Multiobjective Algorithms to Optimize Broadcasting Parameters in Mobile Ad-hoc Networks

R. Pérez-Pérez ramon.perez@uc3m.es
C. Luque cristobal.luque@uc3m.es
A. Cervantes alejandro.cervantes@uc3m.es
P. Isasi pedro.isasi@uc3m.es

Department of Computer Science
Universidad Carlos III de Madrid
Avda. de la Universidad 30, 28911-Leganés, Madrid, Spain.

Abstract- A mobile adhoc network (MANETs) is a self-configuring network of mobile routers (and associated hosts). The routers tend to move randomly and organize themselves arbitrarily; thus, the network's wireless topology may change fast and unpredictably. Nowadays, these networks are having a great influence due to the fact that they can create networks without a specific infrastructure. In MANETs message broadcasting is critical to network existence and organization. The broadcasting strategy in MANETs can be optimized by defining a multiobjective problem whose inputs are the broadcasting algorithm's parameters and whose objectives are: reaching as many stations as possible, minimizing the network utilization, and reducing the makespan. The network can be simulated to obtain the expected response to a given set of parameters. In this paper, we face this multiobjective problem with two algorithms: Multiobjective Particle Swarm Optimization and ESN (Evolution Strategy with NSGAII). Both algorithms are able to find an accurate approximation to the Paretooptimal front that is the solution of the problem. ESN improves the results of MOPSO in terms of the set coverage and hypervolume metrics used for comparison.

1 Introduction

With recent performance advancements in computer and wireless communications technologies, advanced mobile wireless computing is expected to see increasingly widespread use and application, much of which will involve the use of the Internet Protocol (IP) suite [1]. The objective of mobile ad hoc networking, called MANETs, is to support robust and efficient operation in mobile wireless networks by incorporating routing functionality into mobile nodes.

According IETF (Internet Engineering Task Force), the MANETs are autoconfiguration structures which *stations* (also called *terminals*) are temporally connected without a pre-existing infrastructure or a centralized administration [2]. MANETs are wireless network where one station can communicate directly with the other stations.

In the last years, the MANETs have increased the interest, as path to the 3G networks: a) they allow to extend the coverage, b) they don't need new infrastructure. In this way, a terminal user can use another users' terminals as route to obtain multistep way towards the kernel of the network.

In order to allow the communication between stations that aren't directly connected, the networks establish routing protocols ad hoc as such LMR (Land Mobile Radio) [3], Link Reversal [4], DSR (Dynamic Source Routing) [5], OLRS (Optimized Link State Routing) [6], DSDV (Dynamic Destination Sequenced Distance-Vector Routing) [7], TORA (Temporally-Ordered Routing Algorithm) [8], or DFCN (Delayed Flooding with Cumulative Neighborhood) [9] to search routes towards a receptor. Hence, having a well-tuned broadcasting strategy will produce a major impact in network performance.

In this paper we consider 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 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 [9], and the target networks are metropolitan MANETs. Since manipulating such networks is difficult, we must rely on software simulations for evaluating the scenarios from the designer point of view.

The DFCN defines five parameters (minGain, lowerBoundRAD, upperBoundRAD, proD and safeDensity) to determine the objective values in the broadcasting strategy (minimizing the makespan, maximizing the network coverage, and minimizing the bandwidth used).

In multiobjective optimization (MO) there is not a single optimum, since several objectives must be considered. Thus, there are several solutions which are not comparable, usually referred to as Pareto-optimal solutions.

A multiobjective minimization problem with n variables and m objectives can be formulated, without loss of generality, as

$$min \quad y = f(\bar{x}) = min(f_1(\bar{x}), f_2(\bar{x}), \ldots, f_m(\bar{x})) \qquad (1)$$

, where

$$\bar{x} = (x_1, \ldots, x_n) \text{ and } y = (y_1, \ldots, y_m) \qquad (2)$$

In most cases, the objective functions are in conflicts, so that is not possible to reduce any of the objective functions without increasing at least one of the other objective

1-4244-1340-0/07/$25.00

functions. This is known as the concept of pareto optimality [10] [11].

