Constraint Handling in Evolutionary Search: A Case Study of the Frequency Assignment[*]

Raphaël Dorne and Jin-Kao Hao

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

Abstract. In this paper, we present the constraint handling techniques developed to tackle a real world optimization application: the frequency assignment problem (FAP) in cellular radio networks. The main goal of this application consists in finding assignments which minimize electromagnetic interference due to frequency reuse and minimize the number of frequencies used. An acceptable assignment of the FAP must satisfy a set of multiple constraints such as traffic constraints and interference constraints. We present how each type of constraint can play a different role in EAs. In particular, we show how co-station constraints can be used to reduce the search space efficiently. The effectiveness of the constraint handling techniques combined with a regeneration technique is tested on big and hard FAP instances. Results show that limiting the degree of unfeasible solutions is beneficial for this application.

1 Introduction

Constraint handling is an important issue in evolutionary algorithms (EAs) for tackling complex applications such as optimization and constraint problems. There are essentially three approaches for handling constraints [8]. The first totally forbids unfeasible solutions (which violate constraints) during the search and uses constraints to limit the search space by eliminating unfeasible solutions. This is often done via specialized genetic operators tailored to the application at hand. The second approach consists in using some repair mechanisms to produce feasible solutions from unfeasible solutions [2]. The third approach tolerates unfeasible solutions and uses constraint violation as a means of defining penalty functions [7, 6]. Each approach has relative advantages and disadvantages with respect to particular applications although the third is the most general approach. Finally, it will also be interesting to relate the above constraint handling techniques and recent work on the multi-objective approach to constraint optimization [8, 4].

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

In this paper, we present the constraint handling techniques we developed to tackle a real world optimization application: the frequency assignment problem (FAP) in cellular radio networks in the field of telecommunications. The main goal of this application consists in finding assignments which minimize electro-magnetic interference due to frequency reuse and minimize the number of frequencies used. The difficulty of this application comes from the fact that an acceptable assignment of the FAP must satisfy a set of multiple constraints: traffic constraints which specify the minimum number of frequencies required by each station and interference constraints which specify the minimal distances between frequencies assigned to a station or to adjacent stations. We present how these constraints can play different roles in EAs. In particular, we show how co-station constraints can be used to reduce the search space efficiently. The effectiveness of the constraint handling techniques is tested on large and hard instances having up to 350 stations, 60 available frequencies and 35,000 interference constraints. Results show that our EAs integrating these constraint processing techniques and a population regeneration technique are very efficient for this application.

The paper is organized as follows. In Section 2, we introduce the various constraints in the FAP. In Section 3, we present the techniques for efficiently handling these constraints. We define also our EAs combined with these handling techniques. In Section 4, we give the results of our EAs on a set for real-size problems.

2 Frequency Assignment Problem

2.1 Constraints

The general objective of the frequency assignment problem is to provide the maximal number of communications over the network with a sufficient quality of transmission. More concretely, this involves assigning to each radio station one or more frequency values taken from a limited radio spectrum in such a way that a set of constraints is satisfied. These constraints can be classified into three categories which are defined as follows.

1. The *frequency constraint* specifies the number of available frequencies (channels) in the radio spectrum. This constraint is imposed by national and international regulations.
2. The *traffic constraints* specify the minimum number of frequencies required by each station to serve the communications within its geographic area. These constraints are empirically determined by the telecommunications operators.
3. The *interference constraints* are further classified into two families:
   adjacent-station constraints: the frequencies assigned to two adjacent stations must be sufficiently separated in the frequency domain. Two stations are adjacent if they emit within a common area even if they are not geographically adjacent.
   co-station constraints: any pair of frequencies assigned to a station must have a certain distance between them.

Constraints in the frequency assignment problem are therefore multiple and some of them are conflicting. The most severe constraint concerns the very limited number of available frequencies. This constraint imposes a high degree of frequency reuse by the stations and consequently increases the difficulty of satisfying the interference constraints.

