Parameters Selection in the Discrete Particle Swarm Optimization Algorithm Solving Gate and Runway Combinational Optimization Problem

Jianli Ding, Yong Zhang
College of Computer Science and Technology
Civil Aviation University of China
China
jianlidingyahoo.com.cn, zhangyong03_2008@yahoo.cn

JDIM
Journal of Digital Information Management

ABSTRACT: *Gate and runway combinational optimization (GRCO) problem is of great significance in airport operation. In this paper, experimental analysis is performed on the parameters characteristics of the discrete particle swarm optimization (DPSO) algorithm for combinatorial optimization problems. Inertia weight, acceleration constants and population size all have an important impact on the performance of the algorithm. Optimal values exist in the specific application of the respective parameters. The optimal value interval of the population size does not vary with different acceleration constants when inertia weight is constant. Increasing the population size can improve the solution quality, but the time overhead increases significantly. Finally, each parameters selection guidelines are provided.*

Categories and Subject Descriptors:
D.4.6 **[Security and Protection]**; Information Flow Controls: B.6 **[LOGIC DESIGN]**; Combinational Logic

General Terms:
Particle Swarm Optimization, Combinational Optimization

Keywords: Airport Operation, Gate and Runway Combinational Optimization, Parameters Selection, Discrete Particle Swarm Optimization

Received: 10 January 2013, Revised 3 March 2013, Accepted 12 March 2013

1. Gate And Runway Combinational Optimization Problem

Gates and runways are critical resources provided to flights by civil airport, and reasonable optimization of the allocation of gates and runways can reduce flight taxiing time, improve the operating efficiency of the airport, reduce airline operating costs and reduce flight delays.

Under normal circumstances, taxi route between gate and runway is fixed. Therefore, in cases not considering the sliding blockage, a specific flight can have different taxiing time among different allocations to gate and runway combinations. Traditional gate assignment problem only considers constraints related to gate assignment to determine the flight-to-gate assignment scheme minimizing the objective function, while gates and runways combinatorial optimization should consider constraints related to both gate assignment and runway assignment to make the optimal assignment scheme that minimize the flight taxiing time. Literature [16] gives detailed analysis of the airport gate and runway combinational optimization problem.

Gate assignment problem and runway assignment problem have been conducted extensive research by domestic and foreign scholars. Babic et al. formulated a 0-1 integer programming model for the AGAP and used the branch and bound method to search the optimal assignment scheme, while not considering transfer passengers [1]. Yan and Chang proposed a multi-commodity network flow model and used the Lagrangian relaxation method with sub-gradient and a Lagrangian heuristic function to solve the problem [2]. Yan and Huo proposed a bi-objective 0-1 integer programming model to assignment gates. One goal is to minimize passengers' travelling time and the second objective is to minimize passengers' waiting time, because while an aircraft in flight peak time is waiting for available seats, passengers are also waiting for its target flight [3]. Gu and Chung proposed a genetic algorithm to solve the AGAP problem where the optimization objective is to minimize delay introduced by gate reassignment [4]. Jiefeng Xu et al. have designed a kind of simple heuristic tabu search algorithm to solve the problem by constructing AGAP (Airport Gate Assignment Problem) as a mixed 0-1 quadratic integer programming problem, and then converting it into a 0-1 integer programming problem with a linear objective function and constraints [5]. Ding et al. considers the cases where the total number of flights is greater than the number available gates and makes minimum number of flights not assigned to gate and the three kinds of passengers' the minimum walking distances as the optimization objective. On the basis of neighborhood search technique in [5], the authors proposed a new more efficient neighborhood and a tabu search method to solve AGAP [6] [7]. Kim et al. have proposed an improved tabu search (TS) algorithm with the objective of minimizing the sum of the aircraft movement time and passenger movement time in terminal area. They regarded the movement time of the aircraft in terminal as a kind of movement time of the passengers [8]. Dorndorf et al. gives a detailed overview of the airport gate assignment problem [9].

