CHANNEL ASSIGNMENT FOR CELLULAR RADIO USING PARTICLE SWARM OPTIMIZATION

Hassan M. Elkamchouchi **Hassan M. Elragal** **Mina A. Makar**
Faculty of Engineering, Alexandria University

Abstract:
The channel-assignment problem in cellular radio networks is known to belong to the class of NP-complete optimization problems. So far, this problem has been solved by heuristic assignment strategies or by the application of combinatorial optimization tools like simulated annealing, neural networks or genetic algorithms. In this paper, we propose a new approach to the channel assignment problem by the application of the powerful optimization technique known as Particle Swarm Optimization (PSO). First the problem of channel assignment is converted to a real number problem then the algorithm is applied to the problem combined with the Frequency Exhaustive Assignment (FEA) strategy. The results obtained by the application to a well-known benchmark problem reveal that this new strategy clearly outperforms the already existing algorithms.

1. Introduction

Recent demand for mobile telephone service has been growing rapidly. At the same time, the electromagnetic spectrum or frequencies allocated for this purpose are limited. This makes solving the problem of channel assignment more and more critical. The channel assignment problem involves efficiently assigning channels or frequencies to each radio cell in the cellular radio network, while satisfying the electromagnetic compatibility constraints [1].

This paper considers the following three conditions as the electromagnetic compatibility constraints:

1. Cochannel Constraint (CCC): Same frequency can't be assigned to certain pairs of radio cells simultaneously.
2. Adjacent Channel Constraint (ACC): Frequencies adjacent in the frequency domain can't be assigned to adjacent radio cells simultaneously.
3. Co-Site Constraint (CSC): Any pair of frequencies assigned to a radio cell must have certain distance in the frequency domain.

In 1982 Gamst and Rave defined the general form of the channel assignment problem in an arbitrary inhomogeneous cellular radio network [1]. In their definition, the electromagnetic compatibility constraints in an n-cell network are described by an $n \times n$ symmetric matrix which is called the compatibility matrix **C**. Each nondiagonal element c_{ij} in **C** represents the minimum separation distance in the frequency domain between a frequency assigned to cell #i and a frequency to cell #j. The cochannel constraint is represented by $c_{ij} = 1$, and the adjacent channel constraint is represented by $c_{ij} = 2$. $c_{ij} = 0$ indicates that cells #i and #j are allowed to use the same frequency. Each diagonal element c_{ii} in **C** represents the minimum separation distance between any two frequencies assigned to cell #i, which is the co-site constraint, where $c_{ii} \geq 1$ is always satisfied. The channel requirements for each cell in an n-cell network are described by an n-element vector which is called the demand vector **D**. Each element d_i in **D** represents the number of frequencies to be assigned to cell #i. $f_{ik} = 1$ indicates the kth frequency assigned to cell #i. If the total number of available frequencies is denoted by m, then the channel assignment is represented by the constrained optimization problem ➔ Minimize m under constraints: (1) $f_{ij} = 0$ or 1, for $i \in [1,n]$ and $j \in [1,m]$

$$(2)\ \sum_{j=1}^{m} f_{ij} = d_i, \quad \text{for } i \in [1,n]$$

$$(3)\ |p-q| \geq c_{ij}, \quad \text{for } p,q \in [1,m] \text{ and } i,j \in [1,n] \text{ such that } f_{ip} = f_{jq} = 1$$

Or in other words, the channel assignment problem in the cellular radio network is finding a conflict-free frequency assignment with the minimum number of total frequencies m, where **C** and **D** are given.

The paper is organized as follows. In section 2, the previous work in the area of channel assignment is discussed with details on the algorithm presented in [2] as a base for our work. In section 3, the Particle Swarm Optimization (PSO) algorithm is explained especially its application to real number optimization problems. In

Authorized licensed use limited to: University of Johannesburg. Downloaded on February 24, 2009 at 04:20 from IEEE Xplore. Restrictions apply.

section 4, we discuss our new algorithm. First, we convert the problem of channel assignment from an integer programming optimization problem to a real number optimization problem which facilitates the application of PSO and hence makes it easier to solve. Then the application of PSO to channel assignment combined with the frequency exhaustive assignment (FEA) strategy is presented in this section. Finally, in section 5, we provide the simulation results for the convergence behavior of the problem using our new algorithm compared with the application of Genetic Algorithms (GA) to the same problem. A well known benchmark problem is used for comparison and evaluation.

2. Previous Approaches to Channel Assignment Problem

Investigating the already published approaches to the channel assignment problem, it is possible to divide the existing approximative algorithms into two different groups. On one hand [3], there exist algorithms which first determine an ordered list of all calls in the whole system and then assign frequencies to the calls following a deterministic assignment strategy. On the other hand, many researchers proposed to formulate a cost function which evaluates for example the number of interference constraints violated by a given frequency assignment and then try to minimize this cost function.

