A New Fixed Channel Assignment Algorithm Using Adaptive Mutation Particle Swarm Optimization

mo7amed.saeed@gmail.com, melghoneimy@yahoo.com, hanan_ak2003@yahoo.com
Mohamed Saeed Darweesh, Mona Mahmoud El-Ghoneimy, Hanan Ahmed Kamal
Cairo University-Faculty of Engineering, Giza, Egypt

***Abstract*—The limited availability of channel resources offers a bottleneck on the allocation of channels to subscribers in wireless mobile communication systems. The role of a channel assignment scheme is to allocate channels to cells or mobiles in such a way as to minimize call blocking probability under the EMC constraints. Channel assignment is known to be an NP-hard optimization problem. In this paper we propose a novel and efficient channel assignment approach, adaptive mutation particle swarm optimization. The novel algorithm can find optimal solutions to the 7-benchmark problems more than the most existing algorithms in the literature. The results are compared with those obtained by applying Genetic Algorithm and standard PSO. It has been shown that the developed AHPSO global best algorithm is faster in convergence and the obtained results are proved to have lower blocking probability than the other two algorithms.**

***Index Terms*—Fixed Channel Assignment (FCA), Electromagnetic Compatibility (EMC), Blocking probability, Particle Swarm Optimization (PSO), Genetic Algorithms (GA), Adaptive Mutation Particle Swarm Optimization (AMPSO).**

I. Introduction

With a limited frequency spectrum available, the main task of cell design is to optimize the use of the available frequency bandwidth. This is known as the channel assignment problem. The channel assignment problem can be defined as assigning a minimum number of radio frequencies to a set of cells without violating the given constraints. The channel assignment problem can be classified into three categories [1], Fixed channel assignment, Dynamic channel assignment, and Hybrid channel assignment (a combination of fixed and dynamic channel assignment.).

Fixed Channel Assignment (FCA): Its simple but not adaptive. In this scheme [1], the channels are permanently assigned to the base stations based on a predetermined traffic demand and interference constrains.

Dynamic Channel Assignment (DCA): In this scheme [2], the channels are placed into a central pool and are dynamically assigned upon request by a base station.

Hybrid Channel Assignment (HCA): In this scheme [3], the set of available channels is partitioned into two subsets of convenient sizes, one of which is allocated to the network according to the FCA scheme and the other as per DCA scheme.

The objective of a channel assignment algorithm is to determine a spectrum efficient allocation of channels to the cells while satisfying the traffic demand and electromagnetic compatibility (EMC) constraints [4]:

- The Co-Channel Constraint (CCC): the same channel can not be assigned simultaneously to a pair of cells that are not sufficiently far apart.
- The Adjacent Channel Constraint (ADC): Adjacent channels can not be assigned to adjacent cells simultaneously.
- The Co-Site Channel Constraint (CSC): the distance between any pair of channels used in the same cell must be larger than a specific distance.

In this paper, we represent a new approach, Adaptive Mutation Particle Swarm Optimization (AMPSO), to solve the channel assignment problem and minimize the blocking probability [5]. We consider a general cellular radio network satisfying a given channel demand without violating interference constraints, while the number of frequency channels is minimized.

II. Problem Formulation

The channel assignment problem has been shown to belong the class of NP-complete combinational optimization problems. In order to get a mathematical formulation of the channel assignment problem, we will assume the following; the model considered in this paper is a group of hexagonal cells, as shown in Figure 1. For an N-cell cellular system, an compatibility matrix $N \times N$ symmetric matrix [6] C= $[c_{ij}]$ can be used to represent the three types of constraints, where c_{ij} denotes the minimum channel separation between channels assigned to cell i and cell j to avoid the interference. It is easy to see that $c_{ij} = 0$ means a channel assigned to cell i can be reused at cell j. $c_{ij} = s$ means the co-site channel interference constraint is s channels. $c_{ij} = a$ means the adjacent channel interference constraint is a channels. Let the set of frequency channels be ranked by a set of positive integers $1, 2, 3, \ldots, z$ according to their carrier frequencies. The traffic demand requirement $d = d_1, d_2, \ldots, d_n$, each element d_i denotes the number of channels to be assigned to cell i [6], and the total channels in the whole network is ranked by a set of positive integers $1, 2, 3, \ldots, m$. Finally we can describe the resulted channel assignment by a binary matrix F. The representation is denoted as f_{jk} where:

$$f_{jk} = \begin{cases} 1 \\ 0 \end{cases} \quad \text{if channel } k \text{ is } \begin{Bmatrix} assigned \\ notassigned \end{Bmatrix} \text{ to cell j}$$

