Study of Genetic Search for the Frequency Assignment Problem*

Jin-Kao Hao and Raphaël Dorne

LGI2P
EMA-EERIE
Parc Scientifique Georges Besse
F-30000 Nîmes
France
email: {hao, dorne}@eerie.fr

Abstract. The goal of this paper is twofold. First, we present an evolutionary approach to a real world application: the Frequency Assignment Problem (FAP) in Cellular Radio Networks. Second, we present an empirical study on the effectiveness of crossover for solving this problem. Experiments carried out on a set of real-size FAP instances (up to 300 cells, 30 frequencies and 30,000 interference constraints) show the interest of EAs. At the same time, empirical evidence suggests that the contribution of crossover is marginal for this application.

1 Introduction

The Frequency Assignment Problem (FAP) in cellular radio networks is a very complex application in the field of telecommunications. Although different versions can be defined for FAP, the main goal consists in assigning one or more frequencies, a very limited resource, to each radio cell in a cellular radio network while minimizing electromagnetic interferences due to the reuse of frequencies by adjacent cells. The difficulty of this application comes from the fact that an acceptable solution of FAP must satisfy a set of multiple constraints, some of these constraints being orthogonal. The most severe constraint concerns a very limited radio spectrum consisting of a small number of frequencies (or channels). Indeed, telecommunications operators such as France Telecom must cope with only up to 60 frequencies for their networks whatever the traffic volume to cover may be, and this, in agreement with national and international regulations. In addition to this frequency constraint, two other types of constraints must be satisfied to insure good communication quality:

1. the *traffic constraint* for each cell corresponding to the minimum number of frequencies required by the cell to cover the communications of the cell.

* Supported by the CNET (French National Research Center for Telecommunications) under the grant No.940B006-01.

2. two categories of frequency *interference constraints*:
   - *Co-cell constraints*: any pair of frequencies assigned to a radio cell must have a certain distance between them in the frequency domain.
   - *Adjacent-cell constraints*: the frequencies assigned to two adjacent cells[2] must be sufficiently separated in the frequency domain.

The basic FAP can be shown to be NP-complete because, in its simplest form, it is reduced to the graph coloring problem [9]. So far, many methods have been proposed to solve FAP, including 1) classic methods: graph coloring algorithms [9, 8] and integer programming; 2) heuristic methods: neural networks [12, 7], genetic algorithms (GAs) [4, 11], local search such as simulated annealing (SA) [6, 1] and Tabu search (TS) [10], and constraint programming (CP) [3].

The goal of this paper is twofold. First, we present various (hybrid) evolutionary algorithms (EAs) and their performances on real-size FAP instances (up to 300 cells, 30,000 interference constraints with only 30 frequencies). Second, we investigate empirically the effects of crossover for this application.

The paper is organized as follows. In Section 2, FAP is modelled as an optimization problem. In Section 3, the different components of our EAs are summarized. In Section 4, experimental results are presented and compared. Conclusions and perspectives are given in the last section.

2 Modelling the Frequency Assignment Problem

2.1 Notations and Model

In this paper, the following notations will be used to simplify the presentation.

N the number of cells
NF the number of available frequencies in the spectrum
NCI the number of interference constraints defined for adjacent cells
NC the number of constraints (NC=NCI+N)
$f_{i,m}$ the m^{th} frequency assigned to the cell C_i

FAP can be modelled with a quadruple FAP = $<X,D,C,F>$ representing an optimization problem with:

$X = \{C_i \mid C_i$ is a cell of the network, i $\in$ [1..N]$\}$.
$D = \{F_i \mid F_i$ is an available frequency of the spectrum, i $\in$ [1..NF]$\}$.
$C = T \cup I$

$T = \{T_i \mid T_i$ minimal number of frequencies necessary for C_i, i $\in$ [1..N]$\}$.
$I = \{I_i \mid I_i$ interference constraints of frequencies, i $\in$ [1..NC]$\}$.

$f =$ cost function of a frequency assignment.

Finding a solution to FAP = $<X,D,C,F>$ means assigning one or more frequencies in D to each cell C_i of X in such a way that the constraints of C are simultaneously satisfied and the cost f is minimized.

[2] Two cells are adjacent if they emit within a common area even if they are not geographically adjacent.