2. The DPSO Algorithm Solving Grco Problem

Particle swarm optimization (PSO) Algorithm was first proposed by Eberhart and Kennedy in 1995 to solve the optimization problem of continuous functions [10].Yuhui Shi et al. analyzed the impact that inertia weight and maximum velocity have on the performance of the particle swarm optimizer, and provides guidelines for selecting these two parameters [11]. Liping Zhang et al. investigated the characteristics of the each parameters of the PSO algorithm with convergence factor based on 9 benchmark functions and gave guidelines to determine the optimal parameter values [12]. Kennedy and Eberhart proposed a discrete binary particle swarm optimization algorithm for solving problems with discrete change of variables in the solution space [13]. Kusum Deep et al. proposed a non-deterministic adaptive inertia weight particle swarm algorithm in which the inertia weight dynamically adjusts itself according to the fitness increment over each iteration [14]. Chen Dong et al. proposed a new self-adaptive inertia weight method which is defined in terms of the particle fitness, swarm size and the dimension size of solution space [15].

Jianli Ding et al. proposed a discrete particle swarm optimization algorithm for gate and runway combinatorial optimization problem [16] (GR-DPSO). The algorithm does not have separate concept of particle velocity, instead of using a neighborhood function of position vector as the velocity of the particle. The algorithm is described as follows:

Step 1: Initialize each particle's position vector.

Step 2: Calculate the value of the fitness function of each particle, If this value is better than the fitness value of *pBest*, update fitness value of the *pBest* as the particle current fitness value and update *pBest* as the current position of the particle.

Step 3: Update fitness value of the *gBest* as the particle current fitness value and update *gBest* as the current position of the particle if the current fitness value is better.

Step 4: Update particle's position vector according to the position updating formulation.

Step 5: If the ending condition is not satisfied, then go to step2, else end.

3. Parameters Selection And Experimental Analysis

In this paper, more than 1000 flight data of a domestic airport in one day have been used for test analysis. There are 166 available gates and 3 runways in the airport. The GR-DPSO implementation was written in Java and compiled using the Eclipse Version 3.3 compiler. For purposes of comparison, all the simulations use the same parameter settings for the GR-DPSO implementation except the parameter being observed. For each set of parameters selected, 30 runs were performed and the results were shown in Figure. 1 to 3.

In GR-DPSO, the inertia weight w is constricted by the attenuation factor k, and acceleration constants c_1, c_2 satisfy the following constraint $1 = c_1 + c_2$ and w, c_1, c_2 are greater than 0. However, when the $w * k > 1$, the position of the particle will be changed only by the particle velocity, so the test suite was run by varying *w* from 0.2 to 1.0 with increment of 0.01 and from 1.0 to 3.0 with increment of 0.1. Due to the role of the ⊕ operator in GR-DPSO, $(1 - w * k) * c_1$ and $(1 - w * k) * c_2$ control the influence of the self-cognitive part and the social cognitive part of the particle position separately. Considering $1 = c_1 + c_2$, only c_1 is observed varying from 0.2 to 0.9 with increment of 0.01.

The last parameter is the population size N. The population sizes were varied from 30 to 200 in steps of 10. The ranges and increments of parameters used are listed in Table 1.

Parameter	Range	Increment
w	[0.2-3.0]	0.01/0.1
c_1	[0.2-0.9]	0.01
N	[30-200]	10

Table 1. Parameter ranges

3.1 Inertia Weight

Figure.1 illustrates that the average fitness varied with w from 0.2 to 3.0 for test data selected when c_1 = 0.429 and N = 30, and we can see that the average fitness first decreases and then increases while *w* is increasing. So the best range of *w* is obvious in this case, and [0.4-2.0] may be proper. Two other curves are also shown in Figure.