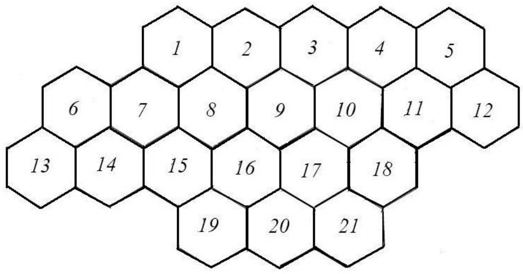

Figure 1. The 21-cell system

III. Proposed Algorithms

A. Adaptive Cauchy Mutation Particle Swarm Optimization (AMPSO)

Particle swarm optimization (PSO) is an efficient tool for optimization and search problems. However, it is easy to be trapped into local optima due to its information sharing mechanism. Many re-search works have shown that mutation operators can help PSO prevent premature convergence. In this thesis, a mutation operator that is based on the global best particle are investigated and compared for PSO. Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. (The fitness value is also stored) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called lbest. when a particle takes all the population as its topological neighbors, the best value is a global best and is called gbest. The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its pbest and lbest locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward pbest and lbest locations. An individual particle i is composed of three vectors: its position in the D-dimensional search space $\overrightarrow{x_i} = x_{i1}, x_{i2,}, \ldots, x_{iD}$, the best position that it has individually found $\overrightarrow{p_i} = p_{i1}, p_{i2,}, \ldots, p_{iD}$, and its velocity $\overrightarrow{v_i} = v_{i1}, v_{i2,}, \ldots, v_{iD}$. Particles were originally initialized in a uniform random manner throughout the search space; velocity is also randomly initialized. These particles then move throughout the search space by a fairly simple set of update equations. The Classical PSO [7] algorithm updates the entire swarm at each time step by updating the velocity and position of each particle in every dimension [8-14] by the following rules:

$$v_{id}(t+1) = v_{id}(t) + c\epsilon_1(p_{id}(t) - x_{id}(t)) + c\epsilon_2(g_{id}(t) - x_{id}(t)) \tag{1}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{2}$$

where w is the inertia weight, at every iteration i is calculated using the following equation:

$$w_i = w_{max} - ((w_{max} - w_{min})/N) * i; \tag{3}$$

and c_1, c_2: two constants multiplier terms known as "self confidence" and "swarm confidence" which respectively determine the influence of $\overrightarrow{p}(t)$ and $\overrightarrow{g}(t)$ on the velocity update formula, and ϵ_1 and ϵ_2 two uniformly distributed random numbers are independent random numbers uniquely generated at every update for each individual dimension $d = 1$ to D, and $\overrightarrow{p_g}$ is the best position found by any neighbor of the particle.

Table I
PSO with inertia Settings

Parameter	Label	Setting
Maximum Inertia weight	w_{max}	0.8 [7]
Minimum Inertia weight	w_{min}	0.4 [7]
Self Confidence	c_1	1.49618 [9]
Swarm Confidence	c_2	1.49618 [9]
Total number of iterations	i	100

An adaptive mutation operator is designed. The results show that this mutation operator can greatly enhance the performance of PSO. The adaptive mutation operator shows great advantages over non-adaptive mutation operators on a set of benchmark problems [7]. Different mutation operators can be used to help PSO jump out of local optima. However, a mutation operator may be more effective than other ones on a certain type of problems and may be worse on another type of problems. In fact, it is the same even for a specific problem at dierent stage of the optimization process. That is, the best mutation results can not be achieved by a single mutation operator, instead several mutation operators may have to be applied at dierent stages for the best performance. This thesis designs a mutation operator that can adaptively select the most suitable mutation operator for dierent problems. The mutation operator designed for the global best particles are described as follow [10]:

- Cauchy Mutation Operator

$$\overrightarrow{V_g^{'}} = \overrightarrow{V_g} exp(\delta) \tag{4}$$

$$\overrightarrow{X_g^{'}} = \overrightarrow{X_g} + \overrightarrow{V_g^{'}}\delta_g \tag{5}$$

where $\overrightarrow{X_g^{'}}$ and $\overrightarrow{V_g^{'}}$ represent the position and velocity of the global best particle. δ and δ_g denote Cauchy random numbers with the scale parameter of 1.

