SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2016, Vol. 59 072201:1–072201:17
doi: 10.1007/s11432-015-5495-3

# A large-scale flight multi-objective assignment approach based on multi-island parallel evolution algorithm with cooperative coevolutionary

Renli LÜ[1], Xiangmin GUAN[1*], Xueyuan LI[2] & Inseok HWANG[3]

[1]*Department of General Aviation, Civil Aviation Management Institute of China, Beijing* 100102*, China;*
[2]*School of Electronic and Information Engineering, Beihang University, Beijing* 100191*, China;*
[3]*School of Aeronautics and Astronautics, Purdue University, West Lafayette, IN* 47907-2023*, USA*

Received May 23, 2015; accepted September 30, 2015; published online May 26, 2016

**Abstract** Due to the rapid increase of air traffic demand, the large-scale flight assignment plays a crucial role in reducing airspace congestion and economic losses via reasonably regulating the air traffic flow of China. In this paper, the large-scale flight assignment problem is formulated as a multi-objective model with consideration of the reduction of airspace congestion and flight delay. However, it is a large-scale combinatorial optimization problem with complex constraints and tightly coupled decision variables, which is difficult to deal with. Hence, an effective multi-objective optimization algorithm is proposed based on the multi-island parallel evolution framework (PEA) with a left-right probability migration topology. Multi-island PEA employs multiple evolution populations for solving the problem simultaneously, and the left-right probability migration topology for exchange individuals among populations to improve the efficiency of the cooperation of populations. Then the cooperative co-evolution (CC) algorithm is introduced for each population to further improve the searching capability. Simulation results using the real traffic data from the China air route network and daily flight plans demonstrate that the proposed approach can improve the solution quality effectively, showing superiority to the existing approaches such as the multi-objective genetic algorithm, the well-known multi-objective evolutionary algorithm based on decomposition, a CC-based multi-objective algorithm as well as other two parallel evolution algorithms with different migration topologies.

**Keywords** air traffic flow management, flight assignment, multi-island parallel evolution algorithm, migration topology, cooperative co-evolution

**Citation** Lü R L, Guan X M, Li X Y, et al. A large-scale flight multi-objective assignment approach based on multi-island parallel evolution algorithm with cooperative coevolutionary. Sci China Inf Sci, 2016, 59(7): 072201, doi: 10.1007/s11432-015-5495-3

## 1 Introduction

The past decade has witnessed the rapid increase of flight operations in China. However, the contradiction between the increasing traffic in air transportation and the capacity-limited airspace has resulted in more and more serious airspace congestion. Moreover, heavy congestion has challenged the airspace safety and cost the airline industry billions of dollars every year [1].

* Corresponding author (email: guanxiangmin@camic.cn)

 

Airspace capacity is related to the number of aircraft operating in the given airspace during the given period of time. The airspace congestion occurs when the traffic demands exceed the available capacity in airspace [2, 3]. The large-scale flight assignment methods via adjusting involved flights in time (slot allocation) or in space (route allocation) are methods to a alleviate congestion in the enroute airspace and terminal area of airport [4]. For instance, when a piece of airspace is predicted to undergo congestion, the involved flights are delayed at the departure airports or their routes are changed in order to avoid the congestion. Accordingly, with the aim to safely accommodate high levels of demand and maximize the use of capacity-limited airspace to keep safety and stability of the aviation transportation, the flight assignment problem has become a major concern to both researchers and practitioners in air traffic management (ATM) [5–8].

However, the flight assignment problem is a large-scale combinatorial optimization problem which is difficult to solve [9]. The flight assignment problem involves thousands of flight plans over a huge airspace. For instance, there are more than ten thousand flights flying over China every day on the air route network with more than one thousand waypoints. Hence, in order to get optimal depar-ture time and optimal routes for all aircraft, the flight assignment problem involves large-scale tightly coupled decision variables and constraints which make the problem high computationally complexity and extremely difficult to deal with. Besides, the problem usually has multiple objectives with consideration of reducing congestion and minimizing the induced delay. The objective functions of the problem are all non-differentiable or even non-continuous [9]. Therefore, many conventional optimization techniques that are capable of handling large scale problems, such as Newton/Quasi-Newton method and conjugate gradient method, are not suitable for the flight assignment problem [9].

Due to the importance of the flight assignment problem, it is attracting more and more attention. Abad and Clarke [10] proposed a routes assignment method for flights based on mixed integer linear programming to reduce airspace congestion. Besides, many works have focused on the slottime allocation, known as ground holding in a single or multiple airports. Vossen and Michael [11] introduced a slot trade optimization model of collaborative ground delay program to optimizing the departure time for flights. Indeed, both the time and space dimensions are essential when reconstructing flight plans for a large number of flights. Bertsimas et al. [12] presented an integer optimization approach to large-scale air traffic flow management with consideration of both the time and routes assignment. Delahaye and Odoni [9] have considered this problem from a stochastic optimization point of view. They proposed a genetic algorithm to resolve the problem. However, it often falls into a local optimum because of the limited searching capability of the traditional genetic algorithm. More recently, the Cooperative Co-evolution (CC) algorithm was introduced to solve the flight assignment problem in a simplified network [13]. It adopted the divide-and-conquer strategy to divide the complex problem into several low dimensional subproblems which become easier to deal with.

These works took the minimization of the airspace congestion or the flight delay as the only objective. However, in real operations, it might be more appropriate to consider both airspace congestion and extra flight cost, and try to seek a good trade-off between them. Delahaye et al. [14] formulated the flight assignment problem as a bi-objective optimization problem and introduced the multi-objective genetic algorithm (MOGA). Besides, the real traffic data in France was used to test the method. Recently, with consideration of more objectives including optimizing the congestion in each sector and the overloaded time of sectors, Tian et al. [15] used the MOGA to solve the flight assignment problem in China. However, because of the high computational complexity of the problem and the limited capability of the classical MOGA, the optimization is more likely to fall into the local optima [14].

Parallel evolutionary algorithm (PEA) is a popular heuristic algorithm, widely used to solve complex combinational optimization problems [16–18]. PEA is more prominent because of its distributed and flexible features, and it also has great potential for substantial improvement in search performance [19, 20]. In the last decade, many PEAs have emerged, which can be categorized into [20, 21]: global single-population master-slave PEAs, single-population fine-grained PEAs, multi-population coarse-grained PEAs. A master-slave PEA employs a single panmictic population, but the evaluation of fitness is distributed among several processors. A fine-grained PEA consists of one spatially-structured population with selection and mating restricted to a small neighborhood, but neighborhoods overlap permitting some interactions among all the individuals [20,21]. Multi-population EAs, also called island model PEA, are more sophisticated, as they adopt several populations which exchange individuals occasionally. This exchange of individuals is called migration. The migration topology determining the destination of the migrants is a key feature of the island model, and it affects the quality of the solutions and the efficiency of the algorithm. Currently, there are two common migration topologies, which are the one-way ring topology and the random topology [20]. Multi-island PEAs have been applied to solve many complex problems including the large scale quadratic assignment problem, the graph partitioning problem, and the synthesis of VLSI circuits [21].

In this paper, with the consideration of minimizing the airspace congestion and total flight delay, we formulate the large-scale flight assignment problem as a bi-objective optimization problem, and propose a multi-island PEA framework with a left-right probability migration topology to solve it. Firstly, multiple evolution populations are constructed to solve the bi-objective optimization problem simultaneously, which can effectively increase the chance to find better solutions. Secondly, in order to improve the quality of the solutions and the efficiency of the cooperation among populations, a left-right probability migration topology is proposed to exchange individuals. Thirdly, for each population, the cooperative co-evolution (CC) algorithm is introduced to further improve the searching capability by dividing the complex problem into several low dimensional sub-problems. Finally, each sub-problem employs the multi-objective evolutionary algorithm based on decomposition (MOEA/D). Simulation results using the real traffic data from the China air route network and daily flight plans demonstrate that the proposed approach can improve the solution quality effectively and efficiently, showing superiority to the existing approaches such as the MOGA, MOEA/D, CC-based multi-objective algorithm as well as other two PEA algorithms with different migration topologies.

