An Approach for Solving the Base Station Placement Problem using Particle Swarm Intelligence

Marcos Talau
CPGEI
Federal University of Technology
Dois Vizinhos, Brazil
e-mail: talau@users.sourceforge.net

Emilio C. G. Wille
CPGEI
Federal University of Technology
Curitiba, Brazil
e-mail: ewille@utfpr.edu.br

Heitor S. Lopes
CPGEI
Federal University of Technology
Curitiba, Brazil
e-mail: hslopes@utfpr.edu.br

***Abstract*— A large amount of studies deals with the base station placement problem (BSP), but few studies involve the selection of base stations to meet a set of users. This paper presents an approach for solving BSP problems in an indoor environment, aiming at meeting a set of users, with a minimum number of base stations, using a binary particle swarm optimization (PSO). A benchmark of four maps of increasing complexity was created for testing the system. Results of the binary PSO were compared with optimal solutions found by an exhaustive search algorithm. The computational results suggest that the PSO algorithm provides a quite efficient approach to obtain (near) optimal solutions with small computational effort.**

Keywords*— *Binary PSO; BSP; CDMA.

I. Introduction

With the growing use of wireless base stations in the indoor environment, to seek meeting the total number of users effectively, it becomes necessary to adopt some computational optimization method in order to provide a service of satisfactory quality with the minimal number of base stations.

The base station placement problem (henceforth, BSP) is characterized as the most important issue to solve in planning a wireless network [1]. This problem was considered *NP*-hard by many authors [1]–[4]. This fact suggests the use of metaheuristics for solving large instances of the BSP.

Only few works about the BSP considered the placement of base stations committed to meet the demands of users in a code division multiple access (CDMA) indoor environment [5] [6]. In these works, only one problem instance (map) was considered, the same for both. This paper presents a model for solving BSP problems in a CDMA indoor environment, concerned with quality of service for a set of users. We propose a model and a benchmark of four instances (maps) with increasing complexity. The instances of BSP were solved using the binary particle swarm optimization (PSO), from evolutionary computation. Solutions were validated with the application of the formulation proposed by [5]. The optimal solutions for this benchmark were found by means of an exhaustive search algorithm. The optimal results were compared with those of the PSO.

This paper is organized as follows. In Section II some related works are presented. Section III describes the variables and the formulation of the problem. In Section IV we present the model for solving the problem using PSO. Section V shows the benchmarks and the results obtained. Finally, Section VI concludes the paper and points to future directions of research.

II. Related Work

Research in [5] proposed a binary integer programming (BIP) formulation for solving the BSP problem. It aimed at finding an optimal configuration, in a CDMA system, using the branch-and-bound (B&B) method. The results were compared with a customized version of a genetic algorithm.

In [6] an algorithm to find the optimal configuration of base stations of a CDMA network in an indoor environment was proposed. The algorithm combines a heuristic technique with a brute force search. The heuristic search was used to find the minimum number of base stations, and the brute force searched for places for the base stations in predefined positions. The algorithm managed to reach the optimal solution in 10 runs, consuming an average time of one second of runtime, in a scenario with 12 base stations, 54 users (25 randomly selected) and with an area of 18.5m x 18.5m. We will show in Section V that the benchmarks proposed here are much more complex than the test scenario proposed by [6].

Several studies that used evolutionary computation techniques for the BSP were based on genetic algorithms, such as [1], [5], [7]. In these studies the genetic algorithm was found to be effective in solving the problem.

The PSO was used for this problem in [4]. Authors adapted the PSO specifically for BSP problems. Solutions considered both coverage and economy for the Pareto curve using the divided range multiobjective particle swarm optimization (DRMPSO). The results showed the efficiency of the method to find the optimum.

The work of [4] deals only with the problem of locating the best position (x, y) for the installation of the base stations. In the present paper, instead, the base stations are installed at fixed locations.

