A Parallel Genetic Approach to Transceiver Placement Optimisation

Patrice Calégari, Frédéric Guidec, Pierre Kuonen

A parallel genetic approach to transceiver placement optimisation

Patrice Calégari, Frédéric Guidec, Pierre Kuonen
Swiss Federal Institute of Technology, DI–LITH, CH-1015 Lausanne, Switzerland
E-mail: {calegari,guidec,kuonen}@di.epfl.ch

September 19,1996

Keywords: Genetic Algorithms, MIMD Parallel Computing, Radio Network Design.

Introduction

One of the issues telecommunication companies must face when deploying a mobile phone network is the selection of good locations for base transceiver stations (BTSs).

The problem comes down to guarantying a maximum percentage of radio coverage in a region with a minimum number of BTSs. The best subset of BTS locations must thus be selected among a set of user-provided potential ones to achieve this goal.

This paper introduces a way to solve this problem. First, a graph-based modelling of the BTS placement problem is defined. Second, an overview of genetic algorithms is given, and a parallel approach is proposed. Finally, the results obtained with the *Paragene*[1] program are discussed.

Problem modelling

It is assumed that the region is discretised on a grid, and that each BTS is associated a cell. A cell is defined as the set of the locations grid points that a BTS covers.

Let us consider two sets B and L, where B is the set of all potential BTS locations, and L is the set of all potentially covered locations. Let us then construct the graph $G = (V, E)$, where $V = (B \cup L)$ is a set of vertices and where E is a set of edges such that each BTS location is linked to the locations it covers. An example is shown in Figure 1.

This modelling is based on the notion of minimum dominating set[2] [Irv91].

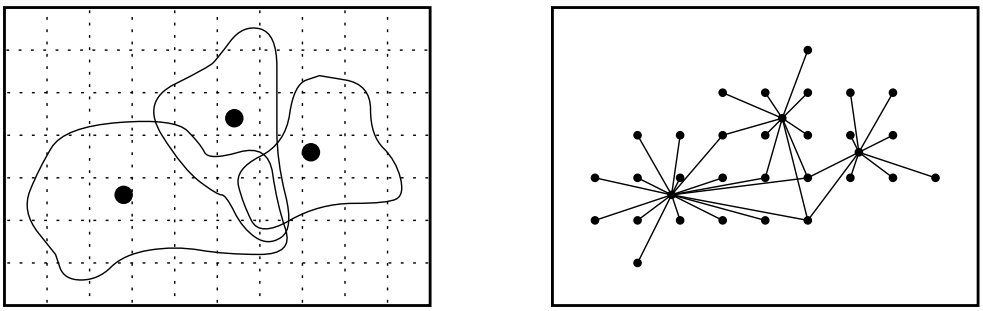
Searching for the minimum subset of BTSs that covers a maximum percentage of a region comes to searching for a subset $B' \subseteq B$ such that $Card(B')$ is minimum and such that $Card(Neighbours(B', E))$ is maximum. $Neighbours()$ is defined by

$$Neighbours(B', E) = (u \in L \text{ such that } \exists v \in B', (v, u) \in E)$$

As the number of locations to consider is likely to be huge, it is not possible to assertain that the optimal solution of such a problem can be found in an acceptable time. A suboptimal algorithm must then be used to find an acceptable solution. To date, we most especially focus on the so-called genetic approach.

[1] *Paragene* stands for PARAllel GENEtic algorithm.

[2] This NP-complete problem originates in the graph theory. It is also known as the vertex-cover problem.



(a) (b)

Fig. 1: Modelling of 3 BTSs and their cells (a) with the graph-based model (b)

Genetic algorithms

A genetic algorithm is a population-based model inspired by evolution, that uses selection and recombination operators to generate new sample points in a search space. There exist many variants of genetic algorithms in the litterature, but these are all based on the algorithm introduced by Holland in [Hol75].

A population is a set of individuals that represent candidate solutions, and that are generally encoded as chromosome-like bit strings. The first step in the execution of a genetic algorithm consists in the generation of an initial population. Each of the following iteration steps can be thought of as a two stage process. Selection is applied to the current population to create an intermediate one. Crossover and mutation operators are then applied to this intermediate population to create the next one.

In our implementation, the chromosome-like bit string represents the whole set of possible BTS locations. Whether a location is actually selected in a potential solution depends on the value of the corresponding entry in the bit string. The initial population is generated randomly. The selection is achieved based on the fitness value associated with each individual, which may be perceived as an allocation of reproductive opportunities: the higher the fitness value of an individual, the likelier it is to be selected. The fitness of an individual is defined by:

$$fitness = \frac{(CoverRate)^{\alpha}}{NumberOfBTSs}$$

where α is a parameter that permits to favour the cover rate with respect to the number of BTSs[3]. The crossover operator we use achieves a one point crossover, that is, it splits two bit strings and recombines them by exchanging their two parts. The mutation operator changes one bit value randomly in the bit string representation of an individual. The execution terminates after a predefined number of iteration steps has been run through (typically, twice the number of individuals in the population).

The parallel genetic approach

As the amount of data to be dealt with the BTS placement problem can be huge, and as a lot of computing resources are needed for the search space to be correctly sampled, a parallel approach was investigated. Moreover, the fact that genetic algorithms are parallel by essence also suggests such an approach.

