Modelling and Optimizing for Departure Scheduling without Taxiing Holding

Zhang Xie, Liu Hongzhi

Key Lab of Operation Planning and Safety Technology of Air Traffic Management, Civil Aviation University of China, Tianjin 300300, China

Corresponding Author: Zhang Xie
Email: xiezhang@cauc.edu.cn

Abstract
Focusing on scheduling of aircraft departures, the departure procedure from pushing back to taking off is studied as a whole, and a scheduling model of aircraft departures without taxiing holding is established. With the introduction of the coevolution to genetic algorithm, the algorithm performance is improved by the competition and the coordination between the populations, an improved coevolution genetic algorithm is proposed to solve the aircraft departures scheduling problem, and a simulation is conducted using operational data. Compared to the First-Come-First-Service algorithm, coevolution genetic algorithm reduces the total departure time by 19.14%, and the taxiing time caused by conflict is reduced to 0. Compared to genetic algorithm, coevolution genetic algorithm has a batter result and lower costs in number of convergence generations, target value variance, running time variance and convergence generations variance, under different simulation scales. In the simulation, it is shown that the scheduling and optimization of aircraft departures without taxiing holding is achieved, the algorithm proposed has better performance in convergence and stability, and the low computation efficiency problem of conventional methods is overcome.
Keywords: AIR TRAFFIC FLOW MANAGEMENT, GENETIC ALGORITHM, DEPARTURE SCHEDULING, TAXIING WITHOUT HOLDING, COEVOLUTION

1. Introduction

The aircraft departure scheduling problem is a hot issue in the air traffic management field. Based on the statistics of America Air Transport Association, the direct loss caused by air traffic delay is about 10 billion USD in 2009, in which the loss on the departure phase is 70% [1]. In addition, about 50% fuel consumption and emissions in departure are caused by taxiing holding [2], so it has important practical significance to study the departure scheduling problem, especially departure scheduling without taxiing holding.

For the departure scheduling problem, scholars have proposed many methods such as single runway threshold sorting model of multi-flight queue [1-5], surface node taxiing model based on dot and line diagram [6], runway scheduling model based on constrained position shifting method [7] and airport departure dynamic scheduling model [8]. Some scholars establish the airport departure process model based on dynamic system theory of the discrete event [9] and departure aircraft optimization scheduling model based on max-algebra [10], and have achieved some achievements.

However, the above methods are for one single phase in departure process such as runway threshold sorting [11,12], taxiing path selection and optimization [13] and back pushing optimization [14,15] etc., and from preparation, pushing back, taxiing, entry to runway to take-off the operation processes of the aircraft are closely related. If any phase changes, it will affect whole departure process. For example, the change of the pushing back time will result in change of the occupancy time of the taxiway and runway, which will generate additional conflict wait. As such, the departure process from pushing back to departure is studied as a whole, which can facilitate better solution of the problem and improve efficiency of aircraft departure scheduling.

The aircraft departure scheduling is a typical NP-hard problem, which will take optimization of taxiing holding into consideration, meanwhile pushing back sequence, pushing back time, take-off sequence, take-off time and taxiing confliction of the aircraft should be identified. As such, the problem is very complicated. While there is a large number of aircraft, solution space will dramatically increase. It is difficult to find the optimal solution in a short time with a general algorithm.

The genetic algorithm is a method to search the optimal solution by simulating the natural evolution process, features concurrency and robustness, and is extensively applied in many fields. However, the original genetic algorithm features slow convergence speed and easy pre-maturity, and it is difficult to get the ideal effect in solving the constraint optimization problem [16]. This paper introduces the collaborative evolution idea, improves the algorithm performance via the population competition and collaboration. Furthermore, an improved co-evolutionary genetic algorithm (CoGA) is proposed to solve aircraft departure scheduling problem.

2. Problem description

This paper studies scheduling optimization of the pushing back sequence and pushing back time of the departure aircraft on a take-off runway under certain constraint conditions given departure aircraft set, preparation time of aircraft, taxiing path and optimization target (without taxiing holding and take-off safe interval). Different parts are described as follows.