This paper is organized as follows. The model is described in Section 2. The framework of the parallel evolution algorithm is presented in Section 3. Section 4 summarizes experimental results with real air traffic data of the national route of China. Finally, conclusion is presented in Section 5.

## 2 Problem formulation

The problem considered in this paper is presented through the illustration of airspace structure in Figure 1. The rectangular airspace is divided into several sectors with dotted lines, and each sector is managed by air traffic controllers on the ground. It is supposed there is a flight from airport A in the left bottom to airport B in the top right corner. In real operations, the aircraft will fly along the waypoints (circle points) between airports A and B, because a waypoint is a navigation marker and the aircraft need the information, such as the desired track and heading direction, which can be provided by the ground navaids. Hence, there are three paths passing through different sectors from A to B in Figure 1. The flight can select its satisfactory path to avoid congestion. For example, if congestion in sector 6 (S6) becomes severe, the flight can choose path 3 which doesnot pass through S6.

### 2.1 Decision variables

Flight assignment methods have been developed in a way to reduce congestion by adjusting involved flights in time and space [12]. Those techniques try to find an optimal route, or an optimal time of departure, or both, for each individual flight [14]. Supposed that there are $n$ flights ($F_1$, $F_2$, $\ldots$, $F_n$) with different flight plans. For each flight, there is a pair of decision variable($\delta_i$,$r_i$), where $\delta_i$ represents time difference (advance or delay) from the original departure time slot and $r_i$ is a new route. Besides, $\delta_i$ should not deviate from the original departure time too much from the operational perspective and $r_i$ should choose shorter paths to reduce cost. So $\Delta$ and $R$ denoting the slots set and the routes set are described as follows:

$$\Delta = -\delta_m, -\delta_m + 1, \ldots, -1, 0, 1, \ldots, \delta_p - 1, \delta_p, \tag{1}$$

$$R = r_0, r_1, r_2, \ldots, r_{\max}, \tag{2}$$

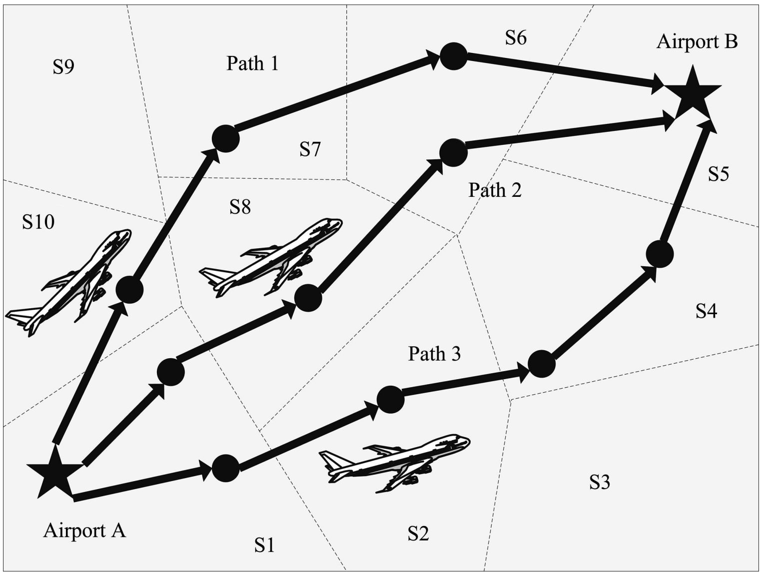


**Figure 1** The structure of the airspace.

where $\delta_m$ is the maximum advance time, $\delta_p$ is the maximum delay permitted for a flight, $r_0$ is the best route and $r_{\max}$ is the worst one.

### 2.2 Objective functions

Safety and efficiency are the two main objectives of air traffic management. Many flight assignment approaches have been developed to reduce the congestion and the total delay cost which can be divided into two categories. The first category in general formulates the minimization of the total delay cost as the single objective with the critical monitoring workload capacity of sectors as constraints. In this case, integer programming models have been proposed to solve this problem [10–12]. In the second category, the reduction of the congestion and the total delay cost are both considered as two objectives to provide the controllers more feasible solutions, and multi-objective optimization methods are applied [13–15].

In this paper, with consideration of safety and efficiency, we formulate the flight assignment problem as a bi-objective problem and try to reduce the congestion and the total delay simultaneously. Different with the total delay objective proposed in [14] without consideration of the aircraft fleet mix, we introduce the delay cost coefficients for different categories of aircraft to make the model more realistic. In this section, the objective functions will be pre-sented in details.

(1) Congestion

The first objective is to reduce congestion and workload of over-loaded sectors by balancing flights among all sectors. For example, if a sector is crowded, flights in this sector will be assigned to other neighboring sectors with the aim to reduce the workload into a safe level.

As mentioned before, workload $W^t_{S_k}$ in a sector $S_k$ at time t includes two parts, the monitoring workload $W^t_{\mathrm{mo}S_k}$ related to the number of aircraft in the sector and the coordination workload $W^t_{\mathrm{co}S_k}$ related to the number of aircraft passing through the boundary of this sector at time $t$. Due to the motivation to reduce congestion of over-loaded sectors as the first objective, in this paper, both $W^t_{\mathrm{mo}S_k}$ and $W^t_{\mathrm{co}S_k}$ are formulated to compute the workload beyond the critical monitoring workload capacity of sectors, and they are assumed to be zero as long as the critical monitoring workload capacity is not reached.

Therefore, it can be expressed by [14]

$$W^t_{S_k} = W^t_{\mathrm{mo}S_k} + W^t_{\mathrm{co}S_k}, \tag{3}$$

where $W^t_{\mathrm{mo}S_k}$ can be defined as

$$W^t_{\mathrm{mo}S_k} = \begin{cases} 1 + W^t_{S_k} - C^t_{\mathrm{m}S_k}, & \text{if} \quad W^t_{S_k} > C^t_{\mathrm{m}S_k}, \\ 0, & \text{else}, \end{cases} \tag{4}$$

where $W^t_{S_k}$ is the number of aircraft in sector $S_k$ at time $t$ , and $C^t_{\mathrm{m}S_k}$ is the critical monitoring workload capacity of the sector.

Similarly, $W^t_{\mathrm{co}S_k}$ in Eq. (3) can be defined by

$$W^t_{\mathrm{co}S_k} = \begin{cases} 1 + C^t_{S_k} - C^t_{\mathrm{c}S_k}, & \text{if} \quad C^t_{S_k} > C^t_{\mathrm{c}S_k}, \\ 0, & \text{else}, \end{cases} \tag{5}$$

where $C^t_{\mathrm{c}S_k}$ is the critical coordination workload capacity of sector $k$ at time $t$, and $C^t_{\mathrm{c}S_k}$ is the number of aircraft passing through the boundary of sector $S_k$ at time $t$.

With consideration of reducing congestion in the most overloaded sectors, the first objective function can be defined as follows [14]:

$$\min f_1 = \sum_{k=1}^{k=P} \left( \left( \sum_{t\in T} W^t_{S_k} \right)^{\phi} \times \left( \max_{t\in T} W^t_{S_k} \right)^{\varphi} \right), \tag{6}$$

where $P$ is the number of sectors, $T$ is the time period we considered, and $\phi$ and $\varphi$ are weight factors. $\sum_{t\in T} W^t_{S_k}$ and $\max_{t\in T} W^t_{S_k}$ indicate the total workload and the maximum workload of sector $k$ over the whole time period respectively. We can find that the objective is to minimize the total workload of all sectors. Besides, the more congested sector $k$ is, the higher probability sector $k$ will have to reduce workload.

