Optimal Placement of Antennae in Telecommunications Using Metaheuristics

E. Alba, G. Molina

March 24, 2006

Abstract

In this article, several optimization algorithms are applied to solve the radio network design problem (RND). The task is to find the best set of transmitter locations and their best configuration parameter values in order to offer coverage to a given geographical region at an optimal cost. This problem is of high interest in today GSM/GPRS, UMTS and in general ad hoc wireless networks, and is also related to many problems arising in sensor networks. A simulated annealing algorithm is developed along with a standard genetic algorithm and a CHC algorithm to understand the basic behavior and needs of this problem; three classes of problem are also considered, each one using one different antenna technologies: ideally-squared, omnidirectional and sector antennae; instances of several sizes are studied for all classes to asses the scaling properties of the algorithms as the problem dimension grows. Results show that all three algorithms manage to solve the problem satisfyingly, being CHC the fittest algorithm in both effectiveness and efficiency.

1 Introduction

Mobile communications is a major area in the telecommunications industry of the twenty-first century. As customers get used to have mobility besides connectivity this kind of services is more and more required. Mobile communications require the use of a mobile device by the end user, the presence of an access network accessible by the mobile device from any place the user has to be, and a backbone network that manages the connections and communications between different users.

Also, ad hoc and sensor networks need to define a cluster responsible for communications to take place, in a dynamic and novel way of assigning data transfer functionalities to a given terminal.

Recently, numerous companies have entered this area and began to compete in order to offer the best services at lowest cost. Therefore, a great number of issues have arisen as problems to be solved in order to optimize the features of the service.

In this work will develop some techniques for solving one major issue of the problem: the design of the access network. As mentioned, this could be needed statically for regular cellular phone networks and dynamically for networks with no infrastructure (such as ad hoc or sensor) hence results could be of very high interest in academics and industry. A generally accepted method for building an access network is to divide the terrain to be covered into small cells, each of which can be covered by a single transmitter conveniently located in a base station (BS); this solution is known as a radio network. The problem we solve is how to achieve maximum coverage of the terrain in order to obtain a valuable service for the customer (ideally the coverage should be complete) by placing the lowest number of transmitters, so that the cost of the service remains competitive. This is equivalent to selecting the optimal positions for placing the transmitters, and this problem is known as the radio network design problem (it will often be referred to as the RND problem).

Our contribution in this work is to compare several algorithms on the same large set of instances to highlight their different advantages. Also, we will include several instances of each class of problem, while addressing three of such classes, one for every type of transmitter type. Finally, we also contribute to research by analyzing the scalability of algorithms, avoiding any bias to just one instance and will show a solid statistical analysis to sustain our claims.

This work is organized as follows: in the next section a quick review of the state of the art in optimization for telecommunication problems and particularly RND problem will be shown. In section three, a formal description of the RND problem will be presented. Section four will describe the principles of the employed optimization techniques for solving the problem. In section five we will discuss the results obtained during the different tests. Finally, the conclusions are presented in section six.

2 State of the Art

A large number of problems have had to be solved in order to bring telecommunication services to a wide public. Most technical and technological problems were faced during the early ages of this industry, which have culminated in the present telecommunication means. Now, the industry is reaching its maturity, and a new kind of problems have to be solved: optimization problems. The technology is well known, now the goal is to obtain the best possible way to administrate the resources so as to get the maximum efficiency.

A large variety of optimization problems have been faced in the telecommunications field, and also a wide variety of techniques have proven to be useful in those tasks. Most recent work in this area has got really good results in real scenarios, and has encouraged further investigation.

In [4], genetic algorithms are applied to optimize telecommunication networks. A set of network nodes and end users are given as input, and the algorithm has to determine which nodes will be multiplexers, which ones will be exchanges, and what links between nodes are to be set up. The objective is to build a minimum cost network that satisfies the requirements. The results obtained by the genetic algorithm surpass the ones obtained by a standard heuristic algorithm.

A clustering problem is treated in [1]: the well known location area management. With the aim of clustering a cellular mobile radio network so as to obtain minimum amount of roaming information (determined by the number of cells in a frontier and the number of users travelling between different clusters), an iterative algorithm is developed and applied. Good results are obtained in very short times.

Some algorithms for determining the coverage of an ad-hoc sensor network are proposed in [7]. Based on graph techniques, these methods achieve to obtain an estimation of the best-case and worst-case coverage with an optimal polynomial effort.

Also, a lot of work has been done in radio network design. In [6] the focus of the study is the parameterization of the base stations in a determined area, whereas in [2] and [8] the main objective is to locate the base stations, so as to obtain high coverages at low costs. Each problem provides a set L of locations for the base stations, a set P of parameters that control the behavior of every base station, and a set T of test points where coverage is checked in order to determine the quality of the service provided by the network. In [6], L is a list of the locations where the base stations are placed, P is the list of parameters that have to be optimized (in this case, orientation and tilt of the sector antennae), and coverage is measured with the help of *domain areas*, which are geographical regions determined by the position of the base stations. Both in [2] and [8], L is a list of locations where base stations can be placed or not, P is the set of parameters that have to be optimized (type of antenna, emitted power, orientation, tilt, etc) and T is a list of discrete points where the received signal has to be calculated in order to determine the degree of coverage.

To solve the optimization problem, an iterative algorithm is proposed in [6], which geographically partitions the problem according to the domain areas and solves iteratively the resulting subproblems using a genetic algorithm. A signal to noise criterium (determined by free-space propagation model) is employed to calculate the coverage regardless of signal interference, for the sake of simplicity. This technique is faced to a global genetic algorithm for the whole problem (without partition) and to a random search technique, and improves the results obtained by both.

A similar problem is treated in [8], where a genetic algorithm is also employed. A discrete set of locations is offered for placing base stations, and a set of reception points is used, divided in three hierarchical categories, depending of the purpose: testing of the signal quality (R), testing of the expected service (ST) and traffic requirements (T). The objectives are to minimize the number of used sites while the amount of traffic held by the network is maximized, and a set of constraints is defined over ST which have to be satisfied. A multiobjective genetic algorithm is used to solve the problem, and a realistic highway area is used for evaluating the proposed technique. The results obtained when using a sharing technique along with the genetic algorithm are better than those obtained with the genetic algorithm alone in terms of quality (Pareto-dominance) and entropy (diversity). The sharing technique consists in applying a penalty to each solution depending on the number of close neighbors (considering a predefined distance in the space of solutions of the problem) it has in the actual solution population, thus promoting solutions that have few neighbors and contributing to the population diversity.

In [2] the location of stations is investigated for UMTS. The third generation for mobile telecommunications requires a different approach since its features allow for more flexibility in its use. The cell capacity is not limited *a priori* -resources are shared all over the network- and the main limitation is interference, therefore a capacity study ought to be made. Hata's propagation model is used to deal with realistic instances over a rectangular service area where a set S of candidate sites is defined and another set TP of test points is randomly determined. Two kinds of power control mechanisms (PC) are considered: the power-based and the SIR-based, and two kinds of techniques are employed: greedy procedures (direct and reverse) and a taboo search algorithm (TS). Experimental results show that TS behaves better than the greedy procedures, although differences are not too strong.

In short, greedy, GAs, TS and in general heuristic methods seem the best tools to solve this problem. We show a chronological summary of all these achievements in Table 1 and pass to make our new proposals.

3 The Radio Network Design Problem (RND)

The radio coverage problem amounts to covering an area with a set of transmitters. The part of an area that is covered by a transmitter is called *a cell*. A cell is usually disconnected. In the following we will assume that the cells and the area considered are discretized, that is, they can be described as a finite collection of geographical locations (taken from a geo-referenced grid, for example). The computation of cells may be based on sophisticated wave propagation models, on measurements, or on draft estimations. In any case, we assume that cells can be computed and returned by an *ad hoc* function.

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 and let be the

Date	Article	Ref.	Comments	Proposed Techniques
2003	Efficient Radio Network Optimization	[6]	Considers coverage of the whole terrain, but BS placement is poorly managed	Hybrid iterative algorithm using GA for local search
2003	Planning UMTS Base Station Location	[2]	Specialized in third generation mobile telephony (UMTS), lacks a general vision of the problem	Taboo Search tested against Greedy procedures
2000	A MultiObjective Genetic Algorithm for Radio Network Optimization	[8]	Evaluates only one technique, high terrain discretization	Genetic Algorithm

Table 1: Existing work on RND in the litterature

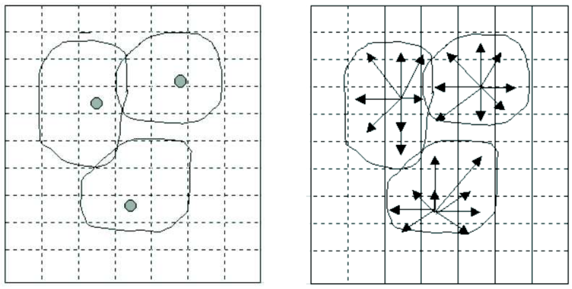

Figure 1: (left) Three potentially transmitter locations and their associated covered cells on a grid, and (right) graph representing covered locations.

vector $\vec{x}$ a solution to the problem where $x_i \in \{0, 1\}$, and $i \in [1, |M|]$ indicates whether a transmitter is being used or not. As the geographical area needs to be discretized, the potentially covered locations are taken from a grid, as shown in the Figure 1.

