International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 4 (2012) pp. 496–505
© MIR Labs, www.mirlabs.net/ijcisim/index.html

Designing Cellular Networks using Particle Swarm Optimization and Genetic Algorithms

Antonio I. S. Nascimento and Carmelo J. A. Bastos-Filho

Polytechnic School of Pernambuco
University of Pernambuco
Recife, Brazil
E-mail: carmelofilho@poli.br

Abstract: **The demand for cellular systems has increased drastically in recent years. To design such networks is not a simple task and computational tools to assist network designers can be very helpful. The design of a cellular network can be divided in two minor problems: the maximum coverage and channel assignment planning. In this paper, we propose a methodology to tackle the former using Particle Swarm Optimization and the latter using genetic algorithms. We had to adapt the Particle Swarm Optimization algorithm, since we have associated the position of the Radio Base Stations to the particle positions in the search space. We also developed two mechanisms to avoid the overlap among the cells and to maximize the coverage of the entire system. We tested our approach in two scenarios in different configurations. For the channel assignment planning, we adapted the crossover and the mutation operators in order to define which channels should be used in each cell. We believe the preliminary results are very encouraging and with future works we can develop a commercial tool to solve the real problem.**

Keywords: Cellular networks; Network design; Particle Swarm Optimization; Genetic Algorithms.

I. Introduction

Nowadays, flexibility and reliability are required characteristics for telecommunication systems. In this scenario, wireless technologies are the most promising alternative to connect people with mobility. Because of this, wireless networks have gained a lot of attention in the last years due to the ever growing demand for mobile telephones and wireless local area networks [1][2]. Wireless networks can be organized in two distinct ways: networks with infrastructure and *ad hoc* networks. In the former, there are Radio Base Station (RBS) in which the mobile devices can connect to, whereas in *ad hoc* networks the devices send information to one another through links created by the devices itselves.

The cellular networks are networks with infrastructure, where each cell [3] is a geographical region with a RBS. Each RBS should connect the users within its coverage area by using a set of predefined channels. This strategy can be used due to the frequency reuse concept, where the frequencies can be reused when they are not being used in adjacent cells. In networks with infrastructure (*e.g.*a cellular network), there are two non trivial tasks during the planning phase. The first is to define how many base stations are necessary to build the infrastructure network and where the designer should position them. This problem is known as the maximum coverage problem. The second one consists on how to define which frequencies should be used in each cell, in order to supply the service demand while avoiding intercell interference.

The maximum coverage problem can be formally stated as: Given a geographical area with a service demand distribution, one has to define the positions of the RBS and the set of parameters for the antennas, such as the transmission power and the antenna height [4]. These parameters influence on the area covered by the antenna, which defines the cell size. One should note that the network designer must avoid to overlap the cells, since it reduces the total area covered by the entire system. However, sometimes it is necessary to shorten the distance between adjacent cells in order to supply regions with higher demand for services. Besides, other practical issues must be analyzed during the network design, such as the cost to position the antennas in certain places and the physical land constraints.

A relevant aspect widely used in models for solving the maximum coverage problem is how the service demand is geographically distributed [5]. In many cases, the demand is higher in some areas and more RBS must be positioned in these regions.

The major issue to design wireless networks with infrastructure is the trade-off between Quality of Service (QoS) and the total Capital Expenditure (CAPEX) to implement the network. One should note that the higher the number of RBS, the higher the availability of the network for the user, but the higher the CAPEX is. Besides, one should notice that a higher number of RBS in a bounded geographical area results in transmitters with lower power. The lower power should be used in order to avoid interference between non adjacent cells.

In this paper, we propose to adapt the Particle Swarm Optimization (PSO) algorithm [6] to solve the maximum coverage problem. The PSO has to determine the position of the RBS considering the demand distribution in order to maximize the QoS. The PSO algorithm had to be adapted since each particle in our approach is associated to a different RBS of the network.

Once the RBS are positioned, it is necessary to define which channels should be allocated in each cell. This task is crucial in the network design process, since it directly impacts on the network QoS and on the amount of generated interference. A bad channel assignment scheme can cause interference among adjacent cells, since the signal transmitted by a RBS can surpass the cell boundary and if an adjacent cell is using the same channel, co-channel interference can occur. Thus, the channel assignment planning must aim to avoid adjacent cells using the same channels. We used genetic algorithms (GA) [7] to solve the channel assignment problem. The GA was applied to define which channels should be used in each cell of the network built by the PSO.

This paper is organized as follows: section II presents the basic PSO algorithm and some related concepts. Section III presents a brief review on genetic algorithms. Our proposal to solve the maximum coverage problem with a modified PSO is presented in section IV. In section V, we present the GA model developed to solve the channel assignment problem. The experiments and results are described, respectively, in sections VI and VII. Finally, in section VIII, we present our conclusions.

II. Particle Swarm Optimization

Some efficient optimization algorithms for hyper-dimensional problems in continuous spaces have been proposed in recent years. Among them, we can cite the Particle Swarm Optimization (PSO) algorithm [6]. The PSO algorithm was proposed by Kennedy and Eberhart in 1995 [8], inspired by the behavior of flocks of birds. PSO was conceived to optimize non-linear functions in multidimensional continuous search spaces.

The PSO algorithm is based on an analogy between the flight of a flock of birds looking for food and a set of artificial entities searching for the best solution in the search space of a problem. The term particle stands for the bird. Each particle has a position, that represents a possible solution within the search space, and a velocity. The position of each particle is updated at each algorithm iteration by using the following equation:

$$\overrightarrow{x_i}(t+1) = \overrightarrow{x_i}(t) + \overrightarrow{v_i}(t+1), \tag{1}$$

where $\overrightarrow{x_i}(t+1)$ is the new position of particle *i*, $\overrightarrow{x_i}(t)$ is the position of particle *i* in the current iteration and $\overrightarrow{v_i}(t+1)$ is the velocity of particle *i* already updated in the current iteration of the algorithm.