Figure 2, shows the Cauchy Distribution

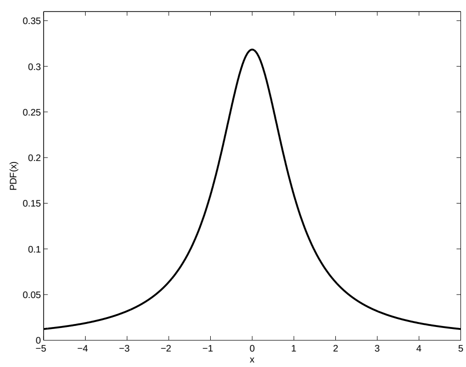

Figure 2. Cauchy distribution

The overall procedure is described as follows:

Algorithm 1 Adaptive Mutation Particle Swarm Optimization Algorithm

---

procedure AMPSO
  repeat
    for $i = 1$ to number of individuals do
      if $G(\overrightarrow{x_i}) > G(\overrightarrow{p_i})$ then $G(\ )$ evaluates fitness
        for $d = 1$ to dimensions do
          $p_{id} = x_{id}$ p_{id} is the best state found so far
        end for
      end if
      $g = i$ arbitrary
        for $j = indexes$ of neighbors **do**
          if $G(\overrightarrow{p_j}) > G(\overrightarrow{p_g})$ **then**
            $g = j$ g is the $index$ of the best performer in the neighborhood
          end if
        end for
      for $d = 1$ to number of dimensions **do**
        $v_{id}(t) = f(x_{id}(t\text{–}1), v_{id}(t\text{–}1), p_{id}, p_{gd})$ update velocity
        $x_{id}(t) = f(v_{id}(t), x_{id}(t\text{–}1))$ update position
      end for
      for $d = 1$ to number of dimensions **do**
        MutateP_g
        Select a best particle P_{gbest} from the N particles after mutation
        if the fitness value of P_{gbest} is better than P_g then $P_g = P_{gbest}$
      end for
    end for
  until stopping criteria
end procedure

B. Cloud Mutation Particle Swarm Optimization (CMPSO)

In CMPSO, the choice of P_m is created from a cloud mutation rate generator. The stable tendency of normal cloud guarantees CMPSO can improve convergent speeds, while the randomness of normal cloud model can, to a large extent, prevent CMPSO from getting stuck at a sub-optimal solution. A normal cloud is defined with three digital characteristics, expected value E_x, entropy E_n and hyper entropy H_e. and are calculated using the following algorithm [11]:

In any given population, let f_i be the fitness value of any chromosome i, f_{max} be the best fitness value and f_{avg} be the average fitness value. Then,

$E_x = f_{avg}$, $E_n = C_n * (f_{max} - f_{avg})$, and $H_e = E_n * C_e$.

For each chromosome i ,
1) If $(f_i \geq f_{avg})$

$E_n^{'}$ =Cloud Generator (E_n, H_e)

$P_m = C_m *$Cloud Generator $(E_x, E_n^{'})$

2) If $(f_i < f_{avg})$

$P_m = P_{fm}^{'}$

Where the Cloud Generator (x,y) gives a normal random number from a normal distribution with mean x and standard deviation y.

In this algorithm, for the individual whose fitness is higher than the average fitness of the population, an adaptive mutation rate are produced using a cloud generator. For the individual whose fitness is lower than the average fitness, fixed mutation rate are given, and $C_n = 1/3$ and $C_e = 1/10$ are adjustment factors that directly affects the diversity and randomness of the cloud model, $P_{fm}^{'} = 0.5$ is the fixed mutation rate, and $C_m = 0.5$ is constant.

IV. Simulation Results And Discussion

The algorithm shows a great efficiency in solving a known seven bench mark problems which are taken from [12]. In the simulation, we use a 21 cell network in the most of the problems (except one problem we use a 25 cell network).

For the 25 cell model, we use the following two demand vectors:

D_2 = [10 11 9 5 9 4 5 7 4 8 8 9 10 7 7 6 4 5 5 7 6 4 5 7 5]

For the 21 cell model, we use the following two demand vectors:

D_3 = [8 25 8 8 8 15 18 52 77 28 13 15 31 15 36 57 28 8 10 13 8]

D_4 = [5 5 5 8 12 25 30 25 30 40 40 45 20 30 25 15 15 30 20 20 25]

The compatibility matrices are shown in Figure 3.