Throughout this work we will consider different versions of the RND problem, which will differ in the type of antennae that might be placed in every location. There will be simple versions with antennae that will require no parameters to determine its coverage, and more complex versions in which antennae will require some parameters (i.e. direction) to determine the area covered by it.

Searching for the minimum subset of transmitters that covers a maximum surface of an area comes to searching for a subset $M' \subseteq M$ such that $|M'|$ is minimum and such that $|Neighbors(M', E)|$ is maximum, where

$$Neighbors(M', E) = \{u \in L \mid \exists v \in M', (u, v) \in E\}. \tag{1}$$

$$M' = \{t \in M \mid x_t = 1\}. \tag{2}$$

The problem we consider recalls the unicost set covering problem (USCP) that is known to be NP-hard. The radio coverage problem differs, however, from the USCP in that the goal is to select a subset of transmitters that ensures a *good* coverage of the area and not to ensure a *total* coverage. The difficulty of our problem arises from the fact that the goal is twofold, no part being secondary. If minimizing was the primary goal, the solution would be trivial: $M' = \emptyset$. If maximizing the number of covered locations was the primary goal, then problem would be the USCP. An objective function $f(\vec{x})$ to combine the two goals has been proposed in [3]:

$$f(\vec{x}) = \frac{CoverRate(\vec{x})^{\alpha}}{Number\ of\ transmitters\ selected(\vec{x})}. \tag{3}$$

where

$$CoverRate(\vec{x}) = 100 \cdot \frac{Neighbors(M', E)}{Neighbors(M, E)}. \tag{4}$$

the parameter α can be tuned to favor the cover rate item with respect to the number of transmitters. Just like Calégari et al. did[3], we will use $\alpha = 2$, and 287×287 point grid representing an open-air flat area.

In every instance of the problem taken here a specific kind of transmitter with an associated cell may be placed in each location. Depending on the shape of the cell the optimal solution will differ. For every instance the optimal solution can be determined and offered as a possible solution to evaluate the quality of the solutions computed by the algorithm. This shall be done by locating the set of optimal locations.

In the first instance, 49 primary transmitter locations are distributed regularly in this area in order to form a 7×7 grid structure to be the optimal set of locations, and each transmitter has an associated 41×41 point cell.

Consequently, the obtained coverage would be total if the algorithm happens to assign one transmitter to these optimal locations known beforehand. A hundred complementary transmitter locations have been then randomly added, associated to 41×41 point cells. By construction, the best solution with total coverage is the one that covers the area with the 49 primary transmitters (giving an optimum fitness value 204.082).

In the second instance, 61 transmitter locations are distributed following an hexagonal grid, and each transmitter has an associated circular cell of radius 22 points (so that the area of the circular cell is similar to that of the square cell of the previous instance). The type of transmitter used in the first two instances do not require any parameter to determine the shape of the associate covered area (the cell). This implies that any solution to the problem consists only of a subset of selected locations for placing transmitters. They represent an academic benchmark and the industry problem of omnidirectional antennae.

The third instance includes parameter-dependent transmitters, more specifically directive coverage ones. Those transmitters have a cell coverage with the shape of a circle sector, with an angle of 60^0. Thus, six of them are required to completely cover an omnidirectional cell. Each transmitter requires a direction to point as a parameter to determine which area it gives coverage to. There are six allowed directions for every single directive transmitter, equally spaced through the 360^0 by steps of 60^0, so that any point (close enough to the transmitter) can be covered by one and only one configuration of the transmitter.

4 Algorithms

The primary goal of this work is to determine which algorithm among the ones analyzed works best for this kind of problem. The comparison will be made between *simulated annealing* (SA), CHC, and a standard *genetical algorithm* (GA). Also, the parameters of the algorithm will be tuned depending of the particular instance of the problem, looking forward to establishing some relationship between the instance of the problem and the optimal values for the algorithm's parameters.

In the following we will briefly describe the algorithms employed in this work.

4.1 Simulated Annealing

The first algorithm we use to solve the problem is known as simulated annealing, or SA. This algorithm works with a single complete solution of the problem at any time, and proceeds to improve it iteratively *moving* towards the optimal solution.

The pseudocode for this algorithm is shown in Figure 2.

Procedure Simulated Annealing

```
begin
    Initialize(T,t,Sa)
    while not end_condition(t,Sa)
        while not cooling_condition(t)
            Sn := Choose_neighbor(Sa)
            Evaluate(Sa,Sn)
            if Accept(Sa,Sn,T)
                Sa := Sn
            end if
            t := t+1
        end while
        Cooldown(T)
    end while
end
```

Figure 2: Pseudocode for Simulated Annealing.

As can be seen in the code, the algorithm keeps a single solution S_a. In every iteration, a new solution S_n is made from the old one, S_a, and depending on some acceptance criterion, it might replace it.

The acceptance criterion is the true core of the algorithm. It works as follows: both the old (S_a) and the new (S_n) solutions have an associated quality value - determined with a *fitness* function applied during the evaluation of the solutions. If the new solution turns out to be *better* than the old one, then it shall replace it. If it is *worse* there is still some chance that it will replace it: the replacing probability is calculated from the quality difference between both solutions and a special control parameter T, namely the *temperature*.

This way the algorithm is biased towards finding better and better solution until ultimately reaching the best one. The acceptance criterion ensures a way of escaping local optima by choosing solutions that are actually worse than the previous one with some probability. The probability of choosing a *bad* solution over a *good* one is larger the larger T is, and is smaller the bigger the quality difference between them is. The probability of choosing a solution that is actually worse is calculated using the Boltzmann's distribution function:

$$P = \frac{2}{1 + e^{\frac{fitness(S_a) - fitness(S_n)}{T}}} \tag{5}$$

As can be seen in Equation 5, the probability of choosing S_n over S_a is 100% if both have the same value of fitness, and decreases as the fitness value for S_n gets lower than the one for S_a.

As iterations go on, the value of the temperature parameter is progressively reduced following a cooling schedule, thus reducing the probability of choosing worse solutions and increasing the biasing of the algorithm towards good solutions.

4.2 CHC

The second algorithm that will be employed is Eshelman's CHC. It is a population based method, which works with a set of solutions (*population*) at any time. The algorithm also works iteratively, producing new solutions at each iteration, some of which will be placed into the population instead of others that were previously included.

The pseudocode for this algorithm is shown in Figure 3.

Procedure CHC

```
begin
    t:=0
    Initialize(Pa,distance)
    while not ending_condition(t,Pa)
        Parents := Selection_parents(Pa)
        Offspring := HUX(Padres)
        Evaluate(Pa,Offspring)
        Pa := Elitist_selection(Offspring,Pa)
        if not_modified(Pa)
            distance := distance-1
            if (distance == 0)
                Restart(Pa)
                Initialize(distance)
            end if
        end if
        t := t+1
    end while
 end
```

Figure 3: Pseudocdigo del CHC.

The algorithm CHC works with a population of solutions that we will refer to as P_a. In every step, a new set of solutions is produced by selecting pairs of solutions (the parents) and recombining them. This selection is made in such a way that individuals that are too similar can not mate each other, and recombination is made using a special procedure known as HUX. This is done in order to preserve the maximum amount of diversity in the population, as no diversity is introduced during the iteration. The next population is formed by selecting the best individuals among the old population and the new set of solutions (elitist criterium).

Because of this, at some point of the execution convergence is achieved, so the normal behavior of the algorithm should be to stall on it. A special mechanism is used as a way of producing diversity when needed: a *restart* mechanism. This works as follows: all of the solutions but the very best ones are significantly modified (*cataclysmically*). This way, the best results of the previous work are maintained and the algorithm can proceed again.

4.3 Genetic Algorithm

Genetic Algorithms (GAs) are archetypical evolutionary algorithms. They include, in major or minor grade, all the characteristics of evolutionary algorithms. A pseudocode for a genetic algorithm is shown in Figure 4.

A genetic algorithm works with a population of solutions and iteratively produces new solutions (*offspring*) from the old ones, adds new diversity to them, and finally selects

Procedure Genetic Algorithm

```
begin
    t:=0
    Initialize(Pa)
    while not ending_condition(t,Pa)
        Parents := Father_selection(Pa)
        Offspring := Reproduction_operator(Parents)
        Offspring := Diversity_operator(Offspring)
        Evaluate(Pa,Offspring)
        Pa := Population_selection(Offspring,Pa)
        t := t+1
    end while
end
```

Figure 4: Pseudocode for a Genetic Algorithm.

the individuals for the next population from the old population and the newly made solutions. Genetic algorithms exist in countless versions, depending on the selection methods used for both selecting the parents and the new population, the operators for making new solutions, and the values of the parameters used by all these methods and operators.

In this work we shall reproduce the results obtained in the previous work [5], using the parameter values shown in Table 2.

Population Size	*512*
Selection	*roulette wheel*
Crossover	*dpx prob = 1.0*
Mutation	*bit-flip prob = 0.00671*
Replacement	*least fitted*
Stop Criterion	*find a solution*

Table 2: Parameters of the algorithm being used.

A slight variation will be made in the study. Whereas in [5] the GA used was a steady-state one (only one offspring per iteration) in this work we will compare two kinds of GA. One will be the steady-state, the other one will be a generational one, in which the number of offspring produced in every iteration will be equal to the population size (512 offspring per iteration). This is intended to be a simple way to determine whether the granularity of the execution has some importance over the algorithm's performance.