2.2 Constraints

There are essentially two big families of constraints: *traffic constraints* and *interference constraints*. The traffic constraint for each cell C_i is represented by an integer T_i coding the minimum number of frequencies necessary for C_i to cover its maximum traffic volume. In practice, this maximum traffic value is defined by an estimation of the maximum number of communications which can simultaneously arise within this cell. The interference constraints over the network are represented by a symmetric compatibility matrix M[N,N] defined by:

- M[i,j] with i $\neq$ j represents the minimum frequency separation required to satisfy the adjacent-cell constraints between the cells C_i and C_j.
  $\forall$n $\in$ [1..T_i], $\forall$m $\in$ [1..T_j], $|f_{i,n} - f_{j,m}| \geq$ M[i,j]
- M[i,i] represents the minimum frequency separation necessary to satisfy the co-cell constraints:
  $\forall$n, m $\in$ [1..T_i], n $\neq$ m, $|f_{i,n} - f_{i,m}| \geq$ M[i,i]
- M[i,j]=0 means there is no constraint between the cells C_i and C_j.

Several optimization problems can be defined for FAP. For example, given a traffic volume in the network, we may minimize the frequency interference and the number of frequencies used. Otherwise, given a number of available frequencies, we may maximize the traffic volume (assigning more frequencies to a cell) while minimizing the frequency interference. In the latter case, the two optimization objectives are orthogonal. Indeed, requiring a higher degree of reuse of frequencies implies a higher risk of frequency interference. In this paper, we deal with the first problem, i.e. we try to mimimize the frequency interference using a minimum number of frequencies for a given traffic volume.

3 EAs for FAP

The main obstacle of applying EAs to a new application lies essentially in two difficulties. First, we must define an efficient encoding for the given application and a set of tailored genetic operators. Second, we must find an efficient combination of these operators working with the encoding. The first point is related to the application at hand and can be solved by integrating, at different levels of an EA, specific knowledge about the application and various efficient local search techniques. On the contrary, the second point is a methodological issue and must be dealt with independently of applications. Indeed, until now, neither the roles of genetic operators nor their combinations are well understood. From an application point of view, the simultaneous presence of all the genetic components makes it difficult to build efficient EAs and to evaluate the efficiency of each component. For these reasons, to tackle our application with EAs, we take a step-by-step approach which is based on the bottom-up principle.

Instead of integrating all the genetic components in an EA, we begin with a very simple EA which uses only mutation and selection. Using this largely simplified EA, many alternative mutation and selection operators can be experimented with and evaluated. Another interesting point of this simple EA is that it may work with a singleton population. In this case, the EA can be used to simulate numerous (stochastic) hill-climbers. Once the most efficient selection and mutation operators are identified, crossover may be added to the initial EA if this proves to be necessary for the application.

Problem Encoding & Solution Space

Given a problem FAP $= < X, D, C, F >$, a feasible solution $s = < C_1, C_2, ..., C_N >$ corresponds to a complete assignment of frequencies to cells where

- $|C_i| = T_i$, i.e. each C_i is composed of T_i frequencies (genes) assigned to it.

An individual or a chromosome I simply represents a feasible solution. A population of chromosomes represents a part of the total search space composed of all the feasible solutions of FAP noted by $S = \{s \mid s$ is a feasible solution of FAP$\}$. Note that this encoding satisfies implicitly the traffic constraint.

Other encodings are possible. For instance, one may use a NF$* \sum_{i=1}^{T_i}$ boolean matrix, each element of the matrix indicating if a frequency is assigned to a cell. One may also use a heterogeneous 2 dimensional encoding whereby a number of frequencies equal to its traffic corresponds to each cell. However, the chosen encoding has some desirable properties compared with others. First, the number of genes in a chromosome is minimized. Second, mutation can be directly applied. Third, crossover can also be directly applied or be applied with a minimum constraint of choosing crossover points at the first gene of each cell.

Fitness Function

Two functions are used to evaluate an individual's fitness: the first is for evaluating the initial population and offspring produced by crossover and the second is for offspring produced by mutation.