(A) Call Orderings: A quite old approach to the channel-assignment problem consists of the idea of first generating a list of all calls in the system [3]. Denoting the total number of calls in the whole system by k where k is given by a summation over the demand vector **D**, first all calls are continuously numbered and then these numbers are entered in a list. This list can then be identified with a corresponding vector **L** which contains the k different call numbers as described by $\mathbf{L} = (l_i), \quad l_i, i \in [1,k], \quad \text{and } l_i \neq l_j \text{ for } i \neq j$. Once given such a list of calls, some authors propose to assign the frequencies to the calls following a deterministic strategy [3,4]. Such a well-known assignment algorithm is for example the so called "frequency exhaustive assignment (FEA) strategy". Starting at the top of list **L**, this algorithm assigns to each call the lowest possible frequency consistent with previous assignments, i.e., without violating the interference constraints. It should be obvious that the number of frequencies m which is necessary to allocate a proper channel for each call depends significantly on the order of the calls within the list. So in literature there can be found proposals to sort the calls according to some heuristic measures of the difficulty on assigning frequencies to the calls [4]. Unfortunately, even such a process of sorting in general does not lead to an optimal order of the calls and consequently no optimal frequency assignments can generally be found this way. The Pseudo-code for this method is:

```
Loop (L1) over all call numbers l_i of list L
    Loop (L2) over all m frequencies (sorted in ascending order)
        Assign the current frequency to call number l_i and quit loop (L2), if this assignment does not lead
        to interferences with previous assignments. Otherwise the call number l_i is considered a blocked
        call.
    End (L2)
End (L1)
```

(B) Minimization of Cost Functions: Many scientists tried to get improved solutions for the channel assignment problem by formulating proper cost functions. Such a cost function takes a value of zero if all constraints of the frequency assignment problem are fulfilled by a determined channel assignment. Most recently published papers only differ in the method used to minimize the generally nonlinear cost function. A quite popular method for the minimization of the cost function consists in the application of neural networks [5-8]. Unfortunately, the minimization of the described cost function is a quite difficult problem because the danger of getting stuck in local minima constitutes a very big problem. So some authors applied simulated annealing to overcome this problem [9]. The main drawback of such algorithms consists in the fact that the convergence behavior strongly depends on the seed of the used random number generator and an appropriate choice of several parameters. A much more powerful approach to cope with the problem of local minima consists in the application of genetic algorithms. Consequently, some authors proposed to minimize the cost function this way [10-12]. Nevertheless, for very hard channel assignment problems with the minimum number of frequencies, it seems to be quite impossible to minimize the cost function to the desired value of zero, i.e., some of the interference constraints are violated by the generated frequency assignments.

Authorized licensed use limited to: University of Johannesburg. Downloaded on February 24, 2009 at 04:20 from IEEE Xplore. Restrictions apply.

(C) Combined Genetic Algorithm (CGA) method: The heuristic approaches (FEA strategy) applied a simple assignment strategy based on a previously determined call list. The great advantage of such methods consists in the fact, that there are only frequency assignments derived which fulfill all constraints concerning the demand in the single cells and also the interferences between different calls. Exactly this ability constitutes the main problem of the second group of algorithms presented before. In spite of the application of quite powerful optimization tools, in the case of large and difficult frequency-assignment problems, it is nearly impossible to find a solution which fulfills all given constraints (i.e. the value of the cost function can not be minimized to zero). On the other hand, the application of tools like genetic algorithms makes it possible to explore the solution space efficiently and lowers the danger of getting the nearest local optimum as the final solution as it is the case for the heuristic algorithms discussed first.
The observation that the advantages of the first group of algorithms correspond to the disadvantages of the cost-minimizing approaches and vice versa led to the new idea [2] to combine both methods in order to also combine their advantages. The strong power of genetic algorithms is used to solve combinatorial optimization problems for the determination of an optimal call list **L**. After the determination of a satisfying call list, frequencies are assigned with the FEA strategy. This assures that in contrast to all existing cost-minimizing methods, we only get legal frequency assignments without any interferences. This strategy to derive only channel allocations which do not violate any of the interference constraints during the search process leads to a desirable reduction of the search space and therefore facilitates the optimization problem. The search space is restricted to solutions of a higher quality because a certain solution quality is assured by the application of the FEA strategy.
The authors in [2] called their new method "Combined Genetic Algorithm (CGA)". This method is shown in fig. (1). It can be seen that new call lists are generated by the genetic algorithm. Then the FEA strategy is used to evaluate the quality of the generated call lists. This means that for each single call list **L** the FEA strategy is applied in order to determine the number of calls without an allocated frequency, i.e. we check how many calls have to be blocked because for the given number of frequencies m the FEA strategy can not assign a proper frequency without violating the interference constraints. The number of blocked calls is denoted by b. The problem is solved when $b = 0$ which means that all the calls in the list are properly assigned to frequencies. The genetic algorithm generates new, hopefully better, call lists during each iteration. This process is repeated until the application of the FEA strategy to a proper call list leads to a frequency assignment with the desired minimum number of frequencies.