The velocities of the particles are updated according to the equation (2).

$$\overrightarrow{v_i}(t+1) = \overrightarrow{v_i}(t) + c_1 r_1[\overrightarrow{p_i}(t) - \overrightarrow{x_i}(t)] + c_2 r_2[\overrightarrow{n_i}(t) - \overrightarrow{x_i}(t)], \tag{2}$$

where $\overrightarrow{v_i}(t+1)$ is the new velocity for particle *i*, $\overrightarrow{v_i}(t)$ is the velocity in the previous iteration for particle *i*. r_1 and r_2 are random values generated by a uniform distribution in the interval [0, 1]. $\overrightarrow{p_i}(t)$ is the position of the best solution found by the particle *i* and $\overrightarrow{n_i}(t)$ is the position of the best solution found by the neighborhood of particle *i* during the search so far.

The first term in the equation (2) is the inertial term and takes into account the previous velocity of the particle. The other two components are called cognitive and social terms, respectively. The cognitive component represents the experience of the particle itself during the search, whereas the social component is related to the experience of the swarm. c_1 and c_2 are constants used to weight the influence of the cognitive component and the social component, respectively. It has been demonstrated empirically that when $c_1 = c_2 = 2.0$, the PSO algorithm can achieve good results [8].

Other proposals have been presented aiming to accelerate the convergence of the algorithm, most of them proposing to change the equation used to update the velocities or to create a new communication mechanisms among the particles. One of the adjustments was the insertion of a coefficient ω in the inertial term [9], which was widely accepted by the scientific community. This approach is known as Inertia approach and the velocity update equation for this is presented in equation (3):

$$\overrightarrow{v_i}(t+1) = \omega\overrightarrow{v_i}(t) + c_1 r_1[\overrightarrow{p_i}(t) - \overrightarrow{x_i}(t)] + c_2 r_2[\overrightarrow{n_i}(t) - \overrightarrow{x_i}(t)]. \tag{3}$$

It has been demonstrated that if one decreases ω along the algorithm iterations, the algorithm can achieve better results. The most used strategy is to start the search process with ω close to 1 and decrease it linearly during the search process. By doing this, the particles begin the search in an exploration mode and change gradually to an explotation mode along the iterations [9].

Regarding the mechanisms to exchange information among the particles, called the communication topology, there are different strategies to perform it. The most used topology is the Global topology. In the Global topology, each particle can obtain information directly from all other particles. Despite the PSO with Global topology converges quickly, it has a higher chance to get trapped in local minima.

Other topologies have been proposed based on local neighborhoods. The most famous is the ring topology, where each particle just acquires information from the two direct neighbors [6]. Generally, a PSO with ring topology achieves better results than a PSO with global topology. However, the convergence time tends to increase, since the information obtained during the search process is transmitted indirectly through the particles.

III. Genetic Algorithms

Genetic Algorithms (GA) were inspired in the Charles Darwin theory of evolution and natural selection. This theory states that individuals with some competitive advantages are more likely to survive and reproduce. These advantages can be transmitted from the parents or can appear due to variations caused by environmental changes. Then, to eliminate the weaker individuals through the iterations may result in a population with more chances to survive [7]. Valuable information is transmitted from generation to generation through the genetic code of the individuals. This idea was firstly modeled by Holland in 1975. Nowadays, it is largely used in search and optimization problems. In this model, each individual is represented by a set of genes and each gene represents an attribute of a problem, the set (chromosome) represents a possible solution for the problem. The fitness of a chromosome is determined by the quality of the encoded solution. The goal is to simulate the principle of Darwin searching for new and better solutions. The evolutionary process proposed by Holland can be implemented by successive repetition of three operations: crossover, mutation and selection. The pseudocode of a generic GA can be viewed in algorithm 1.

Algorithm 1: Pseudocode of a Generic Genetic algorithm.

```
Create a population with N individuals;
Initialize the population by defining random values to
  the genes;
while stop condition is not reached do
    Evaluate the quality of the individuals;
    Select parents;
    Perform Crossover;
    Perform Mutation;
    Eliminate the weaker individuals;
end
```

The crossover operator generates new individuals by combining chromosomes of two individuals chosen from the population (parents). The occurrence of crossovers is determined by a parameter called *crossover rate*. Figure 1 presents one example of crossover, the one point crossover. There are some other possible strategies to perform the crossover.

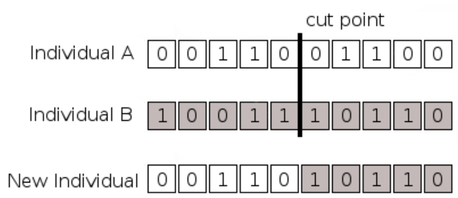

Figure. 1: Example of one point GA crossover operator.

