Radio Coverage Optimization with Genetic Algorithms

Kai Lieska[1], Erkki Laitinen[1] and Jaakko Lähteenmäki.[2]

[1]Department of Mathematical Sciences, University of Oulu, Infotech Linnanmaa, 90570 Oulu, Finland. E-mail klieska@cc.oulu.fi.
[2]VTT-Information Technology, Telecommunications, 02044 Espoo, Finland.

ABSTRACT

Cellular network operators are dealing with complex problems when planning the network operation. In order to automatize the planning process, the development of simulation and optimization tools are under strong research. In this paper the genetic algorithms with three different approaches are studied in order to optimize the BS's sites. This research shows that proper approach in developing the individual structure and fitness function has crusial importance in solving practical base station siting problems with genetic algorithms.

I. INTRODUCTION

Radio coverage planning is currently being accomplished by using commercial network simulation tools. Nowadays, simulation is probably the best and most efficient way to get answers to the many and crucial questions of the operator. These tools allow the computing of the radio coverage supplied by a given configuration of the base stations, for example. However, the radio coverage optimization of the GSM cellular mobile telecommunication network faces a very difficult optimization problem, which is not practically solvable using only simulation tools. For this reason, the development of optimization tools, not only for coverage planning, but also for many other areas of mobile telecommunication, is under heavy research. Combining simulation and optimization tools will enable radio coverage planning to be performed automatically. The aim of this work is to examine the suitability of genetic algorithms with the problem of optimizing total radio coverage (coverage area achieved with some amount of BS). Due to the complexity and size of the networks, other possible optimization methods considered in literature are combinatorial methods or heuristic methods. The advantage of a genetic algorithm is in this that, rather than manipulating the mathematical formulations (object functions), the algorithm processes the computer representation of the potential solutions directly. Another important property of genetic based search methods is that they maintain a population of potential solutions while other methods process a single point of the search space. Genetic algorithms do not guarantee the optimum value, but by randomly choosing sufficiently many "witnesses" the probability of error may be as small as we like. The main idea of genetic algorithms comes directly from evolution and genetics. The vocabulary that we use comes from natural genetics as well. Individuals, made of genes, in a population represent a potential solution to a problem. Each individual can be presented as a binary vector i.e. genes can have the value 0 or 1. However, genetic algorithms need some numerical measure of fitness to rate solutions. This fitness value must be non negative, and formulated such that it reaches its maximum at the optimal solution. In this paper the suitably of tree variations of simple genetic algorithm, developed for this problem, are examined and discussed.

II. FORMULATION OF THE OPTIMIZATION PROBLEM

The problem considered in this paper is to choose the locations for a fixed number k of base stations (BS's) so, that the criterion set for total radio coverage is fulfilled. Such a criterion could be, for example, to maximize the total radio coverage area. The total radio coverage area is defined as the area where the power received from the base stations is more than -60 dBm. Moreover, one must chose the k base stations from the set of n admissible BS's. Two examples of urban radio coverage areas, S_i $i \in I = \{1, ..., n\}$, used in numerical calculations are presented in Figure 1.

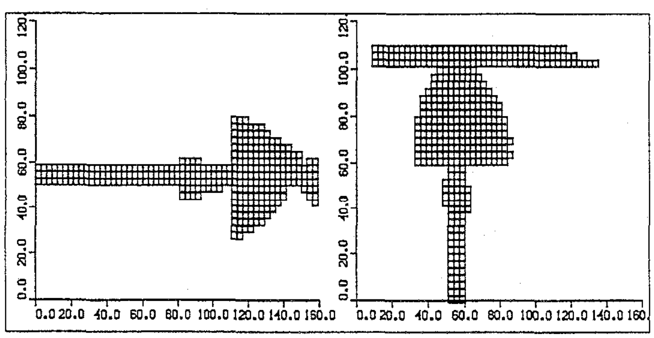

Figure 1.