For any individual *I*, one natural way to calculate its fitness is to apply the following function *f*: $S \rightarrow NC$ (the number of interference constraints).

$$f(I) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{T_i} \sum_{p=1}^{T_j} CI(i,j,k,p) + \sum_{i=1}^{N} \sum_{k=1}^{T_i} \sum_{p=k+1}^{T_i} CO(i,k,p) \quad (1)$$

$$\begin{aligned} CI(i,j,k,p) &= 1 \text{ if } \mid f_{i,k} - f_{j,p} \mid < M[i,j] \\ &= 0 \text{ otherwise} \\ CO(i,k,p) &= 1 \text{ if } \mid f_{i,k} - f_{i,p} \mid < M[i,i] \\ &= 0 \text{ otherwise} \end{aligned}$$

It is easy to see that this function examines in pairs all the cells of the networks in order to count the total number of interference constraints violated by *I*. Given that T_i (i $\in$ [1..N]) are bounded by *NF*, the time complexity of the fitness function (1) is $\Theta(N^2 * NF^2)$.

When mutation is applied to an individual I producing I', only one cell in I is affected. More precisely, only one frequency of the cell is changed. Thus, to calculate the fitness of I', only cells adjacent to this modified cell need to be re-examined to calculate the fitness difference between I and I'. This can be realized by using the following formula which counts the number of violated constraints induced by the frequency $f_{i,k}$ of the cell C_i.

$$f(Ci, k) = \sum_{j=1, j \neq i}^{N} \sum_{p=1}^{T_j} CI(i, j, k, p) + \sum_{p=1, p \neq k}^{T_i} CO(i, k, p) \quad (2)$$

CI and *CO* are the same as defined in (1). Since the traffic of the cells is bounded by the number of frequencies *NF*, the time complexity of this function is $\Theta(N * NF)$. Note finally that, in practice, the complexity is much lower since the number of adjacent–cells of C_i is usually much smaller than *N-1*.

Selection

For the purpose of investigating the effects of crossover for FAP, a single selection mechanism, Baker's SUS (Stochastic Universal Sampling) selection algorithm [2] is used for all our EAs (with and without crossover). Similar to the standard *spinning wheel* method with one pointer, SUS can be considered as a spinning wheel method with K equally spaced pointers (K being the size of population). Hence, all K samples will be achieved in a single spin. Like other probabilistic selection methods, well-fit individuals will have more chances to be selected by SUS.

Compared with other selection methods, SUS has some desirable qualities such as simplexity and efficiency (complexity of $\Theta(K)$), and accuracy and precision (zero bias and minimum spread). Moreover, according to our experiments, SUS gives good results on average.

Crossover

Three crossover operators are studied: two standard crossovers, i.e. one-point and uniform, and one specialized crossover called conflict-based.

- *one-point*: a crossover point is randomly chosen with equal probability and the right portions of the two parents are exchanged, producing two children.
- *uniform*: each gene of the two children receives the corresponding gene from either parent one or parent two according to a given probability [13].
- *conflict-based*: the idea of this specialized crossover is to use specific knowledge about the application. It tries to pass on good genes from the parents to their children. More precisely, frequencies (genes) free of conflict are directly copied and frequencies in conflict are stochastically or deterministically chosen. Figure 1 gives an example whereby deterministic choices are made for genes in conflict.

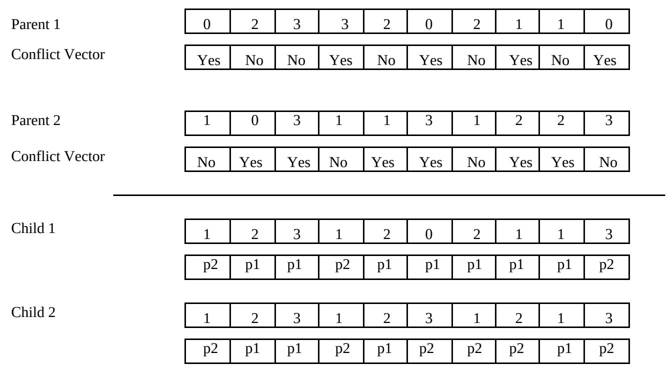

Fig. 1. Conflict-based crossover

These crossover operators can be considered to be sufficiently representative for studying the effects of crossover for FAP. In fact, one-point and uniform crossover are well known and have some complementary properties. One-point crossover exhibits the minimum distributional bias and the maximum positional bias. On the contrary, uniform crossover exhibits the maximum distributional bias and the minimum positional bias. Moreover, uniform crossover is known to be more disruptive, which is particularly suitable for small populations to sustain a highly explorative and diversifying search. On the other hand, specialized crossovers use application knowledge to insure a better transition of useful information from parents to offspring.