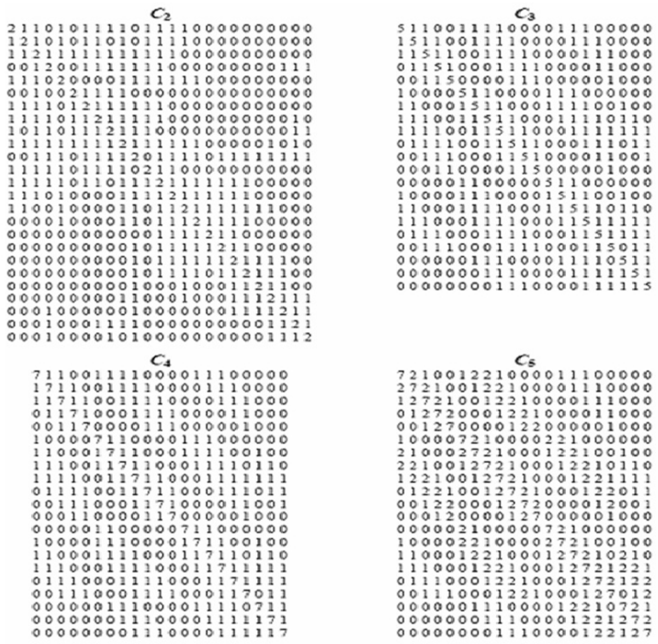

Figure 3. Compatibility matrices used for the investigated problems

Table II, presents the specifications of the case study problems used for algorithm simulation.

Table II
SPECIFICATIONS OF CASE STUDY PROBLEMS

Problem Number	Number of radio cells (n)	Minimum number of frequencies (z)	Compatibility matrix (C)	Demand vector (D)	Number of calls (m)
1	25	73	C_2	D_2	167
2	21	381	C_3	D_3	481
3	21	533	C_4	D_3	481
4	21	533	C_5	D_3	481
5	21	221	C_3	D_4	470
6	21	309	C_4	D_4	470
7	21	309	C_5	D_4	470

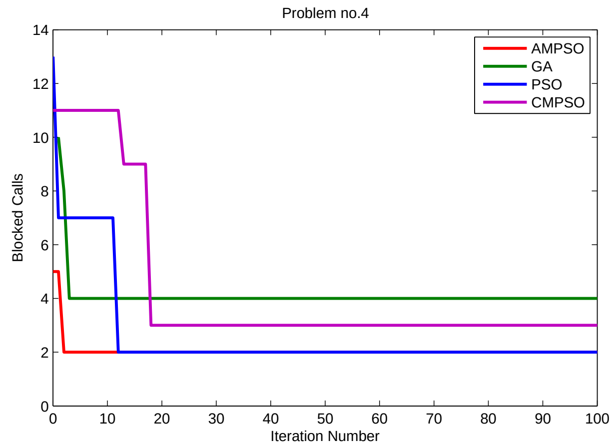

Figure 4. Results of Problem # 4

As a result for the algorithm efficiency, we chose only 100 iterations as the maximum number of iterations for our algorithm. We refer to the solutions that achieve zero blocking probability for the whole system (no call is blocked through the whole pattern).

Table III
SIMULATION RESULTS FOR THE CASE STUDY PROBLEMS

Problem Number	Number of blocked calls	Overall Blocking Probability (%)	CPU Time (sec.)
1	0	0	128.2055
2	1	0.00262	563.3603
3	1	0.00187	816.2842
4	2	0.00375	831.9336
5	0	0	545.5212
6	0	0	711.0261
7	2	0.00647	782.7361

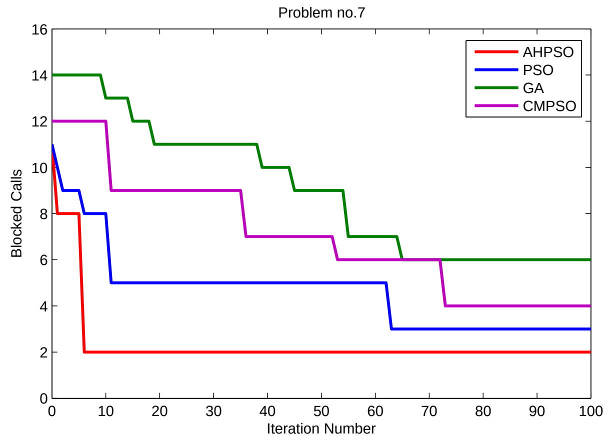

Figure 5. Results of Problem # 7

In Table IV, resulted blocking probabilities are compared with an approach using Improved Genetic Algorithm to solve the same benchmark problems [13]. As we have seen from this table, the AMPSO can solve each of the seven benchmark problems optimally. Our algorithm can achieves a zero blocking probability for 3 Problems (2, 5, 6) from the 7 Problems.