Thus, our problem is to find k coverage areas S_j, $j \in I$ such, that $\bigcup_k S_j$ give the maximum total radio coverage. This problem is NP-complete, so it is not possible to solve it every case in polynomial time. Usually combinatorial optimiza-

0-7803-4872-9/98/$10.00 © 1998 IEEE

tion algorithms, i.e. heuristic or genetic algorithms are used to solve such problems. Usually the methods are very sensitive with regard to the dimension of the problem and input data of the problem. Moreover, in practice, the most useful methods include many practical modificatios of the algorithms. An essential part of optimizing total radio coverage is the calculation of the radio coverage for each admissible BS location. The radio coverage areas of the BS's can be calculated by any available network planning tool. In this case the coverage areas are synthetically generated.

Suppose we have an area, where we have n admissible locations for BS's, and n radio coverage areas, respectively. The element approach is used to represent radio coverage area data. The whole area considered is embedded into a rectangle, which is in turn divided into m small rectangle elements. A radio coverage area of a BS is a union of elements, where the power recived from the BS is more than -60 dBm (see Fig. 1). For computational reasons, we denote such an element by the number 1 and all other elements by the number 0. Now the set of all admissible radio coverage areas can be presented as a matrix $M = [a_{ij}]_{m\times n}$, $a_{ij} = \{0,1\}$, where each column represent the elements of one radio coverage area. The structure of the original, and the transformed matrices are presented in Figure 2.

$$\begin{pmatrix} -103.3 & -20.2 & \dots & -120.6 \\ -63.1 & -35.8 & \dots & -59.9 \\ \vdots & \vdots & \ddots & \vdots \\ -42.4 & -55.9 & \dots & -55.6 \end{pmatrix}_{m\times n} \Rightarrow \begin{pmatrix} 0 & 1 & \dots & 0 \\ 0 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}_{m\times n}$$

Figure 2.

The genetic approach for the problem is obvious. First, we denote an individual by the bit string I. This string represents the sites of BS's in a potential solution. The individual I and the fitness function $f(I)$ must be formulated carefully in order to get good convergence. In addition, the parameters of the genetic algorithm, such as population size, mutation rate, crossover rate, etc. must be chosen carefully. Now the basic simple genetic algorithm for solving the problem can be presented step by step as follows:

Algorithm 1

Step 0. Set iteration number i=0. Set the initial population $P_i = \{I_1, ..., I_{ps}\}$, ps is the population size.

Step 1. Apply reproduct, crossover and mutation to the population P_i.

Step 2. Calculate the fitness value $f(I)$ for every $I \in P_i$.

Step 3. If (not termination-condition) then
 Set $i = i + 1$ goto step 1
 else
 Solution is the best individual achieved during running. STOP

III. SOLVING ALGORITHMS FOR THE OPTIMIZATION PROBLEM

Genetic algorithm is used to solve the optimization problem. In literature, ([2],[3]), many kinds of genetic algorithms with different operators have been developed. In this work we use a simple genetic algorithm with 3 operators: reproduction, crossover and mutation. Reproduction is a process in which individual strings are copied according to their fitness value. The higher the fitness value, the higher the probability of it having one or more offspring in the next generation. We use biased roulette wheel selection. An individual who survives reproduction, goes to the next operation, which is crossover. Crossover is an operation where genes of two individuals are swapped from point K, which is selected randomly. This has to be done for every pair. We use also probability (crossover rate) weather a pair goes to the crossover operation or not. The third operation is mutation. Because we use binary genes, it is very simple operation. It changes the bit, which is the target, to mutation, with some probability (mutation rate). The convergence and the computer efficiency of the algorithms depend strongly on the representation of the structure of the individual and on the fitness function. Following are presented three different approaches to form the individual's structure and fitness function. In all three approaches, we use Algorithm 1 to solve the problem.