5 Tests and Results

In this section we present the results of performing an assorted set of tests by using the three described algorithms to solve the RND problem.

We will solve all kind of instances (varying in the type of transmitter used) with different sizes. The size of an instance is the number of transmitter locations that can be selected by the algorithm for placing those transmitters. The primary instance will have 149 transmitter locations, and there will be bigger instances increasing by steps of 50 locations until reaching the final 349 transmitter location size. We will focus on the number of visited locations and wall clock time required by the algorithm to measure the effort needed to solve the problem.

In the primary instance a parametric study will be made for the SA and CHC algorithm, determining how their execution's parameters influence the algorithm's performances, while the GA algorithm will be let in a standard configuration for comparing against the first two. Each result will be averaged over 50 independent runs.

The parameters and their ranges are shown in Table 3.

Algorithm	SA
Independent runs	50
Number of evaluations[1]	2500000
Markov-chain length	50
Temperature decay	0.9∼0.9999
Probability of mutation(%)	0.5∼97

Algorithm	CHC
Independent runs	50
Number of evaluations[1]	2500000
Crossover probability	0.8
Diverge(%)	1∼35
Size of population	50∼2000

Algorithm	GA
Independent runs	50
Number of evaluations[1]	2500000
Size of population	512
Size of offspring	1 ∼ 512
Selection of parents\offspring	roulette\best
Probability of mutation(%)	0.671∼1

Table 3: Values of the parameters of the algorithms.

The results for each experiment will be considered to be the mean number of visited solutions among the 50 independent runs (and the time spent during the execution) as well as the percentage of successful executions (the proportion of executions that actually reach the optimal solution of the total number of executions). Thus we consider both the effort and the effectiveness of the algorithm.

Also, a statistical analysis of the data will be realized in order to confirm that the differences between the algorithms are really meaningful and not a random dropout of a non-deterministic process. The results of that analysis will be mentioned only when they add some important information to the comments.

5.1 RND with Squared shaped Cell Transmitters

The first instance of the problem is solved using all algorithms. Here we analyze the importance of parameters for CHC and SA since for GA the have been studied elsewhere [5].

In this instance, each transmitter placed offers coverage to a square shaped area centered in the location site. Figure 5 shows an example of a terrain with such transmitters placed.

5.1.1 Parametrical study for the basic instance (149 transmitter locations)

The results of the experiments are shown in tables 4 and 5 for the algorithm SA. The perturbation parameter is hold while the decay has been swept in the experiments shown in Table 4; then the best value found for decay has been kept and the perturbation parameter has been swept in the study shown in Table 5.

During our studies, both the decay of the temperature and the perturbation have proved to be important parameters for the execution of the algorithm. Considering the ranges of variation employed, the larger variations of performance occur when changing the probability of mutation rather than the decay.

The best combination found for SA is a value of decay $\alpha = 0.99$ and a perturbation of $p = 1\%$. Using this configuration, SA can solve the RND at full reliability in 68.76 seconds after making 86,781 fitness function evaluations. Any distance from this configuration

[1]Fitness function evaluations

Figure 5: Sample terrain coverage offered by square shaped cell transmitters

Decay	***Fitness***	**Evaluations**	**Time (sec)**	**Hit ratio (%)**
0.9	204.082	118085	185.09	100
0.98	204.082	101798	82.17	100
0.99	204.082	**86781**	**68.76**	100
0.995	204.082	92231	75.49	100
0.999	204.082	92203	76.96	100
0.9995	204.082	113226	96.77	100
0.9999	204.082	287487	266.59	100

Table 4: Performance of SA algorithm depending on the decay for a perturbation of 1% (Evaluations).

Perturbation (%)	***Fitness***	**Evaluations**	**Time (sec)**	**Hit ratio (%)**
0.5	204.082	202250	159.87	100
1	204.082	**86781**	**68.76**	100
1.5	204.082	91650	73.09	100
2	204.082	106505	85.11	100
3	204.082	205349	168.11	100
4	204.082	496175	400.26	100
5	203.068	872289	707.89	56
25	133.940	2500000	-	0
50	116.859	2500000	-	0
97	103.359	2500000	-	0

Table 5: Performance of SA algorithm depending on the mutation for a decay of 0.99. (Evaluations)

will result in a loss a reliability (almost 50% reliability is lost if the perturbation is set at 5% instead of 1%), efficiency, or both.

The results of the experiments with the algorithm CHC are shown in tables 6 and 7. Table 6 shows the performance of CHC depending on the values used for the divergence and the population size whereas Table 7 presents the hit percentages obtained for those values.

We can conclude from the results that the size of the population is a much more

div\pop.	50	100	200	400	600	800	1200	2000	2800	10000
0.01	639055	483867	334925	53291	38339	40015	55992	91600	127288	441600
0.05	-	110006	80479	33614	34223	42255	56712	92240	125944	437400
0.15	-	135245	70995	**30319**	34043	41023	56400	91280	127344	441400
0.25	-	138864	82037	50805	39683	48415	57024	90840	126056	442000
0.35	188193	181370	71200	50429	31667	40223	55656	91280	126952	440600

Table 6: Performance of CHC algorithm (Evaluations).

div\pop.	50	100	200	400	600	800	1200	2000	2800	10000
0.01	74	100	100	100	100	100	100	100	100	100
0.05	-	100	100	100	100	100	100	100	100	100
0.15	-	100	100	100	100	100	100	100	100	100
0.25	-	100	100	100	100	100	100	100	100	100
0.35	100	100	100	100	100	100	100	100	100	100

Table 7: Effectiveness of CHC algorithm (Hit ratio(%)).

relevant parameter than the divergence. The best values are 400 for the size of the population with a divergence of 15%, resulting in a number of solutions visited of 30,319 showing 100% hit ratio. Bigger populations result in progressively greater required effort but has no effect on the quality of the solution produced by the algorithm (as long as we allow CHC the perform that effort). Smaller populations quickly result in bigger solving effort, and often in worse solution quality.

5.1.2 Scalability Study

In this section we will study how the algorithms react to an increase in the problem's dimension. We want to find out the robustness of exactly the same algorithm when faced to unseen instances of large dimensions. Hence, we will be able of better understanding the importance of our final claims. Starting from the basic instance of 149 we will consecutively solve instances of 199, 249, 299 and 349 transmitter locations.

There is a twofold objective for this study:

- Analyze the relationship between the size of the problem and the effort required to the algorithms.
- Analyze the relationship between the size of the problem and the best values for the parameters of the (SA and CHC) algorithms.

In order to achieve both objectives, a quick parametrical study will be done for every problem size in order to find a reasonably good estimation of both the best combination of values for the parameters and the slightest effort required to solve the problem. The results are shown in Table 8 and illustrated in figure 6. .

As expected, the effort required to solve the RND increases when the size of the instance grows, no matter which algorithm we are employing. Both SA and CHC manage to solve the RND for all instance sizes with 100% hit ratio. None of the implemented GA has been able to do so, given the restrictions made upon the algorithms's executions (5 million evaluations at maximum).

CHC has always got the best effort results, roughly solving RND at 50% of the effort required with SA, considering the best parametric configurations for both algorithms. Both of our genetic algorithms and the distributed genetic algorithm from Chicano & Alba [5] achieve efforts that are several orders of magnitude worse than those obtained by tuned SA and CHC. We can state therefore that CHC is the best algorithm to solve the RND problem, when squared coverage antennae are to be used.

Numerical analysis has shown that the number of evaluations required for CHC to solve the problem grows as a cubic function on the size of the instance.

Algorithm	Size	Fitness	Effort (evaluations)	Normalized effort	Time (sec)	Normalized time
SA	149	204.082	86761	1.00	68.76	1.00
SA	199	204.082	196961	2.27	158.32	2.30
SA	249	204.082	334087	3.85	282.98	4.12
SA	299	204.082	637954	7.35	541.11	7.87
SA	349	204.082	810755	9.34	729.61	10.61
CHC	149	204.082	**30319**	1.00	**25.59**	1.00
CHC	199	204.082	**78624**	2.59	**76.66**	3.00
CHC	249	204.082	**148595**	4.90	**146.21**	5.71
CHC	299	204.082	**228851**	7.55	**237.38**	9.28
CHC	349	204.082	**380183**	12.53	**427.81**	16.72
ssGA	149	204.082	239305	1.00	525.50	1.00
ssGA	199	204.082	519518	2.17	1181.79	2.25
ssGA	249	204.082	978573	4.09	2496.94	4.75
ssGA	299	204.082	1872463	7.82	5219.36	9.93
ssGA	349	204.034	3460110	14.46	10461.82	19.91
gGA	149	204.082	141946	1.00	118.19	1.00
gGA	199	204.082	410531	2.89	346.21	2.93
gGA	249	204.082	987074	6.95	818.62	6.93
gGA	299	204.082	1891768	13.33	1551.39	13.13
gGA	349	204.038	3611802	25.44	3424.11	28.97
dssGA8	149	-	785893	1.00	80.00	1.00
dssGA8	199	-	1467050	1.87	174.00	2.17
dssGA8	249	-	2480883	3.16	378.00	4.73
dssGA8	299	-	2997987	3.81	463.00	5.79
dssGA8	349	-	4710304	5.99	927.00	11.59

Table 8: Results of the scalability study.