Different optimization versions of the FAP could be developed such as maximizing all the traffic, minimizing the number of frequencies used and minimizing the interference over the network. In this paper, we minimize the frequency interference while using a minimum number of frequencies.

2.2 Modeling

Let *NS* be the number of stations in the network, *NF* the number of available frequencies in the radio-spectrum, *NA* the number of adjacency constraints, and *NC* the number of co-station constraints. The FAP can be modeled with a quadruple <X,D,C,F> representing a constraint optimization problem (COP) with:

X = $\{S_i \mid S_i$ is a station in the network, i $\in$ [1..NS]$\}$.
D = $\{F_i \mid F_i$ is an available frequency in the spectrum, i $\in$ [1..NF]$\}$.
C = TC $\cup$ AC $\cup$ CC
TC = $\{T_i \mid T_i$ minimal number of frequencies necessary for Si, i $\in$ [1..NS]$\}$.
AC = $\{A_i \mid A_i$ adjacent-station constraints for any pair of frequencies assigned to two adjacent stations, i $\in$ [1..NA]$\}$.
CC = $\{C_i \mid C_i$ co-station constraints for any pair of frequencies assigned to the same station, i $\in$ [1..NC]$\}$.
F = the number of frequencies used to cover all the traffic in the network.
To solve the FAP problem, assignments satisfying all the constraints in C must be found and F must be minimized.

3 Constraint Handling

In this section, we show how the different types of constraints can be efficiently handled in evolutionary algorithms. Some constraints may be solved easily and others may be used to guide the search process. Note that this study have been done with EAs without a crossover operator. A study concerning crossover for the FAP has been already carried out in [5] and showed a marginal efficiency of this operator for this problem.

3.1 Frequency Constraint

This constraint requires the minimization of the frequencies used by all the stations. To do this, we proceed as follows: beginning with a certain number K of frequencies, we look for an interference-free assignment. If such an assignment is found, we start again with K-1 frequencies and so on. Details are given in Section 3.4.

3.2 Traffic Constraints

These constraints are static, that is, they will not change during the optimization process. Therefore these constraints could be directly included in the chromosome encoding as follows:

A frequency assignment is associated with a chromosome and one frequency value of a station is associated with one gene. The length of a chromosome is thus equal to the total traffic of all the stations in the network. Fig.1 gives an example where three stations require respectively two, one and four frequencies and $f_{i,j}$ represents the j^{th} frequency value of the i^{th} station S_i.

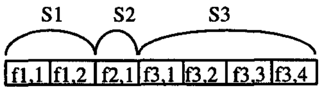

Fig. 1. Chromosome Encoding

3.3 Interference Constraints

Each interference constraint describes the minimal distance between any pair of frequencies assigned to two adjacent stations or to the same station. This set of constraints is represented by a symmetrical matrix M of NS*NS :

- M[i,j] with i $\neq$ j represents the minimal number of channel separations required to satisfy the adjacency constraints between the stations S_i and S_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 channel separations necessary to satisfy the co-station 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 interference constraint between the stations S_i and S_j.

In general, both co-station and adjacent-station constraints can be used to define the fitness function. This is perhaps the easiest technique to deal with them. However, as explained below, there are much more efficient techniques for co-station constraints. In what follows, we show first how co-station constraints can be used by EAs to reduce the search space and then how adjacent-station constraints are used to *guide* the search process.

Co-station Constraints

With the chromosome encoding presented above, different ways of handling co-station constraints will lead to a search space of different sizes. In fact, given S_i a station requiring T_i frequencies, and NF the available frequency values, the search space SP can be defined by:

$$SP = \prod_{i=1}^{NS} NP(i)$$

with $NP(i)$ being the number of possible combinations of frequencies for S_i.

