

## Research Article

# Whether and How to Select Inertia and Acceleration of Discrete Particle Swarm Optimization Algorithm: A Study on Channel Assignment

Min Jin,<sup>1</sup> Xiangyuan Zhong,<sup>1</sup> and Xudong Zhao<sup>2,3</sup>

<sup>1</sup> School of Information Science and Engineering, Hunan University, Changsha, Hunan 410082, China

<sup>2</sup> College of Information Science and Technology, Bohai University, Jinzhou 121013, China

<sup>3</sup> College of Information and Control Engineering, China University of Petroleum, Qingdao 266555, China

Correspondence should be addressed to Min Jin; [jinmin@hnu.edu.cn](mailto:jinmin@hnu.edu.cn)

Received 12 September 2013; Accepted 11 December 2013; Published 5 January 2014

Academic Editor: Lixian Zhang

Copyright © 2014 Min Jin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

There is recently a great deal of interest and excitement in understanding the role of inertia and acceleration in the motion equation of discrete particle swarm optimization (DPSO) algorithms. It still remains unknown whether the inertia section should be abandoned and how to select the appropriate acceleration in order for DPSO to show the best convergence performance. Adopting channel assignment as a case study, this paper systematically conducts experimental filtering research on this issue. Compared with other channel assignment schemes, the proposed scheme and the selection of inertia and acceleration are verified to have the advantage to channel assignment in three respects of convergence rate, convergence speed, and the independency of the quality of initial solution. Furthermore, the experimental result implies that DPSO might have the best convergence performance when its motion equation includes an inertia section in a less medium weight, a bigger acceleration coefficient for global-search optimum, and a smaller acceleration coefficient for individual-search optimum.

## 1. Introduction

The ever-increasing popularity of mobile communication services gives rise to the need for efficient use of the limited frequency spectrum [1, 2]. The channel assignment problem (CAP) is to obtain a conflict-free channel assignment scheme, which satisfies both the electromagnetic compatibility (EMC) constraints and regional demand for channel. Generally, there are three types of electromagnetic compatibility constraints. The cochannel constraint (CCC) restricts the assignment of the same channel to certain pairs of cells simultaneously. The adjacent channel constraint (ACC) restricts the assignment of channels adjacent in number to adjacent cells simultaneously. The cosite constraint (CSC) specifies that any pair of channels assigned to the same cell must be separated by a certain number.

Currently, much effort has been made to solve the CAP, such as neural network (NN), simulated annealing (SA), and genetic algorithm (GA). However, NN based algorithms

typically yield only suboptimal solutions. The SA approach, although it may be more flexible, is easily trapped in a local minimum, which cannot escape without spending a lot of computation time [3]. GA can effectively locate the neighborhood of the global optimum, but it has the problem of converging to the optimum itself. In other words, the algorithms mentioned above are not particularly efficient in local search [4, 5]. Particle swarm optimization (PSO) is an intelligent algorithm introduced by [6]. It does not have the complex genetic operator and thus the computation complexity is reduced greatly. Moreover, due to the parallel computation of particle swarms, the convergence speed of PSO is much faster than GA.

In [7], a discrete particle swarm optimization (DPSO) based approach is applied to solve the CAP. The particle's position is defined as an  $N$ -dimensional vector to represent the call orderings ( $N$  is the total number of calls in the cellular system). The velocity of particle is defined as a sequence of movements from source particle to target particle, which is

randomly selected to update the particle's position according to the preset inertia factor and acceleration factor. This representation describes the distance between particles properly and reflects the mechanism of evolution. That means the new position is updated by the particle's inertia position, the particle's historical best position, and the swarm's best position. As the swarm's best acceleration has more influence on the new velocity than the particle's inertia velocity and historical best acceleration in the motion equation, the global search performs better than the local search. For this reason, a strategy of local search is introduced during the process of iteration to speed up the convergence. However, the computation complexity is increased and more iteration is needed.

In whole, there are two major problems in terms of the application of DPSO to CAP. On one hand, the selection of inertia section and acceleration coefficients has a significant effect on the performance in the motion equation. However, some selection experiences in other fields [8–12] may not be suitable for the CAP. On the other hand, the scheme of frequency exhaust assignment (FEA), a call orderings based strategy, can firstly assign channels to the most difficult cells. Thus, FEA has been proved to be an efficient scheme to solve the CAP with large and unevenly distributed traffic demand. In [5], a call orderings based strategy is proposed. From the result of several benchmark instants for CAP, the assigned channel number is disordered, while the channel number for each cell is sorted in ascending manner by FEA.

