Using Omnidirectional BTS and Different Evolutionary Approaches to Solve the RND Problem

Miguel A. Vega-Rodríguez[1], Juan A. Gómez-Pulido[1], Enrique Alba[2], David Vega-Pérez[1], Silvio Priem-Mendes[3], Guillermo Molina[2]
[1] Dep. Techn. of Computers and Communications, Univ. of Extremadura, Caceres, Spain
{mavega, jangomez}@unex.es, dvperez@indra.es
[2] Dep. of Computer Science, Univ. of Malaga, Malaga, Spain
{eat, guillermo}@lcc.uma.es
[3] Polytechnic Institute of Leiria, School of Technology and Management, Leiria, Portugal
smendes@estg.ipleiria.pt

Abstract. RND (Radio Network Design) is an important problem in mobile telecommunications (for example in mobile/cellular telephony), being also relevant in the rising area of sensor networks. This problem consists in covering a certain geographical area by using the smallest number of radio antennas achieving the biggest cover rate. To date, several radio antenna models have been used: square coverage antennas, omnidirectional antennas that cover a circular area, etc. In this work we use omnidirectional antennas. On the other hand, RND is an NP-hard problem; therefore its solution by means of evolutionary algorithms is appropriate. In this work we study different evolutionary approaches to tackle this problem. PBIL (Population-Based Incremental Learning) is based on genetic algorithms and competitive learning (typical in neural networks). DE (Differential Evolution) is a very simple population-based stochastic function minimizer used in a wide range of optimization problems, including multi-objective optimization. SA (Simulated Annealing) is a classic trajectory descent optimization technique. Finally, CHC is a particular class of evolutionary algorithm which does not use mutation and relies instead on incest prevention and disruptive crossover. Due to the complexity of such a large analysis including so many techniques, we have used not only sequential algorithms, but also grid computing with BOINC in order to execute thousands of experiments in only several days using around 100 computers.

Keywords: Omnidirectional BTS, RND, PBIL, DE, SA, CHC.

1 Introduction

The Radio Network Design problem is a kind of telecommunication network design problem. When a set of geographically-dispersed terminals needs to be covered by transmission antennas (also called base station transmitters or base transceiver stations -BTS-), a capital subject is to minimize the number and locations of those antennas while covering the largest possible area.

RND is an NP-hard problem; therefore its solution by means of evolutionary algorithms is appropriate. In this work we use several different evolutionary approaches in order to solve this problem: PBIL, DE, SA and CHC.

Finally, since our interest is not only studying these techniques, but also to open new research lines, we have applied grid computing for the realization of our experiments (not in an exhaustive manner for space constraints in this paper). In particular, we have used BOINC, a very interesting proposal for volunteer computing and desktop grid computing.

The rest of the paper is organized as follows: Section 2 briefly explains the RND problem with omnidirectional BTS. After that, in the following section we introduce the PBIL, DE, SA and CHC algorithms. Then, in section 4 we show the most interesting results of this work, including comparisons among the different studied techniques, finally leading to the conclusions and future work in the last section.

2 Radio Network Design with Omnidirectional BTS

The RND problem [1,2] consists in covering the largest area with a minimal set of transmitters. In order to mathematically define this problem, let us consider the set L of all potentially covered locations and the set M of all potential transmitter locations. Let G be the graph, $(M \cup L, E)$, where E is a set of edges such that each transmitter location is linked to the locations it covers. As the geographical area needs to be discretized, the potentially covered locations are taken from a grid, as shown in figure 1a. In our case, we focus on a 287×287 point grid representing an open-air flat area and we will be able to use a maximum of 349 available locations for placing antennas.

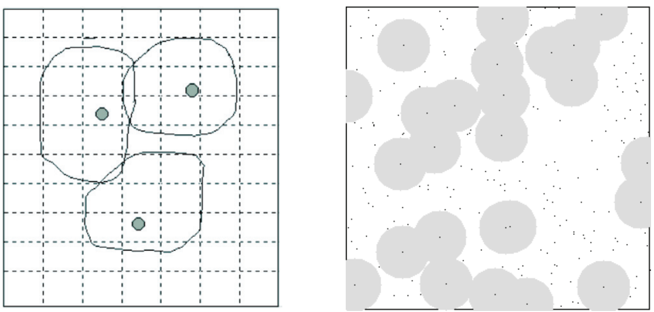

Fig. 1. (a) Associated covered cells of 3 potential transmitters. (b) Omnidirectional BTSs.

