



HAL
open science

A New Multiobjective A \star Algorithm With Time Window Applied to Large Airports

Bosheng Ba, Ye Yu, Ruixin Wang, Jean-Baptiste Gotteland, Yunqi Gao

► **To cite this version:**

Bosheng Ba, Ye Yu, Ruixin Wang, Jean-Baptiste Gotteland, Yunqi Gao. A New Multiobjective A \star Algorithm With Time Window Applied to Large Airports. *Journal of Advanced Transportation*, 2024, 2024, 10.1155/atr/7536217. hal-04934065

HAL Id: hal-04934065

<https://enac.hal.science/hal-04934065v1>

Submitted on 7 Feb 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Research Article

A New Multiobjective A^* Algorithm With Time Window Applied to Large Airports

Bosheng Ba ¹, Ye Yu,² Ruixin Wang ¹, Jean-Baptiste Gotteland,³ and Yunqi Gao ⁴

¹CAUC-ENAC Joint Research Center of Applied Mathematics for Air Traffic Management, Sino-European Institute of Aviation Engineering, Civil Aviation University of China, Tianjin, China

²Consulting Department II, China Civil Aviation Engineering Consulting Co., Ltd., Beijing, China

³ENAC Lab, 7 Avenue Edouard Belin, BP 31055, Toulouse Cedex 4, France

⁴Hangzhou International Innovation Institute, Beihang University, Hangzhou 311115, China

Correspondence should be addressed to Ruixin Wang; ruixin.wang@recherche.enac.fr

Received 29 May 2024; Accepted 22 October 2024

Academic Editor: Indrajit Ghosh

Copyright © 2024 Bosheng Ba et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Current airport ground operations, relying on single and fixed aircraft taxiing rules, struggle to handle dynamic traffic flow changes during peak flight times at large airports. This leads to inefficient taxiing routes, prolonged taxiing times, and high fuel consumption. This paper addresses these issues by proposing a new adaptive method for dynamic taxiway routing in airport ground operations. This method aims to reduce ground taxiing time and fuel consumption while ensuring the safety of aircraft taxiing. This study proposes a multiobjective A^* algorithm with time windows which takes into account the allocation of resources on airport taxiways and introduces factors such as turning angles, dynamic turning speeds, and dynamic characteristics of the ground operations. Experiments conducted over the 10 busiest days in the history of Tianjin Binhai International Airport demonstrate that the algorithm excels in minimizing total taxiing time, differing only by 0.5% from the optimal solution. It also optimizes multiple objectives such as fuel consumption and operates at a solving speed approximately three orders of magnitude faster than the optimal solution algorithm, enabling real-time calculation of aircraft taxiing paths. The results of the study indicate that the proposed multiobjective A^* algorithm with time windows can effectively provide decision support for dynamic routing in airport ground operations.

Keywords: airport ground movement; dynamic routing; multiobjective A^*

1. Introduction

The civil aviation transportation system is complex, encompassing numerous elements and intricate influencing factors. Within this system, airports are not only the primary supporting structures but also the key hubs connecting air and ground transportation. Given that the limited capacity of airport infrastructure may constrain the growth of air transport and its economic contribution, the development of more efficient airport operation decision support systems becomes particularly urgent.

Airport Ground Movement (AGM) is a core component of airport operations, responsible for planning the route of

aircraft from the gate to the runway, and closely related to other key subsystems of airport operations, such as runway scheduling and apron allocation. Furthermore, AGM issues are also considered a key challenge in the planning of aircraft paths within the airport. Thus, the optimization of AGM is crucial for enhancing airport transportation efficiency. Currently, air traffic controllers arrange routes for aircraft in a predetermined order, as shown in Figure 1, where each path from the apron to the runway is preset. Although this rule simplifies the controllers' monitoring of intersecting paths (i.e., "hot-point"), reduces their workload, and helps ensure flight safety, it leads to unnecessary increases in aircraft taxiing distance and time due to the artificial