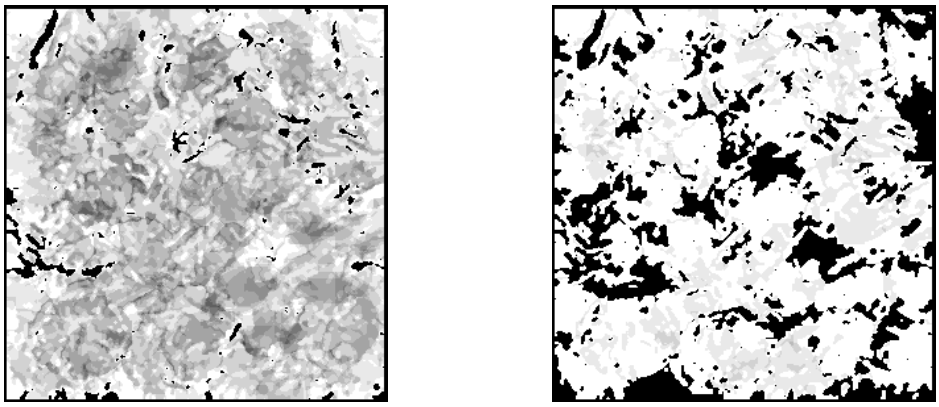
[3] So far, experiments were achieved using $\alpha = 2$.

We decided to use the "island model" [Whi93]. In this model, sub-populations, called islands, evolve independently. From time to time, some individuals are allowed to migrate from island to island. A MIMD[4] approach was achieved, first because it suits well to this model, and second because it permits to use networks of workstations which are more usual than parallel supercomputers.

In our implementation, the population is distributed on islands that are positioned on an oriented ring. After every iteration step, a copy of the best individual (the one with the greatest fitness value) ever met by each island is sent to the next island on the ring. Each island thus receives a new indivividual that replaces one of its individual selected randomly. The number of iteration steps is fixed, and an implicit synchronism is established by the communications between islands. The *Paragene* program is written in C++. It uses the PVM library for communications.

First results

Tests were performed on a network of Sparc-5 workstation. The population was distributed in such a way that each workstation was in charge of one island.



(a) (b)

Fig. 2: (a) Coverage C obtained with 150 potential BTSs. (b) Coverage obtained with 44 BTSs selected by the algorithm among the potential ones (80% of C). Black regions are not covered, white region are covered by exactly one BTS, grey regions are covered by more than one BTS (dark grey regions are covered by more BTSs than light grey ones)

The quality of the solutions, obtained by applying *Paragene* on data provided by TDF[5], is very encouraging. These data model a part of the French region "Les Vosges", with 150 potential BTS locations. An example of a solution proposed by *Paragene* is shown in Figure 2. On this example which is representative of the solutions obtained, *Paragene* managed to cover 80% of the region with only 44 BTSs. Moreover, no location is covered by more than 4 BTSs, and only 1.7% of the covered locations are covered by more than 2 BTSs. Finally, it is interesting to notice that, with *Paragene*, large cells are more likely to be chosen by the algorithm than small ones. All these characteristics are commonly perceived as "good" characteristics by telecommunication operators.

The fitness of the best individual found by the algorithm increases with the number of islands. A possible explanation of this phenomenon is that, if the population of an island tends

[4] Multiple Instruction Multiple Data.
[5] Tele Diffusion de France, Metz.

to converge too quickly towards a local optimum, it can be "saved" by an individual migrating from another island. In other words, a good individual arriving in a homogenous population can lead to some interesting diversification.

Figure 3 shows the speedup performance of *Paragene*. It must be noticed that the algorithm behaves differently when it runs on different numbers of workstations. The reason is that, as the population is distributed on islands, its behaviour is not exactly the same in all cases. This may explain why the $\frac{T(1)}{T(N)}$ function (where $T(N)$ is the time spent by *Paragene* on N workstations) is superlinear.

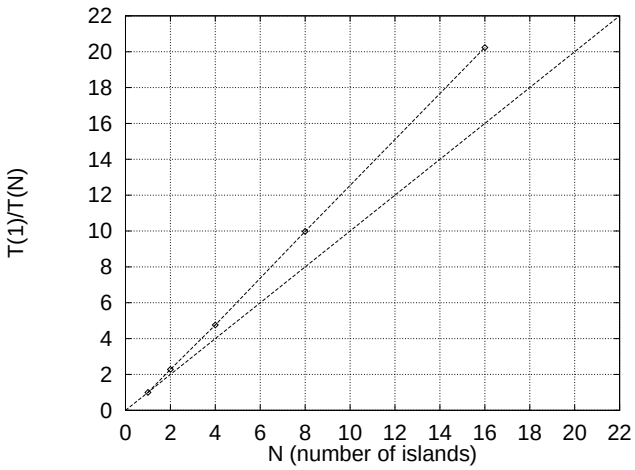


Fig. 3: $\frac{T(1)}{T(N)}$ of the parallel algorithm with 160 individuals distributed on N islands and 320 iteration steps of the population

Conclusion

Compared to another method based on the search of stable set using a greedy algorithm [Act96], *Paragene* gives better results (when using more than 2 islands), but it remains slower.

Since better results are obtained with several islands than with one, future works shall focus on the study of the quality of the solutions with respect to the number of islands. Moreover, comparing the current *Paragene* with an equivalent, sequential algorithm using N islands, would permit to refine the speedup evaluation.

References

[Hol75] J. Holland, "Adaptation in natural and artificial systems", University of Michigan Press, 1975.

[Irv91] R. W. Irving, "On approximating the minimum independent dominating set", Information Processing Letters 37, 197-200 North-Holland, 1991.

[Act96] first deliverable of the radio network optimisation workpackage in the STORMS project (4th ACTS project), 1996.

[Whi93] D. Whitley, "A genetic algorithm tutorial", technical report CS-93-103, Colorado State University, 1993.