Mutation

For the same reason we fix the selection operator, we have chosen to fix the mutation operator to be used in our experiments in order to isolate the effects of crossover. More precisely, a special mutation operator which is based on frequency conflict and defined in [5] is used by all our EAs. This conflict-based mutation (CBM) operator is composed of three choices based on specific knowledge of the application: 1) random selection of a cell among the cells in conflict, 2) random selection of a frequency from this cell, and 3) the best frequency value which is different from the current value.

Compared with other mutation operators we have tested, including classic random and specialized mutations of the same class defined in [5], the chosen mutation proves to be the most efficient on average for this application.

4 Experimentation and Results

4.1 Tests

In this section, we present our experimental results for the various EAs on two sets of FAP instances provided by the French National Research Center for Telecommunications. These instances are produced by a generator in such a way that they correspond to real situations encountered in real networks or sub-networks in France. Consequently, some instances have a very big size in terms of the number of cells (variables) and of inteference constraints. The biggest instance has 300 cells (600 genes) and more than 30,000 constraints. Moreover, each gene may take up to 30 possible values (frequencies).

Test Set No.1 (Traffic=1)

The first set of FAP instances has the following characteristics.

- *traffic constraints*: $T_i = 1$ (i $\in$ [1..N]); i.e. each cell is assigned one frequency. Consequently, a co-cell constraint does not exist (M[i,i] = 0 for i $\in$ [1..N]).
- *adjacent constraints*: $|f_i - f_j| \geq$ M[i,j] = 1 (i, j $\in$ [1..N]) for two adjacent cells C_i and C_j; i.e. C_i and C_j must be assigned different frequency values.

It is easy to see that these instances correspond to the classic graph coloring problem. In fact, it will be sufficient to replace frequencies by colors, cells by n-odes and adjacent constraints by edges. Finding an optimal frequency assignment is equivalent to coloring a graph using only a minimum number (the chromatic number) of necessary colors. For these instances, the chromatic number corresponds to the minimum (optimal) number of frequencies necessary for having a frequency assignment without interference (*optimal solution*). The names of these instances consist of three integer parameters *nf.nc.d* where

nf : the optimal number of frequencies needed for an optimal solution.
nc : the number of cells in the network.
d : the density of interference constraints defined as a percentage of all the possible constraints over the network.

For example, the instance 8.150.30 defines a problem composed of 150 cells with 8 available frequencies and 30% of 150*(150-1)/2 total constraints. For large problems having a high density, we obtain up to 13,000 constraints.

Test Set No.2 (Traffic=2)

The second set of FAP instances has the following characteristics.

- *traffic constraints*: $T_i = 2$ (i $\in$ [1..N]); i.e. each cell is assigned two frequencies.
- *co-cell constraints*: $|f_{i,n} - f_{i,m}| \geq$ M[i,i] = 3 (i $\in$ [1..N]); i.e. two frequencies assigned to the same cell must have a minimum distance of 3.
- *adjacent constraints*: $|f_{i,n} - f_{j,m}| \geq$ M[i,j] $\in$ [1,2] (i, j $\in$ [1..N]) if C_i and C_j are adjacent cells; i.e. two frequencies assigned to two adjacent cells must have a minimum distance of 1 or 2 according to the cells.

The names of the instances consist of four integer parameters *nf.nc.d.p*. The first three parameters have similar meanings to those presented before. The fourth parameter indicates the average degree for the nodes of the graph (the average number of constraints associated with a cell). Due to the way the instances are generated, the exact number of frequencies for an optimal solution is no longer known in advance. However, this optimal number is bounded by a value given by the generator.

Compared with the first set, these instances are naturally much harder due to the doubled traffic and more constraints. For large instances having a high density, we obtain up to 300 cells (therefore 600 genes) and 30,000 constraints[3].

4.2 Measure Criteria

Two criteria are used: *excess of frequencies* and *number of fitness evaluations*.

Excess: the number of frequencies added to the minimum of the optimal solution. For instance, for the problem 8.150.20 which requires 8 frequencies, an excess of 2 of a method means that the method can only find an optimal solution by adding 2 extra frequencies. This criterion is essential because adding even one frequency may make the initial problem much easier to solve. This criterion reflects the *quality* of a solution found by an algorithm.