The goal of RND is to search for the minimum subset of transmitters that covers a maximum surface of an area, therefore, we are searching for a subset $M' \subseteq M$ such that $|M'|$ is minimum and such that $|Neighbours(M', E)|$ is maximum, where:

$$\text{Neighbours}(M', E) = \{u \in L \mid \exists v \in M', (u, v) \in E\} \quad (1)$$

To achieve this, we use the fitness function shown in equation 2 [3].

$$f(x) = \frac{CoverRate(x)^2}{NumberTransmittersUsed(x)} \tag{2}$$

An important constraint in this problem consists in determining the list of available locations for the antennas, because there are some places where the antennas can not be placed (public recreation areas, etc.). In our case, for the predefined set of available locations we selected the one included in [4]. This will make easy the comparisons among the different evolutionary techniques.

In our experiments we consider omnidirectional BTS (see figure 1b) and each transmitter has an associated coverage of a 22-sector-radius circle.

3 Different Evolutionary Approaches

In this section we briefly introduce the different evolutionary algorithms used for solving the RND problem.

3.1 PBIL

Population-Based Incremental Learning (PBIL) is a method that combines a genetic algorithm with competitive learning for function optimization. Instead of applying operators, PBIL infers a probability distribution from the present population and samples the new population from the inferred distribution [5,6].

3.2 DE

Differential Evolution (DE) is an algorithm that targets continuous optimization problems and has been used in the past with satisfactory results [7,8]. DE is a simple population-based stochastic function minimizer/maximizer, used in a wide range of optimization problems, including multi-objective optimization [9]. It has been modified in this research to work with discrete representations [10].

3.3 SA

Simulated annealing (SA) is a generic probabilistic meta-algorithm for the global optimization problem, namely locating a good approximation to the global optimum of a given function in a large search space. It was independently invented by S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi in 1983 [11], and by V. Černý in 1985 [12]. SA is a trajectory based optimization technique (i.e., only one tentative solution is manipulated in contrast with the rest of algorithms here, where a population of solutions is used). It is commonly found in industry and provides good results; therefore it constitutes an interesting method for comparison.

3.4 CHC

The fourth algorithm we propose for solving the RND problem is Eshelman's CHC [13], which stands for Cross-generational elitist selection, Heterogeneous recombination (by incest prevention), and Cataclysmic mutation. CHC is a kind of Evolutionary Algorithm (EA), where mutation is not used. As a mechanism for preventing convergence and maintaining diversity, CHC employs incest prevention [14] plus a special recombination procedure known as HUX, combined with a special restarting mechanism for adding diversity through mutation when stagnation is detected (cataclysmic mutation).

4 Results

In this section we present the most interesting results coming from using each of the evolutionary techniques we have proposed to solve the RND problem.

4.1 Results with Population-Based Incremental Learning (PBIL)

In this work we have first evaluated different configuration parameters for our PBIL algorithm in order to solve the RND problem since this application to this problem is relatively new. In particular, the parameters we can adjust in PBIL are: number of samples (the population size), mutation probability, mutation shift (intensity of the mutation that affects the probability vector), learning rate, and whether an elitist strategy is used or not (the best individual in the previous population is transferred to the current generation unaltered). As we can see, the total number of possible combinations is high, for this reason we have used the middleware system BOINC [15,16] (Berkeley Open Infrastructure for Network Computing) in order to perform massive computations/experiments in a parallel way for PBIL. In this way, we can do a deep survey about which are the best parameter values for solving the RND problem with PBIL, which is needed to guide future research after this first study. Researchers interested in learning more about our platform RND-BOINC (RND@home), or wanting to join in this project (volunteer donation of CPU cycles), can access it via the website http://arcoboinc.unex.es/rnd. At present, around 100 computers are available in this project, executing hundreds of experiments at the same time.

Table 1 shows the most important results using PBIL. As we can see, PBIL obtains a reasonable result (85% of coverage with only 62 transmitters) with a low computational effort (only 333,045 evaluations).

Table 1. Results with PBIL for solving the RND problem (omnidirectional BTSs).

	Config. best result
# Generations	2500
# Individuals	135
Mutation probability	0.02
Mutation shift	0.05
Learning rate	0.10
Elitism	NO

	Best result
Fitness function	116.95
# Transmitters	62
Coverage	85%
Execution time	2 h, 31', 9''
Execution on	Pentium IV – 2.8 GHz
# Evaluations	333,045

4.2 Results with Differential Evolution (DE)

In order to compare the results we have obtained from DE with other evolutionary approaches, it is necessary to apply the experiments on the same predefined set of BTS available locations. In the case of DE, the parameters we can adjust are: number of generations, population size, and the crossover function to use. Two crossover functions have been considered: FA and SA. Let be two set of locations (individuals), named A and B (the parents). Let be the S individual (the offspring) obtained from the application of the crossover function to A and B. FA function chooses the first half of A to build the first half of the offspring. The second half of the offspring is then built with the first half of B, but if a repeated location appears, successive locations of the second halves of B and A are taken. SA function chooses the second half of A to build the first half of the offspring, and the second half of the offspring is then built with the second half of B, but if a repeated location appears, successive locations of the first halves of B and A are taken.