Definition 1 (Pareto Optimal): A point $\bar{x} \in X$ is **Pareto optimal** if for every $\bar{x^*} \in X$ and $I = \{1, ..., m\}$ either

$$\forall_{i \in I}(f_i(\bar{x}) = f_i(\bar{x^*})) \quad (3)$$

or, there is at least one $i \in I$ such that

$$f_i(\bar{x}) > f_i(\bar{x^*}) \quad (4)$$

In other words, this definition means that $\bar{x^*}$ is Pareto optimal if there exists no feasible vector $\bar{x}$ that decrease some criterion without increment in at least one other criterion.

Definition 2 (Pareto Dominance): A vector $\bar{u} = (u_1, \ldots, u_m)$ is said to dominate $\bar{v} = (v_1, \ldots, v_m)$ (denoted by $\bar{u} \preccurlyeq \bar{v}$) if and only if $\bar{u}$ is partially less that $\bar{v}$, i.e.,

$$\forall i \in \{1, \ldots, m\}, u_i \leqslant v_i \wedge \exists i \in \{1, \ldots, m\} : u_i < v_i. \quad (5)$$

A solution a is said to be non-dominated regarding a set $X' \subseteq X$ if and only if, there is no solution in X' which dominates a. The solution a is Pareto-optimal if and only if a is non-dominated regarding X.

The set of all non-dominated solutions constitutes the Pareto optimal set. Therefore, our goal is to find the best Pareto front and near to Pareto optimal.

In this paper, we study the multiobjective problem of MANETs and broadcasting strategy with different techniques. Those algorithms are: Particular Swarm optimization (PSO) [12] and a new algorithm called ESN (Evolutionary Strategy with NSGA-II philosophy) inspired in evolutionary strategy and adapted for multiobjective problems.

In the next section, we show MANETs environments and broadcasting strategy. Moreover, we define the multiobjective problem to optimize the broadcasting strategy. We also introduce the metrics to evaluate the results. In Section 3 and 4 we describe the PSO and ESN algorithms. In the next section, results obtained are shown, and finally, the conclusions and future works are given in section 6.

2 Problem Definition

The different environments in MANETs and broadcasting strategy define a great investigation area. To continue, we study the Metropolitan MANETs and the broadcasting DFCN we used to optimize the multiobjective problem.

2.1 MANETs

Essentially, though a MANET is a spontaneous, self configuring network of devices that connect wireless. Different types of environments in mobile ad-hoc networks can be find:

1. Mall environment: The mall environment aims at modelling a commercial shopping center. The stations are distributed between the center and in corridors. There are zones with more density than others. So, stations have mobility in the areas of the mall.
2. Metropolitan environment: it simulates a metropolitan area. The places are connected with street. The pedestrians and vehicle are in continuous movement by the places and streets. In this scene, factors as buildings and large distance reduce the signal strength.
3. HighWay environment: it aims at modelling a MANETs out of the cities. We may have a low density of stations per square kilometer, moving all them quickly. So, in this context, there aren't obstacles to attenuate the signal strength.

This paper focused on Metropolitan environment. We can consider Mall environment as a subcase of the metropolitan environment where places are stores and streets are corridors. To deal with such kind of networks, we have to work on software simulators. In this work we have used *Madhoc* [13], a Metropolitan MANET simulator. It aims at providing a tool for simulating different level services based on distinct technologies on MANETs for different environments. The main parameters of Madhoc used for defining the network characteristics are the following:

- Size: defines the area in terms of square meters.
- Density: is the density of nodes per square kilometers.
- Environment: determines the mobility model for the stations and the radio wave propagation model.

2.2 Broadcasting strategy

A broadcasting operation consist of the emission of a message from one station to a set of other stations in the network. Generally, broadcasting strategy is presented as a building block for other network layer protocols [14]. Moreover, it is generally implicitly supposed that the topology is connected. However, the Metropolitan MANETs is quite different because the topology may change quickly and in an unpredictable way.

The Delayed Flooding with Cumulative Neighborhood (DFCN) algorithm is specially designed as an algorithm for message broadcasting on metropolitan ad hoc networks. Enabling the propagation of information when the topology of the network is made of changing sets of ad hoc networks that may merge and disjoin dynamically during the operation.

The DFCN is based on 1-hop neighborhood information. Its behavior is twofold: stations carry out the broadcasting strategy both on message reception and on notification of a new connection [9].