Using this definition for the search space, we now investigate three ways of dealing with co-station constraints, in increasing order of their capacity to reduce the search space.

case 1: *Co-station constraints are not taken into account at all by the chromosome.* In this case, each gene may independently take any of the NF possible values. Therefore, for each station S_i, we have $NP(i) = NF^{T_i}$ possible combinations of frequencies. This implies a total search space of size

$$SP = \prod_{i=1}^{NS} NP(i) = NF^{\sum_{j=1}^{NS} T_j}$$

case 2: *Co-station constraints are taken into account by the chromosome, but an order is introduced among the frequencies of the same station.* That is, for a station S_i having a traffic of T_i, if a *list* of frequencies $(f_{i,1}, f_{i,2} ... f_{i,T_i})$ satisfying the co-station constraints is assigned to S_i, there are still other $T_i! - 1$ valid frequency assignments (lists) for S_i using the same frequencies. In this case, for each station S_i, the possible combinations of frequencies are reduced to

$NP(i) = nbposs(T_i, NF, i) * T_i!$ with

$$nbposs(T_k, N, k) = \begin{cases} N \text{ if } T_k = 1 \\ \sum_{j=1}^{N-(M[k,k]*(T_k-1))} nbposs(T_k - 1, N - (j + M[k,k] - 1, k) \text{ otherwise} \end{cases}$$

case 3: *Co-station constraints are taken into account by the chromosome and no order is imposed among the frequencies of the same station.* That is, for a station S_i having a traffic of T_i, if a *set* of frequencies $\{f_{i,1}, f_{i,2} ... f_{i,T_i}\}$ satisfying the co-station constraints is assigned to S_i, other $T_i! - 1$ possible frequency assignments for S_i using the same frequencies will be considered to be *equivalent* to this assignment. In this case, for each station S_i, the possible combinations of frequencies are reduced to

$NP(i) = nbposs(T_i, NF, i)$

For example, given a FAP instance of 75 stations with 8 available frequencies ($NF = 8$), a uniform traffic of 3 ($T_i = 3, \forall i \in [1..75]$) and a minimal separation distance of 3 for frequencies assigned to the same station ($M[i,i] = 3, \forall i \in [1..75]$), we obtain for each case above these results for the search space:

- **case 1**: $NP(i) = 8^3 = 512 => SP = 512^{75}$
- **case 2**: $NP(i) = 4 * (3!) => SP = 24^{75}$
- **case 3**: $NP(i) = 4$, which are: {1,4,7},{1,4,8},{1,5,8},{2,5,8} $=> SP = 4^{75}$

It is clear that, for this example, the application of co-station handling in cases 2 and 3 allows us to greatly reduce the initial search space by a factor of about 21^{75} and 128^{75} respectively. In general, the larger the initial search space, the bigger the reduction which may be obtained. It is also clear that the difficulty of implementation increases when we go from case 1 to case 3. In what follows, we outline how cases 1, 2 and 3 may be implemented in EAs.

For case 1, there is no special treatment for co-station constraints, and they are used together with adjacent constraints to build the fitness function. Since each gene may take any value from [1..NF], there is no need to adapt our genetic operators.

For case 2, the implementation consists in modifying the mutation operator[2] and the way in which the initial population is generated. For the initial population, we ensure that all of the co-station constraints are satisfied by each chromosome, i.e. any two frequencies assigned to the same station have the required minimal distance. It should be remembered that the mutation operator is based on a three-step choice strategy [3]:

- *choice of a station* S_i: randomly take one from among conflicting stations. A station is conflicting if some of its frequencies violate interference constraints;
- *choice of one frequency for* S_i: randomly take one from 1 to T_i;
- *choice of a frequency value* $f_{i,k}$: take the best one from 1 to NF, which is different from the current value.

In order to carry out co-station constraint processing of case 2, we enhance the third choice for selecting the frequency value; we now refuse to assign a value which violates co-station constraints. More precisely, this choice becomes:

- *choice of a frequency value* $f_{i,k}$: take the best one from 1 to NF which is different from the current value *and* which does not violate co-station constraints.