(2) Total delay

The second objective is to reduce the total delay of all flights. The total delay is the extra flight time caused by congestion compared with the original flight plan. It consists of the delay on the ground and the delay in the air. For flight $i$, the ground delay can be expressed as $\delta_s(i) = |t_n - t_k|$, where $t_k$ is the planned departure time slot and $t_n$ is the actual departure time slot. On the other hand, the cost of air delay is modeled as three times of the ground delay [14], so the air delay can be presented as $\delta_r(i) = 3 \times |T_r - T_0|$, where $T_r$ is the actual flying time and $T_0$ is the shortest flying time. In general, aircraft can be categorized to the light aircraft, the medium aircraft, and the heavy aircraft. The larger an aircraft is, the more cost the delay will cause. Hence, with the consideration of the aircraft fleet mix, the second objective function for all flights is formulated by

$$\min f_2 = \sum_{i=1}^{N} (\lambda_i(\delta_s(i) + \delta_r(i)))^2, \tag{7}$$

where $\lambda_i$ is the cost coefficient for different categories of aircraft. The second objective function indicates the total delay cost of all flights including the ground delay cost and the air delay cost. In addition, it can be found that the flight with a larger cost coefficient will have a higher probability to reduce its total delay.

The search space of the flight assignment problem will be huge, and its computational complexity can be obtained by

$$|\mathrm{space}| = \prod_{i=1}^{N} (|R_i| \cdot |\Delta_i|), \tag{8}$$

where $|S|$ denotes the cardinality of the set $S$, $\Delta_i$ and $R_i$ indicates the slots set and the routes set of flight $i$, and $N$ is the number of flights. Besides, the objective functions are nonlinear and not continuous. The decision variables and constraints are tightly coupled.

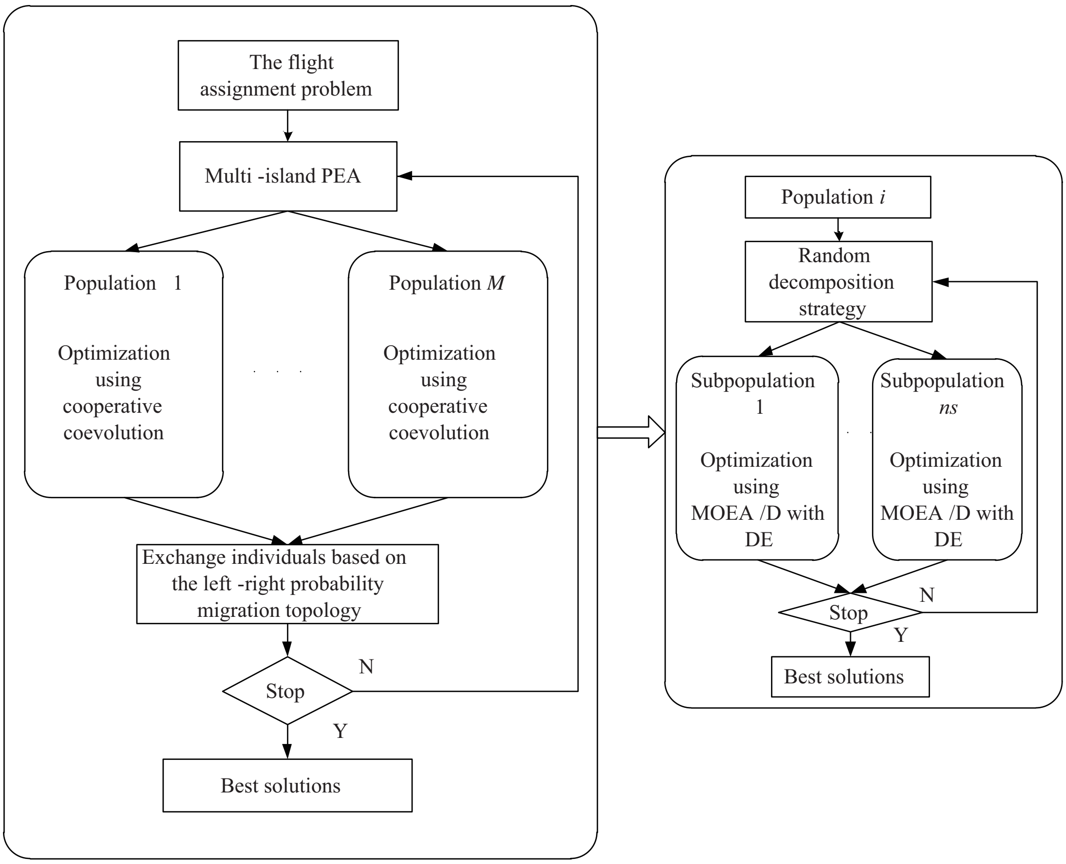


**Figure 2** The framework of the proposed algorithm.

## 3 Optimization framework

To avoid premature convergence and get high quality solutions to this complex problem, we propose a multi-island parallel evolution algorithm (PEA) framework with a left-right probability migration topology. Multiple evolution populations are employed to increase the chance to find better solutions. Then, in order to improve the efficiency of the cooperation among populations, a specially designed left-right probability migration topology is proposed to exchange individuals. In addition, the cooperative co-evolution (CC) algorithm is introduced to optimize each population by dividing the complex problem into several low dimensional sub-problems. In each sub-problem, the MOEA/D with differential evolution (DE) is employed. The framework is presented in Figure 2.

### 3.1 Multi-island parallel evolution algorithm

The classical evolutionary algorithms (EAs) could not solve the large-scale and complex problem efficiently, and the computational cost rises rapidly with the increase of problem size. As a consequence, there have been multiple efforts to make EAs more effective and faster [17, 20, 21]. One of the most promising methods is to use parallel technologies. As parallel computers become more common-place in scientific computing, it becomes more feasible to harness their power for use with EAs [17]. Multi-island PEA consists of several populations, optimizing simultaneously, to avoid premature convergence and explore better solutions. The exchange of individuals occurs occasionally which can guide the optimization via cooperation among evolution populations. Multi-island PEAs can get better results when dealing with complex combinational optimization problems due to its distributed feature. They have been applied successfully to find acceptable solutions to problems in different engineering domains, such as the quadratic assignment problem, the graph partitioning problem, and the synthesis of VLSI circuits [17,21].

Suppose that there are $M$ islands and $N$ flights. Then each population can be denoted as

$$\mathrm{pop}_i = \{\mathrm{id}_{i1}, \mathrm{id}_{i2}, \ldots, \mathrm{id}_{i\mathrm{ps}}\}, \quad 1 \leqslant i \leqslant M, \tag{9}$$

where $\mathrm{id}_{ij}$ is defined by

$$\mathrm{id}_{ij} = \{r_{ij1}, \delta_{ij1}, r_{ij2}, \delta_{ij2}, \ldots, r_{ijN}, \delta_{ijN}\}, \quad 1 \leqslant i \leqslant M, \tag{10}$$

where ps is the size of population, and $\delta_{ijk}$ $(1 \leqslant k \leqslant N)$ are the time difference and the path of the flight $k$ of the chromosome $j$ in population $i$.

### 3.2 The left-right probability migration topology

The migration topology is a key feature of the island model which determines the destination of the migrants, and it could greatly affect the quality of the solutions and the efficiency of algorithms. For instance, if two populations were connected with each other via exchanging good individuals, it could help the two populations improve solution quality quickly, but it will decrease the diversity of populations, and finally results in a local optimum. If two populations rarely communicate with each other, it is difficult for the best solution to spread which may stop populations finding better solutions.

Currently, the main migration topologies are the one-way ring topology and the random topology [17]. In the one-way ring topology, populations are numbered, and the worst individual of a population is replaced by the best individual of the next population. The random topology delivers the migrants to a randomly selected population. It is found that the one-way ring topology is better than the random topology. However, after many times of migration under the ring topology, the population diversity will decrease obviously.

In this paper, populations are numbered from 0 to $M - 1$, and we propose a left-right probability topology which is defined by

$$SL(m) = \begin{cases} (m + M - 1)\%M, & \text{rand} < r, \\ (m + M + 1)\%M, & \text{rand} \geqslant r, \end{cases} \tag{11}$$

where rand is a random number between 0 and 1, $r$ is a real number between 0 and 1, $m$ is an integer and $0 \leqslant m < M$. Different from the one-way ring topology, the left-right probability topology makes the best individual of a population migrate into its left or right population with different probability.