In DFCN there are three different tasks [15]:

1. New message reception (*reactive* behavior).
2. Detection of a new neighbor (*proactive* behavior).
3. The decision making of the station for emission as a follow-up of one of the two previous events (new message, new neighbor).

When a station s_1 sends a packet to s_2, it attaches the set $N(s_1)$ (neighbors of s_1) to the packet. At reception, s_2 hence knows that each station in $N(s_1)$ has received the packet. Furthermore, s_2 knows the possible stations not yet received the packet is then $N(s_2) - N(s_1)$.

In order to minimize the network overload caused by a possible packet reemission. This reemission happens only if the number of newly reached stations is greater than a given threshold.

If the threshold is exceeded, the station s_2 sends the packet after a random delay defined by RAD. The threshold function depends on the size of the neighborhood p, as given by:

$$threshold(p) = \begin{cases} 1, & p \leqslant safeDensity; \\ minGain * p, & \text{otherwise}; \end{cases} \quad (6)$$

where $safeDensity$ is the maximum safe density below which DFCN always rebroadcasts and $minGain$ is the minimum gain for broadcasting.

In summarize, the DFCN algorithm has a set of parameters that influence the behavior of the network:

1. **minGain** is the minimum gain for rebroadcasting. It ranges from 0.0 to 1.0.
2. **[lowerBoundRAD, upperBoundRAD]** defines the RAD value (random delay for rebroadcasting in milliseconds). It ranges from 0.0 to 10.0 milliseconds.
3. **proD** is the maximal density, ($proD \in [0, 100]$) for which it is still needed using proactive behavior for complementing the reactive behavior.
4. **safeDensity** defines a maximum safe density of the threshold which ranges from 0 to 100 devices.

Those parameters characterize the search space. Here, the objectives to be optimized are: minimizing the makespan (in seconds), maximizing the network coverage, and minimizing the bandwidth used. Thus, the goal is to obtain the Pareto front to optimize these three objectives. We use Madhoc simulation [13] to obtain the objective values.

2.3 Metrics

The solutions resulting from different executions of the multiobjective algorithms must be compared using quantitative metrics that measure the success of the algorithms towards the MO problem objectives: distance to the "true" Pareto-optimal front, distribution of solutions over the obtained front, and spread.

It is generally admitted that there is no single metric that can be used to evaluate those objectives simultaneously; this is specially true when the best Pareto-optimal front is not known. A detailed description of available metrics can be found in [16].

We have chosen the following metrics to compare solutions (fronts) obtained by the algorithms:

- **Set Coverage (SC)**. Coverage of a set of points A over a set of points B ($SC(A, B)$) is defined as the fraction of the points in set B that are weakly dominated by a point in set A. This measure has to be calculated in both directions (A vs. B and B vs. A) because $SC(A, B) \neq 1 - SC(B, A)$.
- **Hypervolume (HV)**. This metric, also called size of the dominated space, is the volume enclosed by the union of the of the points in the set. The volume dominated by any point is calculated as the volume of the hypercube defined by each point. The value of this metric is usually calculated after normalization of the points in the solutions.

Set Coverage is a measure of how a set dominates the other in terms of number of points. However, it does not take into account the actual distance between the points in both sets; that is, for how much a point in a set dominates the rest. It requires two sets of points (two fronts) for comparison, so it cannot be used to assign a performance measure to a single front except if the "true" Pareto-optimal front is known and used as reference.

Hypervolume assigns a quantitative value to a given set (where greater values mean better performance) but it cannot be used to derive dominance of a set over the other and its actual value depends on the normalization used.

3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) [12] has been successfully adapted to multiobjective optimization by several authors. The most successful versions are the ones that modify PSO to include concepts from NSGA-II [17] (as in [18]), or PAES [19] (in [11]); both versions show very competitive results over benchmark functions, surpassing the original evolutionary methods (NCSA-II and PAES). There are also different approaches to the problem such as [20, 21, 22].

For the MANETS optimization problem we have used the basic version of MOPSO [11].