The mutation operator simulates the "variations caused by environmental changes in living conditions" suggested by Darwin. It changes the chromosome of an individual randomly, *i.e.*, it chooses a gene (or several genes) and changes their values. The mutation operator is responsible to maintain the diversity during the search, enabling the algorithm to escape from local optima. The mutation occurrence is determined by a parameter called *mutation rate*. Figure 2 shows an example mutation for a binary chromosome encoding.
The selection operator incorporates the core of Darwin's theory: individuals with higher fitness have more chances to remain in the population in the next generations and reproduce. In the search process, the selection leads the solutions from worse regions to best regions of the search space. The selection can be performed in different manners. Some examples are the simple deletion of the N worst individuals, a tournament selection or a selection by roulette wheel. In the tournament selection, a subset of the population is chosen with equal probability and the best individuals within this subset

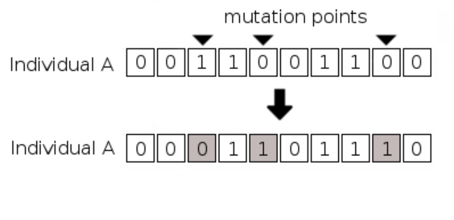

Figure. 2: Example of GA mutation operator.

proceeds to the next generation. In the roulette wheel selection, the individuals are chosen probabilistically, where the probability of selection is proportional to the fitness.
One should notice that the mutation can discard the best solutions. One alternative to avoid this is to copy the N best individuals to next generation. This is called elitism [7].

IV. Our Proposal to Solve The Maximum Coverage Problem

In this section, we present our proposal to adapt the PSO algorithm to solve the maximum coverage problem. In this first proposal, we have decided to reduce the scope of the problem by not including the configuration of the parameters of the RBS in the optimization process. The coverage for all the antennas are the same and all the parameters of the antennas are combined in one parameter, called *RBS coverage radius* (R).
The search space represents the region where it is desired to install the cellular network. In this first model, the region is free of any physical obstacle and it is possible to place the RBS at any point of the search space. The service demand distribution in the search space is described by a function that maps each position of the search space to a value that represents the amount of clients covered if the RBS is placed at this point. The mapping **position** → **demand** is defined by a $F(\vec{x_i}, R)$ function, where $\vec{x_i}$ indicates the position of the RBS i. One should notice that higher values for *R*, will lead to more clients covered by a cell, as can be seen in figure 3. One can observe that $F(\vec{x_i}, R) = 5$ when the *coverage radius* assumes a value R, and the amount of covered clients increases $F(\vec{x_i}, R) = 10$ when the *coverage radius* assumes a value R'. On the other hand, higher values for *R*, will cause more interference among the cells.

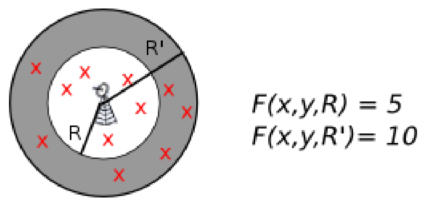

Figure. 3: Example to illustrate the service demand function.

Differently from the conventional applications for the PSO algorithm, where the n search variables are encoded into the n dimensions of the particle, each RBS is a particle "flying" through the search space in our approach. Each particle is guided by the demand function to the higher demand regions. Thus, we have a maximization problem.

However, the convergence of all the antennas to a single point is not interesting since all the cells will be overlapped, and the total area covered by the entire system will be minimized. Because of this, we developed two mechanisms to avoid the agglutination of particles over a single point.

In the former, the particles consider the overlapped areas, evaluated by using the coverage radius, to upgrade their position. Every particle evaluates if the next movement will result in overlapping areas with other particles. If it occurs, an opposite vector, called anti-collision vector, is generated for each collision. Then, the velocity of the particle is updated considering the anti-collision vectors. The weight of the velocity vector is always unitary, whereas the weights of the anti-collision vectors are determined by the normalized area of intersection between the coverage areas generated by the two particles. An anti-collision vector has an unitary weight if two cells are totally overlapped.

An illustration of the anti-collision vector effect can be observed in figure 4. Consider the particle A moving to point X, then an overlapped area will be created. Because of this, a vector ($\vec{V}_{col}$) is generated and applied to the particle A. One should note that the particle A will have a different position as shown in the bottom of Figure 4 and the collision will be avoided in this case.

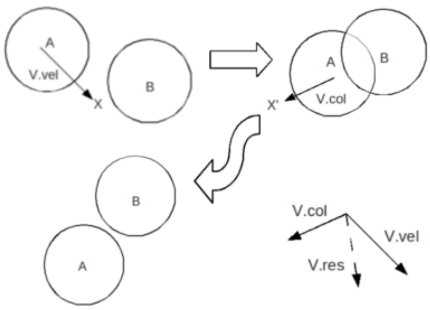

Figure. 4: Illustration of the PSO anti-collision mechanism.

Sometimes, if the overlap occurs for many particles, different anti-collision vectors can nullify or greatly minimize the influence of one another. Because of this, we developed another mechanism to avoid agglutination.

In the second mechanism, we introduced a new term in the equation used to update the velocities of the particles, called repulsion term. It means that the velocity equation for our approach is given by equation (4).

$$\begin{aligned}\vec{v_i}(t+1) &= \omega\vec{v_i}(t) + c_1 r_1[\vec{p_i}(t) - \vec{x_i}(t)] \\ &+ c_2 r_2[\vec{n_i}(t) - \vec{x_i}(t)] - c_3 r_3[\overrightarrow{l_{worst_i}}(t) - \vec{x_i}(t)]. \qquad (4)\end{aligned}$$

where l_{worst_i} represents a position in the neighborhood region of a particle *i* with higher density. The concept of neighborhood to evaluate l_{worst_i} is defined by dividing the search space in a squared grid. A particle *i* evaluates the density by checking how many particles exist in each adjacent regions of the grid. Once one region is selected, a particle *j* is randomly selected from this region and the current position of this particle j is the l_{worst_i} for particle i. This new term of the velocity equation has a negative signal in order to separate particles *i* and *j*. Figure 5 shows an example of this process, where the grid represents the search space. In this scenario, a particle *i* positioned in the region E will check the regions A, B, C, D, E, F, G, H and I. As the region G presents the higher density, one of the 4 particles of the region G will be chosen as the l_{worst_i}.

