# A New Meta-Heuristic Approach for Aircraft Landing Problem

*ZHANG Junfeng**, *ZHAO Pengli*, *YANG Chunwei*, *HU Rong*

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P. R. China

(Received 10 March 2020; revised 30 March 2020; accepted 10 April 2020)

**Abstract:** A new meta-heuristic approach is proposed in this paper based on a new composite dispatching rule to tackle the aircraft landing problem (ALP). First, the ALP is modeled as a machine scheduling problem with the objective of minimizing the total penalty, i. e., total weighted earliness plus total weighted tardiness. Second, a composite dispatching rule, minimized penalty with due dates and set-ups (MPDS), is presented to determine the landing sequence. Then, an efficient heuristic approach is proposed to solve the problem by integrating the MPDS rule and CPLEX solver. In the first stage, the landing sequence is established based on the proposed MPDS rule. In the second stage, landing time is optimized using CPLEX solver. Next, a new meta-heuristic strategy is introduced into the heuristic approach by conducting the local search from the potential landing sequences, which are generated by the proposed MPDS rule. Finally, the performance of the proposed approach is evaluated using a set of benchmark instances taken from the OR library. The results demonstrate the effectiveness and efficiency of the proposed approaches.

**Key words:** arrival scheduling; air traffic control; decision support; meta-heuristic; local search

**CLC number:** V355 **Document code:** A **Article ID:** 1005-1120(2020)02-0197-12

## 0 Introduction

The rapid growth of air traffic has led to a mismatch between traffic demand and scarce supply resources. Demand-supply mismatch results in airport congestion problems with substantial flight delays, excessive fuel consumption, and consequent air pollutant emissions. Countermeasures could be supply-based, such as adding runways to provide more capacity, or demand-based, such as demand management to control air traffic, or combined operational management to improve the efficiency of the system given the same demand and supply. The pure supply-side solution is capital-intensive and time-consuming. Demand management, ranging from legislative instruments to market-based measures, sacrifices the accessibility of some communities. They are not the focus of this study; instead, we focus on operational management of improving airspace system efficiency. In particular, we propose a new meta-heuristic approach to schedule arrival aircraft efficiently.

The problem of arrival scheduling (aircraft landing problem, ALP) has attracted considerable attention[1-4]. To tackle the ALP, one should seek to determine the sequence and time of aircraft landing on available runways by optimizing given objectives while subject to a variety of operational constraints. Previous research generally focused on one of the following objectives: (1) Minimizing the total penalty[1-10], i. e., total weighted earliness plus total weighted tardiness; (2) minimizing the total delay[11-13]; and (3) minimizing the completion time of the last aircraft (or maximizing runway throughput)[14-15]. Concerning the solution algorithms to solve the ALP, CPLEX can be used to solve small-scale ALP. As ALP is an NP-hard problem, the computation time to find an exact solution grows exponentially with the increase of the number of aircraft. Therefore, dynamic programming (DP)[9,16]

*Corresponding author, E-mail address: zhangjunfeng@nuaa.edu.cn.

**How to cite this article**: ZHANG Junfeng, ZHAO Pengli, YANG Chunwei, et al. A new meta-heuristic approach for aircraft landing problem[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2020, 37(2): 197-208.

http://dx.doi.org/10.16356/j.1005-1120.2020.02.003

and branch and bound (BB)[1,17], have been implemented to solve the ALP. Moreover, some researchers have sought assistance from the heuristic and meta-heuristic algorithms to tackle the ALP, such as cellular automata optimization (CAO)[6], simulated annealing (SA)[7-8], genetic algorithm (GA)[13], and ant colony optimization(ACO)[18].

In summary, studies of ALP encompass the following elements: Choosing an appropriate objective, setting a variety of operational constraints, modeling the problem, and implementing a solution algorithm. The above review revealed that there were usually two optimization strategies for ALP—one to optimize the landing time directly and one to determine the aircraft sequence first and then to allocate landing time. In comparison, the former is time-consuming, especially when the number of arrival aircraft increases. For the second strategy, however, the neighborhood generation method of a candidate sequence should be specifically designed to reduce the computation time. Hancerliogullari et al.[19] took ALP as a machine scheduling problem and combined the composite dispatching rule and SA algorithm to solve the ALP. However, the composite dispatching rule only played a unique role as a hot start for the SA algorithm. In this study, we tackled the ALP by (1) proposing a new composite dispatching rule, (2) presenting a new heuristic approach based on such proposed rule, and (3) developing a new meta-heuristic approach through generating the neighborhood by the proposed rule.

The remainder of this paper is organized as follows. The problem formulation is defined in Section 1. Section 2 presents the proposed rule and the new meta-heuristic approach. The results and discussion are illustrated in Section 3. Concluding remarks are provided in Section 4.

## 1 Problem Formulation

### 1.1 Definition and description of ALP

Nowadays, parallel runways are the most common runway configuration of busiest airports around the world, especially in China. Each runway could be treated separately since the regular operation mode is independent parallel approach or segregated parallel operation. So, the ALP with a single runway is considered in this paper. As a result, the ALP can be defined as follows. To be given a set of arrival aircraft, the goal is to assign a landing sequence and time for each aircraft by optimizing the given objective while subject to a variety of operational constraints[20].

As illustrated in Fig.1, ALP is trying to schedule the arrival aircraft (jobs) on the runway (machine) where the scheduled time is constrained by the earliest (release date) and the latest landing time (deadline), ought to land on the runway at the target landing time (due date). The time window constraints could be obtained through trajectory prediction[21]. In the machine scheduling problem, setup time should be considered, which is similar to the wake vortex (WV) separations (time separation) in ALP. Table 1 summarizes the notation and variables used in this study.