The structure of the left-right probability topology for five populations is shown in Figure 3. In the left-right probability topology, for instance, the best individual of population 1 may be delivered to population 2 or 5 with different probabilities.

Under the left-right probability topology, migration occurs locally between the adjacent populations which can improve the solution quality, and the separate populations are relatively independent so that they can evolve towards different directions, which can further avoid premature convergence.

### 3.3 Cooperative co-evolution for each population

The flight assignment problem is difficult to solve, because it involves large-scale tightly coupled decision variables, and the objective functions are nonlinear and not continuous as presented in the previous sections. Though the multi-island PEA uses several populations simultaneously, which can obtain a better solution, in fact each population is hard to avoid falling into a local optimum. Hence, we introduce the cooperative coevolution algorithm for the optimization of each population to further improve the solution quality. The cooperative co-evolution (CC) algorithm, adopting the divide-and-conquer strategy, divides the complex problem into several low dimensional sub-problems [22–24]. Then, each sub-problem is solved by an evolutionary algorithm. At last, the optimal solutions are obtained through cooperation between different sub-components. There are three critical issues in this approach.

(1) Decomposition strategy. The decomposition strategy is a key feature of the cooperative co-evolution framework which can greatly affect the capability and the efficiency of algorithms. In this

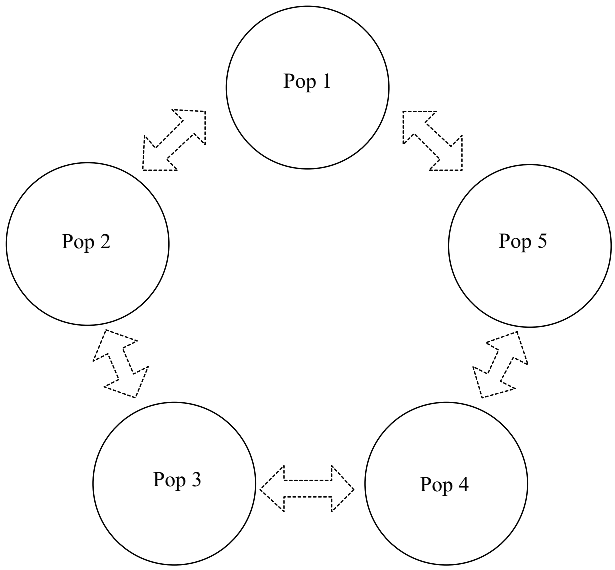


**Figure 3** The structure of the left-right probability topology.

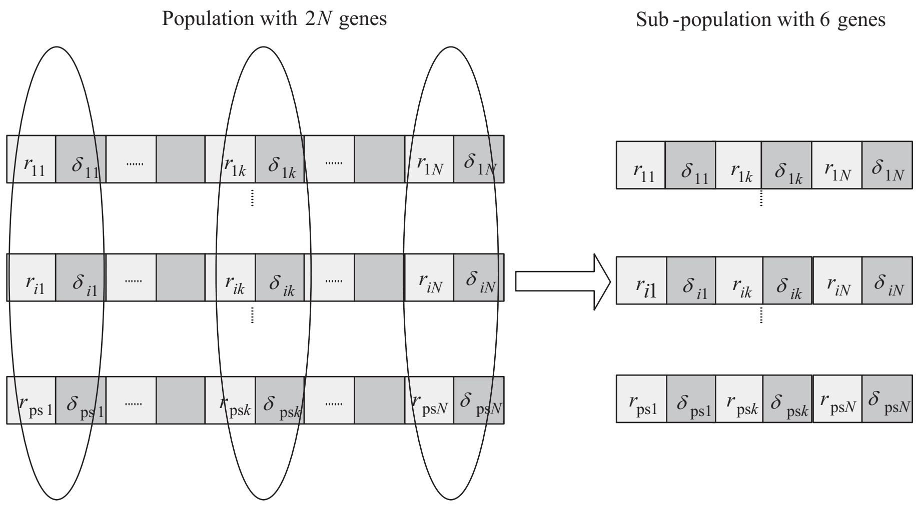


**Figure 4** Generation of sub-population.

work, the random grouping strategy is used which has been both theoretically and experimentally proved to be effective for the large-scale complex problem [22]. At each generation, each population is randomly divided into ns sub-populations with the same population size,

$$\mathrm{pop}_i = \{\mathrm{sp}_i^1, \ldots, \mathrm{sp}_i^{\mathrm{ns}}\}, \quad 1 \leqslant i \leqslant M, \tag{12}$$

$$\mathrm{sp}_i^j = \{\mathrm{sid}_i^{j1}, \mathrm{sid}_i^{j2}, \ldots, \mathrm{sid}_i^{j\mathrm{ps}}, \}, \quad 1 \leqslant i \leqslant \mathrm{ns}, \tag{13}$$

$$\mathrm{sid}_1^{jk} = \{r'_{jk1}, \delta'_{jk1}, r'_{jk2}, \delta'_{jk2}, \ldots, r'_{jk(N/\mathrm{ns})}, \delta'_{jk(N/\mathrm{ns})}\}, \quad 1 \leqslant j \leqslant \mathrm{ps}, \tag{14}$$

where sid denotes the individual of each sub-population, and $\mathrm{sp}_i^j$ indicates the sub-population $j$ of population $i$. Figure 4 shows an example of the generation of a sub-population.

(2) Sub-population optimization. Another critical point is optimization for each sub-population. In this paper, the well-known multi-objective evolutionary algorithm based on decomposition (MOEA/D) is employed by each sub-population [25]. It decomposes a multi-objective optimization problem into a set of scalar optimization sub-problems with neighborhood relations [26] which are defined by the distances between their aggregation coefficient vectors. In this case, the fitness evaluation is the same as single

objective optimization. The diversity can be maintained by the diverse search directions determined by the uniformly distributed weight vectors.

A general framework of MOEA/D is proposed in [25]. The first step is to convert the approximation of the Pareto front (PF) into $n$ scalar optimization sub-problems by de-composition. In this paper, the Tchebycheff approach is used [27]. The scalar function is defined by

$$\min g(x|\lambda, z^*) = \max_{1\leqslant i\leqslant m, x\in\Omega}\{\lambda_i|f_i(x) - z_i^*|\}, \tag{15}$$

where $\lambda = (\lambda_i, \ldots, \lambda_m)$ is a weight vector, and $\sum_{i=1}^{m}\lambda_i = 1$. $f_i(x)$ is the $i$th objective, and $m$ is the number of objectives. $\Omega$ is the solution space, and $z^* = (z_1^*, \ldots, z_m^*)$ is the reference point, where $z^*$ is the best value of the objective function $i$. If $n$ is large enough, and $\lambda^i, \ldots, \lambda^n$ is properly selected, the optimal solutions to those scalar functions will provide a good approximation to the Pareto set (PS) and PF. The major components in MOEA/D are its neighborhood concept and its population replacement mechanism [25, 26].

Besides, DE [28, 29] is used in the MOEA/D framework to generate new solutions, because it is a simple yet effective algorithm for global optimization. DE is a randomized parallel searching algorithm. It begins with a random population, according to specific rules, for example selection, crossover and mutation. An optimized resolution is reached by retaining good individuals and discarding bad individuals. Compared with other optimization algorithms, DE has the advantages in global optimization as well as easy operation. Then, the details of the MOEA/D with DE are described in Algorithm 1. A general framework of MOEA/D can be found in [26].

(3) Cooperation of sub-populations. The array archive[$i$] ($i \leqslant i \leqslant M$) is defined to store non-dominated solutions of population $i$, and if elements of archive[$i$] can be dominated by new solutions found in sub-population optimization, they will be replaced.

### 3.4 Computational complexity analysis

The computational complexity of Step 2 of MOEA/D in Algorithm 1 is $O(\mathrm{ps}(T+m))$ [25]. In this paper, there are $M$ populations with ps/$M$ individuals in each population. And at each generation, there are $M$ migrations among the populations. Hence, the computational complexity of the proposed algorithm is about $O(\mathrm{ps}/M(T+m)+M)$. Besides, in general $M$ is much smaller than ps/$M$, so the proposed PEA has lower computational complexity than MOEA/D at each generation.