Most of the works cited consider only the placement of antennas [1], [4], [7]. In [5] [6] it was considered not only the base station placement, but also the location of the users. However, only one scenario (with twelve antennas and measured twenty-five users) was used for tests, running for ten times the branch-and-bound and genetic algorithm methods.

III. Problem Formulation

This section describes the base station placement problem as presented in [5]. Considering a CDMA indoor wireless network environment, we assume that there exists a finite number of potential base station sites (B) and a finite number of user locations requiring service (U). All users are identical and have a fixed known location. Each potential base station site has its own installation cost (in order to reflect the ease of installation and maintenance at the different sites, for instance). The loading of a base station corresponds to the number of users that are connected to that particular base station (with maximum capacity of L users).

The problem is defined as selecting the least cost set of base station sites from B potential sites, so as to provide service to U users, while ensuring that the signal-to-interference ratios on both the forward and reverse links exceed a predefined threshold. The complexity of a problem of an environment is defined by the number of B base stations plus the number of U users.

The following notation is necessary for developing a mathematical formulation for the BSP problem:

b – the index for potential base station sites.
u – the index for user locations.
l – the index for base station loading.
A_{bu} – the attenuation (pathloss) between b and u.
P_{t1} – the transmission signal strength at a base station when one user is being served.
P_{tar} – the signal strength received at the base station from one of its users under ideal conditions.
P_{min} – the minimum signal strength required to maintain adequate communications as received by a user.
P_{max} – the maximum transmission power output on a handset.
G_p – the processing gain of the system.
L_b – the loading level (number of users) at base station b.
F_b – the cost of installing a base station at potential site b.

The BSP problem is formulated using BIP, as in [5]. Assuming a base station can host up to L users, the decision variable Y_{bl} determines whether a base station is to be deployed at a site b with a loading of l users, and X_{bul} determines whether a link is to be established from potential base station site b to a user location u. Thus, the decision variables are X_{bul} = 1 (0) if a link is (is not) established between base station b and user location u, and there are a total of l users communicating with this base station; and Y_{bl}= 1 (0) if a base station is (is not) deployed at potential site b with a loading of l users.

The objective function, to be minimized, evaluates the total cost of a solution and it is given by Equation (1):

$$Z = \sum_{b=1}^{B}\sum_{l=1}^{L} F_b Y_{bl} \tag{1}$$

Accordingly, a feasible solution must satisfy the following constraints:

1) The signal received by each user is required to be Q_F times stronger than the sum of the interference on the forward link. Thus, for each mobile receiver location u:

$$\sum_{b=1}^{B}\sum_{l=1}^{L} G_p P_{t1} A_{bu} X_{bul} \geq Q_F \times \left(\sum_{b=1}^{B}\sum_{l=1}^{L} (l P_{t1} A_{bu} (Y_{bl} - X_{bul}) + (l-1) P_{t1} A_{bu} X_{bul} \right), \quad u = 1, \ldots, U \tag{2}$$

2) The signals arriving at each base station from each of their users are required to be Q_R times stronger than the sum of the interference on the reverse link. Thus, for each selected base station location b:

$$G_p P_{tar} \geq Q_R \times \left(\sum_{i=1}^{B}\sum_{u=1}^{U}\sum_{l=1}^{L} \left(\frac{A_{bu}}{A_{iu}} P_{tar} X_{bul} \right) - P_{tar} \right), \quad b = 1, \ldots, B \tag{3}$$

3) The signal strength for any chosen link, after accounting for the effects of propagation and system processing gain, is required to be above the receiver noise floor, P_{min}. Therefore:

$$X_{bul} = 0 \;\; \forall\, b = 1, \ldots, B, \;\; l = 1, \ldots, L, \;\; u = 1, \ldots, U : G_p P_{t1} A_{bu} \leq P_{min} \tag{4}$$

4) The transmitted signal strength from any user location must not exceed the maximum power output of a handset, P_{max}. Thus:

$$X_{bul} = 0 \;\; \forall\, b = 1, \ldots, B, \;\; l = 1, \ldots, L, \;\; u = 1, \ldots, U : \frac{P_{tar}}{A_{bu}} \geq P_{max} \tag{5}$$