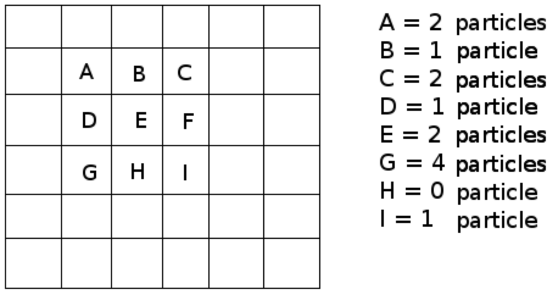

Figure. 5: Example to illustrate how to determine the l_{worst_i}.

An important aspect of our proposal is the behavior of the coefficients c_1, c_2 and c_3. We defined that they can vary along the iterations, such as ω. Initially, the algorithm should act in order to find good regions in the search space and, after that, the algorithm should try to maximize the total coverage area by separating the particles. For this, we performed some tests and the best results were achieved when c_1 and c_2 decrease along the iterations, while c_3 increases along the iterations. One should note that c_1 and c_2 are responsible for attracting the particles to the regions with maximum demand, whereas c_3 is responsible to repulse the particles covering the same region. In our simulations, c_1, c_2 and c_3 vary linearly along the iterations and each one has an initial value and a final value (c_i and c_f).

The result achieved by our proposal is not the best point of the search space ever found during the search process so far, as the standard PSO. In this case, the solution is the set of the final positions of the particles in the search space.

For the maximum coverage problem, the particles should keep a distance close to the RBS radius. Because of this, the size of the grid used to determine the l_{worst} should be chosen based on this parameter. A large grid can result in sparse positioned RBS and can difficult handoff operations. This effect may happen due the repulsion mechanism in unnecessary situations and it can be comprehended by observing the figure 6, where the area of each square in the grid is nearly four times greater than the RBS coverage area. In the scenario of figure 6, the repulsion mechanism will elect particle Y or Z as the l_{worst} for particle X. However, it is not necessary because particle X is not near to Y nor Z.

Figure 7 shows a scenario where the grid square area and the RBS coverage area have closer values. In the case of figure 7, the repulsion mechanism for particle X, will not act because there is no other particle at neighborhood.

On the other hand, if the grid square area is much smaller than the RBS coverage area, the opposite effect can occur, the particles will not be separated when it is necessary. Figure 8 illustrates an example of this situation. Although there is an overlap between particles Y and Z, one can observe in figure 8 that the particles Y and Z will not be separated for this case.

We evaluated the algorithm performance using two metrics: the average fitness per particle (D_{cover}) and the average per-

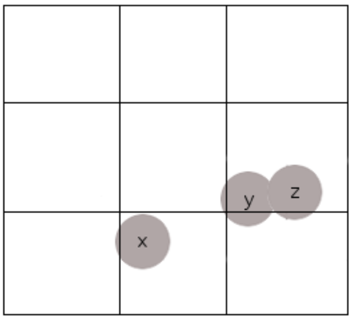

Figure. 6: Example with a grid square area nearly four times greater than the RBS coverage area.

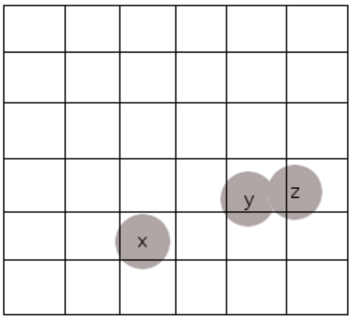

Figure. 7: Example with a grid square area close to the RBS coverage area.

centage of overlap area per particle ($A_{overlap}$). The former is evaluated by using $F(\overrightarrow{x_i}, R)$. It indicates how many clients can be covered by the entire network infrastructure. The second metric is directly related to the total coverage area achieved by the RBS.
The pseudocode of our proposal for the Maximum Coverage Problem is presented in algorithm 2.
The stop condition used in our approach is based on the exponential moving average (EMA) of the overlapped area variation between the iterations (ΔA_{over}). EMA is evaluated by equation (5), where N defines the first samples of ΔA_{over} to be the first value of $EMA_{previous}$ (evaluated by using an arithmetic average).

$$EMA = [1 - (\frac{2}{N+1})]EMA_{previous} + (\frac{2}{N+1})\Delta A_{over}. \tag{5}$$

Since it is possible to estimate the next value of this series, the algorithm terminates when the predicted value for the overlapped coverage area has an average value of 10%. We used this value since the particles should not disperse too much in order to provide a good handoff service [10]. If the algorithm does not reach the stop condition after 1,000 iterations, it finishes.

V. The model developed for the channel assignment problem

The assignment of channels for the RBS is the second stage of the cellular network design. This task is performed after the positioning because it requires the RBS neighborhood information to define the amount of needed channels in each RBS. The number of channels in a RBS determines how many users can use the service simultaneously within the cell. Therefore, the total number of required channels for a RBS is evaluated using the demand function ($F(x, y, R)$) and a parameter that indicates the average percentage (P) of users which use the service simultaneously within the cell. The total number of required channels for a RBS is evaluated by equation (6). The parameter P was introduced in order to avoid the cells to have more channels than it is necessary.