## 4 Experimental studies

To evaluate the efficacy of the proposed multi-island PEA framework and the migration topology, two experiments have been carried out. First, the proposed multiisland PEA framework was evaluated with the real air traffic data of China, and was compared with three existing flight assignment methods. Then, the performance of the leftright probability migration topology was analyzed and compared with two common migration topologies.

The national route network of China is shown in Figure 5. It consists of 1706 airway segments, 940 waypoints and 150 airports. Note that, the takeoff and landing parts of flights are truncated within a given radius (usually 10 NM) around airports since enroute airspace operation is considered in this paper. The traffic around airports is managed with specific procedures by the terminal control area (TMA) control services in these zones. The airspace is divided into many sectors, and Figure 6 shows the sectored airspace in China. The flights are classed as light, medium and heavy with different speeds of 700 km/h, 800 km/h and 900 km/h, and with different delay coefficients of 0.8, 1 and 1.2.

The parameters are set as follows: the number of populations $M$ = 5, $\phi$ =0.9, $\varphi$ =0.1, $r$ = 0.3 and ns = 10. The mutation probability and the crossover probability of DE are 0.15 and 0.85.

The algorithms, such as our proposed method, MOGA, MOEA/D [25] and cooperative co-evolution based algorithm, in this work were implemented in C++, and the simulations were performed on a server

**Algorithm 1** Algorithmic flow of MOEA/D with DE

**Input:**

(1) A stopping criterion.

(2) $n$: the number of the sub-problems.

(3) An uniform spread of n weight vectors: $\lambda^i, \ldots, \lambda^n$.

(4) $T$: the number of the weight vectors in the neighborhood of each weight vector.

**Output:** Approximation to the PF and PS.

**Procedure:**

**Step 1 Initialization:**

**Step 1.1** Compute the Euclidean distances between the weight vectors and work out the $T$ closest weight vectors to each weight vector. For each $i = 1, \ldots, n$, set $\{i_1, \ldots, i_T\}$, where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the $T$ closest weight vectors to $\lambda_i$ .

**Step 1.2** Generate an initial population $x^i, \ldots, x^n$. Calculate the fitness values of the population.

**Step 1.3** Initialize $z = (z_1, \ldots, z_m)$, where $z_j = \min\limits_{1 \leqslant j \leqslant n} f_j(x^i)$.

**Step 2 Update:**

For$i = 1, \ldots, N$,

**Step 2.1 Selection of the mating pool:**

Generate a random number which is uniformly distributed in [0, 1]. Set

$$P = \begin{cases} B(i), & \text{if} \quad \text{rand} < \delta, \\ 1, \ldots, n, & \text{otherwise.} \end{cases}$$

**Step 2.2 Reproduction:**

Set $r_1 = i$, and randomly select two indexes $k$, $l$ from $P$, and then generate a new solution $y$ using mutation and crossover operators of DE.

**Step 2.3 Repair:**

If an element of $y$ is out of the bound of $\Omega$, its value is reset to be a randomly selected value inside the boundary.

**Step 2.4 Update of the reference point:**

For each $j = 1, \ldots, m$ , if $z_j > f_j(y)$ , then set $z_j = f_j(y)$.

**Step 2.5 Replacement of solutions**

**Stopping criterion:**

If the optimization generation gen = 200, then stop the algorithm and output PF and PS. Otherwise, go to **Step 2**.

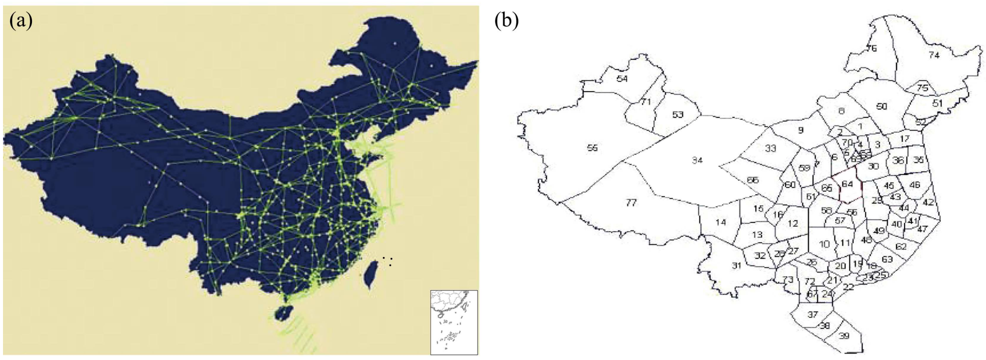


**Figure 5** (Color online) The structure of the national route network of China (a) and the sectored airspace (b).

**Table 1** Parameters of the experiments

| Parameters | Description | MOGA | MOEA/D | CCMA | PEA |
|---|---|---|---|---|---|
| ps | Population size | 100 | 100 | 100 | 20·(IM=5) |
| maxgen | Max generation | 150 | 150 | 150 | 150 |
| $p_{\mathrm{c}}$ | Crossover probability | 0.9 | 0.9 | 0.9 | 0.9 |
| $p_{\mathrm{m}}$ | Mutate probability | 0.1 | 0.11 | – | – |

with an E5620 2.4 GHz CPU with 12 GB RAM. For each algorithm, the results were collected and analyzed on the basis of 15 independent runs. Besides, the proposed approach was realized by multithreaded programming. Then, the optimization of all islands and all sub-components of each population can proceed separately and simultaneously which can reduce the computation time.

The parameters used in all experiments are listed in Table 1.

### 4.1 Comparison with other algorithms

The experiment aims to evaluate the efficacy of the proposed multi-island PEA by comparing it with some existing algorithms. Specifically, the classical multi-objective genetic algorithm (MOGA) and multi-objective evolutionary algorithm based on decomposition (MOEA/D) [25] were chosen for comparison because they are two most well-known MOEAs in the recent years. In addition, a CC-based multi-objective algorithm (CCMA) was also selected for comparison which is designed to randomly and equally divide the decision variables into several groups and uses an EA to optimize each group.

In order to better describe the difference between the algorithms' performance, we consider two scenarios: 960 flights (the busiest one hour), and 1664 flights (the busiest three hours). Results were obtained based on 15 independent runs and analyzed statistically.

It is well acknowledged in the area of evolutionary multi-objective optimization that a single performance metric is insufficient for comparisons between MOEAs. Hence, to better evaluate the performance of the solutions obtained by these algorithms, three typical metrics are used: the distance from the reference set ($I_{\mathrm{D}}$) [28], the spread ($\Delta$) [29], and the Hypervolume ($I_{\mathrm{H}}$) [30]. ID suggests the average Euclidean distance from the non-dominated solution set to the actual Pareto front. $\Delta$ indicates the diversity of solutions along the Pareto front. $I_{\mathrm{H}}$ can evaluate the convergence and the extent of spread of the solutions simultaneously without the real Pareto front. Note that it is difficult to find the actual Pareto front for most realworld optimization problems, in our study, the nondominated solutions obtained by all the algorithms in 15 runs were combined, and those solutions remained non-dominated in this set were used as the reference set for $I_{\mathrm{D}}$. Accordingly, the leftmost (in the objective space) and rightmost solutions obtained over all the 15 runs of the four algorithms were used to calculate $\Delta$.

Tables 2 and 3 show the average value of $I_{\mathrm{H}}$, ID and $\Delta$ over the 15 independent runs of the algorithms when the number of flights is 996 and 1664 respectively. For each performance metric, the Wilcoxon rank sum test [31] has been carried out to compare the proposed algorithm with the other three methods. In each row of the table, the best value is highlighted in boldface. It is seen from the tables that PEA achieves the best performance and is significantly better than all the three algorithms in terms of $I_{\mathrm{H}}$ and $I_{\mathrm{D}}$. MOEA/D gets the best result on $\Delta$ and is much better than PEA on this metric.

