A Bilevel Programming Approach for Optimization of Airport Ground Movement

JIANG Yu[1*], *HU Zhitao*[1], *LIU Zhenyu*[1], *ZHANG Haoyu*[1], *WANG Zhe*[2]

1.College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P.R. China;
2.Beijing Capital International Airport Co., Ltd., Beijing 100621, P.R. China

(Received 10 July 2020; revised 30 November 2020; accepted 3 December 2020)

Abstract: This paper proposes an optimization model for the airport ground movement problem (GMP) based on bilevel programming to address taxi conflicts on the airport ground and to improve the operating safety and efficiency. To solve GMP, an iterative heuristic algorithm is designed. Instead of separately investigating each problem, this model simultaneously coordinates and optimizes the aircraft routing and scheduling. A simulation test is conducted on Nanjing Lukou International Airport (NKG) and the results show that the bilevel programming model can clearly outperform the widely used first-come-first-service (FCFS) scheduling scheme in terms of aircraft operational time under the precondition of none conflict. The research effort demonstrates that with the reduced operating cost and the improved overall efficiency, the proposed model can assist operations of the airports that are facing increasing traffic demand and working at almost maximum capacity.

Key words: airport ground movement (GMP); aircraft routing and scheduling; bilevel programming; iterative heuristic; air transportation

CLC number: V351.11 **Document code:** A **Article ID:** 1005-1120(2021)05-0829-11

0 Introduction

Airports are experiencing ever more congestions on ground movements, due to the long-term growth of air transportation. It has increased air traffic controllers' workload, safety risks, operating costs and decreased passenger comfort. More noise pollution and exhaust emissions have also impeded airports' sustainable growth. Currently, increasing the throughput capacity and efficiently using existed infrastructure are two major methods in airport surface operations to relieve congestions[1]. As the former is always restricted by stringent land use and financial investment[2], optimizing the usage of airport terminal areas has become the leading approach. Airport surface resources consist of gates, taxiways and runways. As taxiways are the physical links between gates and runways, optimization of ground movement on taxiways directly relates to other problems, like gate assignment and runway sequencing[3-5]. Therefore, the taxiway operation is crucial to the holistic airside efficiency.

Atkin et al.[6] attributed the airport ground movement problem (GMP) to a fundamental routing and scheduling problem. An aircraft is instructed to taxi from the source to its destination following a specific path (taxi routes) in a timely manner, e.g. pushback delays, holding patterns and speed profiles. In the case where only a few aircraft need to be dealt with at one time, a shortest path algorithm, like Dijkstra's algorithm or A*, can be used to generate taxi routes. For larger hub airports with more intensive traffic flows on the ground, an alternative routing and scheduling strategy is required to avoid conflicts between moving aircraft as well as to increase overall efficiency.

Some studies have dealt with GMPs using the mixed integer linear programming (MILP), where objective functions and constraints are formulated as

*Corresponding author, E-mail address: jiangyu07@nuaa.edu.cn.

How to cite this article: JIANG Yu, HU Zhitao, LIU Zhenyu, et al. A bilevel programming approach for optimization of airport ground movement[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2021, 38(5): 829-839.
http://dx.doi.org/10.16356/j.1005-1120.2021.05.011

linear expressions with the presence of both integers and continuous variables. Time variables are either discretized through an exact position approach, usually under a time-space network model[7], or described as continuous ones when binary variables are used to represent the ordering of passing taxiway intersections or sectors[8]. In Ref.[3], a comprehensive MILP model were incorporated with the receding horizon scheme in conformance with the dynamic nature of airport GMPs, and an iterative algorithm was designed to fasten the computation. As a GMP is essentially an NP-hard problem, the enlargement of the problem scale will dramatically increase the complexity of the algorithm. Other practices in MILP for GMPs can be found in Refs.[9-11].

Considering the difficulty of developing an exact algorithm in terms of MILP formulation, some researchers adopted either heuristics or metaheuristics in searching for an acceptable approximate optimization rather than a perfect theoretical one. These approaches usually compensate for the loss of optimality by relatively short time span. Nogueira et al.[12] used the ant colony algorithm to optimize taxi routes of aircraft. They searched optimal taxi routes and minimized taxi time in accordance with safety constraints. Weiszer et al.[13] presented an integrated optimization approach for airport ground operation. They adopted an active routing (AR) strategy in the model to obtain the shortest taxi time. Other studies involving the application of heuristics in GMPs can be found in Refs.[14-18].

In recent studies on airport GMPs, multiple aspects of interest have been considered, including the incorporation with runway sequencing[3-4,19-20], pollution emission[1,21] and the issue of uncertainty[22-25]. These efforts essentially focused on the routing and scheduling problem, despite their different research scopes.

To our best knowledge, a critical issue that has not been adequately addressed in this domain is the integrating and coordinating both routing and scheduling problems simultaneously, instead of previous attempts on only one problem. Biased focus on either side may lose the global optimality of the overall GMP and the airport airside capacity is potentially decreased. Therefore, we propose a bilevel programming model dealing with routing and scheduling simultaneously from the collaborative perspective. To the best of our knowledge, no other research has introduced the use of bilevel programming on GMP though this method has been applied to other transport optimization problem, as illustrated in Refs.[26-29]. An iterative heuristic is then designed to solve the problem in an efficient way.