Adopting channel assignment as a case study, this paper systematically conducts experimental filtering research on the selection of inertia and acceleration in the motion equation in order for DPSO to show the best convergence performance. The rest of this paper is organized as follows. Section 2 introduces the mathematical model of CAP and the eight well-known benchmark problems. Section 3 describes the proposed channel assignment scheme which integrates the optimal DPSO with the FEA strategy in order to meet the usually large and unevenly distributed channel demand in many regions. With the experiments on the eight well-known benchmark problems, the results of the inertia choice and the acceleration adjustment are deeply analyzed and a comparison between the proposed scheme and other channel assignment schemes is further made in Section 4. A conclusion is given in Section 5.

## 2. Mathematical Formulation and Benchmark Problems

The CAP is specified by the triple  $(X, D, C)$ , where  $X$  is a cell system,  $D$  is a demand vector, and  $C$  is an  $n \times n$  symmetric compatibility matrix. Each element  $d_i$  in  $D$  represents the number of channels required by cell  $i$ . Each element  $c_{ij}$  in  $C$  represents the minimum separation distance between any two frequencies assigned to cell  $i$  and cell  $j$ . The solution space  $F$  is represented as an  $n \times m$  binary matrix, where  $m$  is the total number of available channels and  $n$  is the number of cells. Each element  $f_{ij}$  in the matrix is either 1 or 0 if  $f_{ij} = 1$  indicates the channel  $j$  is assigned to cell  $i$ . So,

TABLE 1: Specific description of the benchmark problems.

Instant	Cell	Channel	$C$	$D$
1	4	11	$C_1$	$d_1$
2	25	73	$C_1$	$d_2$
3	21	381	$C_3$	$d_3$
4	21	533	$C_4$	$d_3$
5	21	533	$C_5$	$d_3$
6	21	221	$C_3$	$d_4$
7	21	309	$C_4$	$d_4$
8	21	309	$C_5$	$d_4$

the mathematical formulation is to minimize  $z$  under the following constrains:

$$\sum_{i=1}^n f_{i,j} = d_i \quad i \in [1, n],$$

$$|p - q| \geq c_{i,j} \quad p, q \in [1, z], \quad i, j \in [1, n] \quad (1)$$

when  $f_{p,i} = f_{q,j} = 1$ .

In order to prove the performance of the algorithm, some well-known benchmark problems are simulated. Table 1 is the specific description of the benchmark problems, and all  $C$  and  $D$  used here can be found in [13].

## 3. DPSO and FEA Combined Channel Assignment Scheme

**3.1. FEA.** Compared with the method of minimum cost function [14–16], FEA has been proved to be an efficient scheme to solve the CAP with large and unevenly distributed traffic demand [17–19]. Denote the number of available frequencies by  $z$  and the total number of calls in the system by  $m$  with  $m = \sum_{i=1}^n d_i$ . The ordered call list can then be identified with a corresponding vector  $L$  which contains  $m$  different call numbers described as

$$L = (l_i) \quad i \in [1, m],$$

$$l_i \neq l_j \quad \text{for } i \neq j. \quad (2)$$

FEA starts with the first call in the ordered list. Each call with unsatisfied channel requirement is assigned a channel with the lowest rank so that the assignment satisfies all constraints with previous assignment. The pseudocode illustrating the algorithm of FEA is presented in Algorithm 1. After applying FEA, the number of blocked calls  $c$  is computed to evaluate the fitness of the ordered list.

### 3.2. DPSO

**3.2.1. Position of the Particle.** Represent the ordered list of  $n$  calls as the position of the particle by vector  $X$  with  $X = (x_1, x_2, \dots, x_i, \dots, x_n)$ , where  $x_i \in \{1, 2, \dots, n\}$ , and  $i \neq j$  when  $x_i \neq x_j$ .

```

Loop (L1) for each call  $l_i$  in list  $L$ 
  Loop (L2) for each frequency  $f$  of all  $z$  frequencies (sorted in increasing order)
    If assign the current frequency  $f$  to  $l_i$  without violation of any EMC constraints
      assign  $f$  to  $l_i$ 
    Else
      continue;
  End (L2)
End (L1)

```

ALGORITHM 1: Algorithm of FEA.