Additionally, the non-dominated solutions with the least delay time cost (DTC) and the ones with the least airspace congestion (AC) obtained by the four algorithms over 15 runs are listed in Tables 4 and 5 where the number of flights is 996 and 1664, respectively. It can be observed that both the non-dominated solution with the least DTC and the non-dominated solution with the least AC obtained by PEA dominate the corresponding solutions of the other three algorithms.

Figure 6 provides a clearer demonstration of the performance of the four compared algorithms. In Figure 6, the non-dominated solutions obtained by MOGA, MOEA/D, CCMA and PEA in all 15 runs are plotted, i.e. all the solutions obtained throughout the experiments were put together, and then the non-dominated solutions were extracted. Although it can be seen that the solutions obtained by MOEA/D and CCMA indeed spread nicely in the objective space, they are dominated by the solutions obtained by PEA. From Figure 6, it can be found that PEA outperforms other three algorithms in terms of both the total delay and the airspace congestion. Besides, MOGA has the worst performance, and CCMA performs better than MOEA/D.

From the experimental results, we can conclude that PEA performs better than the other three methods for the two scenarios. While the flight assignment problem has a large searching space which also increases exponentially over the number of flights, MOGA has difficulty in finding feasible solutions in this problem. Though MOEA/D can get the most feasible solutions, it can easily fall into a local optimum. CCMA divides the complex problem into several low dimensional sub-problems which become easier to solve, and thus it performs better than MOEA/D. However, the variables and the constraints are too tightly coupled to find better solutions. The proposed PEA adopts a multi-island PEA framework which can improve the optimization capability. Besides, the left-right probability migration topology can improve the efficiency of the cooperation among populations and avoid premature convergence. Therefore, PEA is more efficient in finding good solutions than others.

**Table 2** Comparison of different algorithms for 960 flights ($I_{\rm H}$, $I_{\rm D}$, $\Delta$)

| Algorithm | $I_{\rm H}$ | $I_{\rm D}$ | $\Delta$ |
|---|---|---|---|
| MOGA | 1.0776e+13 | 6.3544e+06 | 1.2446 |
| MOEA/D | 1.8664e+13 | 1.7614e+06 | **0.9035** |
| CCMA | 3.1167e+13 | 5.6585e+05 | 0.9647 |
| PEA | **3.2326e+13** | **7.3409e+04** | 0.9956 |

**Table 3** Comparison of different algorithms for 1664 flights ($I_{\rm H}$, $I_{\rm D}$, $\Delta$)

| Algorithm | $I_{\rm H}$ | $I_{\rm D}$ | $\Delta$ |
|---|---|---|---|
| MOGA | 1.3690e+12 | 2.1414e+07 | 1.0412 |
| MOEA/D | 7.2003e+13 | 1.1160e+07 | **0.9413** |
| CCMA | 1.9501e+14 | 3.3108e+06 | 1.0115 |
| PEA | **2.2314e+14** | **3.9088e+05** | 1.0123 |

**Table 4** Non-dominated solutions with the least AC and the least DTC for 960 flights

| Compared algorithms | Solutions with least AC | | Solutions with least DTC | |
|---|---|---|---|---|
| | AC | DTC | DTC | DTC |
| MOGA | 2.0521e+006 | 1.0251e+007 | 2.0841e+006 | 1.0034e+007 |
| MOEA/D | 1.8887e+006 | 1.0961e+007 | 3.0276e+006 | 3.3000e+006 |
| CCMA | 1.3003e+006 | 5.6795e+006 | 1.6825e+006 | 2.8065e+006 |
| PEA | **1.2836e+006** | **4.2880e+006** | **1.5632e+006** | **2.4430e+006** |

**Table 5** Non-dominated solutions with the least AC and the least DTC for 1664 flights

| Compared algorithms | Solutions with least AC | | Solutions with least DTC | |
|---|---|---|---|---|
| | AC | DTC | DTC | DTC |
| MOGA | 1.9434e+007 | 2.6591e+007 | 1.9677e+007 | 2.5973e+007 |
| MOEA/D | 1.5847e+007 | 2.7569e+007 | 2.1769e+007 | 1.1667e+007 |
| CCMA | 1.1315e+007 | 1.2667e+007 | 1.3277e+007 | 7.5794e+006 |
| PEA | **1.1181e+007** | **7.3979e+006** | **1.2250e+007** | **5.4164e+006** |

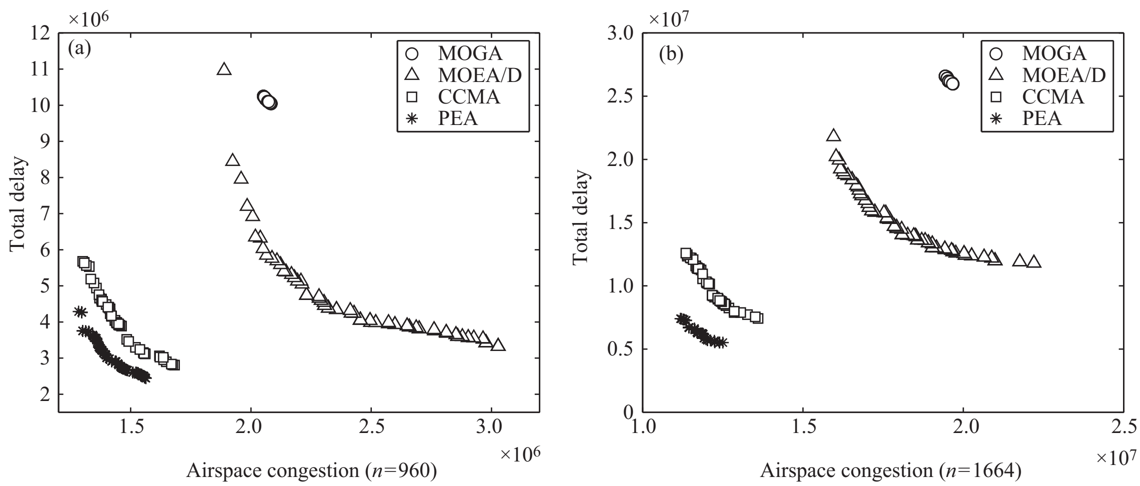


**Figure 6** Comparison of different algorithms for 960 flights (a) and 1664 flights (b).

## 4.2 Comparing our migration topology with others

In the previous section, the first set of experiments has justified the superiority of PEA to over existing methods. The next experiment is designed to further investigate whether the proposed left-right probability migration contributes to the success of PEA.

Tables 6 and 7 show the results obtained with the four methods in terms of the values of the metrics

**Table 6** Comparison of different algorithms for 960 flights ($I_{\mathrm{H}}$, $I_{\mathrm{D}}$, $\Delta$)

| Algorithm | $I_{\mathrm{H}}$ | $I_{\mathrm{D}}$ | $\Delta$ |
|---|---|---|---|
| PEA-one way | 3.1532e+13 | 1.2421e+05 | 0.9746 |
| PEA-random | 3.1454e+13 | 1.3532e+05 | **0.9632** |
| PEA-left right | **3.2326e+13** | **7.3409e+04** | 0.9956 |

**Table 7** Comparison of different algorithms for 1664 flights ($I_{\mathrm{H}}$, $I_{\mathrm{D}}$, $\Delta$)

| Algorithm | $I_{\mathrm{H}}$ | $I_{\mathrm{D}}$ | $\Delta$ |
|---|---|---|---|
| PEA-one way | 2.0211e+14 | 1.1160e+06 | 1.0436 |
| PEA-random | 2.0143e+14 | 3.9857e+06 | 1.0442 |
| PEA-left right | **2.2314e+14** | **3.9088e+05** | **1.0123** |

**Table 8** Non-dominated solutions with the least AC and the least DTC for 1664 flights

| Compared algorithms | Solutions with least AC | | Solutions with least DTC | |
|---|---|---|---|---|
| | AC | DTC | DTC | DTC |
| PEA-one way | 1.1257e+007 | 0.2355e+007 | 1.2351e+007 | 6.2452e+006 |
| PEA-random | 1.1303e+007 | 0.4532e+007 | 1.3142e+007 | 6.4326e+006 |
| PEA-left right | **1.1181e+007** | **7.3979e+006** | **1.2250e+007** | **5.4164e+006** |

**Table 9** Non-dominated solutions with the least AC and the least DTC for 960 flights

| Compared algorithms | Solutions with least AC | | Solutions with least DTC | |
|---|---|---|---|---|
| | AC | DTC | DTC | DTC |
| PEA-one way | 1.2986e+006 | 4.4573e+006 | 1.6023e+006 | 2.5425e+006 |
| PEA-random | 1.3001e+006 | 4.5732e+006 | 1.6211e+006 | 2.5753e+006 |
| PEA-left right | **1.2836e+006** | **4.2880e+006** | **1.5632e+006** | **2.4430e+006** |

over 15 independent runs of the algorithms when the number of flights is 996 and 1664, respectively. It is seen from the tables that PEA almost always outperforms the other three algorithms in terms of $I_{\mathrm{H}}$, $I_{\mathrm{D}}$ and $\Delta$. The random topology achieves the best result on $\Delta$ and is much better than PEA on this metric.