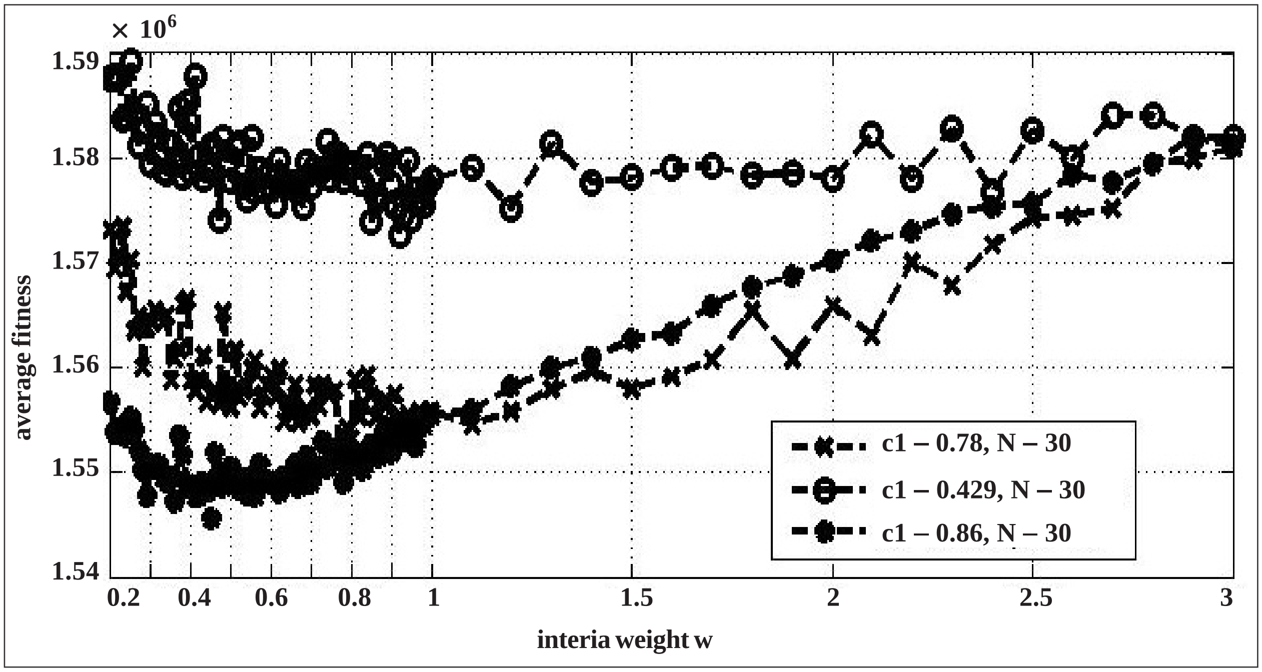

Figure 1. The Average Fitness Variation with w When $c_1 = 0.429, 0.78, 0.86$ and $N = 30$

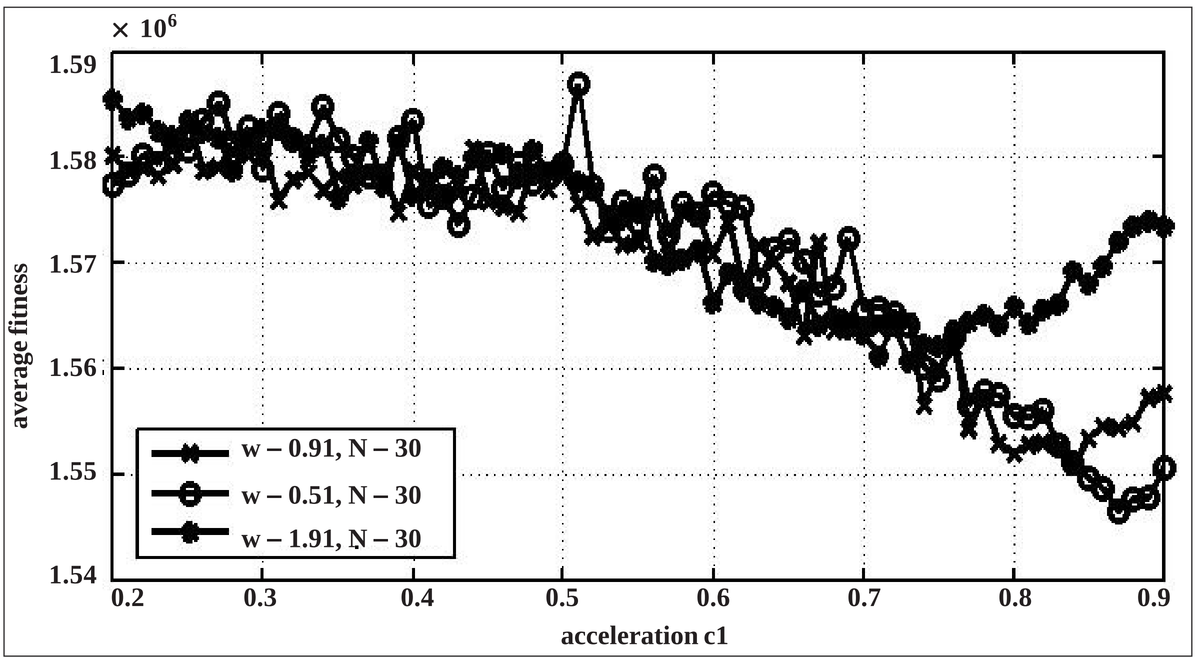

Figure 2. The Average Fitness Variation with c_1 When $w = 0.51, 091, 1.91$ and $N = 30$

1 which represent the situations in which $c_1 = 0.78$ and $N = 30$, $c_1 = 0.86$ and $N = 30$ respectively. Again the best ranges are obvious, and ranges [0.6-1.6] and [0.4-0.6] may be proper for each. However, the fitness changes more dramatic in these two cases and the intervals decrease in turn. More importantly, the c_1 corresponding to the bottom of these three curves decreases in turn. It is concluded that higher the c_1 is, lower the w when N is constant.