A. Approach 1

One genetic representation for the individual structure is very obvious. We have n possible BS locations, so let the individual be a bit string with length n. Those BS's which exist in the solution are denoted by 1, and all others are denoted by 0. Hence every possible BS-combination can be represented by using this kind of individual. The fitness function has to be such that it guides the algorithm in the direction where exactly k BS's exist in the solution. In this case, the fitness function $f(I)$, for the individual I, is defined as follows:

$$f(I) = \frac{c(I)}{|k - o(I)| + 1},$$

where $c(I)$ is total radio coverage area of individual I, k is the desired number of BS and $o(I)$ is number of BS's in individual I.

This fitness function is not good for simple genetic algorithm ([2],[3]) because it tends to converge prematurely in the earily phases of running, and on the other hand it causes problems with convergence later in a run. To prevent these problems the following linear scaling is used.

$$f'(I) = a \cdot f(I) + b,$$

where the coefficients a and b are chosen such, that the average fitness in a population remains the same and the fitness of the best individual obeys the following equation:

$$f'_{max} = c_{mult} f_{avg},$$

where c_{mult} is the value of the expected copies of the best population member and f_{avg} is average fitness of the population. This linear scaling must be done during each iteration step i.e. for every population. Care should be taken that $f'_{min} \geq 0$. If this is not possible by using c_{mult}, we scale as much as possible ($f'_{min} = 0$). We use $f'(I)$ as a fitness function in Approach 1.

B. Approach 2

The object of optimization is to find the best k sites of n BS's. In Approach 1 the fitness function guides the solution in such a direction where the individual contains exactly k BS. In Approach 2, a priori knowledge about the desired number of BS's is used, when we define the individual structure. Because we have only $\binom{n}{k}$ different possible solution to examine, all the possible solutions can be represented by using l genes long individual, where l is defined such, that

$$2^{l-1} < \binom{n}{k} \leq 2^l.$$

Because the number of individuals 2^l, can be bigger than the number of possible solutions $\binom{n}{k}$, it is not possible to join each individual to different unique solution. This problem is presented schematically in Table 1, where I_{10} is the decimal value of individual I.

Table 1.

I_{10}	Individual I		BS's sites
0	00...00	$\Leftrightarrow$	111100...00
1	00...01	$\Leftrightarrow$	111010...00
...	....	$\Leftrightarrow$	....
$\binom{n}{k}-1$	1011....	$\Leftrightarrow$	00...001111
...	1011....	$\Leftrightarrow$	?

However, for computational reasons each individual must correspond to exactly one possible solution. So, we use the following linear scaling

$$M(I) = INT(I_{10} \cdot \frac{\binom{n}{k} - 1}{2^l - 1}),$$

where $0 \leq M \leq \binom{n}{k} - 1$ is the index of a possible solution and INT means rounding to the nearest integer. By using the scaling each of the 2^l individuals can be joined to exactly one possible solution. In Approach 2 the fitness function used is as follows:

$$f(M(I)) = a \cdot c(M(I)) + b,$$

where $c(M(I))$ is total coverage area for the possible solution M and a, b are chosen as in Approach 1.

C. Approach 3

In this approach a different structure for individual is used. The individual is divided into k parts, where k is the desired number of BS's. Every part corresponds to the index of BS to be selected. Because the total number of BS's is n we can represent every part of the individual by using a string of l bits in length, where l is defined as:

$$2^{l-1} < n \leq 2^l.$$

Now the length of an individual will be $k \cdot l$ bits. As in Approach 2 we have to join every part of the individual to a exactly one BS location. This is done by using the linear scaling

$$M^p = INT(I^p_{10} \cdot \frac{n-1}{2^l - 1} + 1).$$

where p refers to the part index and $1 \leq M^p \leq n$ is the index of the BS. Table 2 represents the individual structure and the corresponding BS combination schematically, for n=40, k=4 and I = 110101111111110110100110.

