optimal Broadcasting Strategy in Metropolitan MANETs. In: IPDPS-NIDISC'05. (2005) 192
3. Nebro, A.J., Luna, F., Dorronsoro, B., Alba, E., Beham, A.: AbSS: An Archive-based Scatter Search Algorithm for Multiobjective Optimization. European Journal of Operational Research (2005) Submitted
4. Glover, F.: A Template for Scatter Search and Path Relinking. In: Third European Conf. on Artificial Evolution. Volume 1363 of LNCS. Springer Verlag (1997) 3–54
5. Glover, F., Laguna, M., Martí, R.: Fundamentals of Scatter Search and Path Relinking. Control and Cybernetics **29** (2000) 653–684
6. Glover, F., Laguna, M., Martí, R.: Scatter Search. In: Advances in Evolutionary Computing: Theory and Applications. Springer, New York (2003) 519–539
7. Beausoleil, R.P.: MOSS: Multiobjective Scatter Search Applied to Nonlinear Multiple Criteria Optimization. Eu. J. of Operational Research **169** (2005) 426–449
8. da Silva, C.G., Clímaco, J., Figueira, J.: A Scatter Search Method for the Bi-Criteria Multi-Dimensional {0,1}-Knapsack Problem using Surrogate Relaxation. Journal of Mathematical Modelling and Algorithms **3** (2004) 183–208
9. Nebro, A.J., Luna, F., Alba, E.: New Ideas in Applying Scatter Search to Multiobjective Optimization. In: EMO 2005. LNCS 3410 (2005) 443–458
10. Williams, B., Camp, T.: Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks. In: Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC). (2002) 194–205
11. Knowles, J., Corne, D.: The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC. (1999) 9–105
12. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6** (2002) 182–197
13. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical report, Swiss Federal Inst. of Technology (2001)
14. Deb, K., Agrawal, B.: Simulated Binary Crossover for Continuous Search Space. Complex Systems **9** (1995) 115–148
15. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, Swiss Federal Institute of Technology (ETH) (1999)
16. Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms – A Comparative Study. In: PPSN V. (1998) 292–301