Table 2 shows the most important results, considering the same instance of the problem used by the other evolutionary algorithms presented in this paper. The main conclusion is that the desired optimal coverage (100%) has not been reached. Also, it can be observed that DE algorithm is very fast.

Table 2. Results with DE for solving the RND problem (omnidirectional BTSs).

	Config. best result
# Generations	4000
# Individuals	2000
Crossover	FA

	Best result
Fitness function	100.54
# Transmitters	52
Coverage	72.30%
Execution time	5', 33''
Execution on	Pentium IV –1.7 GHz
# Evaluations	4,914

4.3 Results with Simulated Annealing (SA)

SA has been used on the same instance as the previous algorithms, in order for the obtained results to be comparable.

SA has only three parameters the programmer needs to tune (we use 1 as initial temperature): mutation probability, length of the Markov chain, temperature decay

(α). The length of the Markov chain and the temperature decay have been proven to work in the same manner, thus to be equivalent. Therefore, we decided to keep the first at a constant value of 50, and allow the tuning of the latter.

Table 3 shows the results. The tests have been performed in a 16 machine cluster named in dedicated mode, and the code has been developed using the MALLBA library [17]. This resource code is available at the web page http://neo.lcc.uma.es/mallba/easy-mallba/index.html.

SA has been able to solve the problem with 90% coverage (the best result until now), but presents a high computational effort with 4,152,235 evaluations.

Table 3. Results with SA for solving the RND problem (omnidirectional BTSs).

	Config. best result
# Evaluations	50,000,000
Mutation probability	0.005
Markov chain length	50
Temperature decay	0.99998
Initial temperature	1

	Best result
Fitness function	157.77
# Transmitters	52
Coverage	90%
Execution time	2 h, 16', 20''
Execution on	Pentium IV – 2.4 GHz
# Evaluations	4,152,235

4.4 Results with CHC

When using the CHC algorithm on the same instance that the previous methods, we have considered two parameters that can be tuned (the rest of configuration parameters are fixed to their typical values): population size and cataclysmic mutation probability. Table 4 shows the best configuration, and the results obtained. The tests have been performed in a 16 machine cluster named in dedicated mode, and the code has been developed using the MALLBA library [17]. This resource code is available at the web page http://neo.lcc.uma.es/mallba/easy-mallba/index.html.

During the tests we have concluded that the mutation probability (second parameter tuned) has little effect on the algorithm's performance, and can be kept at a value of 35% without any significant loss of efficiency.

CHC solved the problem with 90% coverage (a very good result), but it presents a quite high computational effort (though lower than SA).

Table 4. Results with CHC for solving the RND problem (omnidirectional BTSs).

	Config. best result
# Evaluations	50,000,000
# Individuals	8,000
Mutation probability	0.35
Incest distance	25% vector length
Crossover probability	0.8
Elitism	YES

	Best result
Fitness function	157.77
# Transmitters	52
Coverage	90%
Execution time	31', 17''
Execution on	Pentium IV – 2.4 GHz
# Evaluations	1,400,000

5 Conclusions and Future Work

In this paper we have solved the RND (Radio Network Design) problem with different evolutionary approaches. In particular, we have focused on the RND using omnidirectional BTS. Our aim was to solve the problem efficiently and at the same time research in the results of a wide spectrum of modern techniques. All these evolutionary algorithms have obtained reasonable results (from 72.30% to 90% of coverage). However, there are important differences.

If we look for the best fitness value (relation between coverage and number of antennas), the best alternatives are CHC and SA (see figure 2). Between them, CHC has proven to be better suited than SA to solve the problem instances, since it obtains an equivalent result, but the computational effort (#evaluations) is smaller. On the other hand, DE obtains the worst fitness value. However, DE is the evolutionary approach that needs the lowest number of evaluations (computational effort) in order to obtain a reasonable result. The difficulties in DE's accuracy may be originated by its intrinsic continuous nature, originating the stagnation of its progression [18].