Now, we want to make some notes on the CGA algorithm, for more information the reader is referred to [2]:

1. The optimization problem in CGA method is an integer programming problem as the fitness function is evaluated for a certain call list **L** which contains call numbers from 1 to k arranged randomly.
2. The fitness function depends on both the number of blocked calls and whether the assigned frequencies are of high or low values (i.e. the algorithm gives credit to assigning frequencies with low values).
3. The algorithm uses two modified mutation operators which are different from the simple mutation operator used in the basic GA. Also, the algorithm uses a modified crossover operator which is again different from the simple crossover operator used in the basic GA.
4. Many of the optimization problems that the algorithm was tested on were considered easy problems w.r.t. the CGA method. CGA method solved these problems in the first iteration of the GA. This is because of the power of the FEA strategy in channel assignment. The first generation already contained a call list that made $b = 0$. We may say that the power of the GA was not obvious in these easy problems.

3. Particle Swarm Optimization (PSO) Algorithm

Particle Swarm Optimization (PSO) is a Swarm Intelligence method for global optimization [13]. It differs from other well-known Evolutionary Algorithms (EA). As in EA, a population of potential solutions is used to probe the search space, but no operators, inspired by evolution procedures, are applied on the population to generate new promising solutions. Instead, in PSO, each individual, named *particle*, of the population, called *swarm*, adjusts its trajectory toward its own previous best position, and toward the previous best position attained by any member of its topological neighborhood [13,14]. In the global variant of PSO, the whole swarm is considered as the neighborhood. Thus, global sharing of information takes place and the particles profit from the discoveries and previous experience of all other companions during the search for promising regions of the landscape. For example, in the single-objective minimization case, such regions

Authorized licensed use limited to: University of Johannesburg. Downloaded on February 24, 2009 at 04:20 from IEEE Xplore. Restrictions apply.

possess lower function values than others, visited previously. In the local variant of PSO, the neighborhood of each particle in the swarm is restricted to a certain number of other particles but the movement rules for each particle are the same in the two variants of the algorithm.

The basic algorithm of PSO has two different versions. The first version is the one where the particles are represented by binary strings and the other is the one where the particles are represented by real numbers in n-dimensional space where n is the dimension of the optimization problem under consideration. The application of PSO to integer programming optimization problems is done by the application of the version with real (continuous) number particles and then the resulting updated particles are approximated to the nearest integer. For more information about the use of PSO for integer programming, the reader is referred to [15]. As we will see in the next section, in order to avoid the approximations done in integer programming problems, we will convert the channel assignment problem to a real number optimization problem and thus the particles of the swarm will contain continuous numbers. Now we describe the basic algorithm of PSO in real number problems. The Pseudo-code for this algorithm is:

```
Loop
  For i = 1 to number of individuals
    If G(x_i) > G(p_i) then do          // G() evaluates fitness
      For d = 1 to dimensions
        p_id = x_id                     // p_id is best so far
      Next d
    End do

    g = i                               // arbitrary
    For j = indices of neighbors
      If G(p_j) > G(p_g) then g = j     // g is index of best performer in the neighborhood
    Next j
    For d = 1 to dimensions
      v_id(t) = v_id(t-1) + φ1(p_id - x_id(t-1)) + φ2(p_gd - x_id(t-1))
      v_id ∈ (-V_max, +V_max)
      x_id(t) = x_id(t-1) + v_id(t)
    Next d
  Next i
Until criterion
```

Note that x_i is the position of particle i, v_i is the velocity of particle i, φ_1 is a random number that gives the size of the step towards personal best, φ_2 is a random number that gives the size of the step towards global best (the best particle in the neighborhood) and G is the fitness function which we try to minimize in our optimization problem.

In order that the algorithm settles on the best solution after a sufficient number of iterations, the particles of the swarm have to lower their velocities as they go more and more in their search. This led to the application of what is called the *inertia weight* term. This term serves as a memory of previous velocities. The inertia weight controls the impact of the previous velocity: a large inertia weight favors exploration, while a small inertia weight favors exploitation. Now the update equations in the previous algorithm will be:

$$v_{id}(t) = \alpha \cdot v_{id}(t-1) + \varphi_1(p_{id} - x_{id}(t-1)) + \varphi_2(p_{gd} - x_{id}(t-1))$$
$$x_{id}(t) = x_{id}(t-1) + v_{id}(t)$$

where α is the inertia weight. We always begin the algorithm with a large value for α to ensure exploration of the whole search space, then we decrease its value gradually to ensure that the particles settle on the best solution that they find. Finally, we have to say that there are many other different versions and modifications for the PSO algorithm including other ways for damping the velocity, hybrid techniques with other optimization approaches and so on. [13,14].

Authorized licensed use limited to: University of Johannesburg. Downloaded on February 24, 2009 at 04:20 from IEEE Xplore. Restrictions apply.

4. Proposed PSO-based Algorithm for Channel Assignment Problem

In this section, we present our new PSO-based algorithm for solving the channel assignment problem. A detailed study was made on the previous algorithms (Section 2) in order to compare their results. We found that the CGA method presented in [2] and discussed in Section 2.C gave excellent results compared to other algorithms based on the application of optimization techniques and the minimization of a cost function. We decided to base our algorithm on a similar approach but instead of applying the GA, we applied the basic PSO technique. The PSO algorithm is used to generate a group of call lists then frequencies are assigned using the FEA strategy. After that, the quality of each call list is evaluated and PSO is used to update the call lists to better ones in each iteration of the algorithm. Now, we discuss our algorithm in more details.