Also, Tables 8 and 9 show the non-dominated solutions with the least delay time cost (DTC) and the ones with the least airspace congestion (AC) obtained by the four compared algorithms over 15 runs when the number of flights is 996 and 1664. It can be seen that the non-dominated solution with the least DTC and the non-dominated solution with the least AC obtained by PEA dominate the correspoding solutions of the other three algorithms.

Furthermore, like the first experiment, the non-dominated solutions of the algorithms are shown in Figure 7. It shows that the left-right probability topology performs much better than the others and its non-dominated solutions can dominate the solutions obtained by the others. Moreover, it can be seen that as the number of fights grows the superiority of PEA is more obvious. For at the scenario of 1664 flights, PEA has the most non-dominated solutions and spreads nicely in the objective space. Besides, the one-way ring topology and the random topology almost have the same performance in the flight assignment problem.

The migration topology determines the destination of the migrants. Too sparse or too intensive connection between populations may prevent populations evolving towards the right direction. The one-way ring topology can spread good solutions through the neighbor population which can improve the solution quality of all populations in the initial phase. However, it can also decrease the diversity of populations which results in a local optimum. On the other hand, the random topology causes insufficient cooperation among populations which can decrease the optimization capability. The left-right probability topology can ensure enough local communication between neighbor populations, and also it can avoid decreasing the diversity of all populations. Hence, this is the likely reason why the left-right probability topology performs best.

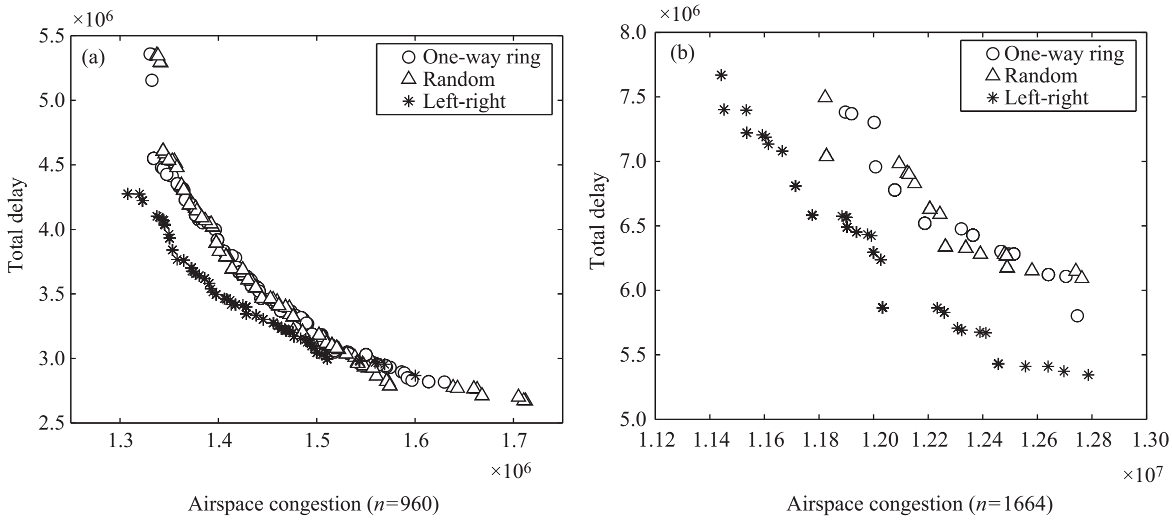


**Figure 7** Comparison of different migration strategies for 960 flights (a) and 1664 flights (b).

### 4.3 Application to real operations

The last two experiments have demonstrated that the proposed approach can get better solutions compared with the existing methods. Next we further check whether the proposed approach can be applied to real operations and used to provide feasible solutions for the air traffic controllers to update flow regulations.

In general, congestion can be predicted based on the evaluation of flight plans in several hours even in a few days. The flight assignment method can be used as a strategic approach. The computations time needed to get all non-dominated solutions for 960 and 1664 flights are about 20 min and 31 min respectively. About 50 non-dominated solutions in scenario 1 and 30 non-dominated solutions in scenario 2 are obtained. However, in real operation, controllers may need several feasible solutions. In that case, the computation time can be reduced. Besides, using more advanced parallel computation technology, the computation time can be further reduced. Hence, in view of computation time, the proposed approach can be applied to real operations.

Then, the sum of the workload beyond the safety critical value of each sector under different methods is given in Figure 8. We can find that the proposed method outperforms the other methods.

In addition, we further analyze the reason why the proposed approach can effectively reduce congestion and total extra workload of sectors. Figure 9 shows the number of flights in 7 busiest sectors under the original flight plans of scenario 2 and the number of flights in these sectors under the flight plans optimized by the proposed approach. We can find that under the original flight plans sectors 2 and 5 are very congested with more than 200 flights, but sectors 4 and 7 both have some underused resources, assuming the safe workload of all sectors are 150 flights. On the contrary, the proposed flight assignment approach can effectively relieve congestion via balancing the number of flights among sectors.

In conclusion, we can find that the proposed PEA can avoid local optimal and improve the optimization capability. Besides, it is more effective to deal with the large-scale flight assignment problem, and provide feasible solutions for strategy management and pre-tactical strategy. However, due to the computation time, the algorithm cannot be used for tactical management for air traffic controllers to provide real time solutions.

## 5 Conclusion and future work

In this paper, we have proposed a multi-island parallel evolutionary algorithm (PEA) framework with a left-right probability migration topology to solve the flight assignment problem which is a large-scale