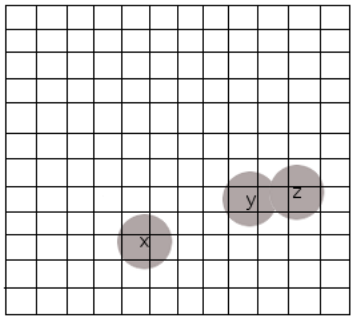

Figure. 8: Example with a grid square area nearly 25% of the RBS coverage area.

Algorithm 2: Pseudo-code of the PSO approach to solve the Maximum Coverage Problem.

for *each particle i of the swarm* **do**
  Assign randomly the position $\overrightarrow{x_i}$;
  Assign randomly the velocity $\overrightarrow{v_i}$;
  Evaluate the fitness using $F(\overrightarrow{x_i}, R)$;
  Evaluate $\overrightarrow{p_i}$;
  Evaluate $\overrightarrow{n_i}$;
end
while *stop condition is not reached* **do**
  for *each particle i of the swarm* **do**
    Determine $\overrightarrow{l_{worst_i}}$;
    Update the velocity according equation (4);
    Update the position according equation (1);
    Check collisions;
    if *there is any collision* **then**
      Apply the anti-collision mechanism;
    end
    Evaluate the fitness using $F(\overrightarrow{x_i}, R)$;
    Update $\overrightarrow{p_i}$ if it is necessary;
    Update $\overrightarrow{n_i}$ if it is necessary;
  end
  Update the Coefficients c_1, c_2 and c_3;
end

$$N = F(x,y,R)\frac{P}{100}. \tag{6}$$

In our genetic algorithm approach, each individual represents the channel assignment planning for the whole network, *i.e.*, a single chromosome encodes the channels used in all RBS. The individual is composed by subsets of genes where each subset represents the channels used by a specific RBS. Figure 9 presents a simple scenario of a network with up to 9 available channels and figure 10 shows examples of individuals for this scenario.

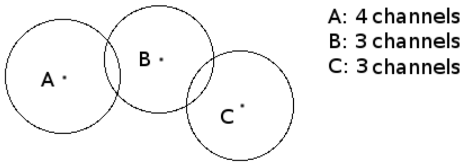

Figure. 9: Hypothetical example of a network.

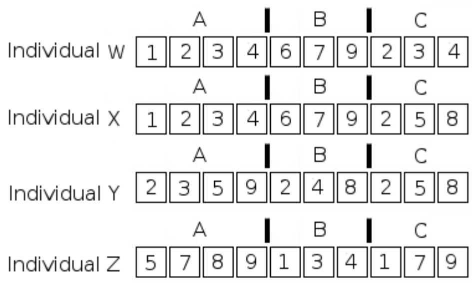

Figure. 10: Examples of GA individuals for the channel assignment problem.

The fitness evaluation is performed by counting the total number of channels used simultaneously by adjacent cells. The lower this number, the lower is the co-channel interference. In this model, two cells are adjacent if the distance between their centers is lower than the sum of their radius. Table 1 shows the fitness function results for the individuals of figure 10.

Table 1: Fitness function results for the individuals of figure 10 individuals.

Individual	Fitness
W	0
X	0
Y	3
Z	1

The GA aims to find combinations that minimize the fitness function. In the example of figure 10, W and X are the fittest individuals. The stop condition is to find an individual with fitness function equal to zero or to reach a maximum number of iterations.
The strategy developed for the crossover operator was to define N cut points, where N is the number of RBS. The position of each cutting point is randomly selected within the gene sequence that represents the RBS and the crossover is performed into this sequence. Figure 11 illustrates our crossover operator. If the crossover operator results in a individual with repeated channels within a RBS, the generated individual is penalized.
The mutation operator developed, defines N mutation points where each point is randomly chosen within a gene sequence

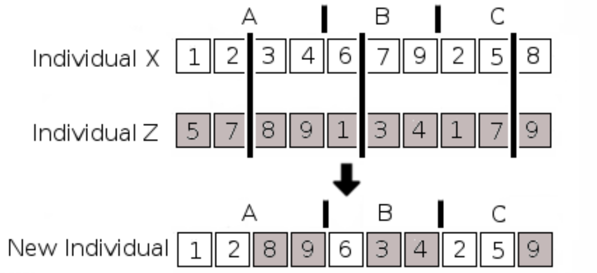

Figure. 11: Crossover operator for the channel assignment problem.

that represents a RBS. The gene mutation is done by assigning a channel that is not present in the RBS to avoid repetition. This value is chosen randomly among the remaining channels available for each RBS. Figure 12 illustrates the mutation operator. The arrows above the chromosome indicate the mutation points.
We decided to use the roulette wheel strategy for the selection operator.

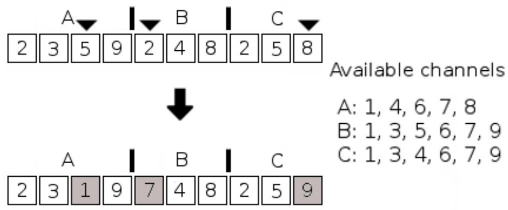

Figure. 12: Mutation operator for the channel assignment problem.