(A) Conversion of the problem to a real number optimization problem:

As every particle in the swarm (or every chromosome in the CGA method) will represent a call list, this particle must contain integer numbers from 1 to k arranged randomly in order to be evaluated by FEA strategy. Through every update during the iterations of the algorithm, we have to restrict the updated outputs as follows (1) No outputs are allowed to take non-integer values, (2) No integer from 1 to k is repeated and (3) No integer can take value greater than k. These restrictions led to the special kinds of mutation and crossover operators used in the CGA method. Also, if we try to apply PSO algorithm to the problem without modification, we will have to apply these restrictions and this will lead to many approximations in the swarm outputs which results in slower convergence. So, we convert the representation of the call lists in the particles of the swarm to real number values as follows:

1. The dimension of each particle in the swarm is k.
2. The particles are initialized with totally random real numbers.
3. The call list generated by each particle is formed from the integers representing the ordering of an ascending order version of the real numbers present in the particle, i.e. the values in the particle are arranged in ascending order then the index of each element in this ordered version will be an integer from 1 to k arranged randomly. These indices form the call list and we can see that although the particles may contain any real number value, the restrictions on the call list are always satisfied.

(B) Application of a combined (PSO – FEA strategy) approach

The generated call lists are used to assign frequencies using the FEA strategy and then the quality of these call lists is evaluated by testing their ability to minimize a fitness function. After that, the quality of each call list is transferred back to the PSO algorithm in order to update the particles in the swarm. Here, we use the basic algorithm of PSO described in Section 3 to make these updates in each iteration and the inertia weight term is used to damp the velocity of the particles. Now, the values of real numbers in the particles are updated, so their orderings in ascending order are changed and hence every particle generates a different call list compared to the previous iteration. The new generated call list of each particle is approaching the best call list ever generated by the same particle and is also approaching the call list generated by the best particle in the swarm (global best).

(C) Evaluation

We use the number of blocked calls b, which results from the attempt to solve the frequency assignment problem with the FEA strategy, as the decisive quality measure. So, the fitness function that we try to minimize here is nothing but the number of blocked calls b. This fitness function is simpler than the one used in CGA method. We tried to simplify the fitness function as much as possible because the evaluation of the fitness function here is the step that takes most of the time of running the algorithm. As mentioned in the results of the CGA method in [2], nearly 95% of the computing time is spent within the algorithm in order to evaluate the call lists by the FEA strategy. This was noticed again during our simulations and thus a simpler fitness function will result in a great reduction in the computing time. The channel assignment problem is solved if any generated call list can make a frequency assignment with no blocked calls at all. ($b = 0$).

(D) Application of genetic algorithms with our new problem representation (Modified CGA)

In order to compare the effectiveness of the PSO algorithm in solving our optimization problem which is the minimization of the number of blocked calls while assigning frequencies, we had to apply another optimization technique to the same problem. Of course, the best choice was to apply genetic algorithms as this method was adopted before in the CGA method. When we applied GA to our problem, we didn't use the chromosome representation of the CGA method. Instead, we used a chromosome representation similar to that

Authorized licensed use limited to: University of Johannesburg. Downloaded on February 24, 2009 at 04:20 from IEEE Xplore. Restrictions apply.

we used to represent the particles in the swarm while applying the PSO algorithm. The chromosome has k dimensions and it is initialized with random real numbers. The call list is generated from each chromosome is the same way described in Section 4.A. We call this method "The modified CGA method" as it uses the same concept of the CGA method but after converting the problem to a real number optimization problem.

The modified CGA method has an important advantage over the CGA method. When we apply GA using our chromosome representation, we don't have to use the modified mutation operators or the modified crossover operator described in [2] and used in CGA method. Instead, we can use the simple mutation and crossover operators used in the basic form of genetic algorithms. This gives us the ability to solve the channel assignment problem with the modified CGA method using any software package or any hardware implementation for the basic genetic algorithm.

The same advantage is present when we apply PSO algorithm to our problem. We didn't apply any modification on the basic algorithm of particle swarm optimization in order that we can use a general purpose implementation of the algorithm to solve our problem. This real number representation of the problem gives a great flexibility when we concern implementation issues. Also, if we use a hybrid technique (hybrid PSO-GA) after that to minimize the fitness function, again there is no problem in the implementation because of the similarity of the particle representation in the swarm and the chromosome representation in the GA.

5. Simulation Results

We have applied our algorithm to many quite different examples discussed in papers related to the frequency assignment problem. In the following we present the results for the application to a 21-cell system. This special example has been considered by many authors working in the field of frequency assignment: The cellular layout of the considered 21-cell system is shown in fig. (2). Seven benchmark frequency assignment problems were solved using our proposed algorithm. All of them use the same 21-cell system but they differ in the demand vector **D** which gives the number of required calls in each cell. They differ also in the compatibility matrix **C** and thus they differ in the electromagnetic constraints including CCC, ACC and CSC. The description of these different problems is given in table (1) where the used demand vectors and compatibility matrices are given in [5].