For case 3, the implementation consists once again in modifying the mutation operator and the way in which the initial population is generated. Here, an initial chromosome will be the concatenation of a list of NS sublists $(L_1, L_2, ..., L_{NS})$ where each L_i is one of S_i's $NP(i)$ valid combinations of frequencies. When mutation is applied, a sublist L_i of the chromosome will be replaced by another sublist L_i', which is always taken from S_i's $NP(i)$ valid combinations of frequencies. For the moment, case 3 is not implemented yet.

Adjacent-station Constraints

Having presented the techniques for handling the frequency constraint, traffic constraints and co-station constraints, we now study how adjacent constraints can be used. These constraints can neither be integrated into the chromosome encoding like traffic constraints nor be satisfied like the co-station constraints. However, they are useful to guide the search process. To do this, we use them to define the fitness function *Eval* which associates with each chromosome, an integer value corresponding to the number of interference constraints violated by the chromosome. The fitness will thus be a value in the interval [0...NA+NC] if the co-station constraint handling is not applied; otherwise, this interval becomes [0...NA].

Therefore, a chromosome I is a solution (a frequency assignment without interference) if and only if $Eval(I) = 0$. Evidently, different assignments could have this evaluation value and thus be solutions to the problem.

[2] If crossover is used, it should also be adapted.

3.4 Algorithm

The general algorithm we used to process the tests is described below:

```
begin
    $NF \leftarrow MAXfreq$ ; $success \leftarrow false$ ; $failed \leftarrow 0$ ; $gen \leftarrow 0$ ;
    $generate(P_0)$ ;$evaluate(P_0)$ ;$P_{best} \leftarrow P_0$;
    while ($failed < MAXfailed$) do
        $success \leftarrow false$ ; $gen \leftarrow 0$ ;
        while ($gen < MAXgen$ and $success = false$) do
            $P_{sel} \leftarrow selection(P_{gen})$ ;
            $P_{gen+1} \leftarrow mutation(P_{sel})$ ;
            $gen \leftarrow gen + 1$ ;
            $evaluateP_{gen}$ ;
            if ($Bestof(P_{gen})$) is a solution(without interference) then
                $P_{best} \leftarrow P_{gen}$ ; $success \leftarrow true$ ;
        if ($success = true$) then
            NF $\leftarrow$ NF-1 ; failed $\leftarrow$ 0;
            $P_0 \leftarrow regenerate(P_{gen}, NF)$ ;
        else
            $failed \leftarrow failed + 1$;
            $P_0 \leftarrow regenerate(P_{best}, NF)$ ;
    return $NF + 1$ ;
end;
```

P_{best} is the last population containing at least one individual having an evaluation equal to 0, i.e. an interference-free assignment with a certain number K of fixed frequencies. P_i represents the population of the i^{th} generation.

An initial population is randomly generated with the maximum number of available frequencies $MAXfreq$ (if co-station constraint processing is applied, the initial population will be forced to satisfy all the co-station constraints). This population evolves until one of its individuals becomes a solution to the problem or the maximum number of generation cycles $MAXgen$ is reached.

In the first case, the number of available frequencies NF is decremented and the population is regenerated using the following process: for each gene having a frequency value greater than the decremented NF, it is given a new value which is randomly determined in the interval [1..NF] (if co-station handling is applied, this new value will satisfy all the co-station constraints); other genes remain unchanged. We call this process the *population regeneration* technique.

In the second case, we consider this attempt as a failure. The population is regenerated using the best population P_{best} and then the evolution starts again for $MAXgen$ generations. If we get the maximal number of failed attempts $MAXfailed$ for the same number of available frequencies, the current run is finished and the last minimal number of frequencies used for an assignment which does not contain unsatisfied constraints is returned.

4 Results

Different EAs have been developed and tested on 10 FAP instances: four instances with a uniform traffic of two frequencies per station, and six instances with non-uniform traffic from 2 to 4 frequencies per station.

The name of each instance consists of three numbers *ns.nf.nc* which are respectively the number of stations, the sum of the traffic (frequencies) required by all the stations, and the number of interference constraints. For example, 150.300.12634 defines a problem composed of 150 stations, a total traffic of 300 frequencies and 12,634 constraints.