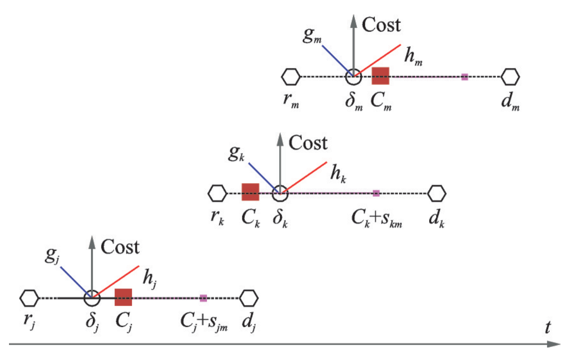



Fig.1 Illustration of aircraft landing problem

### 1.2 Modeling and optimization of ALP

The mixed integer liner program (MILP) formulation for ALP is described as follows

$$\min \quad \sum_{j=1}^{n}(g_jE_j+h_jT_j) \tag{1}$$

$$\text{s.t.} \quad r_j \leqslant C_j \leqslant d_j \quad \forall j \in J \tag{2}$$

$$q_{jk}+q_{kj}=1 \quad \forall j,k \in J; j<k \tag{3}$$

$$C_j \geqslant C_k+q_{kj}s_{kj}-q_{jk}(d_k-r_j) \quad \forall j,k \in J; j \neq k \tag{4}$$

$$E_j \geqslant \delta_j - C_j \quad \forall j \in J \tag{5}$$

$$0 \leqslant E_j \leqslant \delta_j - r_j \quad \forall j \in J \tag{6}$$

$$T_j \geqslant C_j - \delta_j \quad \forall j \in J \tag{7}$$

$$0 \leqslant T_j \leqslant d_j - \delta_j \quad \forall j \in J \tag{8}$$

$$C_j = \delta_j - E_j + T_j \quad \forall j \in J \tag{9}$$

**Table 1 Notation and variables**

| Variable | Single machine scheduling | Aircraft landing problem |
|---|---|---|
| $J$ | A set of jobs | A set of landing aircraft |
| $r_j$ | Release date | The earliest landing time |
| $d_j$ | Deadline | The latest landing time |
| $\delta_j$ | Due date | Target landing time |
| $s_{jk}$ | Set-up time | Wake vortex separation |
| $C_j$ | Completion time | Scheduled landing time |
| $p_j$ | Processing time | Runway occupied time |
| $g_j$ | Earliness weight | Incurred cost for early landing |
| $h_j$ | Tardiness weight | Incurred cost for late landing |
| $E_j = \max(\delta_j - C_j, 0)$ | Earliness of job | Earliness of aircraft |
| $T_j = \max(C_j - \delta_j, 0)$ | Tardiness of job | Tardiness of aircraft |
| $q_{kj} = \{0, 1\}$ | Sequence | Landing sequence |

Eq.(1) minimizes the total penalty of landing deviations from the target landing time. Eq.(2) specifies the time window constraint. Eqs.(3)—(4) ensure that the safe separations between the leading and following aircraft. Eqs.(5)—(8) define the earliness and tardiness of landing. Eq.(9) defines the scheduled time of arrival.

There exist two optimization strategies: One directly tackles the MILP formulation to obtain the scheduled landing time (SLT) and the other first establishes the sequence, then determines the SLT. In the former, once the SLT is obtained, the landing sequence is straightforward. However, it is time-consuming. In the latter, once the sequence is determined, the SLT can easily be calculated by

$$C_{\text{seq}+1} = \max\{\delta_{\text{seq}+1}, C_{\text{seq}} + s_{\text{seq},\text{seq}+1}\} \tag{10}$$

or further optimized by the sub-problem as follows

$$\min \sum_{j=1}^{n} (g_j E_j + h_j T_j) \tag{11}$$

$$\text{s.t. } r_j \leqslant C_j \leqslant d_j \quad \forall j \in J \tag{12}$$

$$\begin{aligned} x_{\text{seq}+1} &\geqslant x_{\text{seq}} + s_{\text{seq},\text{seq}+1} \\ x_{\text{seq}+2} &\geqslant x_{\text{seq}} + s_{\text{seq},\text{seq}+2} \end{aligned} \tag{13}$$

$$E_j \geqslant \delta_j - C_j \quad \forall j \in J \tag{14}$$

$$0 \leqslant E_j \leqslant \delta_j - r_j \quad \forall j \in J \tag{15}$$

$$T_j \geqslant C_j - \delta_j \quad \forall j \in J \tag{16}$$

$$0 \leqslant T_j \leqslant d_j - \delta_j \quad \forall j \in J \tag{17}$$

$$C_j = \delta_j - E_j + T_j \quad \forall j \in J \tag{18}$$

Compared to the original problem, Eqs.(3)—(4) are replaced by Eq.(13). With this change, the number of constraints in the sub-problem is $8n - 3$ while it is $3n(n-1)/2 + 6n$ in the original problem. Such a decrease in the number of constraints significantly reduces the complexity of the MILP programming.

## 2 New Meta-Heuristic Approach

### 2.1 General composite dispatching rules

In the machine scheduling field, dispatching rules are useful when one attempts to find a reasonably good solution in a relatively short time[22]. The most common dispatching rules are earliest release date first (ERD, first come first served rule in ALP) rule, earliest due date first (EDD) rule, minimum slack first (MS) rule, weighted shortest processing time first (WSPT) rule, and so on. The advantages include solving the problem quickly, ease of implementation, and optimal for particular cases. However, a single dispatching rule has limitations of being used in practice and resulting in unpredictably bad solutions because objectives and constraints in the real application could be more complicated. Composite dispatching rule (CDR) is a ranking expression that combines some basic dispatching rules, which could perform significantly better than a single dispatching rule[23-24]. Each basic rule in CDR has its scaling parameter that is chosen to scale the contribution of each basic rules properly.

For the total weighted tardiness minimization problem, the apparent tardiness cost (ATC) rule is a typical CDR. It is a combination of the WSPT rule and the MS rule. By this rule, jobs are scheduled one at a time according to the highest-ranking index

$$I_{\mathrm{ATC}}(t)_j=\frac{w_j}{p_j}\times\exp\left(-\frac{\max(\delta_j-p_j-t,0)}{K_1\bar{p}}\right)\tag{19}$$

where $\bar{p}$ is the average processing times of the remaining jobs, $w_j$ the weight assigned to job $j$, and $K_1$ the scaling parameter. If $K_1$ is very large, the ATC rule changes into WSPT rule. Otherwise, it turns into MS rule.

For the total weighted tardiness minimization problem with sequence-dependent set-ups, the ATC rule can be extended to the ATCS rule[24], which is a combination of WSPT rule, MS rule, and shortest set-up time (SST) rule.

$$I_{\mathrm{ATCS}}(t,k)_j=\frac{w_j}{p_j}\times\exp\left(-\frac{\max(\delta_j-p_j-t,0)}{K_1\bar{p}}\right)\times\exp\left(-\frac{s_{kj}}{K_2\bar{s}}\right)\tag{20}$$

where $s_{kj}$ represents the job $k$ before the job $j$, $\bar{s}$ the average set-up time of the remaining jobs, and $K_2$ the scaling parameter as to set-up time.