Nb_evaluation: the evaluation number needed to obtain an optimal solution, corresponding to the exact number of points in the search space visited by an algorithm. This criterion reflects the *speed* of an algorithm and is the most objective for measuring an algorithm's performance.

The criterion of generation number is not used since two algorithms may have completely different complexities for a generation.

4.3 Results

This section describes two (classes of) EAs and their results on the two sets of FAP instances. The first EA called SM is a simple EA which uses only selection (S) and mutation (M). The second EA called SCM is SM augmented by crossover (C). The set up of SM and SCM is defined as follows.

- *SM*: SM uses the SUS selection and the conflict-based mutation defined in §3. Beginning with an initial population, SM selects first some individuals using SUS favoring well-fit individuals. Then the conflict-based mutation is applied to certain individuals according to the mutation rate.
- *SCM*: SCM uses the same selection and mutation operators as SM, but before mutation is activated, crossover (one of those described in §3) is applied to certain individuals according to the crossover rate.
- *Population size*: the population size is fixed at 50 for all of our experiments.
- *Initial population*: it is generated randomly, i.e. for each gene of an individual, a random value is taken from among all the possible frequency values.
- P_c (the crossover rate): it defines the percentage of the selected individuals that will receive a crossover operation, producing two offspring. According to our experiments, P_c is fixed at 40% in this paper.
- P_m (the mutation rate): it defines the percentage of individuals that will receive a mutation operation, i.e. one frequency (gene) of a cell will be changed. According to our experiments, P_m is fixed at 10% in this paper.

[3] The difficulty of a problem depends not only on the number of variables and the number of possible values for each variable, but also on the number of constraints.

– *Generation*: The maximum generation for each try is fixed between 10,000-50,000 according to the difficulty of the instances.

Remember that the main objective to reach in the application is to minimize the frequency interference with a minimum number of frequencies. In practice, our EAs begin with a certain number K of frequencies (a bigger value than the optimal). If an optimal solution is found within a certain number of tries (fixed at 5 in this paper) with K frequencies, then the procedure will try to solve the problem with K-1 frequencies and so on. This process continues until the procedure can no longer find an optimal solution (an interference-free assignment).

Table 1 gives the results of SM and SCM on the first set of FAP instances.

problems	Opt.	SM Excess /nb_eval.	SCM(O) Excess /nb_eval.	SCM(U) Excess /nb_eval.	SCM(CB) Excess /nb_eval.
8.150.10	8	+0/14280	+0/25510	+0/21500	+0/3000
8.150.20	8	+2/68900	+2/116250	+2/291300	+2/130200
8.150.30	8	+0/605620	+5/478170	+0/351900	+0/437500
15.150.10	15	+0/13400	+0/19700	+0/19170	+0/1630
15.150.20	15	+0/32985	+0/44952	+0/42939	+0/5484
15.150.30	15	+0/101170	+0/427240	+0/554170	+0/465900
15.300.10	15	+0/58890	+0/73280	+0/62470	+0/8540
15.300.20	15	+2/742300	+2/1107000	+2/1034000	+3/1985000
15.300.30	15	+8/756930	+8/1594558	+9/893348	+9/781000
30.300.10	30	+0/78390	+0/71360	+0/59030	+0/4250
30.300.20	30	+0/130110	+0/122900	+0/110350	+0/14150
30.300.30	30	+0/237480	+0/312470	+0/256180	+0/65370

Table 1. Comparative results on test No. 1

SCM(X) indicates the crossover operator used, O, U and CB represent one-point, uniform and conflict-based crossover respectively.

From the data in the table, two remarks can be made. First, although all of our EAs manage to solve most of the instances, they have serious difficulties with 3 instances (in bold) which are intrinsically hard. Second, according to the quality criterion, all the EAs behave similarly. However, according to the speed criterion, SCM(CB) is much more efficient. This seems logical given that the CB crossover operator directly passes on good frequencies (genes) to future generations, which favors the convergence of the algorithm.

Note also that the number of evaluations used in the table does not distinguish Nb_evaluation caused by crossover from that caused by mutation. However, as discussed in §3, evaluating an individual produced by crossover is much more expensive than evaluating an individual produced by mutation ($\Theta(N^2 * NF^2)$ v.s $\Theta(N * NF)$). The same remark remains true for Table 2. Thus, in practice, crossover can justify its utility only if it produces solutions of better quality.