1 Problem Description and Bilevel Modeling

1.1 Ground movement problem

As aforementioned, airport GMPs can be defined as the work of assigning taxi routes and the timings of arriving at taxiway sectors or intersections for both arrival and departure flights. The running procedure for an arrival flight is taxiing from runway exit to a predetermined gate following a specific path, while a departure flight taxies from the gate to an assigned runway threshold. In the taxiing process, a prescribed safety separation should be strictly maintained between any two aircraft in order to avoid conflicts. Fig.1 demonstrates the taxi scheduling flow in a large airport.

In order to clarify the research category and standardize the research content, certain assumptions and simplifications are defined as follows, though it may compromise the fidelity of the model.

(1) Airport ground is abstracted to a directed graph $G=(V,E)$, where V and E represent nodes and edges, respectively. Any edge $(v_1,v_2)\in E$ connecting nodes v_1 and v_2 has a direction, which means aircraft are only allowed to taxi from v_1 to v_2.

(2) In order to constrain the problem scale within a manageable range, it is assumed that the taxi route for each aircraft is selected from a set of predetermined feasible routes.

(3) Aircraft ground taxi is a continuous operation process, which involves massive operating da-

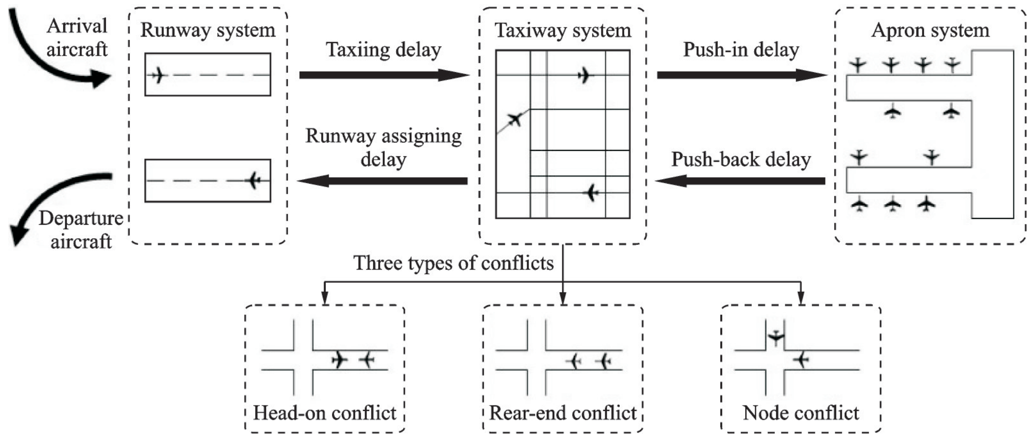

Fig.1 Operation flow on airport ground

ta. In this paper, the ground taxi research is conducted within a busy period during peak hours.

(4) The moving speed of an aircraft is assumed to be constant during taxiing. The process is continuous and unimpeded, which means any stopping and waiting is not allowed. Taxi conflicts are avoided by controlling the time of entering taxiway and selecting the taxi routes.

(5) Some adjacent stands are integrated as an apron. This model only requires aircraft to be assigned to certain aprons. The number of stands in each apron is sufficient for all the departure and arrival aircraft within the planned period, and the stand for each aircraft is assigned in advance.

(6) The departure aircraft taxies from the stand to the runway threshold, and the arrival aircraft taxies from runway exit to the stand. Both the runway threshold and the exit for each aircraft are fixed.

1.2 Bilevel programming model

In the ground movement bilevel programming model, the decision variable at the upper level is the start time of taxiing. By controlling the start time of taxiing, it aims at reducing conflicts during taxiway operation and aircraft's waiting time before taxiing. The decision variable at the lower level is the taxi routes assigned for aircraft. The taxi conflicts and travel distance can be reduced after selecting proper routes. Notations used in this model are presented in Table 1.

Table 1 Notations

F	Set of all aircraft within the planned period, $F=\{f_1,f_2,\cdots,f_k\}$
F^{d}	Set of departure aircraft within the planned period
F^{a}	Set of arrival aircraft within the planned period
N	Set of all nodes, any node $n_p,n_q\in N$
N^i	Taxi route of aircraft f_i, which is composed of several nodes, $N^i=\{n_1^i,n_2^i,\cdots,n_{k_i}^i\}$
N_{up}^i	Taxi route of aircraft f_i at the upper level
N_{low}^i	Taxi route of aircraft f_i at the lower level
L_{pq}	Edge length between nodes n_p and n_q
V_i	Taxi speed of aircraft f_i
T_{ip}	Time of aircraft f_i arriving at node n_p
T_i^{s}	Time of aircraft f_i starting taxiing
R_{ipq}	Equal to 1 if aircraft f_i taxies from node n_p to node n_q, otherwise equal to 0
Z_{ijp}	Equal to 1 if aircraft f_i arrives at node n_p before aircraft f_j, otherwise equal to 1
d_{sep}	Safety separation between aircraft f_i and f_j
ETA_i	Estimated arrival time of aircraft f_i, i.e. start time of taxiing
ETP_i	Estimated pushback time of aircraft f_i
ETD_i	Estimated departure time of aircraft f_i
c^{a}	Waiting cost of arrival aircraft
c^{d}	Waiting cost of departure aircraft