lengthening of paths to reduce the number of hot-plots. This not only affects the economy and environmental friendliness but also reduces airport operational efficiency. To address this issue, introducing efficient decision support systems like A-CDM (Airport Collaborative Decision Making) is particularly important [1]. A-CDM aims to ensure the effective use of airport resources by intelligently optimizing ground movement and takeoff processes, thereby reducing aircraft taxiing and waiting times and enhancing airport operational efficiency. Under the fixed taxiway path rule, the general extension of taxi paths does not meet the requirements for economy and environmental protection. Besides relying on advanced intelligent decision support systems, adopting the shortest path taxiing rule is also a viable solution. However, during peak airport operating periods, this rule may lead to the repeated use of certain taxiways, increasing the frequency of hotspots on the field. More seriously, as shown in Figure 2, there may be irreconcilable conflicts caused by two aircraft taxiing towards each other on the same taxiway.

The existing two routing rules have not been able to fully adapt to the changes in airport dynamic flow, prompting us to seek more intelligent decision-making methods. This method should regard flights as obstacles to each other and fully consider the overall real-time operational status of the airport in order to plan aircraft paths. In this context, dynamic routing algorithms, which aim to calculate the optimal path from the starting point to the destination based on changes in environmental conditions or states, appear particularly suitable for addressing the challenges of AGM.

This study employs a dynamic routing method to solve the challenges of planning aircraft routes on airport grounds. By optimizing flight routes, it not only reduces the frequency of conflicts during taxiing but also enhances airport operational efficiency and the utilization efficiency of airport infrastructure. Furthermore, considering the sustainability requirements of airport operations, this study also delves into the impact of ground transportation costs. As reducing fuel consumption becomes a common goal among stakeholders such as airports and airlines, this study applies a multiobjective optimization method that not only reduces aircraft taxiing distance but also shortens the time aircraft spend waiting and accelerating or decelerating, achieving the goals of minimizing taxiing time and fuel consumption.

The structure of this paper is organized as follows: Section 2 defines and introduces the AGM problem and related research; Section 3 elaborates on the CEPO algorithm and time-window multiobjective A* (TMOA*) algorithm adopted; Section 4 presents the experimental results of the proposed algorithms in a real data case at Tianjin Binhai International Airport; finally, Section 5 summarizes the research findings and looks forward to future research directions.

2. Recent Research

AGM involves complex routing and scheduling issues, with the goal of planning an effective route from the starting point to the destination for each aircraft while avoiding conflicts with the paths of other aircraft [2]. Currently, most

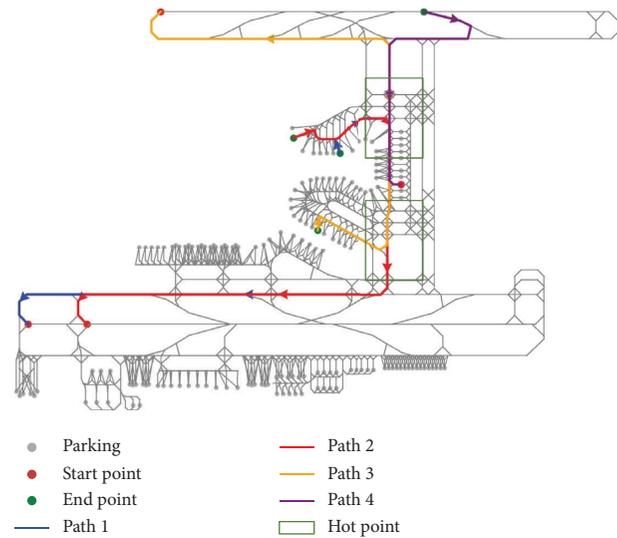


FIGURE 1: Example of hot point area.

algorithms and software used for airport surface routing optimization are based on traditional static routing algorithm, which means the path is predetermined before the aircraft begins taxiing and remains unchanged throughout the process, such as Dijkstra and A* algorithms [3, 4]. The dynamic routing method used in this paper is different in that it considers the current usage of airport taxiways as well as the movement of other aircraft, and it calculates the optimal route in real-time. This method can adapt to congestion or other unexpected situations during taxiing to quickly update the path, ensuring that all aircraft can safely reach their destinations in a short time [5]. Therefore, applying dynamic routing, especially dynamic routing technology with automatic obstacle avoidance capabilities, to airport surface route planning has a certain significance in improving airport operational efficiency.