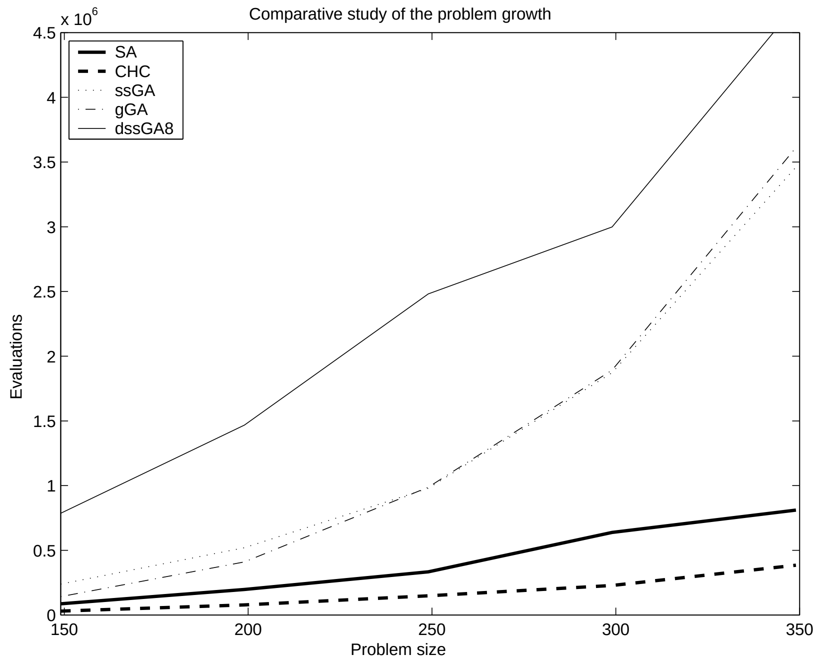

Figure 6: Evolution of the effort required to the different algorithms.

5.2 RND with Omnidirectional Transmitters

To further increase the realism of the problem, the squared-like model for the transmitter coverage is substituted by a realistic model: an omnidirectional transmitter, whose coverage area has the shape of a circle. Figure 7 shows an example of a terrain with such transmitters placed.

Because of the shape of the cell, there is no obvious *best solution* for the problem in

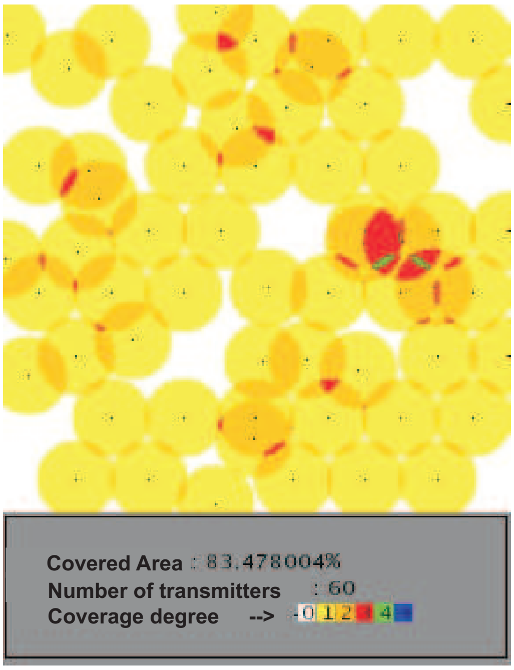

Figure 7: Sample terrain coverage offered by omnidirectional transmitters

this case. As theory says, the fittest solution turned out to be a regular hexagonal net. It is impossible to cover the whole terrain without spilling resources (that is, having some terrain covered by more than one transmitter), so the fitness function has been slightly modified in order to punish that spilling. This has been done by assigning a value to the terrain depending on the coverage degree (the number of transmitters that actually cover that terrain) in such a way that single coverage terrain is the most valuable one, followed by multi-covered terrain, and finally non covered terrain, which has no value.

Let C be the degree of coverage every bit of terrain has, the resulting fitness function is as shown in Equation 6.

$$f(\vec{x}) = \frac{\left(Cover\ Rate(\vec{x})|_{C=1} + 0.5 \cdot \ Cover\ Rate(\vec{x})|_{C>1}\right)^2}{Number\ of\ transmitters} \tag{6}$$

The chosen fitness solution gives terrain covered by more than one transmitter only half the value terrain with coverage by a unique transmitter has. Using this new specification, we will repeat the experiments done for the previous one, and afterwards compare both. This means we will start by making a complete parametrical analysis for SA and CHC algorithms on the basic instance of the problem (149 provided locations), and a after that we will study the behavior of the algorithms when the size of the problem increases, by steps of 50 locations, until having 349 possible locations.

5.2.1 Parametrical Study for the Basic Instance (149 Transmitter Locations)

The results of the experiments are shown in tables 9 and 10 for the algorithm SA. Table 9 shows the analysis done for the decay (maintaining a constant value of 1% for the mutation probability) while in Table 10 we keep the best value found for decay and show the analysis done for the mutation probability.

Decay	*Fitness*	Evaluations	Time (sec)	Hit ratio (%)
0.9	147.029	174188	162.75	78
0.99	147.568	86883	81.98	94
0.995	147.380	95217	90.13	88
0.996	147.442	79523	75.81	90
0.997	147.693	**83175**	**79.69**	98
0.998	147.693	97142	93.48	98
0.999	147.693	127236	123.48	98
0.9991	147.755	135962	131.86	100
0.9992	147.708	128627	135.70	98
0.9993	147.755	155551	151.60	100
0.9994	147.755	162444	158.25	100
0.9995	147.755	184848	194.59	100
0.9999	147.755	613066	598.62	100

Table 9: Performance of SA algorithm on RND with omnidirectional transmitters depending on the decay for a mutation of 1%.

Mutation (%)	*Fitness*	Evaluations	Time (sec)	Hit ratio (%)
0.25	147.630	478185	414.53	96
0.5	147.755	136511	119.86	100
1	147.693	**83175**	**79.69**	98
1.5	147.755	107363	95.46	100
2	147.755	95456	85.20	100
3	147.630	191834	171.20	96
4	147.630	683079	611.25	96
5	147.065	1249574	1126.74	80

Table 10: Performance of SA algorithm on RND with omnidirectional transmitters depending on the mutation for a decay of 0.997.

Both parameters have proven to be important for the performance of SA. We can now better appreciate the influence of the decay (and not only the probability of mutation). The best values found are 0.997 for the decay, and 1% for the probability of mutation. Using this parametric configuration, SA can solve the RND problem with omnidirectional antennae with 98% reliability in 79.69 seconds and making $83,175$ evaluations.

The results of the experiments with the algorithm CHC are shown in tables 11 and 12. Now, both the divergence and the population size are varied within each table, but Table 11 only shows the number of visited solutions resulting while Table 12 shows the hit percentage for each configuration (considering the execution restrictions shown in Table 3).

div\pop.	100	200	400	600	700	800
0.01	543569	496097	209695	125491	105136	50879
0.05	166656	131913	101594	65206	49489	60255
0.15	204437	142029	78836	64546	**45163**	69967
0.25	212815	139158	90819	54707	68137	63567
0.35	293822	212688	125561	60574	59205	73951

div\pop.	1200	2000	4000	6000	10000
0.01	72767	111520	214800	318360	520800
0.05	73175	110480	213520	317400	524000
0.15	72767	110960	214480	318120	522400
0.25	84455	110640	214800	319440	519200
0.35	86039	113240	214720	317040	522400

Table 11: Performance of CHC algorithm on RND with omnidirectional transmitters (Evaluations).

Once again, the size of the population seems to be the most important parameter for CHC. The best results have been obtained for a population of 700 individuals and a divergence of 15%. The average effort needed with the optimal combination of parameters is $45,163$ visited solutions, showing 100% hit ratio.

div\pop.	100	200	400	600	700	800	1200	2000	4000	6000	10000
0.01	66	100	100	100	100	100	100	100	100	100	100
0.05	98	100	100	100	100	100	100	100	100	100	100
0.15	100	100	100	100	100	100	100	100	100	100	100
0.25	96	100	100	100	100	100	100	100	100	100	100
0.35	100	100	100	100	100	100	100	100	100	100	100

Table 12: Effectiveness of CHC algorithm (hit ratio (%)) on RND with omnidirectional transmitters.

5.2.2 Scalability Study

For this new instance of the problem we will repeat the scalability study. The results will be compared to the ones obtained with square-shaped coverage transmitters. The results for this study can be seen in Table 13. The are illustrated in Figure 8.

Algorithm	Size	Fitness	Effort (evaluations)	Normalized effort	Time (sec)	Normalized time
SA	149	147.755	83175	1	79.69	1
SA	199	147.668	**262282**	3.15	**253.59**	3.18
SA	249	147.372	913642	10.98	896.08	11.24
SA	299	147.755	2945626	35.41	3918.92	49.18
SA	349	147.808	6136288	73.78	10346.30	129.83
CHC	149	147.755	**45163**	1	43.71	1
CHC	199	147.755	344343	7.62	366.95	8.40
CHC	249	147.755	**817038**	18.09	**870.82**	19.92
CHC	299	147.755	**2055358**	45.51	**2414.17**	55.23
CHC	349	147.832	**3532316**	78.21	**4009.85**	91.74
ssGA	149	147.567	365186	1	802.48	1
ssGA	199	146.829	1322388	3.62	3169.41	3.95
ssGA	249	144.691	2878931	7.88	7863.59	9.80
ssGA	299	142.075	9369809	25.66	28462.50	35.47
ssGA	349	*141.415*	*9556983*	*26.17*	*30911.70*	*38.52*
gGA	149	147.755	206581	1	189.62	1
gGA	199	147.266	1151825	5.58	1021.03	5.38
gGA	249	145.182	3353641	16.23	3009.53	15.87
gGA	299	144.263	8080804	39.12	7828.76	41.29
gGA	349	141.445	19990340	96.77	19713.43	103.96

Table 13: Results of the scalability study for RND with omnidirectional transmitters.