2.1. Optimization goal

To improve the airport operation efficiency and minimize the total departure time, the following objective function is established:

$$\min f = \sum_{i=1}^{N} \left(t_{\mathrm{D}i} - t_{\mathrm{R}i}\right) = \sum_{i=1}^{N} \left(t_{\mathrm{P}i} + t_{\mathrm{T}i} - t_{\mathrm{R}i}\right) \quad (1)$$

N indicates the number of aircraft which wait for pushing back, taxiing and take-off at the gate. The optimized take-off time of the aircraft i is $t_{\mathrm{D}i} = t_{\mathrm{P}i} + t_{\mathrm{T}i}$. $t_{\mathrm{P}i}$ is the optimized pushing back time of the aircraft i, $t_{\mathrm{T}i}$ is the taxiing time of the aircraft i and $t_{\mathrm{R}i}$ is the preparation time of the aircraft i.

2.2. Take-off interval constraint

To ensure flight safety, any two aircraft taking off on the same runway should keep minimal interval for safe take-off. Generally it is determined by the type of two aircraft. The International Civil Aviation Organization regulates the minimal wake flow safe interval for different aircraft, shown as the table 1:

Table 1. Minimum safe interval (in seconds)

Minimal wake vortex interval		Trailing aircraft Heavy (H)	Middle (M)	Light (L)
Leading aircraft	Heavy(H)	94	157	196
	Middle(M)	60	69	131
	Light(L)	60	69	82

If the flight i takes off before the flight j, the minimal interval constraint can be represented as follows:

$$t_{\mathrm{D}j} - t_{\mathrm{D}i} \ge S_{ij}, i \ne j \quad (2)$$

2.3. Departure time constraint

Before departure, aircraft will send a “ready” signal, which corresponds to ready time $t_{\mathrm{R}i}$ of the aircraft and indicates that the aircraft are ready for pushing back, taxiing and taking off. The pushing back time of the aircraft cannot be earlier than the ready time, so we can get:

$$t_{Pi} \ge t_{Ri} \quad (3)$$

2.4. Taxiing constraint

When an aircraft is taxiing on the airport surface, three possible conflictions may occur and result in holding. To ensure the aircraft taxiing without holding, the following three conflictions should be avoided.

2.4.1. Crossing confliction

When the aircraft i and j pass the same taxiing node crosswise one after another, they should meet certain interval requirement, that is:

$$t_{\mathrm{T}j}^{k} - t_{\mathrm{T}i}^{k} \ge \Delta \quad (4)$$

$t_{\mathrm{T}i}^{k}$ indicates the time when the aircraft i taxies through the node k and Δ indicates the taxiing interval between the sequential aircraft i and j passing the same taxiing node.

2.4.2. Rear-end confliction

To avoid rear-end confliction, the exit node sequence of one taxiing section should keep consistent with the entry node sequence.

$$\left(t_{\mathrm{T}i}^{u}-t_{\mathrm{T}j}^{u}\right)\left(t_{\mathrm{T}i}^{v}-t_{\mathrm{T}j}^{v}\right)>0$$
$$u,v\in W_i,u,v\in W_j \qquad (5)$$
$$W_i(u,v)=1,\quad W_j(u,v)=1$$

W_i indicates the taxiing path of the aircraft i and $W_i(u,v)=1$ indicates that the aircraft i first passes through the node u and then pass the node v.

2.4.3. Head confliction

When two aircraft encounter each other with head to head on the same taxiing section, to avoid confliction, when the aircraft i reaches this taxiing section earlier than the aircraft j, the time when the aircraft j enters this taxiing section should not be earlier than the time when the aircraft i exits from this taxiing section, we can get:

$$\left(t_{\mathrm{T}i}^{u}-t_{\mathrm{T}j}^{v}\right)\left(t_{\mathrm{T}i}^{v}-t_{\mathrm{T}j}^{v}\right)>0$$
$$u,v\in W_i,u,v\in W_j \qquad (6)$$
$$W_i(u,v)=1,\quad W_i(v,u)=1$$

$W_i(u,v)=1,\quad W_j(v,u)=1$ indicates that the aircraft i and j pass the taxiing section (u,v) in the opposite direction.