Table IV
COMPARISON OF SIMULATION RESULTS IN TERMS OF BLOCKING PROBABILITY

	Overall Blocking Probability (%)	
Problem Number	Our Algorithm Results	Results in [13] Improved Genetic Algorithm
1	0	0
2	0.00262	0.1060
3	0.00187	0.1622
4	0.00375	0.0270
5	0	0
6	0	0.0170
7	0.00647	0.0362

V. CONCLUSION

In this paper, we applied AMPSO, which is known to perform better than many other optimization algorithms (such as Genetic Algorithm and Classical Particle Swarm Optimization). The algorithm performance was measured with respect to the blocking probability, and CPU time. From the results we can see that a great improvement has been made on the blocking by using our algorithm (AMPSO) which we achieved zero blocking in problems No. (1), (5), and (6) and the remaining problems the blocking reduced to be near 100%. Finally, we can see that the proposed algorithm was shown to have a great performance and more applicable in real world problems.

REFERENCES

[1] C. Y. Ngo and V. O. K. Li, "Fixed Channel Assignment in Cellular Radio Networks Using a Modified Genetic Algorithm," IEEE Trans. On Vehicular Technology, vol. 47, no. 1, pp. 163-171, 1998.
[2] K. N. Sivarajan, R. J. McEliece., and J. W. Ketchum "Dynamic Channel Assignment in Cellular Radio", Proceedings 40^{th} IEEE Vehicular Technology Society Conference, pp. 631-637, 1990.
[3] T. J. Kahwa and N. D. Georgans, "A Hybrid Channel Assignment Schemes in Large-Scale, Cellular Structured Mobile Communication Systems", IEEE Trans. Commun., vol. COM-26, pp. 432-438, 1978.

[4] K. N. Sivarajan, R. J. McEliece., and J. W. Ketchum “Channel Assignment in Cellular Radio”, Proceedings 39^{th} IEEE Vehicular Technology Society Conference, pp. 846-850, May 1989.

[5] J. S. Kim, S. H. Park, P. W. Dowd, and N. M. Nasrabadi, "Channel assignment in cellular radio using genetic algorithms", Wireless Personal Conunun, vol. 3, no. 3, pp. 273-286, Aug. 1996.

[6] James Kemnedy and Russell C. Eberhart, "Swarm Intelligence", Morgan Kaufmiann Publishers, 2001.

[7] F. Heppner and U. Grenander., “A Stochastic Nonlinear Model for Coordinated Bird Flocks”, In S. Krasner, Ed., The Ubiquity of Chaos, AAAS Publications, Washington, DC, 1990.

[8] Mithun Chakraaborty, Rini Chowdhury, Joydeep Basu, R.Janarthanan, ”A Particle swarm optimization-based approach towards the solution of the dynamic channel assignment problem in mobile cellular networks”, Dept. of ETCE, Jadavpur University, Kolkata-700 032, 2007.

[9] Kunz.D, “Channel assignment for cellular radio using neural networks”, IEEE transactions on Vehicular technology, vol.40, pp. 188-193, February 1991.

[10] C. Li, Y. Liu, L. Kang, and A. Zhou. “A Fast Particle Swarm Optimization Algorithm with Cauchy Mutation and Natural Selection Strategy”, ISICA2007, LNCS4683, pp. 334-343, 2007.

[11] Sarah Deif, “Enhancing genetic algorithms using a dynamic mutation value approach: an application to the control of flexible robot systems”, Master Thesis, Faculty of Engineering, Cairo University, 2012.

[12] N. Funabiki, and Y. Takefuji, “A Neural Network Parallel Algorithm for Channel Assignment Problems in Cellular Radio Networks”, IEEE Trans. Vechular Technology, vol. 41, pp. 430–437, November 1992.

[13] Alaa Ahmed Mohamed Abd El Fattah El Awamy, “Improved genetic algorithm for channel assignment problem using channel borrowing technique”, Master Thesis, Faculty of Engineering, Cairo University, 2010.

[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.