The basic PSO uses a real-valued multidimensional space as search space, and evolves the position of each particle in that space using (7) and (8).

$$v_{id}^{t+1} = \chi(w \cdot v_{id}^{t} + c_1 \cdot \psi_1 \cdot (p_{id}^{t} - x_{id}^{t}) + c_2 \cdot \psi_2 \cdot (p_{gd}^{t} - x_{id}^{t})) \ . \quad (7)$$

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \ . \quad (8)$$

Where the meanings of symbols are: v_{id}^t, component in dimension d of the i^{th} particle velocity in iteration t; x_{id}^t, same for the particle position; c_1 ,c_2:, constant weight factors; p_i, best position achieved so far by particle i; p_{g}, best position found by the neighbors of particle i; ψ_1 ,ψ_2, Random factors in the [0,1] interval; w, inertia weight; and χ, constriction factor.

MOPSO combines PSO with the archiving strategy of PAES. Non-dominated solutions in the swarm are stored in an external repository and used to guide exploration. This repository is used both to store the problem's solution and to maintain the diversity of the population of particles.

The modified algorithm for MOPSO is:

1. Initialize the swarm
2. Evaluate each particle,store fitness as PBestFitness for the particle, and store position as PBestPosition for the particle.
3. Store non-dominated particles in the repository
4. While (number of iterations $< maxIterations$):
   - For each particle,
     (a) Select a leader from the repository
     (b) Update the particle velocity using the standard PSO equations (7,8), using the leader in place of p_g and PBestPosition as best previous position
     (c) Evaluate the particle
     (d) If current fitness dominates PBestFitness, update PBestFitness and PBestPosition, and check for inclusion in the repository

The mechanism of leader selection is as follows: the fitness space is divided in hypercubic sectors, and one of the positions in the repository is randomly selected as leader using a roulette algorithm that favors the sectors that are less populated. The number of sectors becomes a new parameter of the algorithm.

Selected leader replaces the "best neighbor" that was used for social attraction in (7). Therefore, particles are not attracted by the best particles in the swarm, but by leaders located in the areas of the search space where fewer non-dominated positions have been found.

The mechanism for inclusion in the repository ensures that it only stores non-dominated solutions. In order to introduce a new particle position in the repository, it has to be non-dominated by any solution in the repository. Then, if the new solution dominates some of the solutions in the repository, those solutions are removed. Thus, in [11] the maximum number of particles in the repository is fixed, so when this limit is reached, each insertion generates the removal of a particle from the most-populated sector of the repository. This limit has been removed for the current work.

Upon completion of the specified number of iterations, the set of solutions in the repository is reported as the Pareto front.

4 Evolucion Strategy + NSGA (ESN)

The NSGA-II+ES (ESN) algorithm is based on the hybridation of Evolution Strategies and NSGA-II. The algorithm uses the standard Evolution Strategies' steps [23], replacing the selection process by the NSGA-II [17] selection process.

The schemata of the algorithm is:

```
i=0
initialize population
while (i<G)
{
    produce new individuals
    evaluate population
}
```

This algorithm needs two constants to be defined: the number of generations (G) and the population size (μ). For the initialization, random individuals are spawned. The rest of the steps of the algorithm are explained above.

4.1 Produce New Individuals

The main difference between Evolution Strategies and Genetic Algorithms is that crossover operators are not used in ES, and each parent produces one offspring only by mutation.

Each individual in the population of μ generates an offspring by mutation. The mutation process implemented was the standard $(\mu + \lambda)$ process explained in [24], although in our case, $\lambda = \mu$. Being $x = (x_1, ..., x_n, \sigma)$ an individual (where x_i are their coordinates, and σ its variance), the mutation procedure that generates an offspring $x' = (x'_1, ..., x'_n, \sigma')$ can be mathematically described as:

$$\sigma' = \sigma e^{N(0,\Delta)} \tag{9}$$

$$x'_i = N(x_i, \sigma') \tag{10}$$

where $N(X,Y)$ represents a normal random variable with mean X and variance Y. Δ is a standard constant.

After all this process, the offspring is added to the population, that becomes 2μ size.

4.2 Evaluate Population

In this part of the algorithm we need to select the best μ individuals in the 2μ populations in order to be the parents on the next generation. The rest of the individuals will be deleted. In standard Evolution Strategies, the best individuals are selected by its fitness function. This idea can't be applied to multi- objective optimization, because the fitness function is a real function. Despite of that, a multi-objective selection process is required.