3. Solving algorithm

3.1. Basic principle of algorithm

The scheduling of aircraft departure is a discrete and non-linear multi-constraint combined optimization problem. How to efficiently handle constraints and quickly solve it are the difficulties in the study.

The genetic algorithm features better convergence and stronger adaptation, and is extensively used to solve the large-scale combination. The original genetic algorithm is suitable for solving non-constraint optimization. For the constraint optimization problem, the methods commonly used include penalty function method [17, 18]. Generally, it is expressed as follows:

$$\Phi(x)=\sum_{i=1}^{m} r_i G_i(x)+\sum_{i=m+1}^{n} s_i H_i(x) \qquad (7)$$

$G_i(x)$ and $H_i(x)$ respectively express violation degree of the individuals to the ith inequality constraint and equality constraint in optimization. r_i and s_i respectively indicate the corresponding penalty factor. However, it is difficult to select the penalty factor. If the penalty factor is too large, it will lead to the optimization curvature become very complicated and the algorithm is easy to fall into the local optimization. If the penalty factor is too small, it is likely to converge to non-feasible domain.

In view of this, this paper proposes the co-evolutionary genetic algorithm, which contains decision solution of the aircraft departure scheduling and penalty factor population. The algorithm not only considers collaboration and competition between individuals inside two populations, but also considers collaborative influences and interaction between two populations in evolution.For example, when individuals are evaluated, the individual information of other populations should be utilized. CoGA algorithm can adaptively adjust the penalty factor via the collaborative evolution of two populations and get the optimal solution of the aircraft department scheduling.

The CoGA algorithm for aircraft departure scheduling contains two populations. One is the population for the evolutionary decision solutions. The population includes M_1 sub-populations $X_j\,(j=1,2,\ldots,M_1)$. The scale of the sub-population is M_2. The individual $x_i\,(i=1,2,\ldots,M_1\times M_2)$ indicates the decision solutions for the aircraft scheduling. Another population is the penalty factor population Y. The population scale is M_1. The individual $y_j\,(j=1,2,\ldots,M_1)$ indicates the corresponding penalty factor of the individuals in X_j.Each individual of the decision solution sub-population X_j uses the penalty factor of corresponding y_j to compute the fitness of this sub-population, and performs genetic operations. After evolution of t_1 generations, the new decision sub-population will be obtained. The penalty factor (namely individual y_j of the population Y) is evaluated according to the disadvantages and advantage information of all solutions in the decision solution sub-population X_j. The genetic operation is performed on the penalty factor population Y to get the next generation penalty factor populations. After all generations of collaborative evolutions are completed, the decision solution sub-population X_j uses the new generation penalty factor population Y to compute the fitness till the algorithm termination condition is met. Finally all historic optimal solutions are compared to get the solution of the aircraft scheduling. The corresponding optimal individuals of the population Y are the best penalty factor.

This paper introduces the collaborative evolution idea into the genetic algorithm to solve the aircraft departure scheduling model. For this purpose, the genetic algorithm will be improved, including coding scheme, fitness computing and crossover mutation rate. They will be described as follows.

3.2. Encoding scheme

The encoding scheme of the genetic algorithm is the key step in design of the genetic algorithm. The constraints of the aircraft departure scheduling are massive, which is very difficult to solve. Therefore, the encoding scheme based on take-off time is improved and a encoding scheme with constraint handling is proposed to reduce constraint complexity.