We compare below four algorithms with/without Co-station Constraint Handling (CCH) (an implementation of Case 2 described in Section 3.3.) and/or Population Regeneration (PR) described in Section 3.4.

EA: EA with no special techniques.
EA+PR: EA with only population regeneration.
EA+CCH: EA with only co-station constraint handling.
EA+CCH+PR: EA with co-station constraint handling and population regeneration.

These algorithms have the following common characteristics. We used a population of 20 individuals with SUS selection [1], no crossover, and the three-steps choice mutation described in Section 3.3. The maximum number of generation was set at 10,000.

To compare these algorithms, the following criteria are used:

- **NF**: the minimum number of frequencies used to find a frequency assignment without any constraint violation (average over 5 runs).
- **Eval**: the number of evaluations (in thousands) necessary to solve the FAP instance.
- **Time**: the average time per run in seconds excluding failed attempts. Our tests have been carried out on a Silicon Graphic Power Station with a R8000 processor.
- **Fail**: the average of failed attempts per run (see Section 3.4) over 5 runs to find the best solution.

Table 1 shows the performance of the four algorithms according to the chosen criteria. Given that the computing time for some instances is very high (>10 hours), the results are the mean of only 5 independent runs for 10 tests.

In this table, the instances are classified from the easiest to the hardest for the two families (uniform and non-uniform traffic). All the algorithms are compared in terms of solution quality (NF), efficiency (Time and Eval) and robustness (Fail). Several remarks concerning the data in the table may be made.

First, EA and EA+PR perform in a similar way except for two relatively easy instances (119.140.5918, and 168.372.4099) for which the PR option becomes more efficient (up to ten times faster). In general, at the beginning of an evolutionary optimization process, only a small number of generations is needed to satisfy most of the constraints, followed by a long series of generations to satisfy the remaining constraints. Therefore, starting a search process with the

problems	EA				EA+PR				EA+CCH				EA+CCH+PR			
	NF	Eval.	T[sec]	Fail.	NF	Eval.	T[sec]	Fail.	NF	Eval.	T[sec]	Fail.	NF	Eval.	T[sec]	Fail.
75.150.2231	15	1260	600	6.2	15.40	1300	680	5.8	14.20	1200	620	5.0	13.20	1440	740	7.8
75.150.3247	14.60	1400	760	6.8	13.60	1600	1060	6.2	13.20	1280	780	5.4	13.20	600	400	4.8
75.150.3203	18.20	2200	1380	6.4	18.20	1840	1200	6.6	17.20	2980	1880	5.8	17	1300	820	5.2
150.300.12634	25.60	2000	2400	5.8	23.40	2560	3200	9.0	20	11840	12860	6.0	18	3200	3700	5.6
119.140.5918	45	1452	889	0.0	45	190	120	0.0	45	1300	836	0.0	45	168	106	0.0
168.372.4099	45	1301	458	0.0	45	149	53	0.0	45	1186	918	0.0	45	145	54	0.0
343.966.35104	54	11926	10969	0.0	54	1461	1203	0.0	54	10375	15866	0.0	54	1326	1110	0.0
168.538.8197	45.40	3539	1530	0.0	45	585	292	0.0	45	2749	2436	0.0	45	380	176	0.0
119.335.11058	46.50	43362	45923	7.0	45.20	38428	43487	4.4	45	8800	8820	2.0	45	1280	1240	1.6
59.158.6043	56	30868	29267	4.0	56	22036	20766	3.0	56	25451	44348	4.0	55.60	6820	7040	4.6

Table 1. Comparative results

PR option is interesting if the instance is easy or if the last solution obtained is close to a final solution. Note that the PR option is designed to carry out efficient exploitation rather than exploration. Consequently, PR may put the search process in a local optimum. Many generations may be required to release the search process from this local optimum.