For the total weighted tardiness minimization problem with sequence-dependent set-ups and future release times, the ATCS rule can be further extended to ATCSR rule[23] through introducing ERD rule.

$$I_{\mathrm{ATCSR}}(t,k)_j=\frac{w_j}{p_j}\times\exp\left(-\frac{\max(\delta_j-p_j-t,0)}{K_1\bar{p}}\right)\times\exp\left(-\frac{s_{kj}}{K_2\bar{s}}\right)\times\exp\left(-\frac{\max(r_j-t,0)}{K_3}\right)\tag{21}$$

where $K_3$ is the scaling parameter. Such composite rule contains four basic rules—WSPT rule, MS rule, SST rule, and ERD rule. This ranking index establishes a one-to-one relationship between these four specific factors.

## 2.2 The proposed composite dispatching rules

As mentioned, ALP is similar to machine scheduling[25]. However, there are specific features of an ALP when compared with a machine scheduling problem.

Once an aircraft is landing on a runway, it is assumed that the job is completed. Such an assumption indicates that the processing time (runway occupied time, ROT) in ALP can be ignored due to the actual condition that WV separation is more prominent than ROT. The noted composite dispatching rules are mainly concerned with the objective of total weighted tardiness, whereas the objective of total penalty, i.e., weighted earliness and tardiness, is taken into account in this paper.

Considering these differences, we propose a new rule—minimized penalty with due dates and set-ups (MPDS).

$$\begin{aligned}I_{\mathrm{MPDS}}(t,k)_j=&\\ &\exp\left(-\frac{\max(\delta_j-\max(r_j,t+s_{kj}),0)}{K_1}\right)\times\\ &\exp\left(-\frac{s_{kj}}{K_2\bar{s}}\right)\times\exp\left(-\frac{\max(r_j-t,0)}{K_3}\right)\times\\ &\exp(-(g_j\times\max(\delta_j-\max(r_j,t+s_{kj}),0)+\\ &h_j\times\max(\max(r_j,t+s_{kj})-\delta_j,0))/K_4)\end{aligned}\tag{22}$$

The MPDS rule contains four basic rules—improved MS rule, SST rule, ERD rule, and Minimize Penalty rule, as indicated by the four terms in Eq.(22). Furthermore, to obtain good results, the values of scaling parameters should be appropriate for the particular instance of the problem.

$K_1$ is related to the due date range factor $R$, and a study has suggested a guideline for selecting $K_1$

$$K_1=\begin{cases}4.5+R & R\leqslant 0.5\\ 6-2R & R>0.5\end{cases}\tag{23}$$

The due date range factor $R$ is defined as

$$R=(\max_{j\in J}(\delta_j)-\min_{j\in J}(\delta_j))/C_{\max}$$

where the estimated makespan can be

$$C_{\max}\approx\max(\min_{j\in J}(\delta_j)+n\bar{s},\ \max_{j\in J}(\delta_j))$$

$K_2$ is related to the due date tightness factor $\tau$

$$\begin{cases}K_2=\dfrac{\tau}{2\sqrt{\bar{s}}}\\ \tau=1-\dfrac{\sum_{j=1}^{n}\delta_j}{nC_{\max}}\end{cases}\tag{24}$$

where $\sum\delta_j/n$ is the average of the due dates. Values of $\tau$ close to 1 indicate that the due dates are tight while 0 indicates that the due dates are loose.

$K_3$ is related to the release date tightness

$$K_3 = \frac{\max_{j\in J}(r_j) - \min_{j\in J}(r_j)}{\sqrt{s}} \tag{25}$$

$K_4$ is mainly related to the incurred penalty and the number of aircraft

$$K_4 = \frac{\sum_{j\in J} g_j + \sum_{j\in J} h_j}{n} \tag{26}$$

### 2.3 Heuristic and meta-heuristic algorithm

An efficient heuristic algorithm is developed firstly to tackle the ALP in this subsection. As mentioned in Section 1.2, we first determine the landing sequence based on the dispatching rule, single or composite. Then we optimize the landing times (Eqs.(11)—(18)) by using CPLEX software. For a single dispatching rule, like EDD or ERD, it is quite easy. The corresponding heuristic algorithm (EDD_HA or ERD_HA) consists of two significant steps: Sorting and optimizing. For composite dispatching rule, the algorithm (MPDS_HA) is slightly more complicated. Algorithm 1 presents the pseudo-codes of MPDS_HA.

**Algorithm 1**: MPDS based Heuristic Algorithm (MPDS_HA) for ALP

1.Set $t=0; C_j=0; J=\{1,2,\cdots,n\}; \forall j\in J$

2.Calculate scaling parameters using Eqs.(23)—(26)

3.Calculate $I_{\mathrm{MCDS}}(t,k)_j$, according to Eq.(22), $j=\{1,2,\cdots,n\}$ and $s_{kj}=0$

4.Find $j=\{j\in J \mid \max[I_{\mathrm{MPDS}}(t,k)_j]\}$ and put it in the first place

5.Set $C_j=\delta_j,\ t=C_j, k=j$

6.Remove $j$ from $J$

7.While $J\neq\Phi$ do

8. Calculate $I_{\mathrm{MPDS}}(t,k)_j$, according to Eq.(21), $\forall j\in J$

9. Find $j=\{j\in J \mid \max[I_{\mathrm{MPDS}}(t,k)_j]\}$

10. Update $C_j=\max(r_j, C_k+s_{kj})$

11. Update $t=C_j$

12. Update $k=j$

13. Remove $j$ from $J$

14.End While

15.Get the landing sequence and set it into Eq.(13)

16.Optimize sub-problem (Eqs.(11)—(18)) using CPLEX

After setting the initial parameters (line 1), the first landing aircraft is obtained by determining scaling parameters (line 2), calculating ranking index (line 3), finding the highest one (line 4), and updating the decision time (line 5). Then, remove this aircraft (line 6) to construct the remaining set of landing aircraft. Next, MPDS_HA executes the while loop (lines 7—13) for a nonempty remaining set of landing aircraft. Within the while loop, the landing sequence is scheduled once at a time, and the following are executed: Calculating ranking index (line 8), finding the highest (line 9), updating the SLT (line 10), updating the decision time (line 11), updating the scheduled aircraft (line 12) and removing the scheduled aircraft (line 13). Then, the landing sequence is obtained and set into Eq.(13) (line 15). Finally, the SLT is optimized by using CPLEX (line 16).