VI. Simulation Setup

For the maximum coverage problem, two scenarios were considered for the simulations and they are characterized by the functions presented in equations (7) and (8). The graphics for the demand function are presented in figures 13 and 14, respectively. The dimensions of the search space are 100 km x 100 km. The experiments were performed with 10 particles, R equal to 5 km or 10 km and ω decreasing linearly from 0.6 to 0.1 along the iterations. We used four different sets of parameters for c_{1i}, c_{1f}, c_{2i}, c_{2f}, c_{3i} and c_{3f} in the simulations, as shown in table 2. These sets were chosen after a preliminary analysis, aiming to produce different behaviors for the algorithm. We present our results in terms of average value and standard deviation for the two metrics after 30 trials.

$$F_1(x,y,R) = R\{10e^{-[\frac{(x-75)^2}{800}+\frac{(y-70)^2}{800}]} + 20e^{-[\frac{(x-40)^2}{800}+\frac{(y-25)^2}{800}]}\}, \qquad (7)$$

$$F_2(x,y,R) = R\{20e^{-[\frac{(x-65)^2}{578}+\frac{(y-70)^2}{578}]} + 15e^{-[\frac{(x-30)^2}{578}+\frac{(y-30)^2}{578}]} + 10e^{-[\frac{(x-80)^2}{578}+\frac{(y-15)^2}{578}]}\}. \quad (8)$$

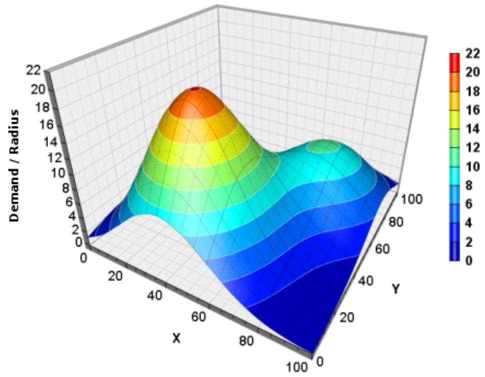

Figure. 13: Graphic of the demand function described by equation (7).

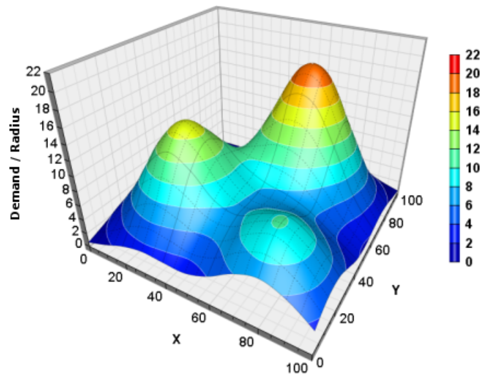

Figure. 14: Graphic of the demand function described by equation (8).

Table 2: Configurations used for c_{1i}, c_{1f}, c_{2i}, c_{2f}, c_{3i} and c_{3f}.

Configuration	Parameters					
	c_{1i}	c_{1f}	c_{2i}	c_{2f}	c_{3i}	c_{3f}
I	0.7	0.4	0.6	0.1	0.1	1.2
II	0.7	0.4	0.6	0.1	0.2	0.6
III	0.7	0.4	1.5	0.4	0.1	1.2
IV	1.6	0.7	0.6	0.1	0.1	1.2

For the channel assignment problem, we performed some simulations considering the network presented in figure 15, where the number within the cell identify the RBS.

In this scenario, 60 channels were available and the number of channels required for each RBS is presented in table 3. These values were determined based on an estimative on the demand functions and the coverage radius used for the maximum coverage problem with P equal to 10% and 20%.

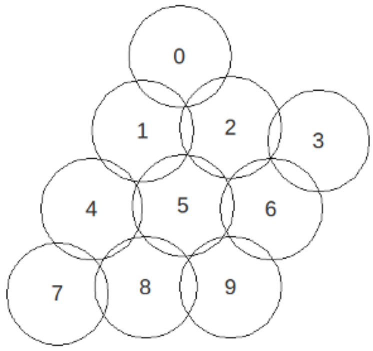

Figure. 15: Network used for the simulation in the channel assignment problem.

Table 3: Number of channels required per RBS.

Cell id.	Number of Channels
0	13
1	12
2	12
3	17
4	11
5	18
6	12
7	16
8	13
9	14

The GA parameters investigated in the experiments were the crossover rate, the mutation rate and the percentage of elitism in the population. The configurations used in our experiments are presented in table 4.

Table 4: GA configurations used in the experiments.

Configuration	Parameters	
	Crossover rate	Mutation rate
I	35	10
II	35	15
III	35	20
IV	40	10
V	40	15
VI	40	20

The results (mean and standard deviation for the best individual) were measured for 30 individuals in the population, 10 trials and a maximum of 3,000 GA iterations were executed.

VII. Simulation Results

For the maximum coverage problem, we performed some tests for two different functions with two different radius of coverage to analyze the results produced by our approach. Tables 5 and 6 present the results for the function F_1 for the configurations defined in table 2 with radius sizes of 5 km and 10 km, respectively. Tables 7 and 8 present the same results for function F_2. One can note from these tables that the configurations I and II achieved better results then configurations III and IV. In all cases, D_{cover} are similar, but configurations I and II obtained lower $A_{overlap}$. One can claim that the configuration I is even better then configuration II, since the standard deviation for D_{cover} and $A_{overlap}$ are lower for almost all the cases. Therefore, it indicates that the algorithm is more stable when c_{3f} is higher.

Table 5: Results for Function F_1 and radius 5 km.

Configuration	D_{cover}	$A_{overlap}$
I	**80.46 ± 3.41**	**15.34 ± 2.66**
II	83.35 ± 7.16	15.10 ± 7.52
III	84.17 ± 3.78	33.06 ± 19.96
IV	83.06 ± 4.70	31.76 ± 13.32

Table 6: Results for Function F_1 and radius 10 km.

Configuration	D_{cover}	$A_{overlap}$
I	**107.20 ± 9.04**	**15.90 ± 3.04**
II	111.70 ± 10.44	16.72 ± 9.16
III	110.91 ± 14.92	34.06 ± 19.15
IV	109.98 ± 14.83	24.61 ± 15.58

Table 7: Results for Function F_2 and radius 5 km.