3.2 Acceleration Constants

Figure.2 illustrates that the average fitness varied with c_1 from 0.2 to 0.9 for test data selected when $w = 0.51$ and $N = 30$, and we can see that the average fitness first decreases and then increases while c_1 is increasing. So the best range of c_1 is obvious in this case, and [0.83-0.89] may be proper. Two other curves are also shown in Figure. 2 which represent the situations in which $w = 0.91$ and $N = 30$, $w = 1.91$ and $N = 30$ respectively. Again the best ranges are obvious, and ranges [0.79-0.86] and [0.68-0.76] may be proper for each. However, the c_1 corresponding to the bottom of these three curves decreases in turn. It is concluded that higher the w is, lower the c_1 when N is constant. It's that w keeps the balance between exploration and exploitation. With the observation of these three best ranges, it's concluded

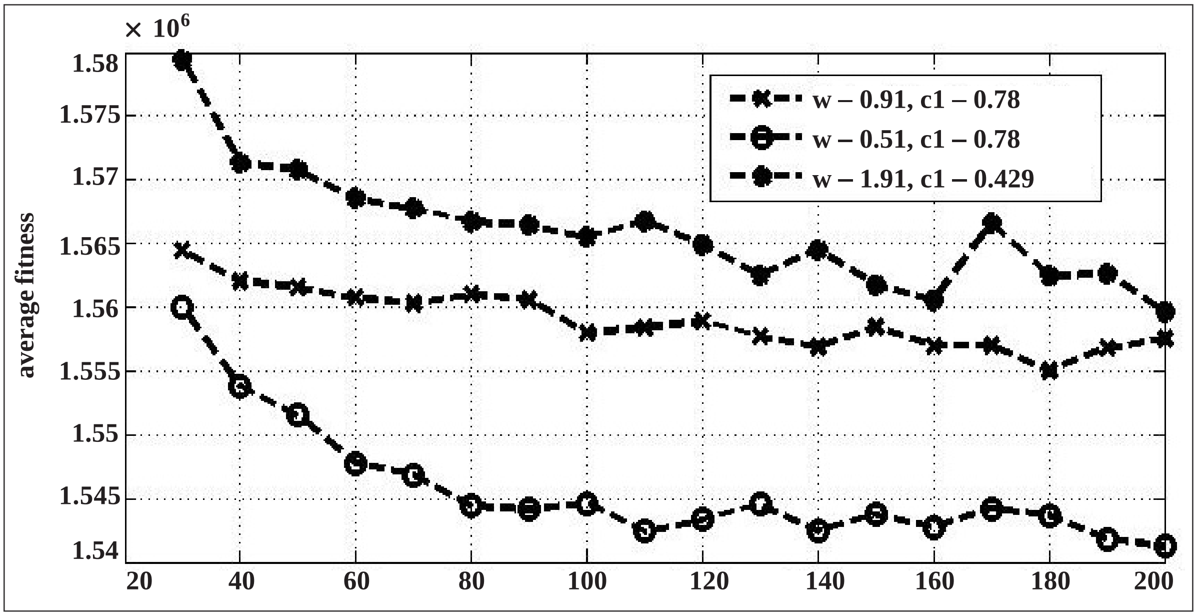

Figure 3. The Average Fitness Variation with N When w = 0.51, 1.91 and c_1 = 0.429, 0.78

that it will make optimal performance of GR-DPSO by keeping the probability of exploiting around the particle's personal best position 1 times higher than the one of exploiting around the swarm's global best position when N is constant. Keeping the probability of exploiting around the particle's personal best position higher than the one of exploiting around the swarm's global best position

3.3 Population Size

Figure.3 illustrates that the average fitness varied with N from 30 to 200 for test data selected when w = 0.51 and c_1 = 0.429, and we can see that the average fitness decreases while N is increasing. When $N > 80$, the average fitness is almost unchanged. Two other curves are also shown in Figure. 3 which represent the situations in which w = 0.51 and c_1 = 0.78, w = 1.91 and c_1 = 0.78 respectively. The average fitness declines in almost identical step when w = 0.51 and c_1 = 0.429, w = 0.51 and c_1 = 0.78. It's concluded that the optimal value interval of the population size does not vary with different acceleration constants when inertia weight is constant. When $N > 60$ (instead of 80), the average fitness is almost unchanged in case where w =1.91 and c_1 = 0.78. It's concluded that high w requires low N. Since increasing N will lead to high time overhead, 80 may be the best value of N.