**3.2.2. Velocity of the Particle.** The best call ordering is determined by applying FEA to the list and evaluating the number of blocked calls. The velocity, used to update the particle's position to inflect the call orderings, is defined as transfer to particle's position. Denote the velocity by vector  $V = (v_1, v_2, \dots, v_i, \dots, v_n)$ , where  $v_i \in \{1, 2, \dots, n\}$ .

**3.2.3. Sum of Position and Velocity.** The sum of position and velocity is denoted as  $X_2 = X_1 + V$ , where  $X_2$  is the new position,  $X_1$  is the original position, and  $V$  is the velocity. When the particle's position is updated, if  $v_i \neq 0$ , exchange the  $v_i$ th call and the  $x_i$ th call in the list and guarantee the feasibility of new position; else keep the  $i$ th call unchanged.

**3.2.4. Subtraction of Position and Velocity.** The result of subtraction of two positions is velocity, denoted as  $V = X_2 - X_1$ . For each dimension of  $X_2$  and  $X_1$ , compare  $x_{1,i}$  and  $x_{2,i}$  one by one. If they are identical, let  $v_i$  equal 0; else let  $v_i$  equal  $x_{2,i}$ .

**3.2.5. External Multiplication of a Coefficient by a Velocity.** The external multiplication of a coefficient by a velocity is represented as  $V_2 = c \cdot V_1$ , where  $c$  is a constant between 0 and 1. For each dimension  $v_{1,i}$  of  $V_1$ , a uniformly distributed random number between 0 and 1 is generated, denoted as  $\text{rand}$ . If  $\text{rand} \geq c$ ,  $v_{2,i} = v_{1,i}$ ; else  $v_{2,i} = 0$ .

**3.2.6. Sum of Velocities.** The result of sum of two velocities is a new velocity, denoted as  $V = V_1 + V_2$ . The new velocity is defined by the equation:

$$v_i = \begin{cases} v_{1,i}, & v_{1,i} \neq 0, v_{2,i} = 0 \\ v_{1,i}, & v_{1,i} \neq 0, v_{2,i} \neq 0, \text{rand} < 0.5 \ (i = 1, 2, \dots, n) \\ v_{2,i}, & \text{others.} \end{cases} \quad (3)$$

**3.2.7. Motion Equation.** Since the peculiarity of CAP, a motion equation is proposed based on [20], in which the inertia section is determined by experience and the coefficients are adjusted finely. Here, is the motion equation

$$V = wV + c_1 (X_{pbest} - X) + c_2 (X_{gbest} - X), \quad (4)$$

$$X = X + V, \quad (5)$$

where  $X_{pbest}$  is the particle's historical best position,  $X_{gbest}$  is the swarm's best position,  $w$  is the inertia weight, and  $c_1$  and

$c_2$  are learning factors. When  $w$  equals 0, the inertia section is canceled.

With the encoding scheme of call orderings, the upper defined operations can guarantee the feasibility of particle's position. Meanwhile, the formulation still reflects the optimal mechanism for PSO.

**3.2.8. Stopping Criteria.** The stopping criterion for DPSO is that the best call ordering meeting all requirements (the number of blocked calls is 0 after applying FEA) is found or the max number of iteration is reached. If the max number is reached and the optimal solution is not found, the algorithm does not converge.

**3.3. Flowchart of DPSO and FEA Combined Channel Assignment Scheme.** The working principle and steps of DPSO and FEA combined channel assignment scheme are described as follows.

**Step 1.** Input parameters. The parameters are divided into two categories. One is the parameters of PSO, including the size of particle swarm, denoted by  $size$ , the maximum number of iterations, denoted by  $iteration$ , inertia weight, denoted by  $w$ , and learning factors, denoted by  $c_1$  and  $c_2$ . The other is the parameters of CAP, including the number of cells in cellular system, denoted by  $N$ , the number of channels, denoted by  $M$ , the matrix of EMC, denoted by  $C$ , and the demand vector, denoted by  $d$ .

**Step 2.** Initialize the position and velocity of particles randomly. If the number of calls is  $n$ , the initial position is an  $n$  dimension vector which represents a randomly ordered call list; and the initial velocity is also an  $n$  dimension vector where the  $i$ th component  $v_i$  equals  $\text{rand}(n+1)$ .  $\text{rand}(n+1)$  is a random number between (0, 1].