Problem Number	1	2	3	4	5	6	7
Compatibility Matrix (C)	C_3	C_3	C_4	C_4	C_5	C_5	C_6
ACC	1	1	1	1	2	2	2
CSC	5	5	7	7	7	7	5
Demand Vector (D)	D_4	D_3	D_3	D_4	D_3	D_4	D_4
Lower Bound for m	221	381	533	309	533	309	≤ 268

Table (1): Specifications of the benchmark problems of frequency assignment

Now, we will specify the parameters used for the PSO algorithm: (1) Number of particles = 20, (2) Max. number of iterations = 200, (3) Initial inertia weight = 0.9, (4) Final inertia weight = 0.4, (5) Inertia weight decreases linearly from initial to final values throughout the 200 iterations, (6) φ_1 and φ_2 are random numbers generated using a random number generator with a uniform distribution and with random values from 0 to 1.4.

Of course, the optimal choice of parameters depends significantly on the given problem and there don't exist exact rules for the determination of suited parameters. We chose 20 particles in the swarm because a lower value will not show the useful interaction between particles and a higher value will increase the simulation time significantly. The values of the inertia weight were chosen to ensure good exploration of the search space at the early iterations and then fine search and settling on a good solution at the final iterations. Here, we want to say that the results of the simulation have less dependence on the parameters of the PSO algorithm than on the parameters of other algorithms like neural networks and simulated annealing. This is because of the power of the FEA strategy that is combined with the PSO to generate the frequency assignment plan. So, small changes in these parameters didn't lead to change in the results.

Referring back to [2], the first six problems were considered so easy to the CGA method that they were solved even before the first generation of the GA, i.e. the FEA strategy found a call list in the initial

Authorized licensed use limited to: University of Johannesburg. Downloaded on February 24, 2009 at 04:20 from IEEE Xplore. Restrictions apply.

population that makes an assignment with no blocked calls. Again, this result was obvious when we applied our PSO-FEA combined method. This of course didn't reveal the power of PSO algorithm, so we repeated our simulation with 15, 10 and 5 particles in the swarm compared to 50 chromosomes in the CGA method. We always got a solution within the first few iterations. This solution depends on both the PSO algorithm and the FEA strategy.

When we consider problem (7), we must say that as the number of the available assigned frequencies (m) decreases, the difficulty of the problem increases significantly. The compatibility matrix (**C**) and the demand vector (**D**) are shown in fig. (3). This benchmark problem was the means we used to compare between the convergence behavior of the modified CGA method (Section 4.D) and the convergence behavior of our method which is dependant on the PSO algorithm. The comparison results may be summarized as follows:

(1) When we considered difficult assignment problems (m = 253, 255 and 257), we found that the PSO-based algorithm generally outperformed the GA-based algorithm. The results using PSO and GA were close to each other, so we may say that PSO has only a small advantage over GA in these difficult problems. PSO led to better convergence in most of the runs we made but we must say that in a few runs GA led to better convergence. Examples of the simulation runs are shown in figures 4, 5 and 6. In fig. (4), a comparison is made for m = 253 frequencies, in fig. (5), a comparison is made for m = 255 frequencies and in fig. (6), a comparison is made for m = 257 frequencies. In figures 4 and 5, we see that PSO gave better results but figure 6 is an example run where GA gave better result.

(2) When we considered relatively easier assignment problems (m = 266, 267 and 268), we found that the PSO-based algorithm clearly outperformed the GA-based algorithm. Here, PSO has a great advantage over GA. Examples of sample runs for these problems are shown in fig. (7). For m = 268, PSO converged to a solution in 4 iterations while GA in 6 iterations. For m = 267, PSO converged to a solution in 4 iterations while GA in 15 iterations. For m = 266, PSO converged to a solution in 6 iterations while GA in 27 iterations. This is because the spreading of the effect of a good solution is faster in PSO than in GA. So, PSO converged to a solution with no blocked calls in fewer number of iterations.

(3) Our new algorithm was applied to problem (7) with different values for the available assigned frequencies (m) starting from m = 268 to m = 253 to evaluate the convergence behavior. Examples of these runs are shown in fig. (8). We see that the results are excellent compared to those given by the CGA method.

(4) The comparison between PSO and GA was based on the convergence behavior and not on the computing time because most of the time (nearly 95%) was spent in the FEA strategy and evaluation of fitness function. So, a time based comparison will be insignificant as both algorithms will need approx. the same time.

Conclusions:

In this paper, a new algorithm for the channel assignment problem is presented. It makes use of the powerful GA-based method called CGA algorithm. The main advantages made in our approach are converting the channel assignment problem to a real number optimization problem and applying the Particle Swarm Optimization (PSO) algorithm combined with FEA strategy to solve it. Experimental simulations proved the efficient performance of the developed technique in terms of producing results that outperformed other existing algorithms based on the minimization of a cost function. Also, PSO outperformed GA when comparing their convergence behavior. Testing hybrid PSO-GA technique and other hybrid techniques combining PSO with different optimization tools is being under investigation.