Table 2.

p	1	2	3	4
I^p	110101	111111	110110	100110
I^p_{10}	53	63	54	38
M^p	34	40	34	25

So individual I corresponds to BS's 34, 40, 34 and 25. We use a fitness function similar to that of Approach 2. It is possible that some parts of an individual correspond the same BS, so this approach can handle also cases where less than k BS's exist in the solution.

IV. NUMERICAL TESTS

In the numerical tests we consider an area of 160 by 110 meters. The area is divided into 1969 rectangle elements or "measure points". Inside the area is 40 potential sites for BS's. The radio coverage areas of BS's are generated synthetically such, that the received power inside elements of the radio coverage area is greater than -60 dBm. The problem is to maximize the total radio coverage area using exactly 4 BS's. In order to check the numerical results, the global maximum of 1506, is calculated by testing all $\binom{40}{4}$ possible solutions. The aim of the numerical calculations is to test the operation of Algorithm 1, with the individual structures and fitness functions presented in Approaches 1-3. Algorithm 1 uses the following probabilities and parameters: population size, crossover rate, mutation rate and scaling coefficient (c_{mult}).

Parameters have a large impact on the optimization result. This makes choosing the best possible combination of parameters very important. Calibrating these parameters to make the algorithm works as effective as possible, is a difficult optimization problem. We did it by simply running Approaches 1-3 with different population sizes and different combinations of parameters.

For solving the optimization problem we used Approaches 1-3 with these "optimal" parameters. In the Algorithm 1 two hundred iterations were done per test, and the test was repeated 20 times to reduce the influence of probability. In order to compare the results, we solved the optimization problem with the random-search method, also. Because Algorithm 1 goes through at moust (population size)×(iteration steps) points of search space, we calculated the same amount of points also with random-search method. Figures 3-5 show the average achieved optimization results for Approaches 1-3 and for random-search method.

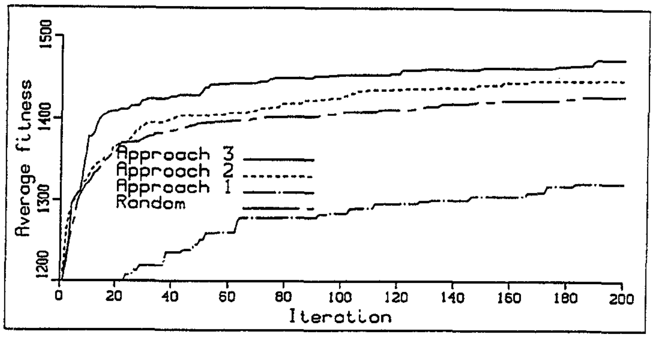

Figure 3. Population size=5; The parameters crossover, mutation rate and scaling coefficient, respectively in Approach 1: 0.4, 0.02, 2; in Approach 2: 0.6, 0.08, 4 and in Approach 3: 0.6, 0.03, 5

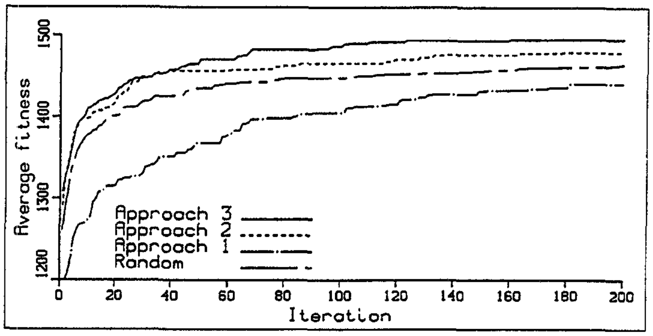

Figure 4. Population size=20; The parameters crossover, mutation rate and scaling coefficient, respectively in Approach 1: 0.8, 0.03, 5; in Approach 2: 0.8, 0.06, 2 and in Approach 3: 0.6, 0.05, 3.