The pushing back time sequence $(t_{P1}, t_{P2}, \ldots, t_{Pn})$ of the departure aircraft set correspond to a take-off time sequence $(t_{D1}, t_{D2}, \ldots, t_{Dn})$. The take-off sequence is expressed as $(r_1, r_2, \ldots, r_n)$ and is an arrangement of $(1, 2, \ldots, n)$. The take-off time difference of adjacent aircraft is $(d_1, d_2, \ldots, d_n)$, wherein $d_i = t_{Dr_i} - t_{Dr_{i-1}}$. $C = (R, D)$ indicates chromosome, namely $(r_1, r_2, \ldots, r_n, d_1, d_2, \ldots, d_n)$.The take-off safe interval $d_i - S_{r_i r_{i-1}} \ge 0, (i = 1, 2, \ldots, n)$ should be met, so the chromosome can be expressed as $C = (R, V)$, namely $(r_1, r_2, \ldots, r_n, v_1, v_2, \ldots, v_n)$, wherein $v_i = d_i - S_{r_i r_{i-1}}$, $V = (v_1, v_2, \ldots, v_n)$. Given:

$$\begin{cases} v_1 = t_{Dr_1}, (i = 1) \\ v_i = t_{Dr_i} - \max\left(t_{Dr_j} + S_{r_j r_i}\right), (j = 1, 2, \ldots i-1, i > 1) \end{cases} \tag{8}$$

The equation (8) can be represented as:

$$\begin{cases} t_{Dr_1} = v_1, (i = 1) \\ t_{Dr_i} = v_i + \max\left(t_{Dr_j} + S_{r_j r_i}\right), (j = 1, 2, \ldots i-1, i > 1) \end{cases} \tag{9}$$

With this encoding strategy with constraint handling, the safe interval constraint can be encoded, so it can successfully reduce the constraint complexity in the aircraft departure scheduling.

3.3. Individual fitness function

The routine individual fitness evaluation method completely depends on the objective function. As such the algorithm diversity dramatically decreases and the algorithm is easy to pre-mature. Therefore, the new individual evaluation method is designed and the fitness function of the individual x_i in the decision population is:

$$F(x_i) = \frac{1}{f(x_i) - q(x_i)\omega_1 - n(x_i)\omega_2} \tag{10}$$

$f_i(x)$ indicates the objective function in the aircraft scheduling optimization, $q(x_i)$ is the constraint violation degree of the individual x_i, $n(x_i)$ is the number of the constraints violated by individuals, ω_1 and ω_2 are the penalty factor of the individual y_j in the population Y.

3.4. Fitness function of penalty factor

N_{Xj} indicates total number of all feasible solutions in X_j and the fitness sum is $\sum_{i=1}^{N_{Xj}} F(x_i)$. The fitness function of the individual y_j in the penalty factor population is:

$$P(y_j) = \frac{1}{N_{Xj} + \sum_{i=1}^{N_{Xj}} F(x_i) + \varepsilon} \tag{11}$$

As shown in the equation (11), the more the number of feasible solutions are or the larger the fitness sum is, the smaller $P(y_j)$ is, then it will facilitate the algorithm to evolve to the area with more feasible solutions and better target.

3.5. Crossover mutation

The crossover and mutation operation with fixed ratio are likely to result in prematurity. As such, an adaptive method is proposed to adjust the crossover rate and mutation rate.

$$P_c = \frac{1}{e^{-k_1\lambda} + 1}, P_m = \frac{e^{-k_2\lambda}}{e^{-k_2\lambda} + 1}, k_1 > 0, k_2 > 0 \tag{12}$$

$\lambda = F_{\max} - \overline{F}$ indicates the prematurity degree, $F_{\max}$ indicates the maximum fitness of the individuals in populations, $\overline{F}$ indicates the mean of the individuals' fitness which is larger than the mean of all the individuals' fitness in the population.

3.6. Flow of CoGA algorithm

Two populations use the same GA operations. The specific algorithm flowchart is shown as the figure 1.