**Step 3.** Evaluate the fitness of each particle in the swarm. After applying FEA, the number of blocked calls is computed and the fitness is evaluated. Each particle's current position and fitness is stored in  $pbest$ , and the best of all  $pbests$  is stored in  $gbest$ . If the stopping criteria are met, output the call ordering represented by  $gbest$  and stop the algorithm.

**Step 4.** Update the velocity and position of each particle by (4) and (5).

*Step 5.* Evaluate the fitness of each particle in the swarm by computing the number of blocked calls.

*Step 6.* Compare the fitness of each particle with its *pbest*. If the current fitness is better, update its *pbest* with the particle's current position.

*Step 7.* Compare each particle's *pbest* with the swarm's *gbest*, update *gbest*.

*Step 8.* If the stopping criteria are met, output the call orderings represented by *gbest* and stops the algorithm, else go to Step 4.

## 4. Filtering Experiments and Result Analysis

*4.1. Inertia Choice and Acceleration Adjustment.* The algorithm is written in C#. In order to prove the performance of the algorithm, some well-known benchmark problems are simulated, which can be found in [13]. The efficiency of algorithm is evaluated by the convergence rate (CR) and the convergence speed (CS). The CR is defined as the number of times that the optimum solution is found by executing the algorithm 100 times. The CS is defined as the average number of iteration that the optimum solution is found by executing the algorithm 100 times. Generally, the selection of inertia and acceleration in motion equation and whether to include inertia may affect the algorithm's performance greatly. In this simulation, the algorithm is performed and evaluated without inertia and with inertia. The value of the inertia weight and the learning factors (namely the acceleration coefficients)  $c_1$ ,  $c_2$  are also determined based on these two conditions by experiment.

*4.1.1. Without Inertia ( $w = 0$  in (4)).* A group of coefficient pairs between  $[0, 1]$  is selected to determine the best acceleration coefficients  $c_1$ ,  $c_2$  by applying this algorithm to single benchmark Problem 8. Table 2 is the CR and CS of Problem 8.

From Table 2, we can draw two conclusions. One is the algorithm cannot reach 100% CR when  $c_2$  equals 1.0. The other is, when  $c_1$  keeps constant, the CS becomes faster as  $c_2$  increases. When  $c_2$  equals 0.8, the CS is fastest. However, the situation is invalid in two marginal cases (0.0 and 1.0). When  $c_2$  keeps constant, the selection of  $c_1$  has no certain impact on CS. When  $c_1$  equals 0.4, the average number of iteration is smallest (1.455) except for  $c_2$  equals 0.0 and 1.0. When the coefficients  $c_1/c_2$  select 0.8/0.8, the best CS is obtained (1.19).

As the result is obtained by only one benchmark problem, we select a group of coefficients pairs between  $[0.2, 0.8]$ , apply these coefficients to all benchmark problems (except for the simple Problem 1 and 2) and execute all the 6 problems for 5 trials to obtain the best average CS under different coefficients. The CS of each problem under different coefficient is shown in Figure 1 and the average CS of Problem 8 for each trial is listed in Table 3.

From Figure 1, we can conclude the following three conclusions. First, for complex problems, like Problems 5 and 8 which are denoted as blue and purple lines, when  $c_2$  keeps constant, the average number of iteration increases as  $c_1$

TABLE 2: CS and CR of Problem 8 without inertia.

$c_1$	$c_2$						
	0.0	0.2	0.4	0.6	0.8	1.0	
0.0	CR	68	100	100	100	100	67
	CS	48.68	1.65	1.45	1.42	1.39	50.17
0.2	CR	100	100	100	100	100	68
	CS	2.24	2.01	1.59	1.22	1.39	48.68
0.4	CR	100	100	100	100	100	75
	CS	2.81	1.66	1.47	1.39	1.3	38.25
0.6	CR	100	100	100	100	100	76
	CS	2.2	2.12	1.37	1.52	1.31	36.76
0.8	CR	100	100	100	100	100	67
	CS	2.88	3.1	1.61	1.59	1.19	50.17
1.0	CR	66	100	100	100	100	70
	CS	51.66	24.87	26.4	5.56	5.59	45.7