These proposed algorithms are indeed efficient and easy to be implemented. However, we could not overlook the drawback of these algorithms, i.e., the shortsightedness. On the one hand, the landing sequence is determined by the highest-ranking index. On the other hand, once the landing sequence is determined, it could not be changed. There will be a particular situation, in which several ranking indexes are very close to each other. Such a situation means there is a good chance that we have different optional landing sequences. Therefore, a new meta-heuristic algorithm based on MPDS rule (MPDS_MHA) is developed to overcome the shortsightedness of Algorithm 1. Algorithm 2 presents the pseudo-codes of MPDS_MHA.

**Algorithm 2**: MPDS based Meta-heuristic Algorithm (MPDS_MHA) for ALP

1.Lines 1—6 of Algorithm 1 $t=0; C_j=0; j,k=\{1,2,\cdots,n\}; k\neq j$

2.While $J\neq\Phi$ do

3. Calculate $I_{\mathrm{MPDS}}(t,k)_j$, according to Eq.(22), $\forall j\in J$

4. Find $j=\left\{j\in J \,\middle|\, \max\left\{I_{\mathrm{MPDS}}(t,k)_j\right\}\right\}$

5. Update $C_j=\max(r_j, C_k+s_{kj})$

6. Update $t=C_j$

7. Update $k=j$

8. Find $N_j\{j\}=\left\{j\in J \,\middle|\, I_{\mathrm{MPDS}}(t,k)_j \geqslant \alpha\cdot \sum_{j\in J} I_{\mathrm{MPDS}}(t,k)_j/|J|\right\}$

9. Remove $j$ from $J$

10.End While

11.Get the initial landing sequence $\mathrm{Seq}_0$

12.Get the initial landing times and objective $\mathrm{Obj}_0$ by solving sub-problem (Eqs.(11)—(18))

13.Get the initial solution $S_0\{\mathrm{Seq}_0, \mathrm{Obj}_0\}$

14.Construct the neighborhood structures (NS) by merging $N_{j\in J}\{j\}, N_j\{j\}\cap N_{j+1}\{j\}\neq\varnothing$
15.Let $K$ be the number of NS
16.Set $i=1$
17.While ($i\leqslant K$) do
18. $\mathrm{Seq}_1 \leftarrow$ Generates a neighborhood of $\mathrm{Seq}_0$ using $\mathrm{NS}_i$
19. Get the landing times and objective $\mathrm{Obj}_1$ by solving sub-problem (Eqs.(11)—(18))
20. If $\mathrm{Obj}_1 < \mathrm{Obj}_0$ then
21. $S_0\{\mathrm{Seq}_0, \mathrm{Obj}_0\} \leftarrow S_1\{\mathrm{Seq}_1, \mathrm{Obj}_1\}$
22. $i=1$
23. Else
24. $i=i+1$
25. End If
26.End While
27.Return the best solution

The several initial steps of Algorithm 2 are the same as Algorithm 1. Next, MPDS_MHA executes the first while loop (lines 2—9) of calculating, sorting and updating to obtain the initial landing sequence. The specific step of MPDS_MHA lies in line 8, which generates the potential neighbors of each scheduled aircraft. At this step, $\alpha$ is a predefined parameter, which could affect the number of potential neighbors. For each scheduled aircraft, there will be at least one potential neighbor, and the adjacent scheduled aircraft may share the same potential neighbors. Then, the initial solution is obtained (lines 11—13). Next, MPDS_MHA adopts a meta-heuristic framework. Within the meta-heuristic framework, the following steps are implemented. As shown in Fig.2, if the adjacent scheduled aircraft have some common neighbors. The neighborhood structures (NSs) are constructed by merging the potential neighbors. If the adjacent scheduled aircraft have totally different potential neighbors, NS is constructed accordingly (line 14). After setting the number of NS (line 15), the main loop of the local search is executed (lines 17—26). At each iteration, generate a neighborhood solution by using $\mathrm{NS}_i$ (line 18), in which the roulette wheel selection is implemented to apply the insertion, reversion or swap operator. As shown in Fig.3, the insertion means getting two indexes randomly and making their position adjacent. The reversion means to invert the old sequence of the neighborhood. And the

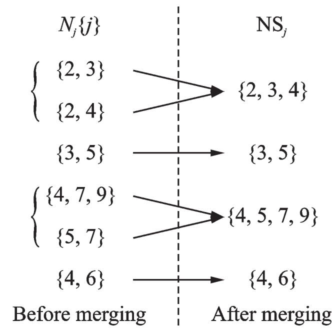



Fig.2 An example of neighborhood construction

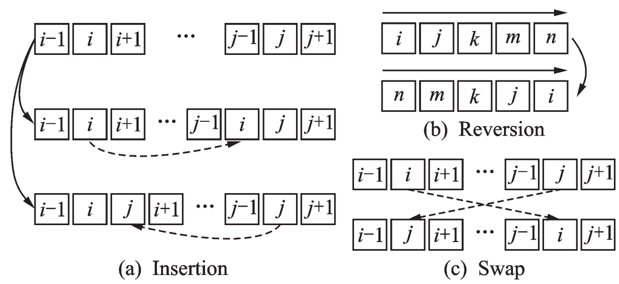



Fig.3 Methods of neighborhood generation

swap means to exchange the position of the two aircraft. Then, the optimized landing times and objective values are produced (line 19). If the generated neighborhood solution ($S_1$) is better than the so-far best solution ($S_0$) (line 20), then replace $S_0$ with $S_1$ (line 21). Otherwise, increase $i$ by one (line 24) to call the next local search. MPDS_MHA will stop the search if the so-far optimum of the $k$ neighborhood structure cannot be improved any further.

## 3 Computational Results and Discussion

### 3.1 Computational scenario

The performance of the proposed method is evaluated using a set of benchmark instances taken from the OR library. Such instances are summarized in Table 2. Also, we split the benchmark instances into small scales involving 10—50 aircraft and large scales involving 100, 200, and 500 aircraft.