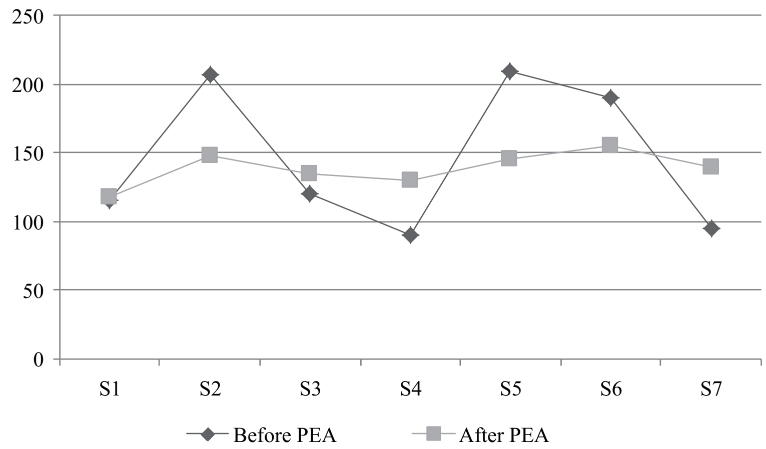


**Figure 8** Algorithmic flow of MOEA/D with DE.

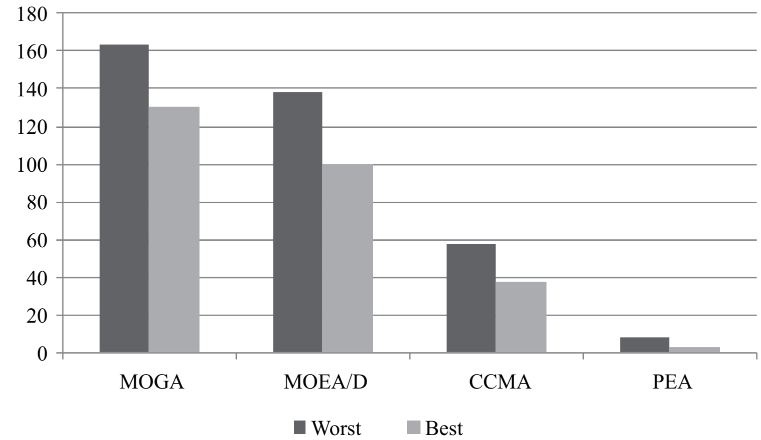


**Figure 9** Algorithmic flow of MOEA/D with DE.

combinatorial optimization problem. Firstly, multiple evolution populations are introduced to optimize simultaneously, which can increase the searching capability. Secondly, in order to improve the efficiency of the cooperation among populations and avoid premature convergence, a left-right probability migration topology is proposed. Thirdly, for each population, a cooperative co-evolution (CC) algorithm is introduced to further improve the searching capability via dividing the complex problem into several low dimensional sub-problems. Finally, the multi-objective differential evolution based on decomposition is employed to solve each sub-problem. Simulation results using the real air traffic data from the China air route network and daily flight plans have demonstrated that the proposed approach can improve the solution quality effectively, showing superiority to the existing approaches such as the multi-objective genetic algorithm (MOGA), the multi-objective evolutionary algorithm based on decomposition (MOEA/D), a CC-based multi-objective algorithm as well as other two PEA algorithms with different migration topologies.

Compared with these algorithms, the proposed optimization approach can also reduce congestion and total delay more efficiently, which are two important goals the air traffic management has been pursuing. In addition, it not only provides insights for research on the flight assignment problem, but also for research on evolutionary algorithms. On one hand, the design of migration topology is a key problem in the multi-island PEA, and the proposed left-right probability migration topology experimentally has shown that it is better than the two existing migration topologies. On the other hand, we can find that it is more effective to deal with the large-scale, combinatorial optimization, multi-objective problem via introducing the CC algorithm to divide the problem into several low dimensional sub-problems and employing MOEA/D in each sub-problem.

For future research, the flight assignment problem with the influence of severe weather will be considered.

**Acknowledgements** This work was supported by National Natural Science Foundation of China (Grant No. U1433203), and Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant No. 61221061).

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

1 Liu W, Hwang I. Probabilistic trajectory prediction and conflict detection for air traffic control. AIAA J Guid Control Dynam, 2011, 34: 1779–1789
2 Alam S, Lokan C, Abbass H A. What can make an airspace un-safe? characterizing collision risk using multi-objective optimi-zation. In: Proceedings of IEEE Congress on Evolutionary Computation, Brisbane, 2012. 1–8
3 Tang M D, Zhang G Q, Sun Yi, et al. Integrating local and partial network view for routing on scale-free networks. Sci China Inf Sci, 2013, 56: 102311
4 Oussedik S, Delahaye D. Reduction of air traffic congestion by genetic algorithms. In: Parallel Problem Solving from Nature — PPSN V. Berlin: Springer, 1998. 855–864
5 Wei P, Cao Y, Sun D. Total unimodularity and decomposition method for large-scale air traffic cell transmission model. Transport Res Part B, 2013, 53: 1–16
6 Ball M, Lulli G. Ground delay programs: optimizing over in-cluded flight set based on distance. Air Traffic Control Quarter, 2004, 12: 1–25
7 Bayen A, Raffard R, Tomlin C. Adjoint-based control of a new Eulerian network model of air traffic flow. IEEE Trans Control Syst Tech, 2006, 14: 804–818
8 Bertsimas D, Patterson S. The air traffic flow management problem with enroute capacities. Oper Res, 1998, 46: 406–422
9 Delahaye D, Odoni A. Airspace congestion smoothing by sto-chastic optimization. In: Evolutionary Programming VI. Berlin: Springer, 1997. 163–176
10 Abad A M, Clarke J B. Using tactical flight level allocation to alleviate airspace corridor congestion. In: Proceedings of AIAA 4th Aviation Technology, Integration and Operations (ATIO) Forum, Chicago, 2004. 1–9
11 Vossen T, Michael B. Optimization and mediated bartering models for ground delay programs. Nav Res Log, 2006, 53: 75–90
12 Bertsimas D, Lulli G, Odoni A. An integer optimization approach to large-scale air traffic flow management. Oper Res, 2011, 59: 211–227
13 Liu H X, Zhu Y B, Cai K Q, et al. Route network flow assignment in the new generation of aviation by cooperative coevolution. In: Proceedings of IEEE 5th International Conference on Cybernetics and Intelligent Systems (CIS), Qingdao, 2011. 175–180
14 Delahaye D, Sofiane O, Puechmorel S. Airspace congestion smoothing by multi-objective genetic algorithm. In: Proceedings of the ACM Symposium on Applied Computing, Santa Fe, 2005. 907–912
15 Tian W, Hu M. Study of air traffic flow management optimization model and algorithm based on multi-objective programming. In: Proceedings of International Conference on Computer Modeling and Simulation, Sanya, 2010. 210–214
16 Tang J, Lim M H, Ong Y S. Diversity-adaptive parallel memetic algorithm for solving large scale combinatorial optimization problems. Soft Comput J, 2007, 11: 873–888
17 Tang J, Lim M H, Ong Y S, et al. Study of migration topology in island model parallel hybrid-GA for large scale quadratic as-signment problems. In: Proceedings of the 8th International Conference on Control, Automation, Robotics and Vision, Kunming, 2004. 3: 2286–2291
18 Zhang X J, Guan X M, Hwang I, et al. A hybrid distributed-centralized conflict resolution approach for multi-aircraft based on cooperative co-evolutionary. Sci China Inf Sci, 2013, 56: 128202
19 Chen Z, Zhang S, Yang L, et al. Optimal phase searching of PTS using modified genetic algorithm for PAPR reduction in OFDM systems. Sci China Inf Sci, 2014, 57: 062305
20 Tang J, Lim M H, Ong Y S. Parallel memetic algorithm with selective local search for large Scale quadratic assignment. J Innov Comput Inf Control, 2006, 2: 1399–1416
21 Cantu-Paz E. A survey of parallel genetic algorithms. Calculateurs Parall Reseaux et Syst Repartis, 1998, 10: 141–171
22 Yang Z, Tang K, Yao X. Large scale evolutionary optimization using cooperative coevolution. Inf Sci, 2008, 178: 2985–2999
23 Potter M, Jong K D. A cooperative coevolutionary approach to function optimization. In: Proceedings of the 3rd Conference on Parallel Problem Solving From Nature, Jerusalem, 1994. 249–257
24 Zhu F M, Guan S U. Cooperative co-evolution of GA-based classifiers based on input decomposition. Eng Appl Artif Intell, 2008, 21: 887–896
25 Zhang Q, Li H. MOEA/D: a multi-objective evolutionary algorithm based on decomposition. IEEE Trans Evol Comput, 2007, 11: 712–731

26 Liu B, Fernandez F V, Zhang Q, et al. An enhanced MOEA/D-DE and its application to multiobjective analog cell sizing. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC), Barcelona, 2010. 1–7
27 Miettinen K. Nonlinear Multiobjective Optimization. New York: Springer, 1999
28 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evolut Comput, 2002, 6: 182–197
29 Zitzler E, Thiele L, Laumanns M. Performance assessment of multiobjective optimizers: an analysis and review. IEEE Trans Evolut Comput, 2003, 7: 117–132
30 Fleischer M. The measure of Pareto optima: applications to multiobjective metaheuristics. In: Proceeding of the 2nd International Conference on Evolutionary Multi-criterion Optimization. Berlin: Springer, 2003. 519–533
31 Wilcoxon F. Individual comparisons by ranking methods. Bio-metrics Bull, 1945, 1: 80–83