The objective of the upper-level programming is to minimize the total waiting time and the number of conflicts for all arrival and departure aircraft. The decision variable is the start time of taxiing T_i^{s}. During the scheduling process, the targets are minimizing waiting time of arrival and departure aircraft and lowering the ground operation costs, as well as resolving conflicts and improving efficiency. Departure aircraft are supposed to start taxiing as early as possible to reduce the delay at stands. Meanwhile, the arrival aircraft should land as early as possible to minimize the costs of waiting and impacts on follow-up flights. Thus, the objective function and constraints at the upper level are formulated as follows

$$\min Z_1=\left(\sum_{i\in F^{\mathrm{d}}}c^{\mathrm{d}}(T_i^s-\mathrm{ETP}_i)+\sum_{i\in F^{\mathrm{a}}}c^{\mathrm{a}}(T_i^s-\mathrm{ETA}_i)+M\cdot\delta\right) \tag{1}$$

$$N_{\mathrm{up}}^i=N_{\mathrm{low}}^i \tag{2}$$

$$T_i^s\geqslant\mathrm{ETP}_i\qquad\forall f_i\in F^{\mathrm{d}} \tag{3}$$

$$T_i^f\leqslant\mathrm{ETD}_i\qquad\forall f_i\in F^{\mathrm{d}} \tag{4}$$

$$T_i^s\geqslant\mathrm{ETA}_i\qquad\forall f_i\in F^{\mathrm{a}} \tag{5}$$

Eq.(1) is the objective function at the upper level. Since the cost of waiting is higher in the air than in the stands, the waiting cost of arrival aircraft c^{a} is set to be greater than that of departure aircraft c^{d}. δ stands for the number of taxiing conflicts happened, and M is a big positive number used as penalty for conflicts. In Eq.(2), the aircraft taxi routes calculated at the lower level are transferred to be a constraint at the upper level. Eq.(3) ensures the departure aircraft must not start taxiing until the estimated pushback time. Eq.(4) ensures the departure aircraft must enter the runway prior to the estimated departure time. Eq.(5) ensures the arrival aircraft must start taxiing after the estimated arrival time.

The objective of the lower-level programming is to minimize the total taxi time of all aircraft by properly assigning taxi routes for the aircraft. The decision variable is the feasible taxi route for aircraft f_i, $N^i=\{n_1^i,n_2^i,\cdots,n_{k_i}^i\}$. The aircraft is supposed to start taxiing according to the time passed from the upper level. The objective function and constraints at the lower level are formulated as follows

$$\min Z_2=\left(\sum_{f_i\in F}(T_{in_{k_i}^i}-T_{in_1^i})+M\cdot\delta\right) \tag{6}$$

$$T_{in_1^i}=T_i^s \tag{7}$$

$$\frac{L_{pq}}{V_i}\cdot R_{ipq}-M(1-R_{ipq})\leqslant T_{iq}-T_{ip}\qquad\forall f_i\in F;\forall n_p,n_q\in N \tag{8}$$

$$Z_{ijp}(T_{ip}+\tau_{ijp})\leqslant Z_{ijp}T_{jp}\qquad\forall f_i,f_j\in F;n_p\in N \tag{9}$$

$$Z_{ijp}-Z_{ijq}\leqslant 2-(R_{ipq}+R_{jpq})\qquad\forall f_i,f_j\in F;\forall n_p,n_q\in N \tag{10}$$

$$Z_{ijp}-Z_{ijq}\geqslant -2+(R_{ipq}+R_{jpq})\qquad\forall f_i,f_j\in F;\forall n_p,n_q\in N \tag{11}$$

$$Z_{ijp}-Z_{ijq}\leqslant 2-(R_{ipq}+R_{jqp})\qquad\forall f_i,f_j\in F;\forall n_p,n_q\in N \tag{12}$$

$$Z_{ijp}-Z_{ijq}\geqslant -2+(R_{ipq}+R_{jqp})\qquad\forall f_i,f_j\in F;\forall n_p,n_q\in N \tag{13}$$

Eq.(6) is the objective function at the lower level, where $T_{in_1^i}$ is the time of aircraft f_i arriving at the first node, and $T_{in_{k_i}^i}$ the time of aircraft f_i arriving at the last node. Eq.(7) transfers the start time of taxiing at the upper level as a constraint at the lower level. For each aircraft, Eq.(8) ensures the taxi time from one node to another is shorter than or equal to the time difference between their arrivings, which reflects the constraint of the continuous operation. Eq.(9) is used to detect conflicts between two aircraft in any node or taxiway chain, in which τ_{ijp} is the time-equivalent safety interval and $\tau_{ijp}=(T_{iq}-T_{ip})/L_{pq}\cdot d_{\mathrm{sep}}$. Eqs.(10—13) are used to detect rear-end conflicts and head-on conflicts. Any dissatisfaction for the inequalities in Eqs.(9—13) will lead to an increasing number of conflict points δ.