5) Each mobile user is served by only one base station to ensure coverage for all the users. Therefore:

$$\sum_{b=1}^{B}\sum_{l=1}^{L} X_{bul} = 1, \;\; u = 1, \ldots, U \tag{6}$$

6) To ensure a valid solution, the loading in each base station must satisfy two conditions, namely, (i) only one loading level is chosen for each of the selected base stations; and (ii) the number of users communicating to the base station equals the loading level selected. Thus:

$$\sum_{u=1}^{U} X_{bul} = l(Y_{bl}), \;\; b = 1, \ldots, B, \;\; l = 1, \ldots, L \tag{7}$$

and

$$\sum_{l=1}^{L} Y_{bl} \leq 1, \;\; b = 1, ..., B \tag{8}$$

IV. Solution Methods

This section describes two solution methods that were used to solve the BSP problem.

A. Particle Swarm Optimization

The PSO is a population-based metaheuristic introduced around a decade and a half ago by Kennedy and Eberhart [8]. PSO belongs to a class of methods known as evolutionary computation and it is inspired in the emerging properties of collective behavior of some animals, such as bird flocking, bee swarming, and schooling fish. For instance, in bird flocking, the velocity of each element is dynamically adjusted according to the velocity of the other surrounding elements. The influence of a bird on another is sensed by keeping an average distance between each other.

At each instant of time (t), the position and velocity of particles are adjusted according to the position (X) and velocity (V) in the previous time ($t-1$), as follows:

$$\begin{aligned} V_i(t) = V_i(t-1) + \varphi_1.r_1.[X_{blp} - X_i(t-1)] + \\ \varphi_2.r_2.[X_{bgp} - X_i(t-1)] \end{aligned} \tag{9}$$

$$X_i(t) = X_i(t-1) + V_i(t) \tag{10}$$

where φ_1 and φ_2 are positive constants; r_1 and r_2 are normally distributed random values in the range $[0,1]$; $X_i(t) = (x_{i1}(t), x_{i2}(t), \cdots, x_{id}(t))$ represents the current location of the i-th particle; d is the dimension of the vector represented by a particle; X_{blp} represents the previous best local position of the i-th particle; X_{bgp} represents the best global position found up to the moment by the whole swarm; $V_i(t) = (v_{i1}(t), v_{i2}(t), \cdots, v_{id}(t))$ is the velocity of the i-th particle.

Equation (10) defines how the position of particles is dynamically adjusted, taking into account the movement in the d-dimensional space given by Equation (9). Notice that this equation consists of three terms. The first term is the *momentum* part, meaning that the velocity cannot be changed abruptly. The second term is the *cognitive* part, that allows the particle to learn from its own flying experience, keeping track of the best position it has encountered up to the moment. The third term is the *social* part, and represents the collaboration among particles, allowing a particle to learn from the swarm's experience. The balance among these three parts determines the global or local convergence of the swarm.

When the sum of the three terms of Equation (9) exceeds a user-defined value $\pm V_{max}$, the velocity is clamped to this value. High values of V_{max} can make the particles potentially fly towards good solutions. On the other hand, low values of V_{max} can lead particles to converge to a local solution.

The original procedure for implementing PSO follows [9]:

1) Initialize a population of particles with random positions and velocities on the d-dimensional search space of the problem.
2) For each particle, evaluate its vector as a possible solution for the problem, by using a fitness function.
3) Compare each particle's fitness with its X_{blp}. If the current value is better than X_{blp}, then set X_{blp} equal to the current value of $X_i(t)$.
4) Among all particles, identify the one with the best fitness, and assign to X_{bgp} the current position of such particle. In some applications, only the surrounding neighbors can be considered instead of the whole swarm for updating velocity.
5) Update the velocity and position of all particles according to Equations (9) and (10) and, if necessary, restrict velocity to $\pm V_{max}$.
6) Loop to step (2) until a stop criterion is met, usually a sufficiently good solution or a maximum number of iterations.