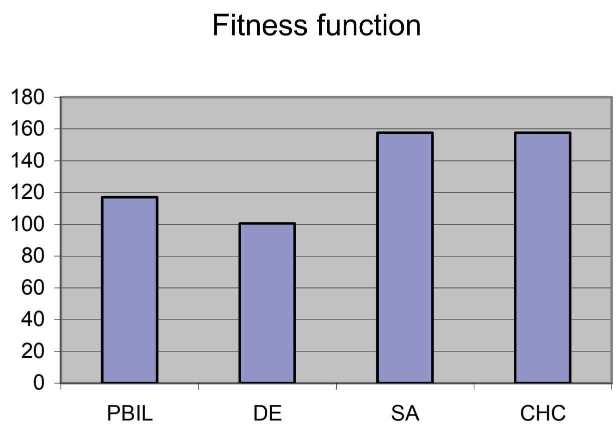

Fig. 2. Fitness value for each evolutionary approach.

Future work includes the study of more sophisticated evolutionary algorithms by communicating information among component parallel agents. Also, more complex and realistic problem instances will be tackled using the best performing techniques. The study of a real-size instance of the coverage of a city is in the current agenda.

Acknowledgments. This work has been partially funded by the Spanish Ministry of Education and Science under contract TIN2005-08818-C04-01 and TIN2005-08818-C04-03. Guillermo Molina is supported by grant AP2005-0914 from the Spanish government.

References

1. P. Calégari, F. Guidec, P. Kuonen, D. Kobler. Parallel Island-Based Genetic Algorithm for Radio Network Design. Journal of Parallel and Distributed Computing, 47(1): 86-90, November 1997.
2. P. Calégari, F. Guidec, P. Kuonen, F. Nielsen. Combinatorial Optimization Algorithms for Radio Network Planning. Theoretical Computer Science, 263(1): 235-265, July 2001.
3. E. Alba. Evolutionary Algorithms for Optimal Placement of Antennae in Radio Network Design. NIDISC'2004 Sixth International Workshop on Nature Inspired Distributed Computing, IEEE IPDPS, Santa Fe, USA, pp. 168-175, April 2004.
4. OPLINK: http://oplink.lcc.uma.es/problems/rnd.html, May 2007.
5. S. Baluja. Population-based Incremental Learning: A Method for Integrating Genetic Search based Function Optimization and Competitive Learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, June 1994.
6. S. Baluja, R. Caruana. Removing the Genetics from the Standard Genetic Algorithm. 12th Int. Conference on Machine Learning, San Mateo, CA, USA, pp. 38-46, May 1995.
7. K. Price, R. Storn. Differential Evolution – A Simple Evolution Strategy for Fast Optimization. Dr. Dobb's Journal, 22(4): 18–24, April 1997.
8. K. Price, R. Storn. DE website. http://www.ICSI.Berkeley.edu/~storn/code.html, May 2007.
9. H.A. Abbass, R. Sarker. The Pareto Differential Evolution Algorithm. Int. Journal on Artificial Intelligence Tools, 11(4): 531-552, 2002.
10. S. Mendes, J.A. Gómez, M.A. Vega, J.M. Sánchez. The Optimal Number and Locations of Base Station Transmitters in a Radio Network. 3rd Int. Workshop on Mathematical Techniques and Problems in Telecommunications, Leiria, Portugal, pp.17-20, Sept. 2006.
11. S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi. Optimization by Simulated Annealing. Science, 220(4598): 671-680, May 1983.
12. V. Cerny. A Thermodynamical Approach to the Travelling Salesman Problem: an Efficient Simulation Algorithm. Journal of Optimization Theory and Applications, 45: 41-51, 1985.
13. L.J. Eshelman. The CHC Adaptive Search Algorithm: How to Have Safe Search when Engaging in Nontraditional Genetic Recombination. Foundations of Genetic Algorithms, Morgan Kaufmann, pp. 265-283, 1991.
14. L.J. Eshelman, J.D. Schaffer. Preventing Premature Convergence in Genetic Algorithms by Preventing Incest. 4th Int. Conf. on Genetic Algorithms, CA, USA, pp. 115-122, 1991.
15. BOINC: http://boinc.berkeley.edu, May 2007.
16. D.P. Anderson. BOINC: A System for Public-Resource Computing and Storage. 5th IEEE/ACM Int. Workshop on Grid Computing, Pittsburgh, USA, pp. 365-372, November 2004.
17. E. Alba, F. Almeida, M. Blesa, C. Cotta, M. Díaz, I. Dorta, J. Gabarró, C. León, G. Luque, J. Petit, C. Rodríguez, A. Rojas, F. Xhafa. Efficient Parallel LAN/WAN Algorithms for Optimization: The MALLBA Project. Parallel Computing, 32 (5-6): 415-440, June 2006.
18. J. Lampinen, I. Zelinka. On Stagnation of the Differential Evolution Algorithm. 6th International Mendel Conference on Soft Computing, MENDEL 2000, Brno, Czech Republic, pp. 76-83, June 2000.