Table 2 gives the results of SM and SCM on the second set of FAP instances. Since now there are two genes for each cell (the traffic is of 2), one-point and uniform are adapted to take this fact into account. We use O2 (U2) to designate the adapted one-point (uniform) crossover and O1 (U1) the original one. Now the crossover point is limited to the first gene of each cell. In particular, for uniform crossover U2, the two genes of each cell will be simultaneously exchanged in the two parents. It should be pointed out that these adapted crossovers can be also considered to be specialized operators.

problems	Opt.	SM Excess /nb eval.	SCM(O1) Excess /nb eval.	SCM(U1) Excess /nb eval.	SCM(O2) Excess /nb eval.	SCM(U2) Excess /nb eval.	SCM(CB) Excess /nb eval.
4.75.05.10	11	+1/104529	+2/98943	+2/125057	+1/260428	+2/359733	+2/14945
4.75.05.30	11	+1/243279	+1/262432	+2/193739	+1/118270	+2/97548	+2/114257
4.75.15.10	11	+6/87593	+5/724826	+6/92105	+6/457314	+5/401066	+7/17880
4.75.15.30	11	+3/507024	+5/274746	+5/957438	+5/112524	+5/425902	+6/300893
8.75.05.10	16	+0/33920	+0/33622	+0/46166	+0/33246	+0/38660	+0/4382
8.75.05.30	16	+0/32928	+0/36514	+0/43146	+0/33946	+0/40214	+0/4642
8.75.25.10	16	+7/165876	+7/130708	+5/1736100	+5/401363	+5/169621	+7/31892
8.75.25.30	16	+3/603915	+3/537415	+4/352140	+3/1017383	+4/131194	+7/22933
8.150.05.30	16	+0/191264	+0/154114	+0/445288	+0/198962	+0/151854	+0/55000
8.150.05.60	16	+0/153216	+0/118812	+0/184286	+0/140346	+0/129712	+0/31000
8.150.15.30	16	+5/1044435	+7/409000	+6/2491000	+6/1945000	+7/2374000	+8/834000
15.300.05.60	30	+0/370000	+0/221000	+0/194000	+0/196000	+0/203000	+0/8000
30.300.05.60	60	+0/413000	+0/205000	+0/193000	+0/224000	+0/179000	+0/6000

Table 2. Comparative results on test No. 2

The second set of FAP instances is, in general, much harder than the first set since the search space is much bigger: the number of genes is doubled and the number of interference constraints increases notably. This can be seen easily from the table. In fact, the EAs have serious difficulties on more instances and several extra frequencies are often needed to find a conflict-free assignment.

As was the case for Table 1, similar observations can be made for Table 2. In terms of solution quality, there is no significant difference between the three crossover operators on the one hand, and no significant difference between SM and SCM on the other. Note also that SCM(CB) is very efficient in terms of evaluation number, though its solution quality is sometime worse than others.

Finally, the resolution time varies greatly following instances. Easy instances, often having a low density of constraints, can be solved instantaneously while the solving of hard instances is, in general, very time consuming and may require several hours CPU time on a SPARC-10 station.

4.4 Discussion

In terms of solution quality, the results of SM and SCM presented in Tables 1&2 are better than those of two complete methods: graph coloring algorithms and constraint programming (CP) presented in [1, 3]. In terms of resolution speed, CP is much faster for easy instances than other methods. Compared with two of the most efficient incomplete methods SA [1] and TS [10], the results of SM and SCM are worse especially in terms of quality. For example, both SA and TS find optimal solutions for all the instances of the first set and require less extra frequencies than SM and SCM for the second set.

However, it should be noted that an efficient EA without crossover, called M&S, exists [5]. Compared with SM, there are two important differences. First, M&S uses the elitist strategy instead of SUS for selection. Second, like a stochastic hill-climber, M&S uses the conflict-based mutation augmented by a deterioration control probability, i.e. non-improved offspring produced by mutation are only accepted stochastically. These two points result in M&S and SM having different behavior and, consequently, producing different results. Indeed, M&S gives competitive or better results compared with SA and TS cited above.