The current mainstream method for implementing dynamic routing is to use routing algorithms with time windows, such as the new QPPTW algorithm (Quick Path Planning with Time Windows) introduced by Ravizza, Atkin, and Burke, based on the Dijkstra algorithm with time windows. It updates the entire graph's time windows according to the path information of each preceding aircraft or the network environment information, achieving conflict prediction and resolution [6]. While addressing dynamic issues, the explored objectives related to economic benefits such as fuel consumption and introduced sequential method based on ranking approaches [7], representing this problem as k-QPPTW [8]. By solving for the top k optimal paths and to find a more acceptable solution, the drawback of this method is the selection of k and whether the solution results are on the Pareto front.

Similarly, in facing real-world multiobjective problems, the development of the A* algorithm has certain advantages. Some researchers evolved the static A* algorithm into a multiobjective A* (MOA*) solving algorithm [9], and after analysis and discussion of the heuristic function in MOA* by Tung and Chew, the structure of MOA* became more

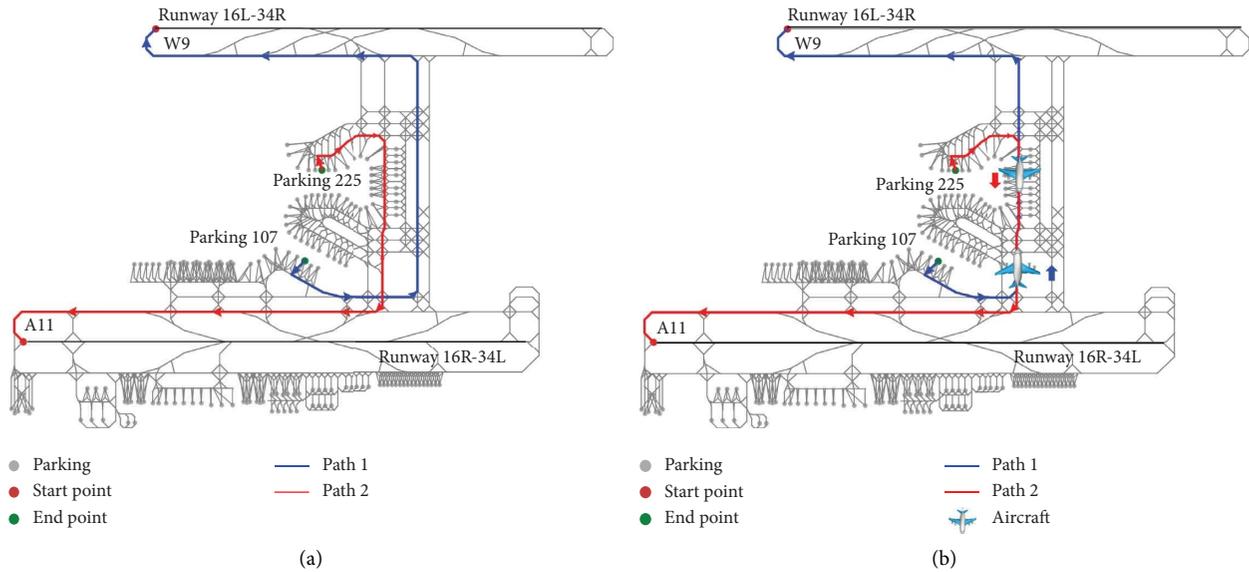


FIGURE 2: Example of two different taxiing rules. (a) Rule of fixing taxi rules path. (b) The shortest path rule.