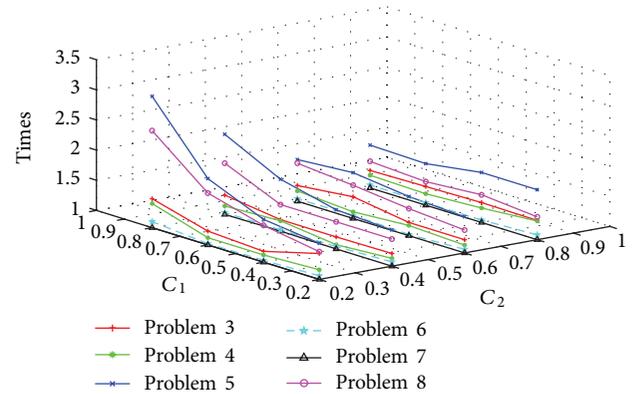


FIGURE 1: Average CS of all problems.

increases. While  $c_2$  equals 0.8, this situation is different. Second, for complex problems like Problem 5 and 8, when  $c_1$  keeps constant, the number of iteration decreases as  $c_2$  increases. Third, for other simple problems, the feature is not obvious as the number of iteration is smaller. From Table 3, we can conclude the following two conclusions. First, for all the benchmark problems, when the coefficient pair  $c_1/c_2$  selects 0.4/0.4, the least average number of iteration (1.16) is obtained. Second, the largest average number of iteration (1.60) is obtained when  $c_1$  equals 0.8 and  $c_2$  equals 0.2. The same conclusion that the average number of iteration increases as  $c_1$  increases when  $c_2$  keeps constant is obtained as the conclusion obtained from Figure 1.

Thus, for single benchmark problem and all benchmark problems, the algorithm performs best when the coefficient pair  $c_1/c_2$  selects 0.2/0.8 and  $w$  equals 0. This selection means that the convergence speed can be improved by choosing bigger acceleration coefficient to control the global-search optimum and choosing smaller acceleration coefficient to control the individual-search optimum without inertia section in motion equation for CAP.

TABLE 3: Average CS of Problem 8 without inertia.

$c_1/c_2$	Trial					avg
	1	2	3	4	5	
0.2/0.2	1.28	1.19	1.17	1.23	1.23	1.22
0.2/0.4	1.25	1.14	1.17	1.13	1.22	1.18
0.2/0.6	1.17	1.17	1.18	1.12	1.20	1.17
0.2/0.8	1.18	1.16	1.27	1.27	1.29	1.23
0.4/0.2	1.15	1.22	1.21	1.23	1.27	1.21
0.4/0.4	1.22	1.16	1.12	1.13	1.18	1.16
0.4/0.6	1.20	1.16	1.19	1.21	1.17	1.19
0.4/0.8	1.18	1.27	1.26	1.19	1.26	1.23
0.6/0.2	1.37	1.31	1.19	1.32	1.33	1.30
0.6/0.4	1.23	1.20	1.24	1.27	1.18	1.22
0.6/0.6	1.26	1.14	1.27	1.23	1.19	1.22
0.6/0.8	1.16	1.19	1.22	1.24	1.14	1.19
0.8/0.2	1.51	1.68	1.53	1.75	1.55	1.60
0.8/0.4	1.38	1.23	1.33	1.36	1.41	1.34
0.8/0.6	1.23	1.14	1.20	1.19	1.37	1.23
0.8/0.8	1.21	1.21	1.27	1.19	1.17	1.21

TABLE 4: CS of all problems with inertia.

$w$	Benchmark problem							
	1	2	3	4	5	6	7	8
0.0	1.00	1.00	1.32	1.31	1.83	1.05	1.00	1.37
0.2	1.00	1.00	1.11	1.02	1.35	1.02	1.00	1.27
0.4	1.00	1.00	1.11	1.03	1.35	1.01	1.00	1.21
0.6	1.00	1.00	1.23	1.06	1.37	1.03	1.00	1.27
0.8	1.00	1.00	1.15	1.07	1.63	1.02	1.00	1.33
1.0	1.00	1.00	1.33	1.18	1.72	1.04	1.01	1.45

4.1.2. *With Inertia ( $w \neq 0$  in (4)).* From the results obtained above, the acceleration coefficients  $c_1, c_2$  are set to be 0.2 and 0.8, respectively. A group of coefficients between  $[0, 1]$  are selected to determine the best inertia weight  $w$  by applying this algorithm to all benchmark problems. Table 4 is the result for average CS obtained when  $w$  equals 0.0, 0.2, 0.4, 0.6, 0.8, and 1.0, respectively.