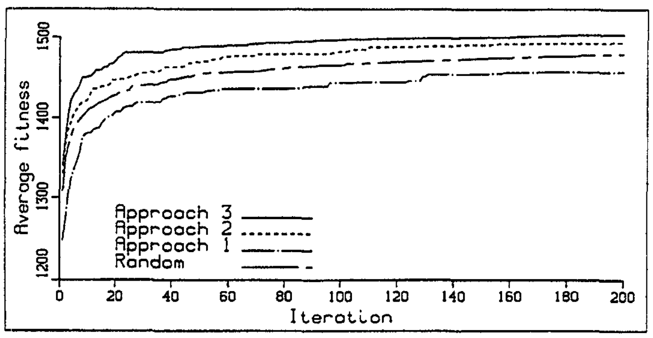

Figure 5. Population size=50; The parameters crossover, mutation rate and scaling coefficient, respectively in Approach 1: 0.8, 0.01, 4; Approach 2: 1.0, 0.04, 3 and Approach 3: 0.4, 0.03, 5.

From Figures 3-5 it can be seen that with every population size Approach 2 and Approach 3 are more effective than the random-search method, while Approach 1 is not. This is not surprising, since the amount of search space in Approach 1 is 2^{40} while in Approach 2 it is only $\binom{40}{4}$ and in Approach 3 $\binom{40}{4} + \binom{40}{3} + \binom{40}{2} + \binom{40}{1}$. In the Table 3 is presented the average results and the best results of 20 test runs for the Approaches 1-3 with population sizes 5, 20 and 50. Moreover in the Table 3 is presented the percentual amount of the search space examined by the algorithm during 200 iteration steps.

Table 3.

Appr.	pop. size	av. result	best result	percents
1	5	1321	1490	$9.1 * 10^{-8}$%
1	20	1441	1506	$3.6 * 10^{-7}$%
1	50	1457	1499	9.09^{-7}%
2	5	1448	1499	1.09%
2	20	1479	1502	4.37%
2	50	1493	1502	10.9%
3	5	1473	1506	0.98%
3	20	1495	1506	3.92%
3	50	1503	1506	9.80%

From the Table 3 it can be seen that, the bigger the size of the population is, the bigger is the obtained average result for every Approach 1-3. This is a very self-evident, but on the other hand the grow of population size means also an increase in the computing time of the algorithm. That is why it can not be said that the bigger the population size is the more effective the optimization method is when we measure the "efficiency" also by time. However, from Table 3 it can be seen that Approach 3 is more effective than the other, no matter what population size is considered.

V. CONCLUSION

The aim in this work was to examine the suitability of genetic algorithms with the problem of optimization of total radio coverage. Three different kind approaches were developed and tested with simple genetic algorithm to solve the radio coverage optimization problem. It was shown that Approaches 2 and 3 worked very well in this case problem and they were more effective than the random-search method. This work shows that the *a-priori* knowledge about desided amount of BS's pays an important role when defining the individual structure and fitness function of the genetic algorithm.

The problem which we solved in this work was very brief. In reality, the optimization problem can be many times larger than the example problem of this paper: the area considered can be larger, and the number of possible BS locations can be easily over 200. The fitness function can also be much more complicated. Besides that of simply maximizing total coverage area, the both BS price and BS sites may be different, or "double" radio coverage may be desirable in some area. Inclusion of such criteria into the problem require only adjustment in the fitness function of the genetic algorithm. Thus, one major benefit of a genetic algorithm is its ability to adapt to different problems and criteria.

REFERENCES

[1] D.E. Goldberg, *Genetic Algorithms in Search, Opti-*

mization, and Machine Learning, Addison-Wesley Publishing Company, inc., 1989.

[2] P. Calégari et al.,"Radio Network Planning with Combinatorial Optimisation Algorithms", in Proc. of *European Mobile Technology towards Global Wireless Infrastructure*, Spain Nov. 27-29, 1996.

[3] Z. Michaelewich, *Genetic Algorithms+Data Structures=Evolution Programs*, Department of Computer Science, University of North Carolina, 1996.