mature [10]; eventually, improvements in the path expansion method within MOA*, enhanced the practical application value of MOA*, which was developed by Weiszer and others into a multiobjective routing and scheduling algorithm based on enumeration methods, AMOA*, aimed at solving the multiobjective, multigraph issues of such airports [11, 12]. Results using the proposed algorithm at a series of international airports were also provided. Compared to other routing and scheduling algorithms, under the constraints of objective functions considering taxi path time and fuel when the aircraft sequence is fixed, this algorithm can find a set of representative optimal or near-optimal solutions in a single run [13]. AMOA* algorithm mainly discusses the relationship between fuel consumption and taxi time under different speed curves during aircraft taxiing at airports, incorporating the concept of time windows from Ravizza's QPPTW algorithm. Although it lacks a detailed discussion on conflict avoidance and has shortcomings in dynamic problem solving, it still considers the time intervals in airport surface routing issues.

A novel dynamic path planning algorithm proposed by Hu et al.'s, the CEPO algorithm based on the Ripple Spreading Algorithm (RSA), employs a concept of co-evolutionary real-time updating [14]. Unlike traditional time window methods, the network information in the CEPO algorithm changes over time, and by predicting the path of obstacles in one go, it achieves an efficient obstacle avoidance function. This feature not only proves the algorithm's optimality in single-objective optimization problems but also ensures its effectiveness in improving resource utilization and aircraft taxiing safety.

While multiobjective and single-objective dynamic routing algorithms can theoretically solve temporal conflict issues, their simulation and solving processes are usually based on ideal network environments. For instance, the CEPO algorithm has only been tested in idealized networks

and has not yet been applied in real airport surface operation. In reality, airport networks are subject to numerous restrictions and rules, making the taxiing process from parking stands to runways far from smooth, potentially involving pushbacks, waiting, and detouring. Currently, realistic simulation of the maximum allowable speed on taxiways, turning angles, and their maximum allowable turning speeds has not been conducted. Especially during high traffic periods, airports face greater usage pressure: with an increased number of aircraft needing to maintain safe intervals, which limits the choice of paths and may lead to waiting or detours. Therefore, the application of dynamic routing algorithms, by providing safe and efficient routes through real-time calculations, is particularly necessary to save time and reduce fuel consumption [2].

This study aims to address the deficiencies of previous research by exploring how to solve the problem of airport ground routing through the limited resources of runways and taxiways, combined with multiobjective and dynamic routing algorithms. The goal is to optimize the routes for aircraft to support the rapid development of the civil aviation transport industry. This is not only of great significance for improving the operational efficiency of current airports but also has an important impact on the planning of new airports and the promotion of energy saving and emission reduction strategies [15, 16].

3. Problem Description

The set of aircraft is composed of the arrival group *ARR* and the departure group *DEP*. Each aircraft flight is equipped with a start time t , a start node s , an end node e , and its belonging weight category w_i , divided into light, medium, and heavy. The weight category of aircraft is used to assign departure flights' takeoff runway starting point and parking stand for arrival flights. The goal of airport routing is to

develop a route from s to e for all aircraft $ight \in \{ARR, DEP\}$, starting at t , and meeting the time window requirements.

The airport can be viewed as a graph structure $G = (N, E)$, where each node n in N represents key points in the airport, including parking stands, runway access and exit points, and points where taxiways intersect. Each edge e in E represents a route between two nodes, which includes three main types: taxiway, pushback way, and runway. The pushback way, as illustrated in Figure 3, is a specific path in the airport used for the aircraft pushback process, which can be divided into four main stages. When an aircraft is preparing to move toward the left side of the diagram, the parking stand directly opposite the terminal building will first conduct a pushback using a tug, pushing the aircraft onto the predetermined path. Once the tug pushes the aircraft onto the taxiway, the pushback process is completed and then the aircraft can taxi in the left direction as shown in Figure 3.

Ordinary taxiways can be divided into one-way and two-way paths according to their direction of use. One-way paths are designed for a specific direction, similar to pushback paths; it allows aircraft to only leave the parking stand and prohibiting entry to the parking stand through this path. Therefore, even if all taxiways in the field are available, the routes for aircraft taking off and landing from the same parking stand to a specific runway takeoff point will differ. This indicates that one-way paths significantly impact the final choice of routes for aircraft, so we need to consider the one-way and two-way problems of the path when building the network.