From Table 4, when  $w$  equals 0.4, the algorithm performs best for all benchmark problems and does better than it performs when  $w$  equals 0.0 (without inertia).

In summary, in the simulation, this algorithm defines a motion equation with inertia section and when  $w, c_1,$  and  $c_2$  select 0.4, 0.2, and 0.8, respectively, the algorithm performs best. Besides, the swarm size of this algorithm is 10 and the max number of iteration is set to be 150 in the simulation.

4.2. *Comparison of CS and CR.* The quality of initial solution has an important influence on the CR and the CS for DPSO. To eliminate the initial solution's impact and evaluate the algorithm's performance objectively, we compute the results with one time and more than one time of iteration respectively, and compare these results with the results obtained from the latest literatures where literature [5] is based on DPSO, literature [21] is based on culture algorithm, and

TABLE 5: Comparison of CS and CR.

Method	Benchmark Problem							
	1	2	3	4	5	6	7	8
One								
CS#	1	1	1.11	1.03	1.35	1.01	1	1.21
CR%	100	100	100	100	100	100	100	100
>one								
CS#	1	1.4	2	3	3.44	4	2.6	3
CR%	100	100	100	100	100	100	100	100
[6]								
CS#	1	5	30	40	55	60	50	60
CR%	100	100	100	100	100	100	100	100
[18]								
CS#	1	2	2	2	2.25	2.75	3.52	4
CR%	100	100	100	100	100	100	100	100
[19]								
CS#	1	2284.6	34.7	12.0	—	136.5	20.2	1977.0
CR%	100	100	100	100	—	100	100	98

literature [22] is based on genetic algorithm. Table 5 is the result.

The result shows that the proposed algorithm achieves 100 percent convergence rate for all benchmark problems. In the experiment, for simple problems, like Problems 1, 2, and 7, the optimum can be obtained by only one time of iteration. Even for the complex problems like Problems 5 and 8, the algorithm achieves a fast convergence speed. Excluding the impact of initial population on the algorithm, 100 percent convergence rate is still achieved for all benchmark problems and the optimum solution is obtained after 5 times of iteration at most.

The result is better than that in [5], where the new velocity is defined as sequential sum of three movement components in the motion equation. For this reason, extra computation is needed and the movement will influence each other, which slows down the CS. The result is also better than culture algorithm and genetic algorithm based approaches which proves that the DPSO and FEA combined scheme with the selection of inertia and acceleration is an efficient approach for CAP.

## 5. Conclusion

Adopting the channel assignment on the eight well-known benchmark problems as a case study, this paper systematically conducts experimental filtering research on the selection of inertia and acceleration in the motion equation of DPSO. A channel assignment scheme integrating the optimal DPSO with frequency exhaust assignment (FEA) strategy was proposed to meet the usually large and unevenly distributed channel demand in many regions. Compared with other channel assignment schemes, the proposed scheme with the optimal inertia weight and acceleration coefficients achieves 100 percent convergence rate for all benchmark problems and the convergence speed is faster than others without

dependence on the quality of initial solution. For cellular systems with high number of cells, this proposed scheme with the optical inertia weight and acceleration coefficients can also be efficiently applied to find the exact solution in an acceptable time of computation. Furthermore, the filtering result on the eight well-known benchmark problems demonstrates that the optimal inertia weight is 0.4 and the two optimal acceleration coefficients are 0.2 and 0.8, respectively. This result implies that DSPO might be of the best convergence performance when its motion equation includes an inertia section in a less medium weight, a bigger acceleration coefficient for global-search optimum, and a smaller acceleration coefficient for individual-search optimum.

### Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

### Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant no. 61374172), National Scientific and Technological Achievement Transformation Project of China (Grant no. 201255), Electronic Information Industry Development Fund of China (Grant no. 2012407), the Military Technology Proliferation Program of China, and the Fundamental Research Funds for the Central Universities, Hunan University.