The proposed algorithms were run on a PC with a 2.3 GHz Intel Core I5-6200U processor and 4 GB RAM. The corresponding MILP model of ALP was solved using CPLEX software (IBM ILOG CPLEX Optimization studio version 12.5.1).

**Table 2 Computational scenarios**

| Scale | Benchmark Instance | Instance number | Number of aircraft |
|---|---|---|---|
| Small scale | Airland #1 | 1 | 10 |
| | Airland #2 | 2 | 15 |
| | Airland #3 | 3 | 20 |
| | Airland #4 | 4 | 20 |
| | Airland #5 | 5 | 20 |
| | Airland #6 | 6 | 30 |
| | Airland #7 | 7 | 44 |
| | Airland #8 | 8 | 50 |
| Large scale | Airland #9 | 9 | 100 |
| | Airland #11 | 10 | 200 |
| | Airland #13 | 11 | 500 |

### 3.2 Small scale instances

As mentioned in section 1.2, for solving the ALP, there exists a strategy of first establishing the landing sequence, then determining the SLT. Also, once the sequence is determined by using ERD, EDD or MPDS, the SLT can easily be calculated by Eq.(10) or further optimized through Eqs. (11)—(18), i.e., by MPDS_HA.

Table 3 provides a comparison between different strategies (calculation or optimization) under different dispatching rules, in which the objective values are taken as performance, and only small-scale instances are considered.

From Table 3, we could find that: (1) The proposed MPDS rule is better than single dispatching rules; (2) the optimization strategy, i.e., MPDS_HA, is far better than the calculation strategy; and (3) the MPDS_HA could obtain the optimal solutions for the small-scale instances, except Instance 8.

**Table 3 Computational results of small scale instances**

| Instance | Optimal | Calculation (Eq.(10)) | | | Optimization (Eqs.(11)—(18)) | | |
|---|---|---|---|---|---|---|---|
| | (CPLEX) | ERD | EDD | MPDS | ERD_HA | EDD_HA | MPDS_HA |
| 1 | 700 | 1 790 | 1 210 | 1 210 | 1 280 | 700 | 700 |
| 2 | 1 480 | 2 610 | 2 030 | 1 720 | 1 790 | 1 500 | 1 480 |
| 3 | 820 | 2 930 | 2 870 | 1 610 | 1 790 | 1 730 | 820 |
| 4 | 2 520 | 7 390 | 4 480 | 4 480 | 4 890 | 2 520 | 2 520 |
| 5 | 3 100 | 8 370 | 7 120 | 4 800 | 6 470 | 5 420 | 3 100 |
| 6 | 24 442 | 24 442 | 24 442 | 24 442 | 24 442 | 24 442 | 24 442 |
| 7 | 1 550 | 3 974 | 3 974 | 3 974 | 1 550 | 1 550 | 1 550 |
| 8 | 1 950 | 31 140 | 4 390 | 4 415 | 18 790 | 2 450 | 2 230 |

While looking into the details of cases, we identified several reasons for Instance 8 being difficult to obtain the optimal solution by MPDS_HA. The first reason is the short scheduled time window per aircraft in Instance 8. The scheduled time window per aircraft is the entire scheduled window divided by the total number of aircraft. We considered only Instances 1—5 and Instance 8, as the same WV separations were used in these cases. The average scheduled window of Instance 8 is around 20 s, which is much lower than the others (40—65 s), which leaves limited flexibility for resorting. The second reason is that the proposed MPDS rule does not always bring about the optimal landing sequence, which inevitably leads to the sub-optimal solution during the optimization by MPDS_HA. Because the MPDS rule only has a shortsighted vision, it determines one aircraft's position each round, as shown in Eq.(22) or line 9 of the Pseudo Codes of MPDS_HA. Therefore, we have developed MPDS_MHA by using meta-heuristic strategy, i.e., to generate the potential sequences for local searching.

Fig.4 provides the scenarios and scheduling results of Instance 8, which consisted of earliest and latest landing time (black star line), TLTs (Target Landing Times, black box), SLTs obtained by CPLEX (black diamond), SLTs by MPDS_HA (black circle) and SLTs by MPDS_MHA (black triangle) with $\alpha$=0.25. Fig.5 displays the deviations between the TLTs and the SLTs obtained from different methods. The objective value of Instance 8 by

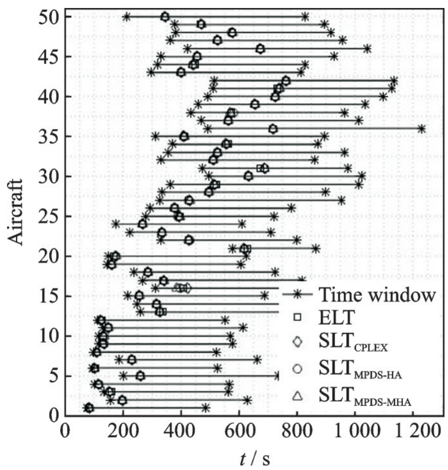



Fig.4 Scheduled results of Instance 8

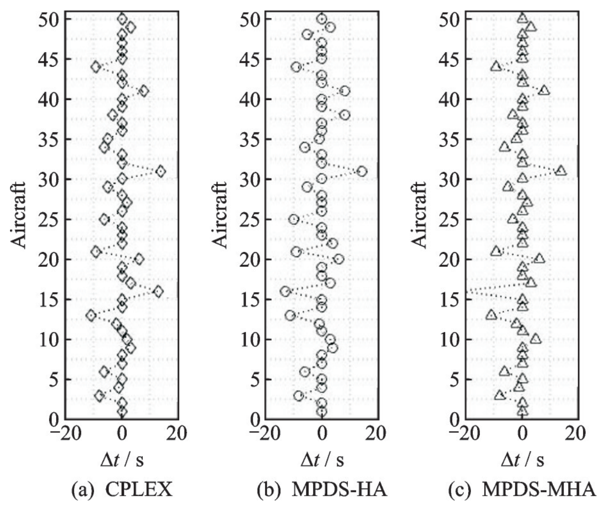



Fig.5 Deviations between TLTs and SLTs of Instance 8

CPLEX, MPDS_HA, and MPDS_MHA are 1 950, 2 230 and 1 950, respectively.

Table 4 illustrates all the results of Instance 8 by different methods.