There are many variations of the algorithm above described. In general, it is useful to have some strategy to preserve diversity in the swarm, so as to avoid fast convergence to local optima. Periodic explosions of the swarm can do this job by observing the agglutination of particles around a local optimum and resetting them to random positions, but keeping X_{bgp} unchanged [10], [11].

Originally, PSO was created to deal with problems with continuous variables. An important extension of the algorithm was the introduction of binary rather than continuous variables, thus allowing PSO to appropriately deal with combinatorial problems [11], [12]. In this implementation, each element of vector $X_i(t)$ is a bit, and the velocity is used as a probability to determine whether each bit will be in 0 or 1. After the evaluation of the n-th element of the velocity vector of the i-th particle (Equation 9, now, with binary values), the following sigmoid function is used to decide which value will have the particle:

$$s(v_{in}) = \frac{1}{1 + \exp(-v_{in})} \tag{11}$$

if $r_i < s(v_{in})$ then $x_{in} = 1$; otherwise, $x_{in} = 0$. r_i is a uniformly distributed random number over the interval [0,1].

B. Applying Binary PSO to the BSP Problem

In order to solve the BSP problem we adopt an approach similar to that proposed by [5], i.e., we split the overall problem in two sub-problems: the *selection* problem and the *allocation* problem. In this paper the PSO is used to solve the selection problem and the heuristic procedure from [5] is applied to handle the allocation of users to the base station sites selected.

In the binary PSO, a particle i has a position $X_i(t)$, at time t. Considering that the location of the potential base stations are known, each element of $X_i(t)$ will correspond to a decision variable $x_{ib}(t)$ = 1 (0) if a base station is (not) placed at location $b = 1, ..., B$.

TABLE I
FEATURES OF THE MAPS USED IN THE EVALUATION OF THE PSO.

map	#base stations	#users	area
1	10	30	20×20 m
2	20	60	40×40 m
3	25	75	50×50 m
4	30	90	60×60 m

At each iteration of the PSO, the current solution of the selection problem is used by the heuristic in order to set the allocation of potential users. Then it is possible to verify if the candidate solution is feasible, and evaluate the objective function. Note that the fitness value, which is an estimation of the solution goodness, is inversely proportional to the objective function value.

In the allocation process, each potential user location is allocated to the base station with the strongest received signal strength amongst the selected ones (by this, the number of users significantly affecting the complexity of the problem). At the end of the allocation, X_{bul} and Y_{bl} are set to 1; X_{bul} indicates that the base station b has a connection with the user u, and there are l users communicating with the base station; Y_{bl} represents that the base station b has a loading of l users.

As in [5], we discourage an infeasible solution by adding a penalty value to the objective function for each user that the solution leaves uncovered. These infeasible solutions arise when the number or placement of the base stations does not give coverage to all the users.

C. Exhaustive Search

To evaluate the performance of our proposed approach, PSO results (and its effectiveness) will be compared with the optimal solutions found by an exhaustive search (ES) procedure. The exhaustive search corresponds to the generation of all the possibilities for active base stations. Each option is evaluated using exactly the same restrictions applied to the PSO.

V. EXPERIMENTS AND RESULTS

A. Benchmarks

Aiming at evaluating the performance of the proposed PSO, four different maps (problems) were created. These maps represent hypothetical situations with increasing number of base stations (assumed omnidirectional), users and obstacles (in this case, walls). The difficulty in solving these problems increases exponentially, by number of base stations and users, thus enabling us to evaluate the performance of the PSO. The maps have the characteristics shown in Table I. Figure 1 shows graphically the maps used as benchmark (solid lines are walls, black squares are users, and white squares are base stations).

Maps were constructed using the software *Inkscape*, in the format SVG. An application in Java was developed to extract data from maps, corresponding to the inputs used in the PSO and exhaustive search algorithm.