Each individual represents a solution of the problem The NSGA-II selection process sorts the solutions in subsets of the population (P) named fronts. These fronts (F_i) can be defined as:

F_1 = Non-dominated individuals of P.
F_2 = Non-dominated individuals of $P \setminus F_1$.
F_3 = Non-dominated individuals of $P \setminus (F_1 \cup F_2)$.
...
F_n = Non-dominated individuals of $P \setminus (F_1 \cup F_2 \cup ... \cup F_{n-1})$.

Solutions in the same front are sorted by a crowding distance (d). After this sorting process, we can define whether an individual is better than another:

$$a \text{ better than } b \iff a \in F_i, b \in F_j \text{ and } \begin{cases} i < j \\ or \\ i = j \text{ and } d(i) > d(j) \end{cases} \tag{11}$$

Therefore, we can select the μ best individuals in the population as parents for the next generation.

5 Experiments

5.1 Experimental Setting

We performed three series of 30 experiments, one using MOPSO and two using ESN with different population sizes. The total number of fitness evaluations was 25000 for each run of the algorithms.

MOPSO used a swarm size of 100 particles, and fixed PSO parameters: $w = 0.4$, $c_1 = c_2 = 2.0$, $X = 0.4$. We used 30 divisions of the adaptive grid as suggested in [2]. The number of non-dominated points in the solution was not limited. In our experiments the size of any solution was never greater than 200.

ESN-200 used a population size of 200, and ESN-500 a population size of 500. In both cases Δ was set to 0.7. In the ES experiments the number of non-dominated points in the solution never reached the size of the population and was never greater than 200.

5.2 Experiment Results

Table 1 shows the results of the Set Coverage metric between each of the pairs of algorithms. To compare two algorithms, SC was calcutated between each pair of individual runs and averaged. To find the value for $SC(A, B)$ find the corresponding row for algorithm A and the column for algorithm B.

Results show that the best algorithm is clearly ESN-200, followed by MOPSO and ESN-500.

	ESN-200	ESN-500	MOPSO
ESN-200		0.4813	0.4631
ESN-500	0.3959		0.4302
MOPSO	0.4063	0.4578	

Table 1: Set Coverage, Comparison between the three algorithms. For $C(A, B)$ the set A is the row, set B is the column

For the Hypervolume metric (HV), results are shown in Table 2. This metric was calculated scaling the points in each solution to the [0,1] interval and then calculating the average HV for that solution. The extreme values used in scaling were the same for the three algorithms to ensure a meaningful comparison.

The HV value for both ESN experiments is similar but always better than the HV value for MOPSO.

	HyperVolume
ESN-200	0.7169
ESN-500	0.7151
MOPSO	0.7067

Table 2: Hypervolume measure for each of the three algorithms.

In Fig. 1,Fig. 2 and Fig. 3 we show some typical solutions found by each of the three algorithms. From the plot it can be seen that the ESN algorithms are able to find solutions closer to the three extremes of the Pareto Front, while the MOPSO solutions are more concentrated toward the central part.

6 Conclusions

A mobile ad-hoc network (MANETs) is defined as a self-configuring network of mobile routers (and associated hosts). Due to the fact that network topology varies dynamically, the operations of broadcasting are of utter importance for the existence and the performance of the network and can benefit from parameter optimization on the broadcasting algorithm used in the network.

We have used *Madhoc* [13], a Metropolitan MANET simulator, to reproduce the behavior of a real MANET. A network is specified in Madhoc using its size, density and mobility model (environment). In this work we have used a predefined network structure with a Mall environment that simulates a commercial shopping center.

A broadcasting operation consists of the emission of a message from one station to a set of other stations in the network. An algorithm specially designed for message broadcasting on metropolitan ad hoc networks is Delayed Flooding with Cumulative Neighborhood (DFCN). This algorithm may be optimized by selecting proper values for five real-valued or integer-valued parameters: minGain, the interval limits for RAD ([lowerBoundRAD, upperBoundRAD]),proD and safeDensity.

We have defined the optimization problem for the broadcasting strategy in MANETs as a multiobjective problem where a single solution is defined as a set of values for those parameters, and the objectives that must be simultaneously achieved are:

- Maximum Coverage, reaching as many stations as possible.
- Minimum Bandwith usage, that is, network utilization.
- Minimum makespan.