As shown in Figs.4, 5, most scheduled results are the same, while some are entirely different. However, by MPDS_HA, there are 21 aircraft whose SLTs are not sticking to the corresponding TLTs, while by CPLEX, 21, and by MPDS_MHA, 20. Furthermore, MPDS_MHA is more likely to schedule those aircraft with a lower penalty for early or late landing. The subsequence of the 23th—27th landing aircraft is a case in point.

By MPDS_HA, the subsequence is aircraft #26, #16, #25, #43, #35.

By CPLEX, the subsequence is aircraft #26, #25, #43, #35, #16.

By MPDS_MHA, the subsequence is aircraft #26, #25, #16, #43, #35.

And the penalties of aircraft #16, #25, #26, #35, #43 are 10, 15, 30, 15, and 25 per second. From Fig.5, we could find that aircraft #16 (penalty 10) is deviated from the TLT most by MPDS_MHA, while aircraft #31 (penalty 15) is deviated from the TLT most by MPDS_HA.

**Table 4 Objective value comparison of Instance 8**

| Literature | Method | Objective value |
|---|---|---|
| Ref.[1] | Heuristic | 2 690 |
| Ref.[3] | DALP | 2 000 |
| Ref.[5] | SS | 2 965 |
| | BA | 2 655 |
| Ref.[6] | CAO | 1 995 |
| Ref.[7] | SA+VND | 1 950 |
| | SA+VNS | 1 950 |
| Ref.[8] | ALNS | 1 950 |
| Ref.[10] | Discretization | (Not include this instance) |
| Ref.[26] | ILS | 1 959 |
| Ref.[27] | HPSO-LS | 1 950 |
| Ref.[28] | SSE | (Not include this instance) |
| Our study | MPDS_HA | 2 230 |
| | MPDS_MHA | 1 950 |

## 3.3 Large scale instances

### 3.3.1 Parameter analysis

Since scaling parameters play an essential role in the MPDS rule, the primary purpose of this subsection is to prove our parameter determination method (Eqs.(23) — (26) ) is as good as a grid search strategy, which is used in the machine scheduling[23].

Take airland#9 as an example, the grids are

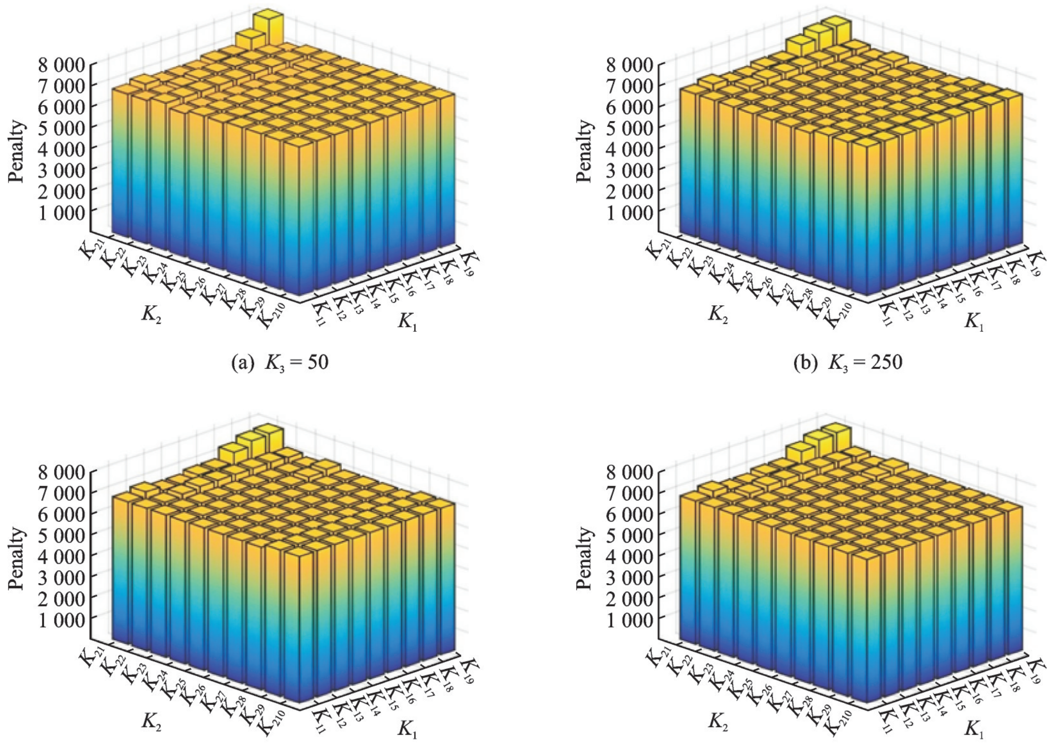
$K_1 = \{1.0, 2.5, 4.0, 4.5, 5.0, 5.5, 6.0, 7.5, 10\}$,

$K_2 = \{0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 1, 2.5, 5, 10\}$,

$K_3 = \{50, 250, 500, 1\,000\}$,

and $K_4 = 2\,000$, while $K_1 = 4.13$ $K_2 = 0.03$ , $K_3 = 136$ and $K_4 = 2\,000$ based on Eqs.(23)—(26).

Fig.6 shows objective values with different scaling parameters. The minimum total penalty is 6 792, and the maximum is 8 073. Fig.7 illustrates the distribution of scheduled results with different scaling parameters for ALP #9. Nearly 97% of the scheduled results are less than 7 097. Meanwhile, 6 841 is our objective value based on MPDS_HA for the scaling parameters. The conclusion could be drawn that tuning the scaling parameters can obtain a better result, but it is time-consuming, while our parameter determination method is a competent way.




(a) $K_3 = 50$ (b) $K_3 = 250$

(c) $K_3 = 500$ (d) $K_3 = 1\ 000$

Fig.6 Results with different scaling parameters for Instance 9

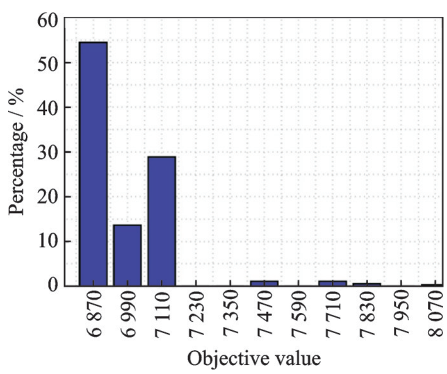



Fig.7 Distribution of results with different scaling parameters

### 3.3.2 Effectiveness of MPDS_MHA

In this section, the effectiveness of MPDS_MHA will be evaluated by large-scale instances. The results obtained with the proposed algorithms and other existing methods are shown in Table 5.