B. Results

The PSO was implemented in ANSI-C programming language, and run in a PC with quad-core processor at 3 GHz and 2 GBytes of RAM. For each map, 100 independent runs of the PSO were done. Each run used the same parameters for the PSO, as follows: number of iterations = 1000, number of particles = 40, constants φ_1 and φ_2 = 2, V_{max} = 3.

The system was configured with the CDMA parameters described in Table II. The cost of installation (F_b) was configured evenly for all base stations. The effects of the environment in the propagation can have a significant influence in the overall performance of the system. In this study, path loss is evaluated by means of the classical formula (12), where d is the distance between base station b and location u (in km) and f is the frequency (in MHz). An extra 3 dB attenuation is added to each drywall existing between b and u.

$$att(dB) = 32.4 + 20\log 10(d) + 20\log 10(f) \tag{12}$$

TABLE II
CDMA PARAMETERS OF A WIRELESS NETWORK.

Frequency	1.8 GHz
P_{t1}	1 mW
P_{tar}	1 nW
P_{min}	1 pW
P_{max}	1 mW
G_p	128 (linear)
F_b	20
Q_F	8.75 dB
Q_R	7.0 dB

Table III shows the final results for the four maps. The first column indicates the instance of the problem (map), the next one is the objective function value (average and standard deviation), next is the success rate (the number of runs that the PSO found the optimal solution), fifth column is the average deviation from optimal, next is the average processing time, and last column the number of iterations to find the best solution (average and standard deviation).

For the maps 1 and 2, the binary PSO found the optimal solution in all runs. With the maps 3 and 4, the optimal solution was obtained in 39% and 0% of runs, respectively, but the average deviation from optimal was very small in both cases.

The optimal solutions found by exhaustive search, as well the processing time are shown in Table IV. The running time indicates the increasing complexity of the maps, two seconds for the first map, and 172 days and eight hours for the last map.

VI. CONCLUSION

In this work we have considered the design of CDMA indoor wireless networks and, in particular, the base station placement problem, where the least cost set of base station sites must be selected from a given number of potential sites to provide service to the users.

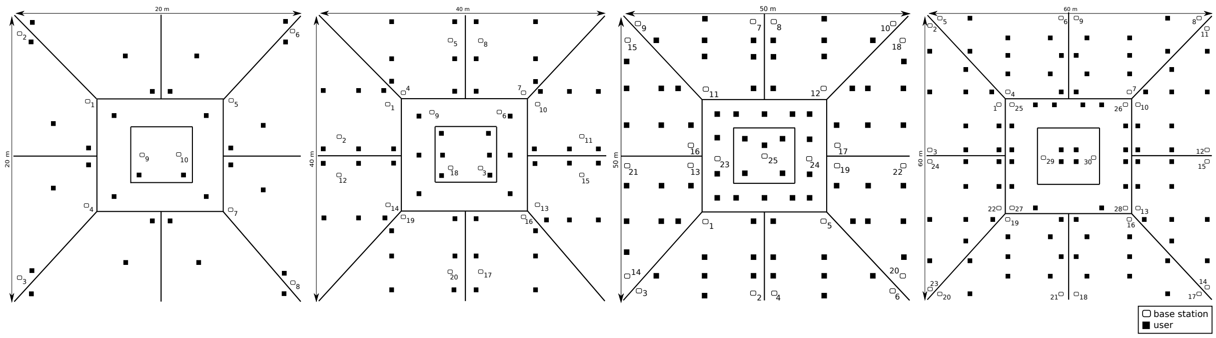

Fig. 1. Benchmark maps used in the experiments.