Second, EA+CCH gives solutions of better quality than EA and EA+PR on all the instances especially on the hardest ones. This point confirms the theoretical expectation about the search space reduction provided by CCH (c.f. Section 3.3). In terms of efficiency, EA+CCH is faster because co-station constraints are no longer a part of the fitness function and solved implicitly. Consequently, for the same number of mutation applications, the number of required evaluations is reduced. EA+CCH is also more robust than EA and EA+PR because even for a solution of better quality this algorithm has comparable failure values. Note, however, that EA+PR is more efficient for solving some easy instances of the second family.

Third, EA+CCH+PR gives far better results than other algorithms for all the instances. This is certainly due to the CCH and PR options. An important example concerns 150.300.12634 which is the hardest instance of the first family. In fact, EA+CCH+PR uses 7.6 fewer frequencies than EA and 2.0 fewer frequencies than EA+CCH. For the same instance, this algorithm divides the number of evaluations and computing time by four compared with EA+CCH. EA+CCH+PR is also robust since it gives much better results with comparable failures. In conclusion, the combination of CCH and PR is definitely very efficient. This may be explained as follows. When co-station handling is used, the co-station constraints which are harder to solve are already satisfied by all the individuals of the population. This will facilitate the search since only adjacency constraints remain. Applied to such individuals, the PR option becomes more efficient for exploitation. Moreover, the fact that CCH is applied means that the co-station constraints are removed from the fitness function. Thus, this function becomes simpler and more refined, which in turn favors a better exploration of the search space.

5 Conclusions

In this paper, we have presented a set of techniques for handling various constraints in the frequency assignment problem. Traffic constraints are solved by a direct integration into the chromosome structure. Interference constraints related to adjacent stations are used to build the evaluation function and guide the search process. Co-station constraints are used to greatly reduce the search space. We have also presented a technique for population regeneration. We have shown that this technique is powerful for exploitation. Results of experiments show that all these techniques integrated into the evolutionary framework give efficient EAs which allow us to solve hard FAP instances. Our results also show that for this application, it is beneficial to limit unfeasible solutions.

We are currently working on the last technique described in Section 3.3 in order to improve co-station constraint handling. As shown in that section, this processing will allow us to further reduce the search space. Finally, we are testing the techniques and ideas presented in this paper on other optimization and constraint problems. We hope to be able to report new results in the near future.

Acknowledgments

We would like to thank the CNET which supported this work. Special thanks go to A. Caminada and R. Mignone from the CNET for their assistance.

References

1. J.E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proc. of Intl. Conf. on Genetic Algorithms (ICGA '87)*, pages 14–21, 1987.
2. L. Davis and D. Orvosh. Shall we repair? genetic algorithms, combinatorial optimization and feasibility constraints. In *Proc. of Intl. Conf. on Genetic Algorithms (ICGA '93)*, page 650, 1993.
3. R. Dorne and J.K Hao. An evolutionary approach for frequency assignment in cellular radio networks. In *IEEE International Conference on Evolutionary Computation (ICEC'95)*, pages 539–544, Perth, Australia, 1995.
4. C. M. Fonseca and P. J. Fleming. Multi-objective optimisation and multiple constraint handling with evolutionary algorithms 1: a unified formulation. Technical Report 564, University of Sheffield, UK, 1995.
5. J.K Hao and R. Dorne. Study of genetic search for the frequency assignment problem. In *Lecture Notes in Computer Science vol. 1063, Artificial Evolution (AE'95)*, pages 333–344, Brest, France, 1995.
6. Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Springer Verlag, Berlin, 1992.
7. J. T. Richardson and M. R. Palmer. Some guidelines for genetic algorithms with penalty functions. In *Proc. of Intl. Conf. on Genetic Algorithms (ICGA '89)*, pages 191–197, 1989.
8. P. D. Surry, N. J. Radcliffe, and I. D. Boyd. A multi-objective approach to constrained optimisation of gas supplay networks: the comoga method. *Lecture Notes in Computer Science vol. 9,AISB Workshop on Evolutionary Computing*, 1994.