Table 5 also shows the percentage gap ($G$) for comparison regarding the best objective values. The percentage gap ($G$) is calculated as

$$G = \frac{\mathrm{Obj} - \mathrm{Obj}'}{\mathrm{Obj}'} \times 100\% \tag{27}$$

where Obj is the best objective value obtained by different methods and Obj′ is the best value so far.

From Table 5, we could find that MPDS_MHA is far better than MPDS_HA since the former one considers meta-heuristic strategy. In comparison to the other existing methods, MPDS_MHA is also a competitive and promising algorithm.

We take Instance 9 as an example to carry out the comparison study about the computational times, as shown in Table 6. The CPU time of MPDS_HA is 1.3 s, significantly shorter than those of the existing methods. The time-effectiveness of MPDS_HA is mostly attributed to the reduction of constraints in the mathematical optimization problem after the landing sequence is determined with the proposed composite dispatching rule, MPDS. Howev-

**Table 5 Objective value comparison of large-scale instances**

| Literature | Method | Instance 9 | | Instance 10 | | Instance 11 | |
|---|---|---|---|---|---|---|---|
| | | Obj | $G$ | Obj | $G$ | Obj | $G$ |
| Ref.[27] | CPLEX[a] | 5 612 | 0.00 | 12 418 | 0.00 | 37 849 | 0.00 |
| Ref.[2] | DALP | 7 848 | 39.84 | 19 327 | 55.64 | 58 393 | 54.28 |
| Ref.[5] | SS | 7 298 | 30.04 | 14 647 | 17.95 | 46 285 | 22.29 |
| | BA | 6 426 | 14.50 | 14 488 | 16.67 | 45 294 | 19.67 |
| Ref.[6] | CAO | 5 612 | 0.04 | 12 439 | 0.00 | 38 573 | 1.91 |
| Ref.[7] | SA+VND | 6 092 | 8.55 | 12 418 | 0.00 | 39 867 | 5.33 |
| | SA+VNS | 6 092 | 8.55 | 12 418 | 0.00 | 41 448 | 9.51 |
| Ref.[10] | Discretization | 6 119 | 9.03 | 13 343 | 7.45 | 41 392 | 9.36 |
| Ref.[26] | ILS | 5 614 | 0.04 | 12 420 | 0.02 | 41 391 | 9.36 |
| Our study | MPDS_HA | 6 841 | 21.90 | 14 709 | 18.45 | 43 608 | 15.22 |
| | MPDS_MHA | 5 806 | 3.46 | 12 841 | 3.41 | 40 228 | 6.29 |

Note: a—The upper bound on the CPU time of the CPLEX solver is set 3 600 s.

**Table 6 CPU time comparison of Instance 9**

| Literature | Method | PC used | CPU time/s |
|---|---|---|---|
| Ref.[5] | SS | 2 GHz, 512 MB RAM | 119.00 |
| | BA | | 554.00 |
| Ref.[6] | CAO | 1.6 GHz | 6.27 |
| Ref.[7] | SA+VND | 2.4 GHz, 512 MB RAM | 11.59 |
| | SA+VNS | | 10.12 |
| Ref.[10] | Discretization | 2.5 GHz, 4G RAM | 29.00 |
| Ref.[26] | ILS | 2.66 GHz, 2G RAM | 7.60 |
| Our study | MPDS_HA | 2.3 GHz, 4G RAM | 1.30 |
| | MPDS_MHA | | 29.62 |

er, MPDS_MHA needs more time, compared with MPDS_HA, to conduct the local search for finding the near-optimal solution.

## 4 Conclusions

A new meta-heuristic approach, based on composite dispatching rule, is put forward in this paper to solve the aircraft landing problem of minimizing the total penalty. Such proposed composite dispatching rule, MPDS, could not only efficiently establish an initial landing sequence but also effectively provide the potential landing sequences by the ranking indexes, as shown in line 8 of Algorithm 2. Thereupon, the proposed approach, MPDS_MHA, could find a good solution within a reasonable time after the local search.

Our proposed methods are evaluated by using a set of benchmark instances taken from the OR library. The computational results show that the MPSD_HA method could get a generally good result in a short time and the MPSD_MHA method could obtain the optimal result in a little bit longer time. Therefore, the combination of CDR and meta-heuristic strategy is an effective way to solve the ALP.

Future work is worth exploring in the following areas—applying the proposed method to solve multi-runway ALP and ALP with arrival time uncertainty, developing a new composite dispatching rule for multi-objective ALP, and tackling the integrated arrival and departure scheduling problem based on our approach.

**References**

[1] BEASLEY J E, KRISHNAMOORTHY M, SHARAIHA Y M, et al. Scheduling aircraft landings—The static case[J]. Transportation Science, 2000, 34(2): 180-197.

[2] BEASLEY J E, KRISHNAMOORTHY M, SHARAIHA Y M, et al. Displacement problem and dynamically scheduling aircraft landings[J]. Journal of the Operational Research Society, 2004, 55(1): 54-64.

[3] BENNELL J A, MESGARPOUR M, POTTS C N. Airport runway scheduling[J]. Annals of Operations Research, 2013, 204(1): 249-270.

[4] BENNELL J A, MESGARPOUR M, POTTS C N. Dynamic scheduling of aircraft landings[J]. European Journal of Operational Research, 2017, 258(1): 315-327.

[5] PINOL H, BEASLEY J E. Scatter search and bionomic algorithms for the aircraft landing problem[J].

European Journal of Operational Research, 2006, 171 (2): 439-462.

[6] YU S, CAO X, ZHANG J. A real-time schedule method for aircraft landing scheduling problem based on cellular automation[J]. Applied Soft Computing Journal, 2011, 11(4): 3485-3493.

[7] SALEHIPOUR A, MODARRES M, MOSLEMI L. An efficient hybrid meta-heuristic for aircraft landing problem[J]. Computers and Operation Research, 2013, 40(1): 207-213.

[8] VADLAMANI S, HOSSEINI S. A novel heuristic approach for solving aircraft landing problem with single runway[J]. Journal of Air Transport Management, 2014, 40: 144-148.