This paper applies two different multiobjective optimization algorithms to find the Pareto-optimal front, that is, the set of non-dominated solutions for this problem. Both algorithms are compared using two standard metrics for evaluation of the solutions to multiobjective problems: Set Coverage (SC) and Hypervolume (HV). The two algorithms are:

- **Multiobjective Particle Swarm Optimization** (MOPSO) [11] is a version of Particle Swarm Optimization (PSO) that has been successfully applied to many problems.
- **Evolution Strategies with NSGA-II** (ESN) is proposed as an alternative that uses Evolution Strategies (ES) [17] mechanism but replaces the selection process by the NSGA-II [17] selection process.

Results show that both algorithms are able to find non-dominated fronts that approximate the "true" Pareto-

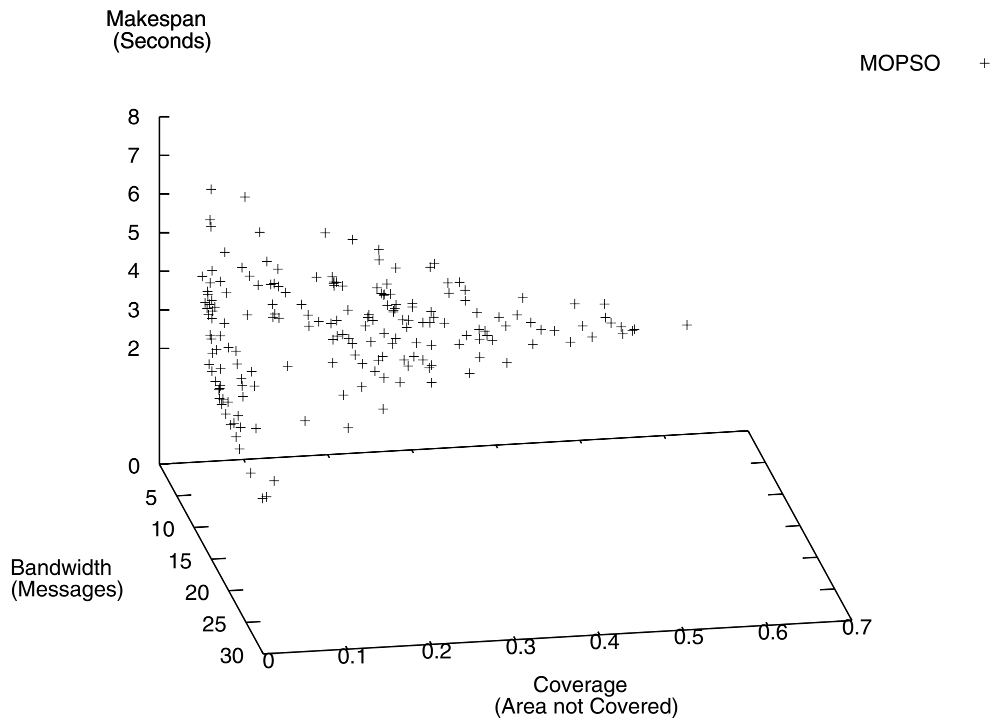

Figure 1: Sample MOPSO solution

optimal front with adecuate accuracy and distribution of the solutions.

The ESN experiments obtain very good values in the Hypervolume(HV) metric, due to the fact that solutions are well distributed over the whole surface of the front. MOPSO, however fails in finding solutions near the extremes of the Pareto front.

MOPSO provides some very good solutions near the central part of the Pareto front, achieving better Set Coverage (SC) metric over the ESN-500 experiment. However, ESN-200 is better than MOPSO in this metric too.

Results also show that an increase in the population size (ESN-500 versus ESN-200) does not produce better results in ESN, when the number of fitness evaluations is fixed, as solutions are quickly spread over the front (HV metric) but a certain number of generations is required to reach solutions that are non-dominated by the other algorithms (SC metric). ESN-200 seems to provide a very good compromise both in quality and spread of solutions.

Acknowledgment

This work has been partially funded by the Ministry of Science, Technology and FEDER under contract TIN2005-08818-C04-01 (the OPLINK project) and TIN2006-15041-C04-01.

Bibliography

[1] S. Corson and J. Macker. Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. Technical report, IETF Secretariat, January 1999. http://www.ietf.org/rfc/rfc2501.txt.

[2] IETF Secretariat. Ieft manet charter. Technical report, February 2007. http://www.ietf.org/html.charters/manet-charter.html.