TABLE III
RESULTS OF THE BINARY PSO FOR 100 INDEPENDENT RUNS.

map	fitness avg $\pm$ stdev	success rate	average deviation from optimal	processing time avg [s]	#iterations avg$\pm$ stdev
1	0.40 $\pm$ 0	100%	0%	16s	747.57 $\pm$ 135.56
2	0.50 $\pm$ 0	100%	0%	2m 33s	940.89 $\pm$ 56.71
3	0.54 $\pm$ 0.02	39%	3.85%	6m 34s	958.29 $\pm$ 37.52
4	0.72 $\pm$ 0.02	0%	2.86%	14m 49s	974.57 $\pm$ 25.25

TABLE IV
OPTIMAL SOLUTIONS FOUND BY EXHAUSTIVE SEARCH.

map	optimal fitness	running time
1	0.40	2s
2	0.50	1h
3	0.52	73h
4	0.70	172d 8h

The problem was formulated as a nonlinear integer programming problem with constraints. A PSO approach was used to obtain feasible solutions, and its performance was compared with an exhaustive search method. Examples of application of the proposed design methodology to different network configurations (of growing complexity) have been discussed.

The computational results suggest that the PSO algorithm provides a quite efficient approach to obtain (near) optimal solutions with small computational effort. When the complexity of the problem increased, the exhaustive search method's computational time tends to increase significantly, and the PSO has to lose a little performance, but there may be observed that the search space rises exponentially as the number of base stations grows.

It is well known that the PSO performance is sensitive to both the number of particles and the number of iterations. Future work will include: the development of a benchmark involving different settings of these two parameters; new simulations to access the performance limits of PSO; and a comparison with other algorithms such as [6].

REFERENCES

[1] J. K. Han, B. S. Park, Y. S. Choi, and H. K. Park, "Genetic approach with a new representation for base station placement in mobile communications," in *Vehicular Technology Conference, 2001. VTC 2001 Fall. IEEE VTS 54th*, vol. 4, 2001, pp. 2703–2707 vol.4.

[2] E. Amaldi, A. Capone, and F. Malucelli, "Planning umts base station location: optimization models with power control and algorithms," *Wireless Communications, IEEE Transactions on*, vol. 2, no. 5, pp. 939–952, Sept. 2003.

[3] L. Raisanen and R. M. Whitaker, "Comparison and evaluation of multiple objective genetic algorithms for the antenna placement problem," *Mob. Netw. Appl.*, vol. 10, no. 1-2, pp. 79–88, 2005.

[4] Z. Yangyang, J. Chunlin, Y. Ping, L. Manlin, W. Chaojin, and W. Guangxing, "Particle swarm optimization for base station placement in mobile communication," in *Networking, Sensing and Control, 2004 IEEE International Conference on*, vol. 1, March 2004, pp. 428–432 Vol.1.

[5] J. Wong, A. Mason, M. Neve, and K. Sowerby, "Base station placement in indoor wireless systems using binary integer programming," *Communications, IEE Proc.-*, vol. 153, no. 5, pp. 771–778, Oct. 2006.

[6] L. K. Pujji, K. W. Sowerby, and M. J. Neve, "A new algorithm for efficient optimisation of base station placement in indoor wireless communication systems," *Communication Networks and Services Research, Annual Conference on*, vol. 0, pp. 425–427, 2009.

[7] L. Nagy and L. Farkas, "Indoor base station location optimization using genetic algorithms," in *Personal, Indoor and Mobile Radio Communications, 2000. PIMRC 2000. The 11th IEEE International Symposium on*, vol. 2, 2000, pp. 843–846 vol.2.

[8] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Micro Machine and Human Science, 1995. MHS '95., Proc. of the Sixth International Symposium on*, Oct 1995, pp. 39–43.

[9] ——, "Swarm intelligence." Morgan Kauffmann, 2001.

[10] H. S. Lopes and L. dos Santos Coelho, "Particle swarm optimization with fast local search for the blind traveling salesman problem," pp. 245–250, 2005. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/ICHIS.2005.86

[11] F. Hembecker, H. S. Lopes, and W. G. Jr, "Particle swarm optimization for the multidimensional knapsack problem," pp. 358–365, 2007. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-71618-140

[12] X. Hu, R. Eberhart, and Y. Shi, "Swarm intelligence for permutation optimization: a case study of n-queens problem," in *Swarm Intelligence Symposium, 2003. SIS '03. Proc. of the 2003 IEEE*, April 2003, pp. 243–246.