Once again, CHC manages to solve the problem with 100% hit percentage for all instance sizes. SA gets close, but does not always reach the optimum, producing high quality solutions even when not getting to the optimum nevertheless. The two configurations of GA manage to solve the smallest instances of the problem quite correctly, but perform poorly for instances with sizes greater than 199. In the biggest instances, they get much worse solutions than SA or CHC after running for much longer times.

In terms of measured effort, CHC gets always the best results. Only for the 199 size instance can SA solve the problem performing fewer evaluations, but not as reliably as CHC does. The efforts required by the GAs are many times worse than either CHC's or SA's.

Numerical analysis has shown that the number of evaluations required for CHC grows as a cuadratic function on the size of the instance.

From the studies of RND with square-shaped and omnidirectional coverage transmitters we conclude that in a general manner CHC is the fittest algorithm to solve RND-like problems. During our experiments, CHC has got simultaneously the lowest effort and the greatest effectiveness (100%). Because of this, we claim that CHC is the best algorithm out of the techniques tried here to solve the RND problem.

The study will now be problem-oriented, as we intend to see how the problem's characteristics affect the resolution capability of the algorithm. For the next instances of the problem, complexity will be increased by rising the degrees of liberty of the problem, so no scalability study will be done.

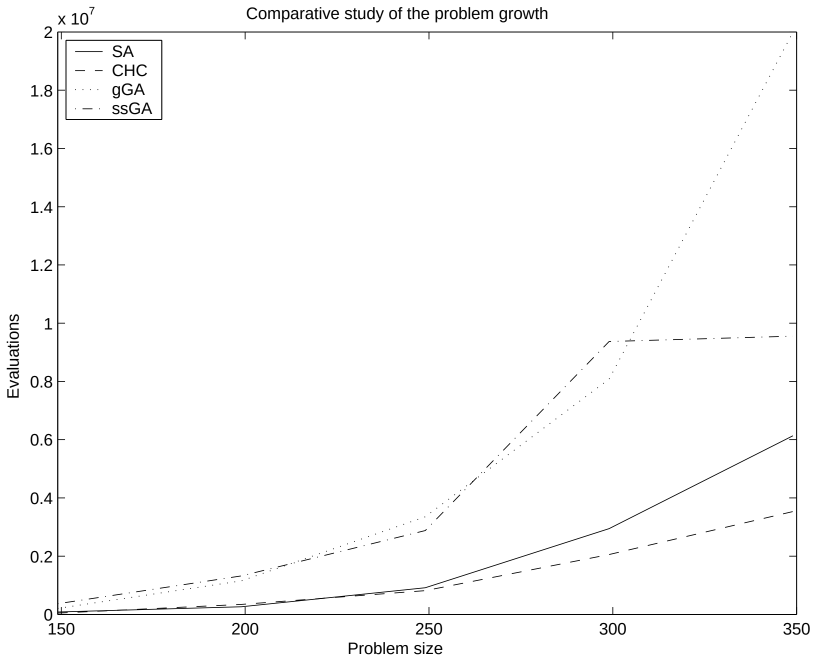

Figure 8: Evolution of the effort required to the different algorithms with omnidirectional transmitters.

5.3 RND with Directive Transmitters

In mobile telecommunications the coverage areas of transmitters have many other shapes out of the ones analyzed here. In general, they are designed to have a directional (sometimes even irregular) coverage, resulting in a sectorial shape (like e.g. in GSM networks). In this section, we will ge a step further into complexity and model our transmitters as directional ones.

Every transmitter used in this section will be assigned a coverage area that amounts to 1/6 of the coverage area of an omnidirectional transmitter. Its shape will be a sector of the circle with an angle of 60^0 and there will be 6 permitted directions, separated by 60^0 also. Several transmitters will be allowed in every location site. To avoid an overgrowth of the space of solutions, we will limit the number of transmitters per location site to 3 or none.

The geometrical features of this instance of the problem hold the same pros and cons than the ones of omnidirectional antennae. Because of this, the fitness function employed in this case will be the same but for one detail: it will count the number of used locations instead of the number of transmitters employed (which is three times bigger). Also, the allowed locations for placing the transmitters will be maintained. Thus, if C is the degree of coverage every bit of terrain has, the resulting fitness function is as shown in Equation 7.

$$f(\vec{x}) = \frac{\left(Cover\ Rate(\vec{x})|_{C=1}\ + 0.5 \cdot\ Cover\ Rate(\vec{x})|_{C>1}\right)^2}{Number\ of\ used\, location\ sites} \tag{7}$$

As the possibility of choosing a direction for the antenna adds one more degree of freedom to the RND problem, several distinct solutions may bring equivalent good results. An initial analysis of the problem has shown us that it can be considered solved for any solution whose fitness value surpasses 75.75. Considering that every location is able to cover only half the area it could cover with omnidirectional transmitters (the ratio between the coverage areas for one omnidirectional antenna to three directive antennas is 2), this value would correspond to fitness value of 151.50 in the previous instance of the problem. This shows that the best solution found using directive antennae is *better* than the best solution found using omnidirectional transmitters.

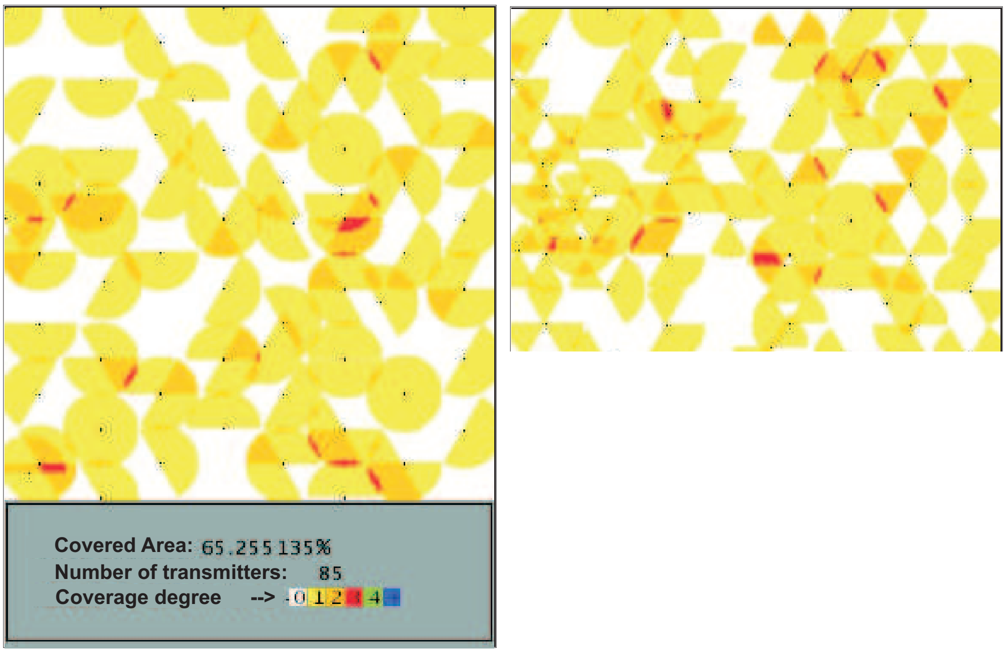

Figure 9: Sample terrain coverages offered by directive antennae. (a) Simple version, (b) Complex version.

We will define two versions of the problem depending on the allowed directions for the transmitters:

- Simple version: the area covered by the three transmitters from one unique location must form a solid half-circle.
- Complex version: the three transmitters from one unique location can point in any direction (out of the 6 possible ones) as long as any two of them do not point in the same direction.

An illustration for each version can be seen in Figure 9.

In the first version, 7 possibilities are offered for each location site: one for every one of the six directions plus another with no transmitter. In the second version, the number of possibilities per location site amounts to 21 (there are 20 different combinations for 6 possible directions if we take 3 at a time plus another with no transmitter).

This means that for a number of permitted locations of 149, the search space size for the problem is:

- $7^{149} = 8.3 \cdot 10^{125}$, for the simple version of the problem.
- $21^{149} = 1.025 \cdot 10^{197}$, for the complex version of the problem.

Those sizes correspond to binary cases (where the only decision to be made is whether a transmitter is placed or not) of **418** and **655** location sites respectively. Thus, the complexity of the instances considered for the RND problem with directive transmitters is greater than the complexity of the same problem with square shaped coverage and omnidirectional transmitters, even for their most complex instances (349 location sites).

Nevertheless, the desired solutions of those instances are not unique. Reconfiguring the hexagonal radio network can be done in many equivalent ways considering the set of available location sites. Every circular cell can be reconstructed in several ways using sectorial cells. In the simple version, each of the 52 cells can be covered in 6 different ways with directive transmitters, in the complex version, there are up to 20 different ways of covering each cell.

If we consider the number of solutions in the solution space for each desired solution, we have:

- $7^{149}/6^{52} = 2.86 \cdot 10^{85}$, for the simple version of the problem.
- $21^{149}/20^{52} = 2.28 \cdot 10^{129}$, for the complex version of the problem.