2 Iterative Solution Method

To solve the bilevel programming model, the first step is to initialize the decision variable at the upper level. Each aircraft is randomly given an initial solution within reasonable ranges, that is, the start time of taxiing for both departure and arrival aircraft is provided. Then at the lower level, within the constraints of this parameter, proper taxi routes are scheduled to optimize the objective function. The solution is fed to the upper level. Afterwards, the upper level makes the better decision once again based on the optimized solution from the lower level and returns the result to the lower level. This process will be repeated till the stopping criterion is fulfilled. Fig.2 illustrates it in detail.

As the genetic algorithm (GA) has the advantages of less computation, rapid convergence and few parameters, this paper imitates the iterative and evolutionary process of it. A solution to ground

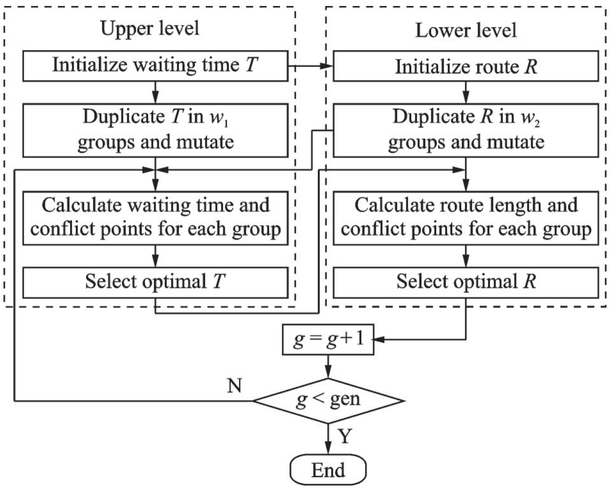

Fig.2 Flow chart of solving process

scheduling problem based on bilevel programming is shown as follows.

Step 1 Initialization. Suppose generation $g=0$, and the waiting time at the upper level is randomly assigned within reasonable ranges. Then the total waiting time T_{D} is calculated and the waiting time solution $\{T_1, \cdots, T_i, \cdots, T_n\}$ and T_{D} are passed to the lower-level programming. To initiate taxi routes at the lower level, routes R_i are randomly selected for each aircraft under the constraints of waiting time $\{T_1, \cdots, T_i, \cdots, T_n\}$, and then the taxi routes solution $\{R_1, \cdots, R_i, \cdots, R_n\}$ are formed. Finally, the total travel distance L is calculated.

Step 2 The time arriving at each node in the selected route is calculated and the number of conflicts δ is detected through constraints (9—13). The routes solution, total travel distance and the number of conflicts are then handed over to the upper level.

Step 3 Suppose generation $g=g+1$. The upper level duplicates the waiting time solution in w groups and adjust the time in each group by introducing a random number $\tau \in (-10, 10)$. Eq.(14) limits the variation range of the waiting time. The waiting time T_i of each aircraft f_i will mutate with a specified probability σ

$$T_i = \begin{cases} 0 & T_i + \tau < 0 \\ T_i + \tau & \text{Other} \\ \text{MaxDelay} & T_i + \tau > \text{MaxDelay} \end{cases} \tag{14}$$

Step 4 The total waiting time T_{D}^w is calculated for each group. Then, based on the time solution, as well as combined with latest route solution, the upper level calculates the time arriving at each node in the route and detects the number of conflicts δ^w.

Step 5 Among the duplicated groups in which $T_{\mathrm{D}}^w \leqslant T_{\mathrm{D}}$ and $\delta^w \leqslant \delta$, the optimal group is selected as the new waiting time solution at the upper level. The total waiting time and the number of conflicts in this group are passed to the lower level.

Step 6 The lower level duplicates the taxi routes solution in w groups and adjusts each route with a specified probability σ. In order to reduce the computing complexity, the alternative taxi routes are selected from a predetermined feasible route set.

Step 7 The lower level calculates the total travel distance L^w for each group. Then, the time arriving at each node in the selected route are obtained based on each route solution and relations with the waiting time from the upper level. Through this approach, the number of conflicts δ^w is detected.

Step 8 Among the duplicated groups in which $L^w \leqslant L$ and $\delta^w \leqslant \delta$, the lower level selects the optimal group as its latest taxi route solution. Similarly, the routes, the total travel distance and the number of conflicts are returned to the upper level.

Step 9 If $g \geqslant \text{gen}$, where "gen" means the times of generation, the loop ends; otherwise goes back to Step 3.

Step 10 If the number of conflicts $\delta=0$ at this time, the solutions to taxi routes and waiting time are taken as the final result of the bilevel programming. Otherwise the solving process fails.

3 Computational Test

In this paper, the ground movement of Nanjing Lukou Airport (NKG) is chosen as a studying objective. We select part of the airport and abstract several adjacent stands as three integrated aprons G1, G2 and G3. After omitting some irrelevant taxiways and nodes, the taxiway layout is drawn with 37 nodes, 48 edges, two runways and three integrated aprons, as shown in Fig.3, where TML means terminal.

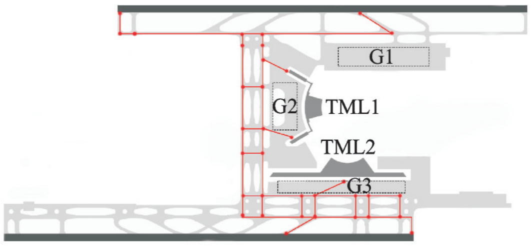

Fig.3 Layout of NKG taxiway network layout

The flight schedule data are chosen between 8:00 and 8:15 during rushing hours, as listed in Table 2. The estimated pushback and landing time of departure and arrival flights are converted into the start time of taxiing. The estimated departure time is then converted into 5 min after starting taxiing. σ is set to be 0.4 and w is set to be 100.