Step 1: Initialize the decision solution population X, $t_1 = 0$ and penalty factor population Y, $t_2 = 0$;

Step 2: Evaluate fitness of X population;

Step 3: Evaluate fitness of Y population;

Step 4: Perform GA operations on Y population;

Step 5: If the extremum of Y population is reached, then update the Y population and skip to the step 6. Otherwise, $t_2 = t_2 + 1$ and skip to the step 3.

Step 6: Perform GA operations with the individuals in corresponding penalty factor population as the penalty factor and update the decision solution population for decision solution sub-populations.

Step 7: If the algorithm termination conditions are met, then output the optimal solution. Otherwise, $t_1 = t_1 + 1$, $t_2 = 0$ and skip to the step 2.

4. Simulation and analysis

Guangzhou Baiyun International Airport is taken as example. 20 departure aircraft within a period of a day are selected as samples for simulation. The simulation sample data is shown in the table 2. The aircraft taxi at constant speed on the airport surface. As symmetric structure and double-runway independent operation mode at Guangzhou Baiyun International Airport, only the aircraft taking off from 02L runway in the west bridges, are simulated. It does not affect validity check of the algorithm. CoGA algorithm is compared with original GA algorithm and first come

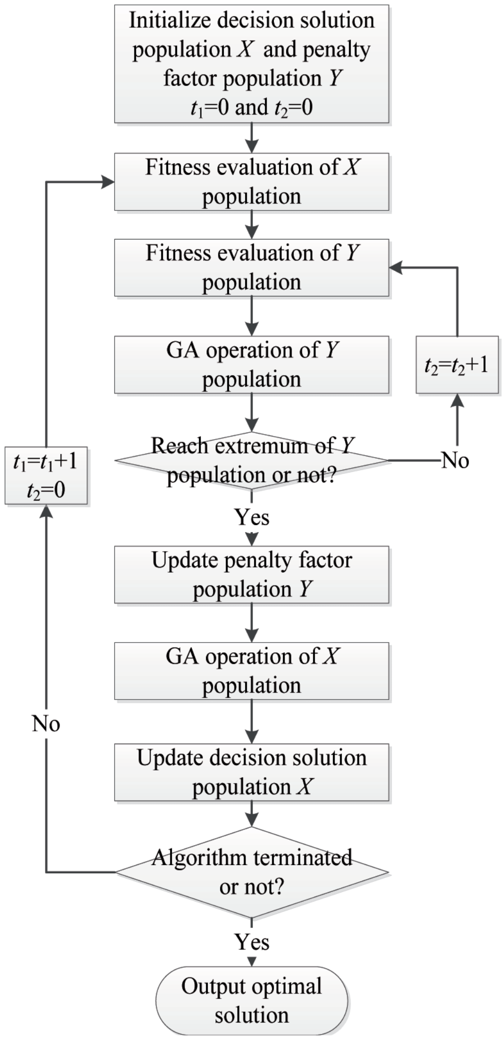

Figure 1. CoGA algorithm framework

first service algorithm. The parameters of the CoGA are shown as follows: the scale of the penalty factor population $M_1 = 20$, the scale of the decision solution sub-population $M_2 = 50$, the iterations of the decision population $T_1 = 100$, the iterations of the penalty factor population $T_2 = 100$, the initial crossover rate of the population $P_c = 0.7$ and the initial mutation rate $P_m = 0.15$.

4.1. Algorithm effectiveness

From the results in the table 3, the simulation algorithm has better effect. Compare to the first come first service algorithm, the total departure time of GA and CoGA reduce by 16.67% and 19.14%. The CoGA algorithm is better. Taxiing holding time reduces from 69min of FCFS to 0 due to no taxiing holding, which validates effectiveness and feasibility. In addition, although taxiing holding time increases, the total holding time significantly reduces and the gate position holding can effectively reduce the fuel consumption and nitrogen oxide emissions. The air traffic control authorities and airlines are happy to see these results.

4.2. Algorithm efficiency

To compare and analyse the algorithm efficiency under different scales, the sample data of 10, 20 and 30 aircraft are simulated. A queue with 100 departure aircraft is generated for each scale randomly and scheduling is optimized. The optimization results are summarized and the mean is computed for comparison. The executive results and efficiency are compared as the table 4.