Configurations	D_{cover}	$A_{overlap}$
I	**75.17 ± 3.14**	**14.68 ± 2.78**
II	78.24 ± 3.25	14.88 ± 6.34
III	78.87 ± 4.99	38.99 ± 20.55
IV	78.39 ± 7.26	33.84 ± 21.75

Figures 16 and 17 show the average values, the average values plus the standard deviation and the average values minus the standard deviation for D_{cover} and $A_{overlap}$, respectively, after the 30 trials using configuration I for function F_1 with coverage radius of 5 km. As expected, the particles are grouped toward high demand areas in the beginning, and the particles spread around after some iterations.

For the channel assignment problem, we performed experiments with the configurations presented in table 4 for 3 values of elitism percentage (0%, 10% and 20%). The results are presented in table 9. One can observe that the configuration with higher mutation rate and elitism percentage achieved better results. The crossover rate did not present a significant difference for this case.

Figures 18 and 19 show examples of solutions found by our approach for the functions F_1 and F_2, using configuration I and 10 particles. Tables 10 and 11 present the channel assignment scheme for these networks found by the GA for $P = 10\%$, GA configuration III and 20% of elitism.

Regarding to the maximum coverage problem, the particles were attracted to the highest demand region of the search space, avoiding the convergence to a single point. We also observed that the particles were arranged only around the highest peak of the function. It suggests the addition of a new mechanism in future work for even better solutions.

VIII. Conclusions

We proposed a new approach to solve the problem of maximum coverage in cellular networks using Particle Swarm Optimization. Since it is not a classical PSO approach, we adapted the PSO algorithm by associating the RBS positions to the particles positions. We developed two mechanisms to create diversity and the results showed that they performed well. We proposed to include a novel term in the equation used to update the velocity in order to disperse the particles

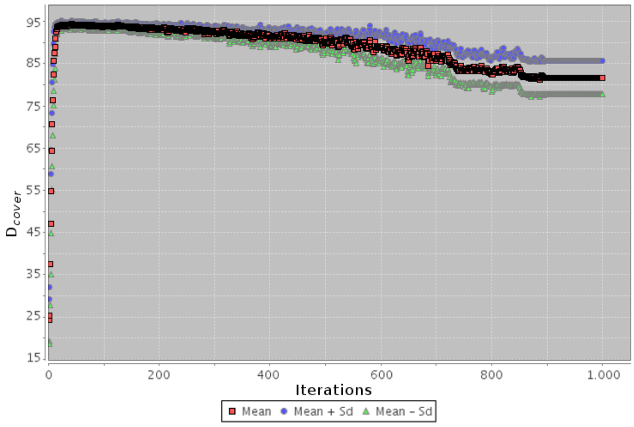

Figure. 16: Evolution of the average value, the average value plus the standard deviation and the average value minus the standard deviation for D_{cover}.

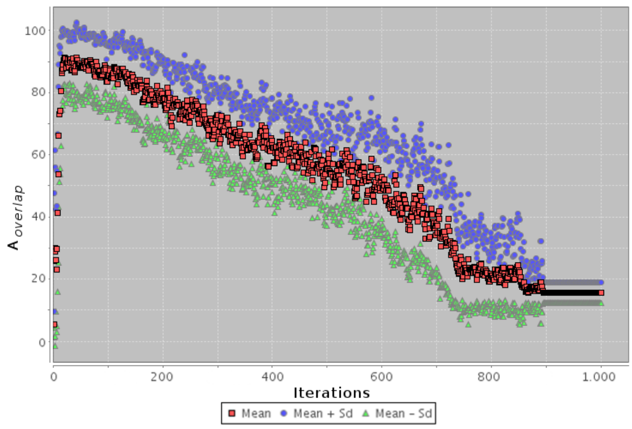

Figure. 17: Evolution of the average value, the average value plus the standard deviation and the average value minus the standard deviation for $A_{overlap}$.

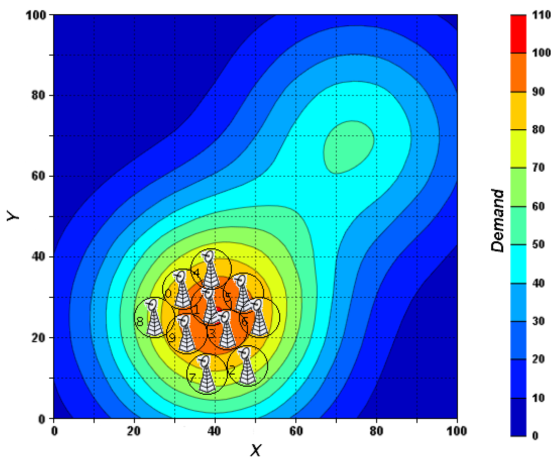

Figure. 18: Example of network obtained for function F_1 with 10 RBS and $R = 5km$ for each RBS.

Table 8: Results for Function F_2 and radius 10 km.

Configuration	D_{cover}	$A_{overlap}$
I	**99.06 ± 7.55**	**15.76 ± 3.01**
II	103.04 ± 8.37	13.22 ± 2.28
III	98.86 ± 13.09	26.58 ± 16.12
IV	96.41 ± 13.20	19.09 ± 12.28

Table 9: Results for GA experiments.

Configuration	elitism (% of population)		
	0	10	20
I	1,7 ± 1,1	1,4 ± 1,2	2,3 ± 1,34
II	8,0 ± 10,54	0,4 ± 0,66	0,3 ± 0,45
III	41,7 ± 2,09	0,1 ± 0,30	0,0 ± 0,0
IV	1,7 ± 1,09	1,8 ± 1,16	2,7 ± 2,75
V	0,6 ± 0,8	0,1 ± 0,30	0,5 ± 0,5
VI	40,4 ± 3,74	0,0 ± 0,0	0,0 ± 0,30

from a crowded region and distribute it in the region around the peaks of demands.