[3] M. S. Corson and A. Ephremides. A distributed routing algoritm for mobile radio networks. In *IEEE Military Communications Conference*, 1989.

[4] E. Gafni and D. Bertsekas. Distributed algorithms for generating loop-free routes in networks with frequently changing topology. *IEEE Transactions on Communications*, 29(1):11–15, jan 1981.

[5] D. Johnson, Y. Hu, and D. Maltz. The dynamic source routing protocol for mobile ad hoc networks. Technical report, IETF Secretariat, February 2007. http://www.ietf.org/rfc/rfc4728.txt.

[6] T. Clausen and P. Jacquet. Optimized link state routing protocol. Technical report, IETF Secretariat, October 2003. http://www.ietf.org/rfc/rfc3626.txt.

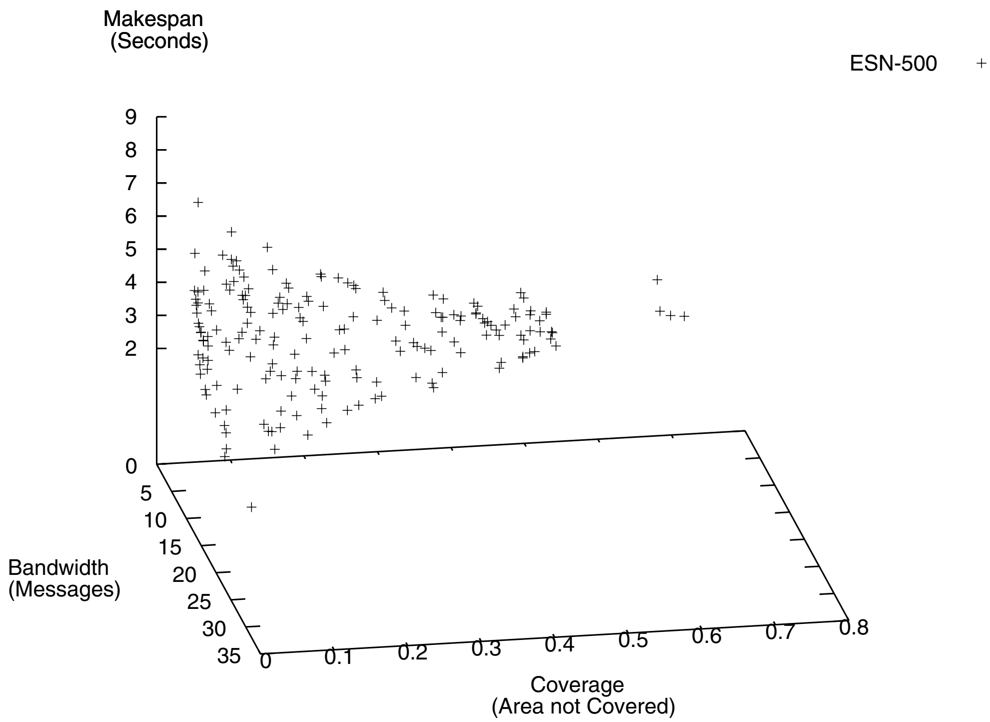

Figure 2: Sample ESN-500 solution

[7] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Conference on Communications Architectures Protocols and Applications*, pages 234–244, 1994.

[8] V.D. Park and M.S. Corson. A highly adaptative distributed routing algorithm for mobile wireless networks. In *IEEE INFOCOM*, 1997.

[9] L. Hogie, F. Guinand, and P. Bouvry. A heuristic for efficient broadcasting in the metropolitan ad hoc network. In *8th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems*, pages 727–733, 2004.

[10] D. Goldberg. *Genetic algorithms in Sarch, Optimization, and Machine Learnig*. Addision-Wesley Plublishing Company, New York, England, Bonn, Tokyo, 1989.

[11] C. A. Coello, G. Toscano, and M . Salazar. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8(3):256–279, jun 2004.

[12] J. Kennedy, R.C. Eberhart, and Y. Shi. *Swarm intelligence*. Morgan Kaufmann Publishers, San Francisco, 2001.

[13] L. Hogie, F. Guinand, and P. Bouvry. The madhoc metropolitan adhoc network simulator. Technical report, Universite du Luxembourg and Universite du Havre, France, mar 2006. http://wwwlih.univ-lehavre.fr/hogie/madhoc/.