Table 2 Aircraft timetable

Aircraft No.	Earliest start time	Source	Destination	Arrival/ departure
1	08:00:00	32	37	A
2	08:00:00	35	33	D
3	08:02:00	36	31	D
4	08:02:00	37	31	D
5	08:04:00	36	33	D
6	08:05:00	34	35	A
7	08:06:00	32	36	A
8	08:06:00	34	36	A
9	08:08:00	32	37	A
10	08:08:00	35	33	D
11	08:10:00	32	35	A
12	08:11:00	34	35	A
13	08:12:00	36	31	A
14	08:14:00	34	37	A
15	08:15:00	35	33	D
16	08:15:00	32	36	A

4 Result Analysis

4.1 Simulation results

Simulation is conducted on MATLAB software. We obtain the optimal solution $Z_1^*=186$ s at the upper level, and $Z_2^*=43\,250$ m at the lower level. The evolutionary processes of optimal solutions at both upper and lower levels are illustrated in Fig.4. For the curve of the upper level programming (ULP), we can find that with the number of itera-

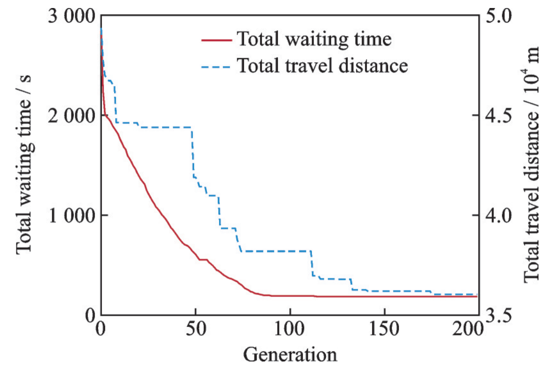

Fig.4 Optimal solution evolution process at the upper and the lower levels

tions increasing, the value of total waiting time declines from the large initial value. The first five generations have a rapid descent as the initial solution has been randomly given a large value, resulting in a higher number of conflicts and a function value. Then, the total waiting time keeps going down as iteration moves forward. The total waiting time starts leveling off since the 90th generation, reaching the optimal solution within the maximum number of iterations.

From the evolutionary curve of the lower level programming (LLP), the total travel distance shows a stepped downward trend as iteration goes further. The value decreases generally faster during the first 100 generations. This phase gains better evolutionary efficiency and a faster decrease of the total length. Whereas only minor reductions appear during the remaining, which is probably related to the increasing difficulty of searching for a better solution as iteration moves forward.

Fig.5 illustrates the evolutionary process of the number of conflicts and the relevant LLP object function value during the 200 generations. As shown in Fig.5, there are six conflict points initially. The number of conflicts decreases to zero until the 113th generation. The result corresponds with the general evolutionary trend of the object function value. Since conflicts are strictly prohibited in the designed ground movement model, only conflicts-free solutions can be considered as feasible, i.e., solutions after the 113th generation. We can conclude

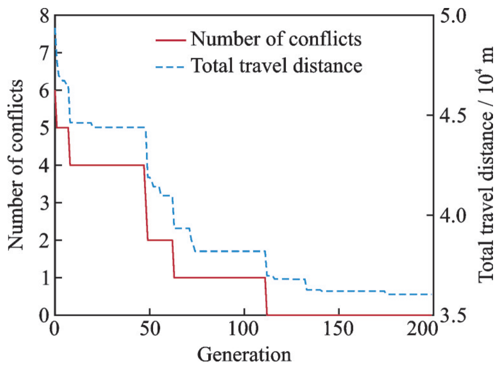

Fig.5 Evolution process of conflict points and total travel distance

that it is adequate to set 200 as the maximal number of iterations as the number of conflicts stalls for over 80 iterations. The setting ensures computational efficiency and the acquirement of stable optimal solutions.

Fig.6 illustrates the taxiing and the waiting time of each aircraft in the optimal result. It is indicated from the travel distance that all the aircraft successfully taxi to their assigned destinations and have balanced distances. Every waiting aircraft calls for rather short waiting time, which is within the acceptable limit. Only No.4 and No.6 need to wait for 106 s and 80 s, respectively, accounting for 12.5% of the total. The flight schedule only suffers a minor disturbance. It is proved by simulation results that the total time can be reduced substantially at the cost of a little extra waiting time. It helps to improve taxiing efficiency and lower operation cost.

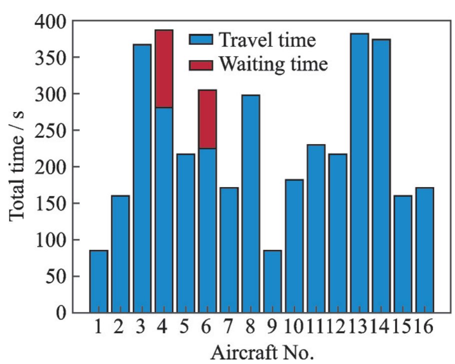

Fig.6 Total time of the aircraft

We perform simulation by the first-come-first-service (FCFS) algorithm on the same flight date. Table 3 compares the simulation results.