4. Conclusion

In this paper, parameters selection on GR-DPSO has been studied through experiment based on real data. The results show that the best range of acceleration constants is [0.68-0.89], and high w requires low c_1, and that the best range of inertia weight is [0.4-2.0], and high c_1 requires low w, and that the optimal value interval of the population size does not vary with different acceleration constants when inertia weight is constant. With consideration of time cost, 80 may be the best value of N.

5. Acknowledgment

This paper is supported by the National Nature Science Foundation of China (60879015, 61139002) and Technology Fund of Civil Aviation (MHRD201013).

References

[1] Babic, O., Teodorovic, D., Tosic, V. (1984). Aircraft stand assignment to minimize walking, *Journal of Transportation Engineering*, 110, 55–66.

[2] Yan, S., Chang, C. -M. (1998). A network model for gate assignment, *Journal of Advanced Transportation*, 32, 176–189.

[3] Yan, S., Huo, C. (2001). Optimization of multiple objective gate assignments, Transportation Research, 35A, p. 413–32.

[4] Gu, Y., Chung, C. (1999). Genetic algorithm approach to aircraft gate reassignment problem, *ASCE Journal of Transportation Engineering*. 125, p. 384–9.

[5] Jiefeng Xu, Glenn Bailey. (2001). The airport gate assignment problem: mathematical model and a tabu search algorithm, proceeding of the 34th Hawaii International Conference on System Sciences.

[6] Ding, H., Lim, A., Rodrigues, B., Zhu, Y. (2005). The over-constrained airport gate assignment problem, Computers & OR, p.1867-1880.

[7] Ding, H., Lim, A., Rodrigues, B., Zhu, Y. (2004). Aircraft and gate scheduling optimization at airports, *In*: Proceedings of the 37th Annual Hawaii International Conference on System Sciences, IEEE. p. 74–81.

[8] Kim, S. H., Feron, E., paul Clarke, J. (2010). Airport Gate Assignment that Minimizes Passenger Flow in Terminals and Aircraft Congestion on Ramps, Aerospace, no.

[9] Dorndorf, U., Drexl, A., Nikulin, Y. and Pesch, E. (2007). Flight gate scheduling: State-of-the-art and recent developments, omega, 35, p. 326-334.

[10] Eberhart, R. C., Kennedy, J. A new optimizer using particle swarm theory, *In*: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, p. 39–43.

[11] Yuhui Shi, Russell C. Eberhart. (1998). Parameter Selection in Particle Swarm Optimization, *In*: Proceedings of the 7th International Conference on Evolutionary Programming, p. 591-600.

[12] Liping Zhang, Huanjun Yu, Shangxu Hu. (2005). Optimal choice of parameters for particle swarm optimization, *Journal of Zhejiang University Science*, 6A (6) 528-534.

[13] Kennedy, J., Eberhart, R. C. (1997). Discrete binary version of the particle swarm algorithm, *In*: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, 5, Orlando, Florida, USA, p. 4104–4108.

[14] Kusum Deep, Madhuri Arya, Jagdish Chand Bansal. (2011). A Non-deterministic Adaptive Inertia Weight in PSO, *In*: Proceedings of the 13th annual conference on Genetic and evolutionary computation, NY, USA, p.1155-1162.

[15] Chen Dong, Gaofeng Wang, Zhenyi Chen, Zuqiang Yu. (2008). A Method of Self-Adaptive Inertia Weight for PSO *In*: Proceedings of the 2008 International Conference on Computer Science and Software Engineering, 1, p. 1195-1198.

[16] Jianli Ding, Yong Zhang. A Discrete Particle Swarm Optimization Algorithm for Gate and Runway Combinatorial Optimization Problem, *Research Journal of Applied Sciences, Engineering and Technology*, in press.