Table 2. Original sample data

Aircraft	Type	Gate	Take-off runway	Ready time
F1	M	220	02L	07:00:00
F2	M	215	02L	07:00:00
F3	M	202	02L	07:01:00
F4	M	223	02L	07:02:00
F5	H	217	02L	07:05:00
F6	H	221	02L	07:06:00
F7	M	206	02L	07:06:00
F8	M	204	02L	07:09:00
F9	H	210	02L	07:10:00
F10	M	218	02L	07:10:00
F11	M	207	02L	07:13:00
F12	M	216	02L	07:15:00
F13	L	403	02L	07:20:00
F14	H	222	02L	07:20:00
F15	M	203	02L	07:20:00
F16	L	401	02L	07:24:00
F17	H	201	02L	07:29:00
F18	M	205	02L	07:33:00
F19	M	214	02L	07:34:00
F20	M	208	02L	07:40:00

Table 3. Comparison of simulation results of FCFS, GA and CoGA algorithms

Algorithm	Total departure time/min	Performance improvement/%	Taxiing holding /min	Gate holding / min	Running time /s	number of convergence generations
FCFS	162	-	69	0	-	-
GA	135	16.67	0	42	15.825	56
CoGA	131	19.14	0	38	7.073	33

Table 4. Optimal results of GA and CoGA under different scales

Aircraft number	Algorithm	Total departure time	Target solution variance/min	running time/s	running time variance/s	number of convergence generations	convergence generations variance
10	GA	66.32	8.43	5.466	1.320	13.52	7.79
	CoGA	63.02	5.22	2.362	0.573	8.67	4.95
20	GA	137.45	11.21	16.167	3.882	53.86	16.32
	CoGA	136.38	8.90	7.981	1.216	32.13	9.45
30	GA	205.72	23.11	129.327	8.633	77.36	18.98
	CoGA	197.26	15.75	27.886	2.901	48.68	11.32

From the simulation results in the table 4, the optimization effects of GA and CoGA are ideal. The convergence generation number of the CoGA algorithm is significantly lower than that of GA algorithm. As such, CoGA has better convergence performance. The objective solution variance, running time variance and convergence generations variance of the CoGA algorithm are lower than that of GA. As such, CoGA algorithm is more stable than GA algorithm. In addition, for 30 aircraft, the average time of GA algorithm is over 2 minutes and the average time of CoGA algorithm is only 27.886s, so it meets the real-time scheduling requirements completely.

5. Conclusions

This paper constructs the aircraft departure scheduling model without taxiing holding and applies the improved collaborative evolutionary algorithm for solution. The simulation results indicate that the model proposed can effectively reduce the departure holding time. The improved CoGA can get better results, features shorter computing time and better stability, can meet the actual application requirements, and provides a new means for solution of the aircraft departure scheduling. The runway taxiway is the intersection location of the taking off and landing aircraft. Considering the influence of the landing aircraft on the departure aircraft scheduling, how to improve the operation efficiency of the runway taxiway will be further studied in future.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. U1333108) and the Fundamental Research Funds for the Central Universities (Grant No. 3122014C020).