Those sizes correspond to binary cases of **284** and **430** location sites respectively.

5.3.1 Codification of the Solutions

Up to this point, binary strings have been sufficient for coding any solution of the problem. For each one of the offered locations, a bit indicates whether the will be an antenna placed in it (with a value of 1) or not (with a value of 0). Mutation has been done by bit reversing and recombination and two point crossover and HUX have been directly applied to the binary string.

This is no longer true. The decision adopted for every location is not binary when directive antennas are used. Besides the decision of placing or not an antenna, the direction of the antenna has to be determined.

We will define a *gene* as the codification of the decision take upon one location site. The gene is constituted by three natural numbers (a, b, c) where each number represents either the direction pointed to by one of the directives antennas, if its value is comprised between 1 and 6 (every one of the six directions is assigned a value in this range), or the absence of that antenna, if its value is 0.

Every solution of the RND problem with directive antennae will be coded by a gene string with length equal to the number of available locations.

As the problem restricts the number of antennae per location to three or none, either all three numbers composing one gene are equal to zero, or none is. Every gene is therefore able to code 7 different situations if the problem is of the simple version, and 21 if the problem is of the complex version.

5.3.2 Parametrical Study

As said before, only CHC algorithm is used to solve the RND with directive transmitters. Once again tuning of the algorithm's parameters is done in order to get both the best results and information about the relation between problem complexity and parameter values.

During the study done with square and omnidirectional coverage transmitters the divergence parameter of CHC didn't show to be a key parameter for the performance of the algorithm. Therefore, the parametrical study will now restrict to the size of the population handled by CHC. The divergence will be kept at a value of 35%, which is an accepted standard value that assures that the restarting process is strong enough to ensure sufficient diversity.

All the tests are done for instances with 149 available location sites. Their results considering the simple version of the problem are shown in table 14, those obtained considering the complex version are shown in table 15.

Population	*Fitness*	Evaluations	Time (s)	Hit ratio (%)
2000	75.758	2536057	4435.79	60
4000	75.832	**2383757**	**4186.38**	96
6000	75.804	3064800	5658.63	94
8000	75.852	3999840	7230.46	100
10000	**75.900**	5492000	10260.17	100

Table 14: Results of the study for CHC using directive transmitters (simple version).

Population	*Fitness*	Evaluations	Time (sec)	Hit ratio (%)
2000	74.022	6699409	12978.32	56
4000	74.049	3964632	7751.32	68
6000	74.088	4994156	10130.26	80
8000	**74.106**	**4736637**	9827.60	**88**
10000	74.056	7195598	15430.55	72

Table 15: Results of the study for CHC using directive transmitters (complex version).

As can be seen, CHC is able to solve reliably both cases. In the simple version, the hit percentage attains 100% if the population size surpasses $6,000$ individuals; in the complex version, full hit percentage has not been attained, nevertheless for populations ranging from $6,000$ to $8,000$ individuals it is above 80% and the expected outcome of the algorithm is of very high quality even when it does not find the optimum.

In the simple version, both the expected solution quality and the hit percentage clearly improve as the size of the population grows, from $2,000$ to $10,000$ individuals. However, if we consider the algorithm's effort for solving the problem, the best configuration seems to be a population of $4,000$ individuals, because it gets very high hit percentage (96%) at very low efforts (only $2,383,757$ evaluations are required). We can save over 40% required evaluations (and over 42% computing time) by sacrificing only 4% of the hit percentage if we employ a population of $4,000$ individuals instead of one of $8,000$ individuals, not to mention the memory requirements improvement.

The behavior of CHC is slightly different for the complex version. The outcome of the algorithm improves as the size of the population increases until it reaches $8,000$ individuals, then it gets worse when the populations goes from $8,000$ to $10,000$ individuals: the average fitness shrinks from 74.106 to 74.056 and the hit percentage decreases by 12%. The best tradeoff between outcome quality and required effort is obtained for a population of $8,000$ individuals, with $4,736,637$ evaluations and 88% hit percentage.

In both cases, we can say that an optimum value exists for the size of the population in CHC, and moving away from it makes the algorithm produce worse results, either in the outcome quality, in the effort required, or both.

The values of the tables 14 and 15 are illustrated by Figure 10.

5.4 RND with All Kinds of Transmitter

Finally, in the last version of the problem we permit all kind of transmitters to be placed in any available location site. This will presumably influence the problem in two opposite ways:

- On one hand, the size of the space of solutions will increase, as the possible combinations for each location site grow. This contributes to further complicate the problem.
- On the other hand, the solver has more freedom to solve the problem since more transmitters are available. This should contribute to simplify the process.

The size of the space of solutions is $23^{149} = 7.9 \cdot 10^{202}$, which corresponds to a binary case problem size of 675 site locations. This means that, considering the space of solutions size, this instance is the most complex in our study.

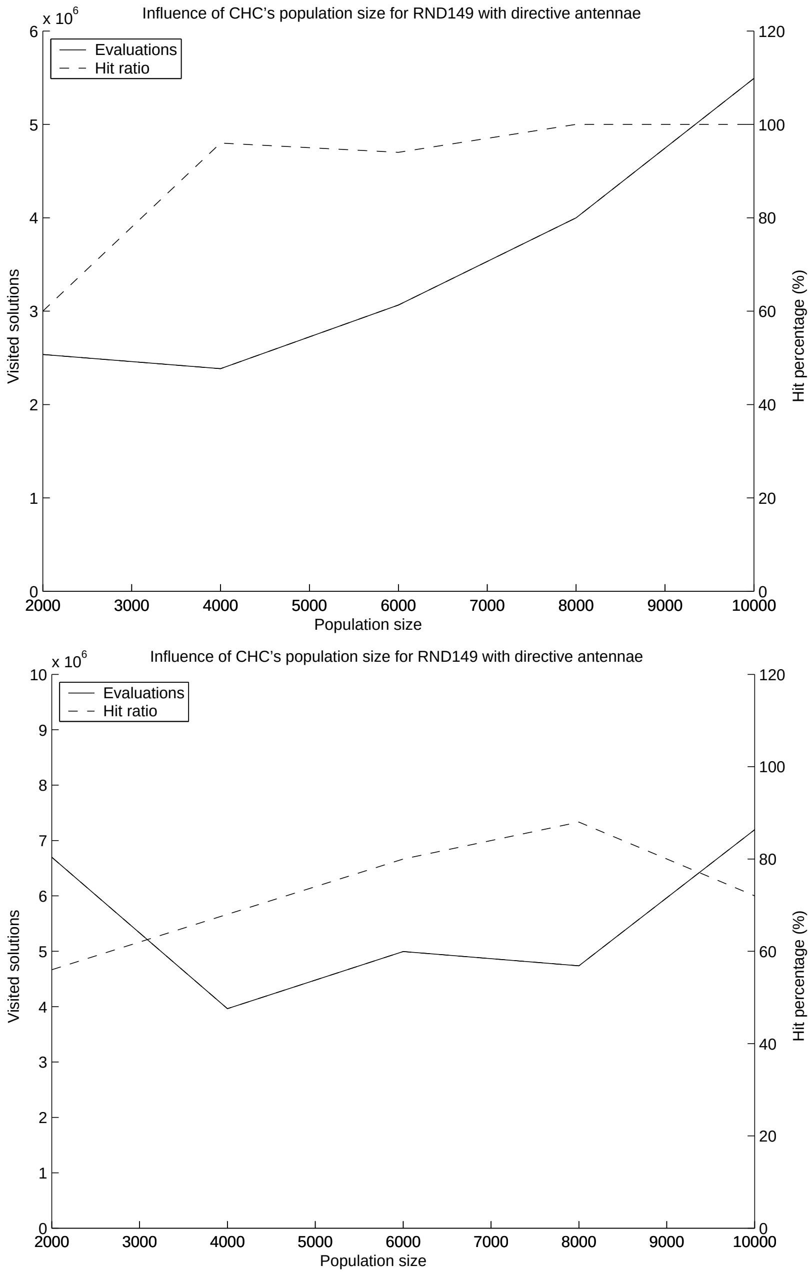

Figure 10: Influence of CHC's population size for solving RND problem with directive transmitters in (top) consecutive directions (bottom) any non-repeated directions.

As we know, the three different kinds of antenna have three different coverage areas (cell). Square and omnidirectional coverage transmitters can offer coverage to the same amount of terrain (only with different shape), but one directive transmitter covers only one sixth of that amount of terrain. Because covering a larger area requires a larger power consumption, it produces higher operating costs for the transmitter that should be considered when designing the radio network.

Thus, in a attempt to give more fairness to the design problem, a new concept is introduced into the fitness function for this version of the problem: *the transmitter cost*.

Every transmitter type has an associated cost which will be the parameter to minimize, instead of simply minimizing the number of transmitters.

The fitness function for this instance of the RND will be as shown in Equation 8.

$$f(\vec{x}) = \frac{\left(Cover\ Rate(\vec{x})|_{C=1} + 0.5 \cdot \ Cover\ Rate(\vec{x})|_{C>1}\right)^2}{Total\ cost\ of\ the\ employed\ transmitters} \tag{8}$$

For our study, we will employ a cost value with a fixed part corresponding to the cost of installing the transmitter, and a proportional part corresponding to the power consumption (which is proportional to the size of the cell corresponding to the antenna). The cost values are shown in Table 16[2].