[9] LIEDER A, BRISKORN D, STOLLETZ R. A dynamic programming approach for the aircraft landing problem with aircraft classes[J]. European Journal of Operational Research, 2015, 243(1): 61-69.

[10] FAYE A. Solving the aircraft landing problem with time discretization approach[J]. European Journal of Operational Research, 2015, 242(3): 1028-1038.

[11] EUN Y, HWANG I, BANG H. Optimal arrival flight sequencing and scheduling using discrete airborne delays[J]. IEEE Transactions on Intelligent Transportation Systems, 2010, 11(2): 359-373.

[12] HU X, CHEN W. Receding horizon control for aircraft arrival sequencing and scheduling[J]. IEEE Transactions on Intelligent Transportation Systems, 2005, 6(2): 189-197.

[13] HU X, DI PAOLO E A. A ripple-spreading genetic algorithm for the aircraft sequencing problem[J]. Evolutionary Computation, 2011, 19(1): 77-106.

[14] BALAKRISHNAN H, CHANDRAN B G. Algorithms for scheduling runway operations under constrained position shifting[J]. Operations Research, 2010, 58(6): 1650-1665.

[15] JI X, CAO X, DU W, et al. An evolutionary approach for dynamic single-runway arrival sequencing and scheduling problem[J]. Soft Computing, 2017, 21 (23): 7021-7037.

[16] BRISKORN D, STOLLETZ R. Aircraft landing problems with aircraft classes[J]. Journal of Scheduling, 2014, 17(1): 31-45.

[17] SÖLVELING G, CLARKE J P. Scheduling of airport runway operations using stochastic branch and bound methods[J]. Transportation Research Part C: Emerging Technologies, 2014, 45: 119-137.

[18] XU B. Ant colony algorithm based on wake-vortex modeling method for aircraft scheduling problem[J]. Journal of Computational and Applied Mathematics, 2017, 317: 157-170.

[19] HANCERLIOGULLARI G, RABADI G, AL-SALEM A H, et al. Greedy algorithms and metaheuristics for a multiple runway combined arrival-departure aircraft sequencing problem[J]. Journal of Air Transport Management, 2013, 32: 39-48.

[20] ZHANG J F, ZHAO P, ZHANG Y, et al. Criteria selection and multi-objective optimization of aircraft landing problem[J]. Journal of Air Transport Management, 2020, 82: 101734.

[21] ZHANG J F, LIU J, HU R, et al. Online four dimensional trajectory prediction method based on aircraft intent updating[J]. Aerospace Science and Technology, 2018, 77: 774-787.

[22] SHARMA P, JAIN A. Performance analysis of dispatching rules in a stochastic dynamic job shop manufacturing system with sequence-dependent setup times: Simulation approach[J]. CIRP Journal of Manufacturing Science and Technology, 2015, 10: 110-119.

[23] PFUND M, FOWLER J W. Scheduling jobs on parallel machines with setup times and ready times[J]. Computers & Industrial Engineering, 2008, 54(4): 764-782.

[24] LEE Y H, PINEDO M. Scheduling jobs on dynamic parallel machines with sequence-dependent setup times[J]. International Journal of Advanced Manufacturing Technology, 2010, 47(5/6/7/8): 773-781.

[25] ZHANG J F, ZHENG Z, GE T. Sequencing approach of arrival aircrafts based on composite dispatching rules[J]. Journal of Traffic and Transportation Engineering, 2017, 17(3): 141-150.

[26] SABAR N R, KENDALL G. An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem[J]. Omega, 2015, 56: 88-98.

[27] GIRISH B S. An efficient hybrid particle swarm optimization algorithm in a rolling horizon framework for the aircraft landing problem[J]. Applied Soft Computing Journal, 2016, 44: 200-221.

[28] JI X, CAO X, TANG K. Sequence searching and evaluation: A unified approach for aircraft arrival sequencing and scheduling problems[J]. Memetic Computing, 2016, 8(2): 109-123.

**Acknowledgements** This work was supported by the Joint Fund of National Natural Science Foundation of China and Civil Aviation Administration of China (No. U1933117).

**Author** Dr. **ZHANG Junfeng** received the B.S. degree in the automated industry from Nantong University in 2002 and the Ph.D. degree in Control Theory and Control Engineering from Nanjing University of Aeronautics and Astronautics in 2008, respectively. He joined in Nanjing University of Aeronautics and Astronautics in July 2008, where he is currently an associate professor in the College of Civil Aviation. His research interests include (1) 4D trajectory generation, prediction and optimization; (2) decision support tool (DST) for the air traffic control, like AMAN and DMAN. These researches are supported by the NSFC, ATMB of CAAC, Boeing, and COMAC. His research works have been published by many journals, such as Aerospace Science and Technology, Journal of Air Transportation Management.

**Author contributions** Dr. **ZHANG Junfeng** designed the study, complied with the models, conducted the analysis, interpreted the results, and wrote the manuscript. Mr. **ZHAO Pengli** contributed to the data preparation and simulation of the study. Mr. **YANG Chunwei** contributed to interpreting the results and designed the algorithm. Dr. **HU Rong** provided the discussion and background of the research. All authors commented on the manuscript draft and approved the submission.

**Competing interests** The authors declare no competing interests.

(Production Editor: SUN Jing)

# 航空器着陆调度问题的一种新型元启发式方法

张军峰，赵鹏力，杨春苇，胡　荣

(南京航空航天大学民航学院，南京 211106，中国)

**摘要：**基于一种新型复合分派规则，提出了一种新型元启发式算法以期求解进场航班排序与调度问题(Aircraft landing problem, ALP)。首先，将ALP等价为最小化加权总延误(加权总提前和加权总滞后)的机器调度问题。其次，提出了一种复合分派规则，即含截止时间约束和顺序决定准备时间约束的最小成本规则(Minimized penalty with due dates and set-ups, MPDS)，以此确定航班的着陆次序。然后，提出一种结合MPDS复合分派规则和CPLEX求解器的高效启发式算法：在第一阶段，由复合分派规则确定航班的次序；在第二阶段，使用CPLEX求解器优化着陆时间。接着，对由复合分派规则生成的潜在可行解进行本地搜索，将新型元启发式策略引入启发式算法得到优化序列。最后，使用从OR Library数据库中获取的多组通用数据来评估所提出方法的性能。结果证明了所提出方法的有效性和高效性。

**关键词：**进场调度；空中交通管制；决策支持；元启发；局部搜索