References

1. Sölveling G, Solak S, Clarke J P B, et al. Scheduling of runway operations for reduced environmental impact[J]. Transportation Research Part D: Transport and Environment, 2011, 16(2): 110-120.
2. Simaiakis I, Balakrishnan H. Impact of congestion on taxi times, fuel burn, and emissions at major airports[J]. Transportation Research Record: Journal of the Transportation Research Board. 2010, 2184(1): 22-30.
3. Rathinam S, Wood Z, Sridhar B, et al. A generalized dynamic programming approach for a departure scheduling problem[C]//AIAA Guidance, Navigation, and Control Conference. 2009: 10-13.
4. Wang Lai-jun, Xiao Run-mou, Hu Da-wei, et al. A new theoretic model and its application for solving open facility departure sequencing problem[J]. Journal of Northwest University (Natural Science Edition), 2011, 41(3): 423-426. (in Chinese)
5. Cao Song, Sun Fu-chun, Hu Lai-hong, et al. Departure aircraft sequence optimization using EDA[J]. Journal of Tsinghua University (Science & Technology), 2012, 52(1): 66-71. (in Chinese)
6. Zhang Ying, Hu Ming-hua, Wang Yan-jun. Optimization model for aircraft taxiing time at airport[J]. Journal of Civil Aviation Flight University of China, 2006, 17(5): 3-6. (in Chinese)
7. Balakrishnan H, Chandran BG. Algorithms for scheduling runway operations under constrained position shifting[J]. Operations research. 2010, 58(6): 1650-1665.
8. Caprì S, Ignaccolo M. Genetic algorithms for solving the aircraft-sequencing problem: the introduction of departures into the dynamic model[J]. Journal of Air Transport Management. 2004, 10(5): 345-351.
9. Hua Ke-qiang, Cai Yu-feng, Li Xiao-bai. Simulation studies on departure process in airports[J]. Journal of Civil Aviation University of China, 2004, 22(6): 15-17. (in Chinese)
10. Liu Chang-you, Zhao Yong-jun. Analyzing Flights Departure Process Based on Max-Algebra[J]. Journal of Civil Aviation University of China, 2009, 27(01): 31-34. (in Chinese)

11. Wang Lai-jun, Shi Zhong-ke. The Implement of Genetic Algorithms in Departure Sequencing Problem[J]. Systems Engineering-theory & Practice, 2005, 24(09): 119-125. (in Chinese)
12. Zhang Qi-qian, Hu Ming-hua, Shi Sai-feng, et al. Optimization algorithm of flight take-off and landing on multi-runways[J]. Journal of Traffic and Transportation Engineering, 2012, 12(06): 63-68. (in Chinese)
13. Clare GL, Richards AG. Optimization of taxiway routing and runway scheduling[J]. IEEE Transactions on Intelligent Transportation Systems, 2011, 12(4): 1000-1013.
14. Solveling G, Solak S, Clarke J-P et al. Runway operations optimization in the presence of uncertainties[J]. Journal of Guidance, Control, and Dynamics. 2011, 34(5): 1373-1382.
15. Li Zhen, Zhang Jun, Zhang Xue-jun. Modeling and Simulation of Departure Scheduling Problem Based on Genetic Algorithm[J]. Tansportation and Computer, 2008, 26(06): 39-42. (in Chinese)
16. Mu Caihong. Co-evolutionary numerical optimization algorithms and their applications[D]. Xi'an: Xidian University, 2010. (in Chinese)
17. Liu J, Zhong W, Jiao L. An organizational evolutionary algorithm for numerical optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2007, 37(4): 1052-1064.
18. Runarsson TP, Yao X. Stochastic ranking for constrained evolutionary optimization[J]. IEEE Transactions on Evolutionary Computation, 2000, 4(3): 284-294.

Improvement of Low Complexity Sparseness Controlled MPNLMS Algorithm Based on Sparse Impulse Response

Huanchang Qin, Zaiqun Wu

Department of Physics and Communication, Baise University,
Baise, Guangxi, 533000, China

Corresponding author is Zaiqun Wu

Abstract
In the work, we obtained probability distribution of error signals with different confidence levels by Hampe censored estimation function. The data collection of "input-expectation" was divided into four data space where the ratios of data disturbed by impulse noise are different. The data space with small data ratio was calculated by SM-SCMPNLMS Algorithm. In data space with large data ratio, the computation complexity was reduced to solve steady-state problems because of pulse noise disturbance by restraining error signal amplitude and using larger error threshold. The effectiveness of this algorithm was proved by simulation.
Keywords: DATA SUBSPACE, SPARSENESS CONTROL, SET-MEMBERSHIP FILTERING, ECHO CANCELLATION