Table 3 Result comparison between the FCFS scheme and the bilevel programming

Parameter	FCFS	Bilevel programming
Total travel distance/m	35 330	36 050
Avg. travel distance/m	2 208.1	2 253.1
Total waiting time/s	0	186
Avg. waiting time/s	0	11.6
Number of conflicts	9	0
Avg. operational time/s	254.6	236.9

In Table 3, for the FCFS scheme, the total travel distance of the 16 aircraft is 35 330 m, and 2 208.1 m on average. As aircraft are operated according to the flight plan, there is no waiting time. In contrast, for the bilevel programming, the total travel distance increases by 2.0% to 36 050 m, and 2 253.1 m on average. The total waiting time is 186 s, and 11.6 s per aircraft, which are acceptable. The intention of avoiding conflicts inevitably leads to the presence of waiting time in the bilevel programming. As a consequence, the taxiing efficiency and the safety performance are improved, and there is no conflict. Although the FCFS scheduling can avoid waiting when the aircraft pushes back, the total wasted time is much higher than that of the bilevel programming, as more time is spent on avoiding conflicts on the taxiway. It is anticipated conservatively that each conflict takes 30 s to be resolved. Given the assumption that the aircraft moves on at a constant speed of 10 m/s, the operational time is the sum of the travel time, the waiting time and the time for solving the conflicts. It turns out the bilevel programming surpasses the FCFS scheme in terms of the average operational time, as illustrated in Table 3.

Fig. 7 further compares the operational time of each aircraft in these two methods. Scatters above the dashed line (slope equals to 1) represents the aircraft that have a longer operational time in FCFS than in the bilevel programming, or vice versa. The operational time in the bilevel programming is generally shown to be shorter due to its significant improvement in avoiding conflicts. Only two aircraft

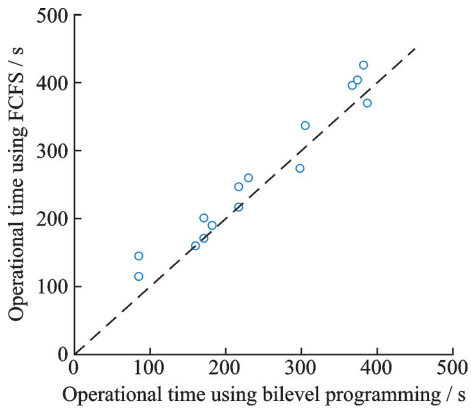

Fig.7 Aircraft operational time comparison between FCFS and the bilevel programming

have an increased operational time after optimization. Alhough it may need further investigation afterwards, a possible explanation would be that their taxi routes are relatively distant from the main traffic flow on the ground, by which they are less likely to conflict with other moving aircraft.

From the above comparisons, we can conclude that the bilevel programming outperforms the FCFS scheduling in terms of the total travel distance, the taxi time and the number of conflicts. This is the because the bilevel programming regularly feedbacks the routes and the waiting time, optimizing the taxi routes as well as the reducing taxi conflicts in both the spatial and the temporal ways. For example, spatially, the aircraft selects a relatively short route and taxies to the destination without passing by congested areas; or temporally, the aircraft chooses the shorter waiting time and then starts to taxi after conflicts. In contrast, the FCFS scheduling is unable to select shorter routes, or to avoid conflicts except waiting. In conclusion, the bilevel programming is superior to the FCFS scheduling in terms of the total travel distance, the taxi time and the number of conflicts.

4.2 Analysis of heuristic parameters

Looking at the solving method in the bilevel programming, the value of mutation probability σ and the number of duplications w have a significant impact on the program execution and results. The following part analyses their impacts on the simulation results.

These two parameters are studied through controlling variables. First, we hold the number of duplications $w=100$ and change the value of mutation probability from 0.2, 0.4 to 0.6. Ensuring all other conditions being equal, we repeat the simulation test and average the test values. Some representative simulation results are chosen for the analysis, as shown in Table 4.

Table 4 Impacts of mutation probability

Mutation probability σ	Avg. travel distance/m	Avg. waiting time/s	Program executing time/s	Avg. iterations for the optimum
0.2	2 264.7	11.9	419	180
0.4	2 253.1	11.6	412	113
0.6	2 312.5	12.2	406	10

In Table 4, the results are very close when the mutation probability equals to 0.2 and 0.4, while the average travel distance and the average waiting time increase to some extent as the mutation probability rises to 0.6. The program executing time is very similar among these three test groups, which is acceptable. Whereas the most remarkable impact of the mutation probability on this method is convergence. When the mutation probability is 0.2, the algorithm presents a slow convergence. It is not until the 180th generation that the optimal solution is found, or even no conflicts-free solutions can be acquired at the end. When the mutation probability is 0.6, the optimal solution appears at the 10th generation, while too fast convergence tends to be trapped into local optimum and lacks stability. Finally, the algorithm convergence lingers when the mutation probability is 0.4. It has a moderate convergence while the speed is not too slow to find a satisfactory optimal solution within the number of iterations set before. From the above analysis, the result is better when the mutation probability equals to 0.2 or 0.4, rather than 0.6.