References:

[1] A. Gamst and W. Rave, "*On frequency assignment in mobile automatic telephone systems*", in Proceedings GLOBECOM'82, pp.309-315, 1982.

[2] Dirk Beckmann and Ulrich Killat, "*A new strategy for the application of genetic algorithms to the channel-assignment problem*", IEEE Trans. Veh. Technol., vol 48(4), pp. 1261–1269, July 1999.

[3] T.-M. Ko, "*A frequency selective insertion strategy for fixed channel assignment*", in Proc. 5th IEEE Int. Symp. Personal, Indoor and Mobile Radio Commun., pp. 311–314, Sept. 1994.

Authorized licensed use limited to: University of Johannesburg. Downloaded on February 24, 2009 at 04:20 from IEEE Xplore. Restrictions apply.

[4] K. S. Sivarajan, R. J. McEliece, and J. W. Ketchum, "*Channel assignment in cellular radio*", in Proc. 39th IEEE Veh. Technol. Conf., pp. 846–850, May 1989.
[5] N. Funabiki and Y. Takefuji, "*A neural network parallel algorithm for channel assignments in cellular radio networks*", IEEE Trans. Veh. Technol., vol. 41, pp. 430–437, Nov. 1992.
[6] J. S. Kim, S. H. Park, P. W. Dowd, and N. M. Nasrabadi, "*Cellular radio channel assignment using a modified hopfield network*", IEEE Trans. Veh. Technol., vol. 46, pp. 957–967, Nov. 1997.
[7] D. Kunz, "*Channel assignment for cellular radio using neural networks*", IEEE Trans. Veh. Technol., vol. 40, pp. 188–193, Feb. 1991.
[8] K. Smith and M. Palaniswami, "*Static and dynamic channel assignment using neural networks*", IEEE J. Select. Areas Commun., vol. 15, pp. 238–249, Feb. 1997.
[9] M. Duque-Anton, D. Kunz, and B. Rueber, "*Channel assignment for cellular radio using simulated annealing*", IEEE Trans. Veh. Technol., vol. 42, pp. 14–21, Feb. 1993.
[10] C. Y. Ngo and V. O. K. Li, "*Fixed channel assignment in cellular radio networks using a modified genetic algorithm*", IEEE Trans. Veh. Technol., vol. 47, pp. 163–172, Feb. 1998.
[11] J. S. Kim, S. H. Park, P. W. Dowd, and N. M. Nasrabadi, "*Channel assignment in cellular radio using genetic algorithms*", Wireless Personal Commun., vol. 3, no. 3, pp. 273–286, Aug. 1996.
[12] W. K. Lai and G. G. Coghill, "*Channel assignment through evolutionary optimization*", IEEE Trans. Veh. Technol., vol. 45, pp. 91–95, Feb. 1996.
[13] James Kennedy and Russell C. Eberhart, "*Swarm Intelligence*", Morgan Kaufmann Publishers, 2001.
[14] Mahamed G. H. Omran, "*Particle Swarm Optimization Methods for Pattern Recognition and Image Processing*", PhD. thesis , University of Pretoria, Pretoria, South Africa, November, 2004.
[15] E.C. Laskari, K.E. Parsopoulos and M.N. Vrahatis, "*Particle Swarm Optimization for Integer Programming*", Proceedings of the IEEE 2002 Congress on Evolutionary Computation, Honolulu (HI), pp. 1582-1587, 2002.