5 Conclusions & Future Work

This study shows the interest of the evolutionary approach for FAP. Meanwhile, empirical evidence suggests that the contribution of the tested crossovers is marginal both in terms of solution quality and resolution speed. This is especially true for hard instances. Whether this conclusion can be generalized is not clear. However, considering its higher complexity for implementation and for fitness evaluation with respect to mutation, we can say that crossover is useful only if it proves to significantly improve the search in terms of solution quality. In the case where low quality solutions are sufficient for practical need, specialized crossovers may be developed.

At the same time, these remarks should be interpreted carefully for several reasons. First, we cannot exclude the possibility of existence of other specialized and efficient crossovers. Second, in this paper, crossover is studied in the context of traditional genetic search, i.e. crossover is used as the main search operator. Other possibilities of using crossover are worthy of investigation. For example, in mutation-oriented algorithms, any specialized efficient mutation operator may be used as the main search mechanism to reach rapidly local optima and crossover is occasionally activated to create new promising individuals in order to help the search to escape from the local optima.

Finally, the EAs presented here can be improved in several respects. First, it is evident that more efficient, application knowledge-based genetic operators may be researched. Second, a more technical and interesting improvement is possible concerning the constraint handling technique. Indeed, the current encoding takes into account only traffic constraints but not co-cell constraints. Consequently, two frequencies may be assigned to one cell during the search even if they violate the co-cell constraint. This conflict situation is only repaired (hopefully) by the search operators and may re-appear later. One way to solve this problem is to include co-cell constraints directly in the solution encoding, i.e. we make sure from the beginning of the search process that conflict frequencies will not be assigned to the same cell. In this way, the search space will be greatly reduced since both the number of constraints to be checked explicitly and the possible combinations of frequencies to be assigned to cells are reduced. Preliminary study on this issue shows very promising performance and a complete study of this technique will be reported in the near future.

Acknowledgements

We would like to thank the referees for their useful comments which helped to improve this paper and the CNET which supported this work. Special thanks go to A. Caminada and R. Mignone from the CNET for their various assistance.

References

1. A. Akrout. Problèmes d'affectation de fréquences: méthodes basées sur le recuit simulé. Technical Report RP/PAB/SRM/RRM/4123, CNET, 1994.
2. J.E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proc. of Intl. Conf. on Genetic Algorithms (ICGA'87)*, 1987.
3. A. Caminada. Résolution du problème de l'affectation des fréquences par programmation par contraintes. Technical Report FT.CNET/CNET BEL/POH/CDI/71-95/CA, CNET, 1995.
4. W. Crompton, S. Hurley, and N.M. Stephen. A parallel genetic algorithms for frequency assignment problems. In *Proc. of IMACS SPRANN'94*, pages 81–84, 1994.
5. R. Dorne and J.K. Hao. An evolutionary approach for frequency assignment in cellular radio networks. In *Proc. of IEEE Intl. Conf. on Evolutionary Computation (ICEC'95)*, Perth, Australia, 1995.
6. M. Duque-Anton, D. Kunz, and B. Rüber. Channel assignment for cellular radio using simulated annealing. In *IEEE Trans. on Vehicular Technololy*, volume 42, pages 14–21, 1993.
7. N. Funakini and Y. Takefuji. A neural network parallel algorithm for channel assignment problems in cellular radio network. In *IEEE Trans. Vehicular Technology*, volume 41, pages 430–437, 1992.
8. A. Gamst. Some lower bounds for a class of frequency assignment problems. In *IEEE Trans. on Vehicular Technololy*, volume 35, pages 8–14, 1986.
9. A. Gamst and W. Rave. On the frequency assignment in mobile automatic telephone systems. In *Proc. of GLOBECOM 82*, pages 309–315, 1982.
10. J.K. Hao and L. Perrier. Tabu search for channel assignment problems. submitted for publication, 1995.
11. A. Kapsalis, V.J. Rayward-Smith, and G.D. Smith. Using genetic algorithms to solve the radio link frequency assignment problem. In *Proc. of Intl. Conf. on ANN and GAs (ICANNGA'95)*, pages 37–40, Alès, France, 1995.
12. D. Kunz. Channel assignment for cellular radio using neural networks. In *IEEE Trans. on Vehicular Technology*, volume 40, pages 188–193, 1991.
13. G. Syswerda. Uniform crossover in genetic algorithms. In *Proc. of Intl. Conf. on Genetic Algorithms (ICGA'89)*, pages 2–9, 1989.