Multiple test results illustrate that when the mutation probability is too small, the program will have a slow convergence, or even no conflicts-free solutions can be acquired within the maximum generation set in advance. This is caused by a lower possibility of generating new excellent individuals through mutation. Alternatively, the crossover operation within the population can hardly produce more good individuals, leading to the algorithm's slowness of convergence and weakness in solving ability. In addition, a too large mutation probability may enhance the randomness of individuals within the whole population, resulting in early convergence and local optimum. The movement plan is not fully optimized as well.

Similarly, we hold the mutation probability $\sigma = 0.4$ and assign w with 50, 100, and 200. As all other conditions being equal, we repeat the simulation tests and average the test values. The results are shown in Table 5.

Table 5 Impacts of number of duplications

Number of duplications w	Avg. travel distance/m	Avg. waiting time/s	Program executing time/s
50	2 230.6	107.1	226
100	2 253.1	11.6	412
200	2 218.0	11.0	812

In Table 5, the aircraft's average travel distance and average waiting time show a downward trend with the number of duplications increasing. Additionally, the program executing time has a positive correlation with the number of duplications and increase linearly.

The increase of the number of duplications enlarges the population scale, so better solutions are more likely to be acquired. The aircraft's average travel distance tends to decrease with the rise of the number of duplications. However, it does not mean "bigger is better" due to the demerits of larger data size and more time spent on computation. So often it takes a fair amount of time to make a little progress. In practical use, it is recommended to select a value in the middle after several tests and comparisons, and to find the optimal results with acceptable executing time.

5 Conclusions

This paper studies optimization of GMPs and discusses the integration of both routing and scheduling. By introducing the bilevel programming, an optimization model is established. An iterative heuristic algorithm is designed for the model. Simulation results indicate that the solving process has good convergence and executing efficiency, and is able to optimize taxi routes and waiting time simultaneously. As the application of the bilevel programming to GMPs is rarely studied at present, research results in this paper can be theory guidance to its future development. The solving process of the bilevel programming is a complete NP-hard problem. When the scale of the problem is enlarged, computational time will significantly increases as well. So the algorithm used in this paper may not be capable of solving the problem anymore. In conclusion, research direction in the future should focus on finding an efficient globally optimal algorithm for the large-scale bilevel programming.

Reference

[1] EVERTSE C, VISSER H G. Real-time airport surface movement planning: Minimizing aircraft emissions[J]. Transportation Research Part C: Emerging Technologies, 2017, 79(3): 224-241.

[2] JIANG Y, XU X, ZHANG H, et al. Taxiing route scheduling between taxiway and runway in hub airport[J]. Mathematical Problems in Engineering, 2015, 2015: 1-14.

[3] CLARE G L, RICHARDS A G. Optimization of taxiway routing and runway scheduling[J]. IEEE Transactions on Intelligent Transportation Systems, 2011, 12(4): 1000-1013.

[4] YU C, LAU, H Y K. Integrated optimization of airport taxiway and runway scheduling[J]. Journal of Automation and Control Engineering, 2014, 2(4): 338-342.

[5] BROWNLEE A E I, WEISZER M, CHEN J, et al. A fuzzy approach to addressing uncertainty in airport ground movement optimisation[J]. Transportation Research Part C: Emerging Technologies, 2018, 92(5): 150-175.

[6] ATKIN J, BURKE E K, RAVIZZA S. The airport ground movement problem: Past and current research and future directions[C]//Proceedings of the 4th International Conference on Research in Air Transportation (ICRAT 2010). Budapest, Hungary: [s. n.], 2010: 1-8.

[7] MARIN A, CODINA E. Network design: Taxi planning[J]. Annals of Operations Research, 2008, 157(1): 135-151.

[8] SMELTINK J W, SOOMER M J, WAAL P R D, et al. An optimization model for airport taxi scheduling[C]//Proceedings of the INFORMS Annual Meeting. Denver, USA:ICAS, 2004:1-14.

[9] SAMA M, D'ARIANO A, TOLI A, et al. A variable neighborhood search for optimal scheduling and routing of take-off and landing aircraft[C]// Proceedings of International Conference on MODELS and Technologies for Intelligent Transportation Systems. Budapest, Hungary: IEEE, 2015: 491-498.

[10] GUEPET J, BRIANT O, GAYON J P, et al. The aircraft ground routing problem: Analysis of industry punctuality indicators in a sustainable perspective[J]. European Journal of Operational Research, 2016, 248(3): 827-839.

[11] BALAKRISHNAN H, JUNG Y. A framework for coordinated surface operations planning at Dallas-Fort Worth International Airport[C]//Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit. Hilton Head, USA: AIAA, 2007.

[12] NOGUEIRA K B, AGUIAR P H C, LI W. Using ant algorithm to arrange taxiway sequencing in airport[J]. International journal of computer theory & engineering, 2014, 6(4): 857-361.

[13] WEISZER M, CHEN J, LOCATELLI G. An integrated optimisation approach to airport ground operations to foster sustainability in the aviation sector[J]. Applied Energy, 2015, 157: 567-582.

[14] JIANG Y, LIAO Z, ZHANG H. A collaborative optimization model for ground taxi based on aircraft priority[J]. Mathematical Problems in Engineering, 2013, 2013: 1-9.