Transmitter Coverage	**Installation cost**	**Operating cost**	**Total cost**
Squared	0.2	0.8	**1**
Omnidirectional	0.2	0.8	**1**
Directive	0.2	0.4	**0.6**

Table 16: Values of the costs employed for the different transmitter types.

5.4.1 Codification of the Solutions

Neither the binary strings used for RND with square and omnidirectional coverage transmitters, or the gene string used for RND with directive transmitters are adequate for codifying the solutions of this version of RND. We will extend the gene definition to make it suitable for this problem.

A gene will now be defined as follows:

Gene = {*Transmitter type, Parameters*}

The first field can get four different values, whose significations are the following:

- 1: Square coverage transmitter.
- 2: Omnidirectional coverage transmitter.
- 3: Directive coverage transmitter.
- 0: No transmitter placed.

The second field will only be meaningful when a directive transmitter is placed in the corresponding location site. It consists in a directive transmitter gene *(a,b,c)* coding the directions of the three directive transmitter. Only the complex case for directive transmitters is considered in this instance. When no directive transmitter is placed, the parameter field is set to *(0,0,0)* by default.

5.4.2 Parametrical Study

We will carry the same study already performed for the directive transmitter instance, in this last and more complex instance of the problem. Only the 149 size instance will be solved.

The divergence parameter of CHC will be maintained at a 35% value while the population size will be swept from $2,000$ to $10,000$ individuals. The set of locations we

[2]For the case of directive transmitter, the cost shown corresponds to a 3-transmitter-pack instead of a single transmitter

will use for this problem is the one employed for the RND with only omnidirectional transmitters. In that version, the best fitness value found was 147.755. Here, the same solution would bring the same fitness value, and the new transmitters are fairly included, so that no one is cost-effectively superior to the others.

The results obtained for this instance are shown in Table 17. Figure 11 illustrates them.

Population	*Fitness*	Evaluations	Time (sec)	Hit ratio (%)
2000	153.599	1806994	2088.40	17
4000	153.984	1107732	1538.84	57
6000	154.032	574400	751.13	77
8000	154.147	774933	1079.30	90
10000	**154.201**	829333	1284.05	**100**

Table 17: Results of the study for CHC using all kind of transmitters.

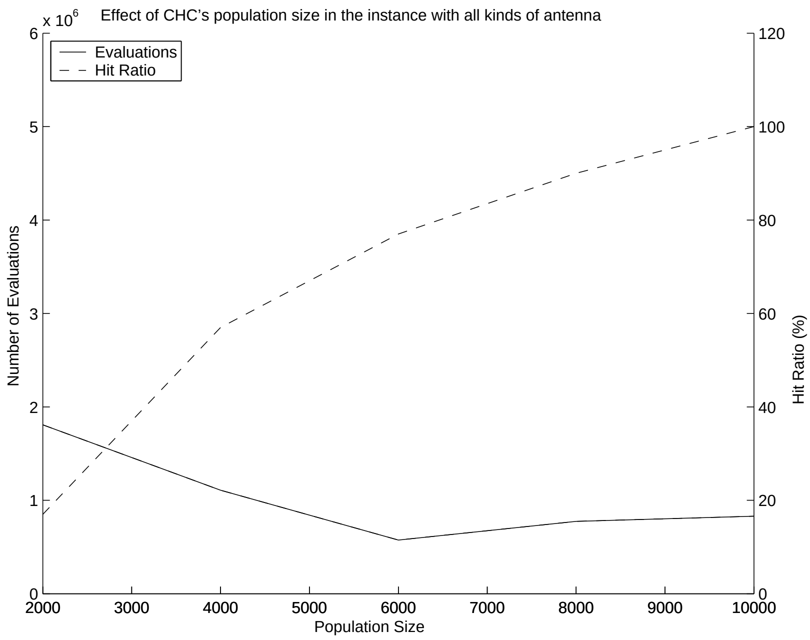

Figure 11: Influence of CHC's population size for solving RND problem with all kind of transmitters.

At first glance we notice that the best solution found by CHC for this instance gets a higher fitness value (154.201) than the best one when only employing omnidirectional transmitters (147.755 for the 149 size instance, 147.832 in the best case for the 349 size instance). The use of more transmitter types help getting better solutions, as expected.

Also, it can be seen that the response of the algorithm improves as the size of the population increases. The best performance is obtained for the highest population size, 10,000 individuals: 100% hit ratio and 829,333 fitness function evaluations.

CHC seems to work best with this instance if the population size is around 10,000 individuals. We can see that there is a stagnation in the effort required for any population greater than 6,000 individuals, and full hit ratio is attained with a 10,000 individual population, so bigger populations[3] are unlikely to improve the algorithm's performance.

[3]Executions with bigger populations couldn't be realized due to computer memory restrictions

5.5 Comparison of Different Problem Instances

In this section we will compare the results obtained by CHC applied to different instances of the problem. Those instances are different both in their formulation (the type of transmitter employed) and in size (number of available location sites). Therefore, they will require different costs from the algorithms to be solved. That means that they are of different *complexity*.

In this section, we will perform a study on the nature of the RND problem (considering it is solved with CHC): the size of the solution space depending on the problem size and the type of transmitter employed, the relationship existing between each instance characteristics, the effort (number of fitness function evaluations) necessary to solve it, and the best parametric configuration of CHC for solving it[4].

The problem instances selected for comparison are the following:

1. RND with square shaped cell coverage transmitters, with sizes 149 and 349. (RND-1).

2. RND with omnidirectional transmitters, with sizes 149 and 349. (RND-2).

3. RND with directive transmitters placed in consecutive directions, size 149. (RND-3).

4. RND with directive transmitters placed in free directions, size 149. (RND-4).

5. RND with all kind of transmitters, size 149. (RND-5).

The effort and hit percentage reached by CHC for the different problem instances is shown in figures 12 and 13, with their values related to the size of the population used in CHC.

We observe the same behavior (with some small differences) for every instance:

- Cost: the cost of resolving the problem improves when the size of the population is increased until it achieves its minimum value, afterwards it increases at a constant rate. This behavior is not observed in the RND-2 instance of size 349, since the cost is still decreasing when the maximum population size is reached.

- Hit percentage: the hit percentage of CHC algorithm increases as the population size increases until reaching full hit ratio (100%), then stays unchanged. There is one exception seen in RND-4 when shifting the size of the population from $8,000$ to $10,000$, the effectiveness falls from 88% to 72%.

When no other information is available, the first way to get some knowledge about the complexity of a problem instance is from its space of solutions, specially its size: the bigger the size of the solution space, the more complex the instance is.

Furthermore, the more complex a problem instance is, the larger the cost needed to solve it.

Besides that, a complex problem usually has many local optima, thus the algorithm employed to solve it will require a big capability of escaping such optima. In CHC this capability is mainly determined by the size of the population (and the divergence parameter which is not studied here for reasons previously explained), meaning a more complex problem will be better solved with a bigger population.

We have hence three ways of measuring the complexity of an instance of a problem:

- The size of the solution space.

[4]Only the population size will be studied, because tuning the divergence parameter does not produce a clear improvement on the algorithm's performance. The former will be kept at a value of 35%

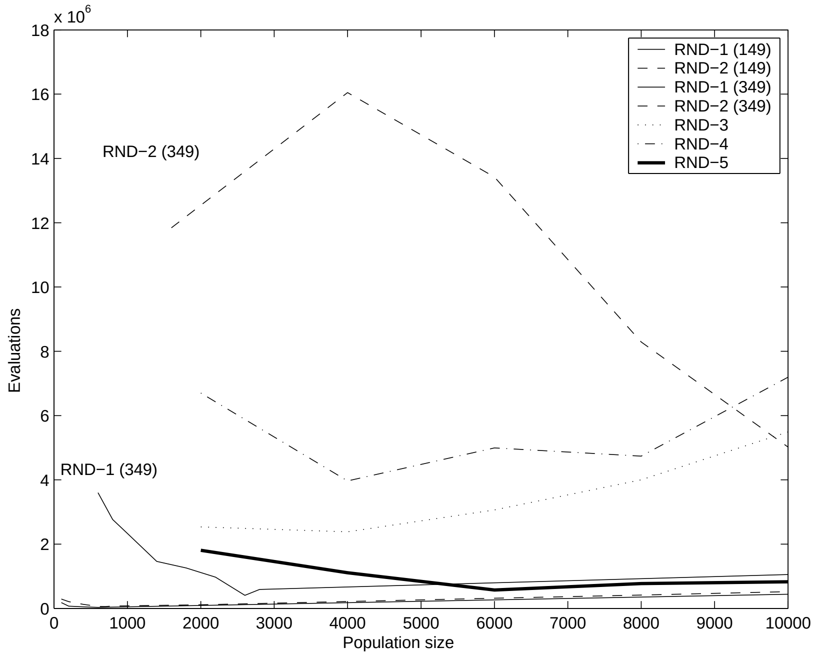

Figure 12: Relation between CHC's population size and RND solving cost (fitness function evaluations) for several problem instances.