[14] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, pages 194–205, 2002.

[15] F. Luna, A.J. Nebro, B. Dorronsoro, E. Alba, P. Bouvry, and L. Hogie. Optimal broadcasting in metropolitan manets using multiobjective scatter search. In *EvoWorkshops*, pages 255–266, 2006.

[16] E. Zitzler and L. Thiele. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Technical Report 43, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 1998.

[17] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[18] Xiaodong Li. A non-dominated sorting particle swarm optimizer for multiobjective optimization. In *Pro-*

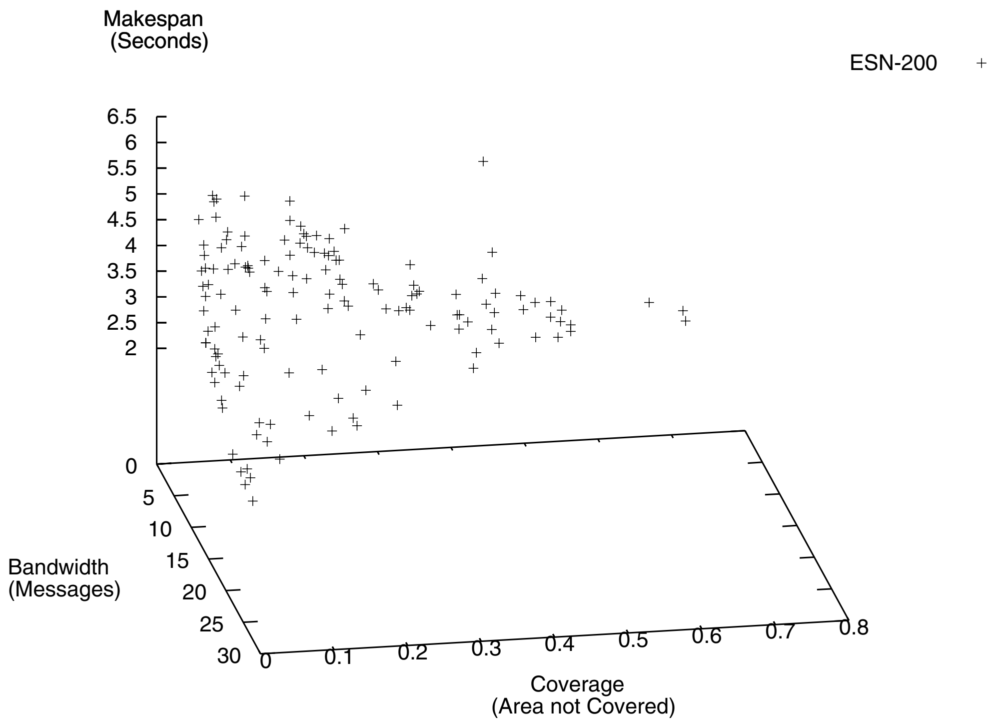

Figure 3: Sample ESN-200 solution

ceedings of Genetic and Evolutionary Computation (GECCO) 2003*, volume 2723/2003, pages 37–48, 2003.

[19] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evol. Comput.*, 8(2):149–172, 2000.

[20] J. Fieldsend and S. Singh. A multi-objective algorithm based upon particle swarm optimisation, an efficient data structure and turbulence. In *The 00 U.K. Workshop on Computational Intelligence*, volume 2723/2003, pages 34–44, 2002.

[21] X. Hu and R.C. Eberhart. Multiobjective optimization using dynamic neighborhood particle swarm optimisation. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 1677–16, 2002.

[22] X. Li. Better spread and convergence: Particle swarm multiobjective optimization using the maximin fitness function. In *Proceedings of Genetic and Evolutionary Computation (GECCO) 2004*, pages 117–128, 2004.

[23] T. Bäck and H.P. Schwefel. Evolutionary algorithms: Some very old strategies for optimization and adaptation. new computing techniques in physics research ii. In *Proc. Second Int'l Workshop Software Engineering, Artificial Intelligence, and Expert Systems for High Energy and Nuclear Physics*, pages 247–254, 1992.

[24] T. Bäck, G. Rüdolph, and H.P. Schwefel. A survey of evolution strategies. In *Proceedings of the 4th International Conference on Genetic Algorithms*, pages 2–9, 1991.