[15] DEAU R, GOTTELAND J B, DURAND N. Airport surface management and runway scheduling[C]//Proceedings of the 8th USA/Europe Air Traffic Management R&D Seminar. [S. l.] : [s. n.] , 2009:1-9.

[16] SAMA M, D'ARIANO A, D'ARIANO P, et al. Optimal aircraft scheduling and routing at a terminal control area during disturbances[J]. Transportation Research Part C: Emerging Technologies, 2014, 47(9): 61-85.

[17] ALONSO-AYUSO A, ESCUDERO L F, MARTIN F J, et al. A VNS metaheuristic for solving the aircraft conflict detection and resolution problem by performing turn changes[J]. Journal of Global Optimization, 2014, 63(3): 583-596.

[18] CHEN J, WEISZER M, LOCATELLI G, et al. Toward a more realistic, cost-effective, and greener ground movement through active routing: A multiobjective shortest path approach[J]. IEEE Transactions on Intelligent Transportation Systems, 2016, 17(12): 3524-3540.

[19] LEE H, BALAKRISHNAN H. A comparison of two optimization approaches for airport taxiway and runway scheduling[C]//Proceedings of Digital Avionics Systems Conference. Williamsburg, USA: IEEE, 2012: 4E1-1-4E1-15.

[20] BENLIC U, BROWNLEE A E I, BURKE E K. Heuristic search for the coupled runway sequencing and taxiway routing problem[J]. Transportation Research Part C: Emerging Technologies, 2016, 71(9): 333-355.

[21] ADACHER L, FLAMINI M, ROMANO E. Airport ground movement problem: Minimization of delay and pollution emission[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 99(1): 1-10.

[22] BOSSON C S, SUN D. Optimization of airport surface operations under uncertainty[J]. Journal of Air Transportation, 2016, 24(3): 84-92.

[23] ZHANG T, DING M, ZUO H, et al. An online speed profile generation approach for efficient airport ground movement[J]. Transportation Research Part C: Emerging Technologies, 2018, 93(5): 256-272.

[24] ZHANG T, WEISZER M, CHEN J. The feasibility of follow-the-greens for 4-dimensional trajectory based airport ground movements[J]. Transportation Research Part C: Emerging Technologies, 2020, 116: 102632.

[25] ANDERSON R, MILUTINOVIC D. An approach to optimization of airport taxiway scheduling and traversal under uncertainty[J]. Journal of Aerospace Engineering, 2013, 227(2): 273-284.

[26] BRACKEN J, MCGILL J. Mathematical programs with optimization problem in the constraints[J]. Operations Research, 1973, 21(1): 37-44.

[27] GAO Z Y, ZHANG H Z, SUN H J. Bi-level programming models, approaches and applications in urban transportation network design problems[J]. Communication & Transportati on Systems Engineering & Information, 2004, 4(1): 35-44.

[28] LIU T, AVISHAI A C. Integrated public transport timetable synchronization and vehicle scheduling with demand assignment: A bi-objective bi-level model using deficit function approach[J]. Transportation re-

search procedia, 2017, 23: 341-361.

[29] FONTAINE P, MINNER S. Benders decomposition for the hazmat transport network design problem[J]. European Journal of Operational Research, 2018, 267 (3): 996-1002.

Acknowledgements This work was supported by the National Natural Science Foundations of China (Nos. U1933118, U2033205).

Author Dr. **JIANG Yu** received her B.S. and Ph.D. degrees in transportation planning and management from Jilin University and Southeast University in 2000 and 2007, respectively. She is currently an associate professor in the College of Civil Aviation of Nanjing University of Aeronautics and Astronautics. Her research interest includes airport design and planning, airport operation and optimization, big data analytics in airport operation.

Author contributions Dr. **JIANG Yu** designed the study, complied the models, conducted the analysis, interpreted the results and wrote the manuscript. Mr. **HU Zhitao** wrote the code and conducted the simulation. Mr. **LIU Zhenyu** and Miss **ZHANG Haoyu** contributed to the simulation results analysis. Mr. **WANG Zhe** contributed to the discussion and background of the study. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: ZHANG Bei)

机场场面运行优化问题的双层规划方法

姜 雨[1], 胡志韬[1], 刘振宇[1], 张皓瑜[1], 王 喆[2]

(1.南京航空航天大学民航学院, 南京 211106, 中国; 2.北京首都国际机场股份有限公司, 北京 100621, 中国)

摘要: 本文提出了基于双层规划的机场场面运行(Ground movement problem, GMP)优化模型,以解决机场场面航空器滑行冲突,提高运行安全性和效率。该模型的特点是改变了以往独立的研究,对航空器路径选择和时序安排进行协同优化。设计了一种迭代启发式算法对问题进行求解。以南京禄口国际机场为对象进行了仿真试验,结果显示,在保证无冲突的前提下,本文提出的双层模型在航空器滑行时间方面明显优于目前广泛使用的先到先服务(First-come-first-service, FCFS)调度方案。研究结果表明,该优化模型能够降低机场运营成本,提高整体运行效率,对达到极限容量并面临需求持续增长的机场具有重要的决策支持作用。

关键词: 机场场面运行;航空器路由调度;双层规划;迭代启发式;航空运输