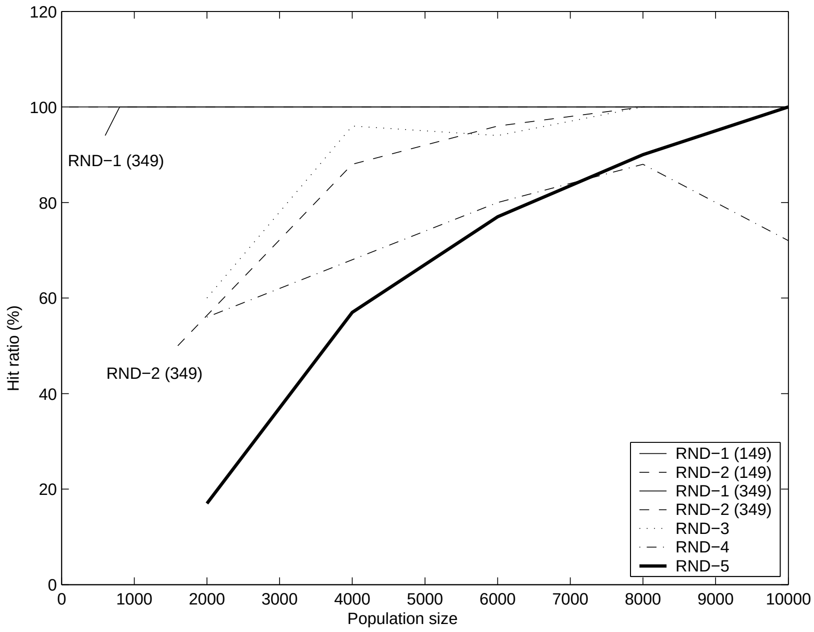

Figure 13: Relation between CHC's population size and RND solving hit percentage for several problem instances.

- The number of fitness function evaluations necessary to solve it (cost).
- The optimum size of CHC's population for solving it.

These three measures are presented for the different selected instances in Table 18.

Instance	Binary size	Best cost achieved	Optimal size of population
RND-1 (149)	149	30319	400
RND-2 (149)	149	45163	700
RND-1 (349)	349	380183	2800
RND-2 (349)	349	3532316	10000
RND-3	418 (284)	2383757	4000
RND-4	655 (430)	4736637	8000
RND-5	675	829333	10000

Table 18: Complexity measures for different RND instances solved with CHC.

At first glance we expect some relationship between the measures of the complexity. Any increase in one of those values is expected to be followed by another increases in the other two. This can be thought of as an intuitive rule for the problem complexity. Clearly this rule does not apply, at least not in every case, as we explain next.

Between different instances of the same problem kind (using one specific kind of transmitter), the rule applies. It can be seen both in RND-1 (square shaped cells), RND-2 (circular shaped cells) and even between RND-3 and RND-4, that an increase in the number of location sites produces an increase in both the cost and the size of the optimum population in CHC. In RND-1 the instance with 149 available location sites is best solved with a population of 400 individuals evaluating $30,319$ solutions, while the instance with 349 locations requires a population of $2,800$ a $380,183$ visited solutions. In RND-2 the instance with 149 available location sites can be optimally solved evaluating $45,163$ solutions if a population of 700 individuals is used, but it is necessary to evaluate $3,532,316$ solutions and use a population of $10,000$ (maximum size allowed by the memory limitations of the computers) to solve the instance with 349 available locations.

However, when compared to each other, RND-1 and RND-2 do not follow the intuitive complexity rule. For a same number of available locations (and hence of the size of the solution space), the other complexity measures of the RND-1 instance are clearly lower than the ones of the RND-2 instance: both the number of evaluation solutions and the size of the optimum population are larger in RND-2 than in RND-1.

In fact, there seems to be two kind of RND problems, regarding their complexity (effort required to be solved). The first and simpler kind is the one where transmitters employ a square geometry cell (i.e. RND-1). The second one includes all instances where circular geometry is employed (RND-2, RND-3 and RND-4). Inside each kind there is some consistency in the relationship between the size of the space of solutions, the optimal population size and the number of evaluated solutions. But when instances of both kinds are compared, a relationship rule for the complexity is hard to determine.

Figure 14 shows all instance solving costs faced to their equivalent binary size. Two major groups can be established: RND instances using squared geometry and RND instances using circular geometry. Numerical approximations have been determined for both groups, showing potential function growths when the binary size of the problem increases. The second group (circular geometry) contains instances that are clearly more difficult to solve, its growth factor is an x^4 factor, while the growth factor for the square geometry group is $x^{2.88}$. The instance with all kinds of transmitters is closer to the squared geometry group than to the circular geometry group.

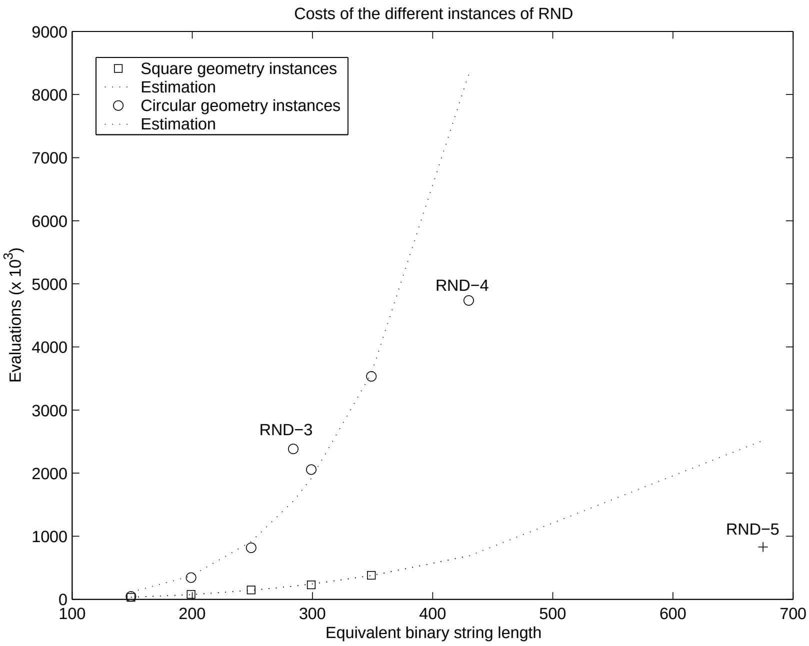

Figure 14: Comparison of the required evaluations with CHC for several problem instances.

6 Conclusions

In this work several metaheuristics have been successfully applied to solve the RND problem.

In the first part of this work, an algorithm-focused study has been carried. Several instances of the problem, differing both in the kind of transmitters permitted and the problem size (number of available location sites) have been solved using three main algorithms: simulated annealing (SA), CHC and genetic algorithm (GA). The parameters of SA and CHC have been tuned for every instance to obtain the best possible results, while GA has been used for means of comparison with existing work done on RND [5]. At the end of this part, it has been concluded that CHC is the fittest algorithm for solving RND-like problems.

In the second part of this work, a problem-focused study has been carried, using only CHC to solve the problem. New instances more complex than the previous ones have been solved, and the focus has been on the nature of the problem. This part has confirmed the robustness of the algorithm, and shown some information about the nature of the complexity of the problem. The cost of solving an instance of RND has shown to be dependant on two features:

- The geometrical restrictions of the coverage cells.
- The size of the solution space (or binary size).

The first feature controls the dependency of the solving cost with the size of the solution space. In our work, two groups have been identified: square geometry instances and circular geometry instances.

Another instance has been solved where all kinds of transmitters could be employed. Its cost has been below both the square geometry and the circular geometry estimated costs for its binary size.

References

[1] S.Z. Ali. *Design of Location Areas for Cellular Mobile Radio Networks.* IEEE 55th Vehicular Technology Conference, Volume 3, Page(s):1106 - 1110. 6-9 May 2002.

[2] E. Amaldi, A. Capone, F. Malucelli. *Planning UMTS Base Station Location: Optimization Models With Power Control and Algorithms.* IEEE Transactions on Wireless Communications, Volume 2, Issue 5, Page(s):939 - 952, Sept. 2003.

[3] P. Calégari, F. Guidec, P. Kuonen, and D. Kobler. *Parallel island-based genetic algorithm for radio network design.* Journal of Parallel and Distributed Computing, 47:86–90, 1997.

[4] G. Celli, E. Costamagna, A. Fanni. *Genetic Algorithms for Telecommunication Network Optimization.* IEEE International Conference on Systems, Man and Cybernetics, 1995. 'Intelligent Systems for the 21st Century' Volume 2, Page(s):1227 - 1232. 22-25 Oct. 1995.

[5] F. Chicano, E. Alba. *Evolutionary Algorithms for Optimal Placement of Antennae in Radio Network Design.* Parallel and Distributed Processing Symposium. 18th International Proceedings, Page(s):168. 26-30 April 2004 .

[6] Y.S. Kim, H. Jung. *Efficient Radio Network Optimization.* The 57th IEEE Semiannual Vehicular Technology Conference, VTC 2003-Spring. Volume 3, Page(s):1546 - 1549. 22-25 April 2003.

[7] S. Meguerdichian, F. Koushanfar, M. Potkonjak, M.B. Srivastava. *Coverage Problems in Wireless Ad-hoc Sensor Networks.* INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE Volume 3, Page(s):1380 - 1387. 22-26 April 2001.

[8] . H. Meunier, E. Talbi, P. Reininger. *A Multiobjective Genetic Algorithm for Radio Network Optimization.* Proceedings of the 2000 Congress on Evolutionary Computation, Volume 1, Page(s):317 - 324. 16-19 July 2000.

[9] C. Vijayanand, M.S. Kumar, K.R. Venugopal, P.S. Kumar *Converter Placement in All-optical Networks Using Genetic Algorithms.* Computer Communications 23, pp. 1223-1234, 2000.