### References

- [1] G. K. Audhya, K. Sinha, and K. Mandal, "A new approach to fast near-optimal channel assignment in cellular mobile networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1814–1827, 2013.
- [2] S. Avallone, "An energy efficient channel assignment and routing algorithm for multi-radio wireless mesh networks," *Ad Hoc Networks*, vol. 10, no. 6, pp. 1043–1057, 2012.
- [3] R.-H. Cheng, C. W. Yu, and T. K. Wu, "A novel approach to the fixed channel assignment problem," *Journal of Information Science and Engineering*, vol. 21, no. 1, pp. 39–58, 2005.
- [4] S.-S. Kim, A. E. Smith, and J.-H. Lee, "A memetic algorithm for channel assignment in wireless FDMA systems," *Computers and Operations Research*, vol. 34, no. 6, pp. 1842–1856, 2007.
- [5] K. N. Zhou and Y. L. Hu, "Improvement of computational efficiency for genetic algorithms," *Control Theory & Applications*, vol. 19, no. 5, pp. 812–814, 2002.
- [6] J. Kennedy, "Minds and cultures: particle swarm implications for beings in sociocognitive space," *Adaptive Behavior*, vol. 7, no. 3-4, pp. 269–288, 1999.
- [7] L. Benameur, J. Alami, and A. El Imrani, "A new discrete particle swarm model for the frequency assignment problem," in *Proceedings of the 7th IEEE/ACS International Conference on Computer Systems and Applications*, pp. 139–144, Rabat, Morocco, May 2009.
- [8] Y.-W. Zhong and R.-Y. Cai, "Discrete particle swarm optimization algorithm for QAP," *Acta Automatica Sinica*, vol. 33, no. 8, pp. 871–874, 2007.
- [9] J. D. Zhang, C. S. Zhang, and S. B. Liang, "The circular discrete particle swarm optimization algorithm for flow shop scheduling problem," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5827–5834, 2010.
- [10] H.-F. Xiao and G.-Z. Tan, "A novel particle swarm optimizer without velocity: Simplex-PSO," *Journal of Central South University of Technology*, vol. 17, no. 2, pp. 349–356, 2010.
- [11] L. X. Zhang, H. J. Gao, and O. Kaynak, "Network-induced constraints in networked control system—a survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 403–416, 2013.
- [12] L. X. Zhang and P. Shi, "Stability, l2-gain and asynchronous  $H_\infty$  control of discrete-time switched systems with average dwell time," *IEEE Transactions on Automatic Control*, vol. 54, no. 9, pp. 2192–2199, 2009.
- [13] N. Funabiki and Y. Takefuji, "A neural network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Transactions on Vehicular Technology*, vol. 41, no. 4, pp. 430–437, 1992.
- [14] G. Chakraborty and B. Chakraborty, "A genetic algorithm approach to solve channel assignment problem in cellular radio networks," in *Proceedings of the 1999 IEEE Midnight-Sun Workshop on Soft Computing Methods in Industrial Application*, pp. 34–39, Kuusamo, Finland, 1999.
- [15] C. Y. Ngo and V. K. Li, "Fixed channel assignment in cellular radio networks using a modified genetic algorithm," *IEEE Transactions on Vehicular Technology*, vol. 47, no. 1, pp. 163–172, 1998.
- [16] M. Asvial, B. G. Evans, and R. Tafazolli, "Evolutionary genetic DCA for resource management in mobile satellite systems," *Electronics Letters*, vol. 38, no. 20, pp. 1213–1214, 2002.
- [17] D. Beckmann and U. Killat, "A new strategy for the application of genetic algorithms to the channel-assignment problem," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 4, pp. 1261–1269, 1999.
- [18] X. N. Fu, A. G. Bourgeois, and P. Z. Fan, "Using a genetic algorithm approach to solve the dynamic channel-assignment problem," *International Journal of Mobile Communications*, vol. 4, no. 3, pp. 333–353, 2006.
- [19] K. L. Yeung and T. P. Yum, "Fixed channel assignment optimization for cellular mobile networks," *IEICE Transactions on Communications*, vol. E83-B, no. 8, pp. 1783–1791, 2000.
- [20] J. S. Hu, C. X. Guo, B. Ye, H. M. Duan, and Y. J. Cao, "Application of discrete particle swarm optimization to transmission network expansion planning," *Automation of Electric Power Systems*, vol. 28, no. 20, pp. 31–36, 2004.
- [21] J. Alami and A. E. Imrani, "Using cultural algorithm for the fixed-spectrum frequency assignment problem," *Journal of Mobile Communications*, vol. 2, no. 1, pp. 1–9, 2008.
- [22] M.-L. Li, Y.-N. Wang, L. Du, and G.-X. Wang, "Channel assignment problem in cellular networks using genetic algorithms," *Journal of Northeastern University*, vol. 24, no. 3, pp. 213–216, 2003.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