We analyzed the influence of the parameter selection in the performance of the algorithm and showed that different sets of acceleration coefficients generate a different behavior for the algorithm during the optimization process.

We also showed that it is possible to assign the RBS channels using a Genetic algorithm, where the main purpose is to avoid co-channel interference.

We believe it is necessary to include some other features to deal with niches in the cases where the demand function is highly multimodal. Furthermore, we intend to incorporate constraints in the search space, for example to represent physical obstacles in the region and other interference issues. Other future work concerns to include the antenna parameters in the optimization process.

References

[1] M. Danziger and F. B. Lima-Neto, "A hybrid approach for IEEE 802.11 intrusion detection based on AIS, MAS and naive bayes," *International Journal of Computer Information Systems and Industrial Management Applications*, 2011.

[2] M. Barbiroli, C. Carciofi, V. Degli Esposti, P. Grazioso, D. Guiducci, V. Petrini, and G. Riva, "Planning criteria to improve energy efficiency of mobile radio systems," in *Antennas and Propagation in Wireless Communications (APWC), 2011 IEEE-APS Topical Conference on*, sept. 2011, pp. 1197 –1200.

Table 10: Channel assignment for the network depicted in the figure 18.

BS	Channels
0	5-10-13-14-20-29-32-33-38
1	1-2-15-16-23-25-27-31-35-49
2	3-8-15-24-30-40-42-48-
3	3-5-7-9-19-26-29-32-41-42
4	0-9-26-36-40-43-44-46-47
5	4-6-7-10-18-21-29-30-38-41
6	13-20-27-31-39-40-43-44-49
7	2-7-8-10-12-24-44-48
8	0-7-17-19-34-38-44-47
9	4-8-10-12-18-22-24-37-43-48

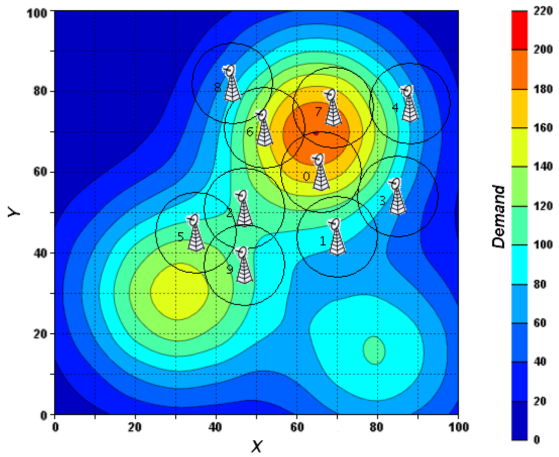

Figure. 19: Example of network obtained for function F_2 with 10 RBS and $R = 10km$ for each RBS.

Table 11: Channel assignment for the network depicted in the figure 19.

BS	Channels
0	1-4-8-13-16-19-26-29-34-35-38-40-45-47-49-51-54-56
1	2-5-6-7-30-43-46-48-55
2	2-5-14-26-29-31-35-45-47-48-54
3	3-10-22-25-26-44-45
4	2-8-17-19-26-31-43
5	8-16-18-20-30-42-46-49-52-57-59
6	2-6-12-17-20-21-22-25-30-32-33-48-53-55-57-59
7	0-3-9-10-11-15-18-23-27-37-39-41-42-44-46-50-52-58
8	3-10-23-24-31-36-39-41
9	0-3-12-21-22-24-27-28-36-37-39

[3] T. S. Rappaport, J. D. Reed, and B. D. Woerner, *Wireless Personal Communications - Research Development*. Springer Verlag NY, 2010.

[4] E. Amaldi, A. Capone, F. Malucelli, and C. Mannino, "Optimization problems and models for planning cellular networks," in *Handbook of Optimization in Telecommunications*. Springer Science, 2006.

[5] K. Tutschku, "Demand-based radio network planning of cellular mobile communication systems," in *In Proceedings of the IEEE Infocom 98*. IEEE, 1998.

[6] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2006.

[7] D. M. Goldberg and K. Sastry, *Genetic algorithms: The design of innovation*. Springer, 2011.

[8] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 1995.

[9] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, may 1998, pp. 69 –73.

[10] N. D. Tripathi, J. H. Reed, and H. F. Vanl, "Handoff in cellular systems," *IEEE Personal Communications*, 1998.

Author Biographies

Antonio I. S. do Nascimento received the B.Sc. degree in computing engineering from the State University of Pernambuco (UPE), Brazil in 2009. He works with computational intelligence since 2007 and his current research interests include computational intelligence applications, optimization algorithms and data mining.

Carmelo J. A. Bastos-Filho was born in Recife, Brazil, in 1978. He received the B.Sc. degree in Electronics Engineering from Federal University of Pernambuco (UFPE) in 2000. He received the M.Sc. and PhD degrees in Electrical Engineering from Federal University of Pernambuco (UFPE), in 2003 and 2005, respectively. In 2006, he received the best Brazilian Thesis award in Electrical Engineering. His interests are related to optical networks, swarm intelligence, evolutionary computation, multi-objective optimization and biomedical applications. He is currently an associate professor at Polytechnic School of the University of Pernambuco. He is the head of the research division of the Polytechnic School of Pernambuco. He is a research fellow of the National Research Council of Brazil (CNPq). He also coordinates the Masters course on Systems Engineering.