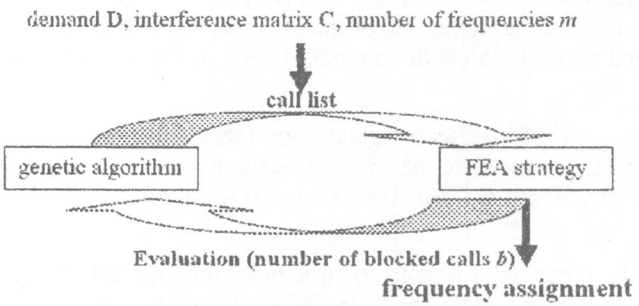

Fig.1. Combined Genetic Algorithm (CGA)method

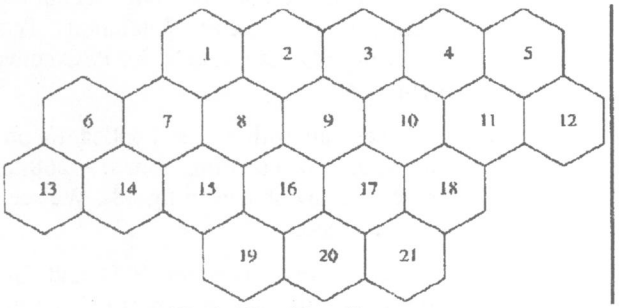

Fig.2. The 21-cell system

$$C=\begin{bmatrix}
5&2&1&0&0&1&2&2&1&0&0&0&0&1&1&1&0&0&0&0&0\\
2&5&2&1&0&0&1&2&2&1&0&0&0&0&1&1&1&0&0&0&0\\
1&2&5&2&1&0&0&1&2&2&1&0&0&0&0&1&1&1&0&0&0\\
0&1&2&5&2&0&0&0&1&2&2&1&0&0&0&0&1&1&0&0&0\\
0&0&1&2&5&0&0&0&0&1&2&2&0&0&0&0&0&1&0&0&0\\
1&0&0&0&0&5&2&1&0&0&0&0&2&2&1&0&0&0&0&0&0\\
2&1&0&0&0&2&5&2&1&0&0&0&1&2&2&1&0&0&1&0&0\\
2&2&1&0&0&1&2&5&2&1&0&0&0&1&2&2&1&0&1&1&0\\
1&2&2&1&0&0&1&2&5&2&1&0&0&0&1&2&2&1&1&1&1\\
0&1&2&2&1&0&0&1&2&5&2&1&0&0&0&1&2&2&0&1&1\\
0&0&1&2&2&0&0&0&1&2&5&2&0&0&0&0&1&2&0&0&1\\
0&0&0&1&2&0&0&0&0&1&2&5&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&2&1&0&0&0&0&0&5&2&1&0&0&0&0&0&0\\
1&0&0&0&0&2&2&1&0&0&0&0&2&5&2&1&0&0&1&0&0\\
1&1&0&0&0&1&2&2&1&0&0&0&1&2&5&2&1&0&2&1&0\\
1&1&1&0&0&0&1&2&2&1&0&0&0&1&2&5&2&1&2&2&1\\
0&1&1&1&0&0&0&1&2&2&1&0&0&0&1&2&5&2&1&2&2\\
0&0&1&1&1&0&0&0&1&2&2&1&0&0&0&1&2&5&0&1&2\\
0&0&0&0&0&1&1&1&0&0&0&0&1&2&2&1&0&5&2&1&0\\
0&0&0&0&0&0&1&1&1&0&0&0&0&1&2&2&1&2&5&2&1\\
0&0&0&0&0&0&0&1&1&1&0&0&0&0&1&2&2&1&2&5&2\\
0&0&0&0&0&0&0&0&1&1&1&0&0&0&0&1&2&2&1&2&5
\end{bmatrix}
\quad D=\begin{bmatrix}5\\5\\5\\8\\12\\25\\30\\25\\30\\40\\40\\45\\20\\30\\25\\15\\15\\30\\20\\20\\25\end{bmatrix}$$
Fig.3. C matrix and demand vector for problem (7)

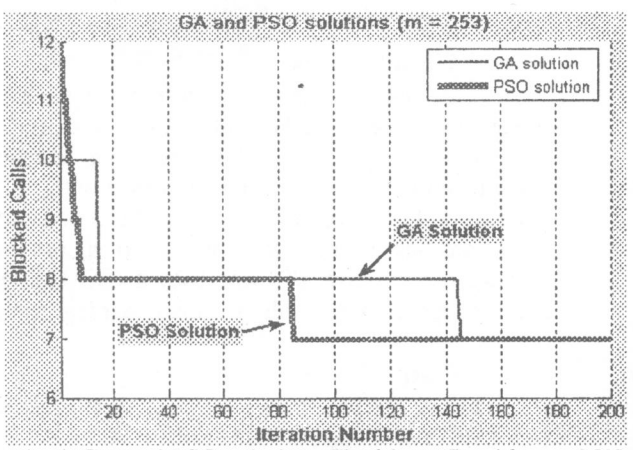

Fig.4. GA and PSO solutions [Problem (7) with m = 253]

Authorized licensed use limited to: University of Johannesburg. Downloaded on February 24, 2009 at 04:20 from IEEE Xplore. Restrictions apply.

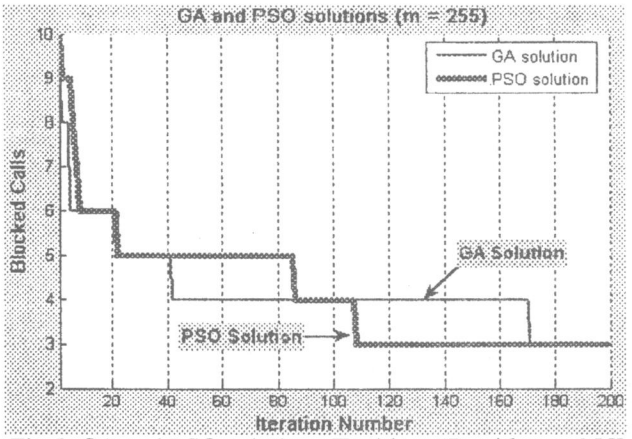

Fig.5. GA and PSO solutions [Problem (7) with m = 255]

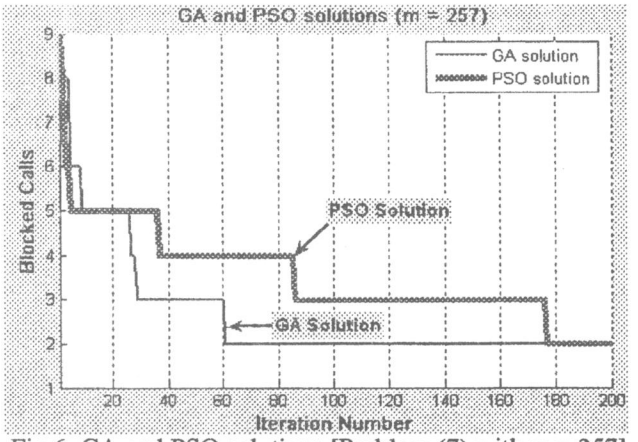

Fig.6. GA and PSO solutions [Problem (7) with m = 257]

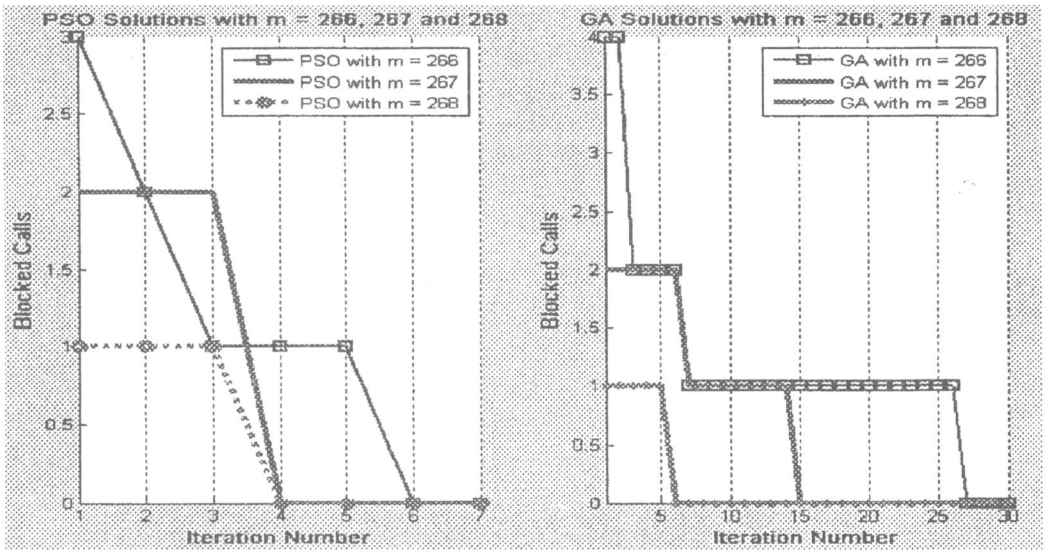

Fig.7. Comparison between GA and PSO solutions for Problem (7) with m = 266, 267 and 268 assigned frequencies. The PSO solutions (left half) clearly outperform the GA solutions (right half).

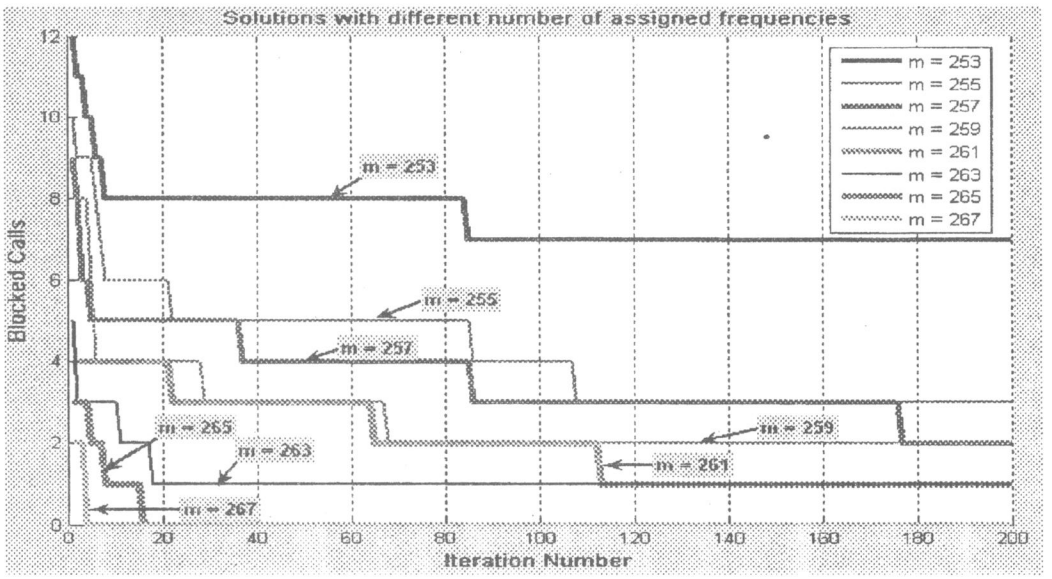

Fig.8. Solutions of Problem (7) using the proposed (PSO-FEA) strategy method. The problem is solved with different values of m (number of assigned frequencies) The figure illustrates the convergence behavior of the PSO algorithm.

Authorized licensed use limited to: University of Johannesburg. Downloaded on February 24, 2009 at 04:20 from IEEE Xplore. Restrictions apply.