When executing routing on the apron, a set of taxiing rules must be established to avoid conflicts. There are mainly three types of conflicts in apron operations. As shown in Figure 4, the first scenario involves two aircraft attempting to enter the same taxiway simultaneously, for example, aircraft A preparing to go straight while aircraft B plans to turn left; the second scenario is when aircraft A and B taxi in the same direction but fail to maintain a safe distance of more than 60 m; the third and most severe conflict occurs when aircraft A and B taxi towards each other. Since aircraft cannot reverse, this situation must be completely avoided.

Based on the types of conflicts and guidance from airport control center, the basic taxiway planning rules established in this study are as follows:

1. Each taxiway portion is only permitted to have one aircraft taxiing on it simultaneously.
2. Aircraft must maintain a minimum safe distance of 60 m, which can be adjusted by updating the network's time window. Each taxiway is assigned a specific allowable passage time period, defined as a time window. When an aircraft begins a taxi segment, the entire network's time window information is updated, blocking adjacent taxiways according to the 60 m safety interval. Under a set order of aircraft taxiing, the time window setting can prevent conflict types (a) and (b), as shown in Figures 5(a) and 5(b). For conflict (a), the lower priority aircraft must wait until the

higher-priority aircraft leaves the conflict zone before it can taxi. Priority is determined by the takeoff and landing order given by air traffic controllers. For example, if A has a higher priority than B, A will go straight first, and B will wait until A leaves before continuing with its taxiing plan. For conflict (b), if the distance between two aircraft is less than 60 m, it is resolved by adjusting taxiing speeds. By setting a safe interval and preblocking the taxiway behind aircraft B, such conflicts can be prevented. However, conflict (c), as a three-dimensional problem, cannot be completely solved by just increasing the time interval, and specific details will be discussed in Chapter 4.

3. Restriction on turning angles, aircraft are not allowed to make turns greater than 90° : Given the complexity of airport structures, the presence of turns in routes is inevitable. To ensure the routes calculated by the algorithm are practically feasible, the determination of angles between taxiways must be strictly enforced, and restrictions on turning angles applied. The three types of turning angles shown below, with Figures 6(a) and 6(b) displaying the settings for turning angles on straight taxiways. According to this study, the angle between the direction of the aircraft's nose along the taxiway and the taxiway itself is defined as the entry angle α , while β is defined as the angle of the next turn after turning at angle α . In practice, the curves in the airport model consist of multiple short segments, as shown in Figures 6(c) and 6(d), where the angle α between the aircraft's nose and the tangent of the first segment represents the turning angle. The primary restriction on turning angles in airports is that they must not exceed 90° , hence for the scenarios in Figures 6(b) and 6(d), the aircraft cannot turn at angle α . In addition to the three taxiway routing rules previously mentioned, aircraft undergo various states such as acceleration, deceleration, and constant speed during taxiing, resulting in changes in taxiing speed.

Given the relatively low speed of aircraft taxiing on the ground, the impact of short periods of acceleration and deceleration on taxi time and costs such as fuel consumption is minimal. Therefore, this paper assumes that aircraft maintain a constant speed on each section of the taxiway, with a maximum taxiing speed set at 10m/s and a minimum speed at 3m/s. Specifically, to accommodate variable-speed taxiing route selection and accurately calculate fuel consumption, the maximum taxiing speed of the aircraft will vary depending on the turning radius, taking into account the need to decelerate when turning to safely pass the curve. Figure 5 shows a schematic of the speed change during turning, where the maximum allowable taxiing speed on a turn is closely related to the degree of the turn, following the formula $V_{\max} = 10 \times R/\text{RADIUS}$, which is the maximum taxiing speed multiplied by the turning radius R divided by a fixed parameter RADIUS, thus determining the specific taxiing speed in the turning area. Furthermore, the determination of the maximum taxiing speed has implications for subsequent cost parameter settings.

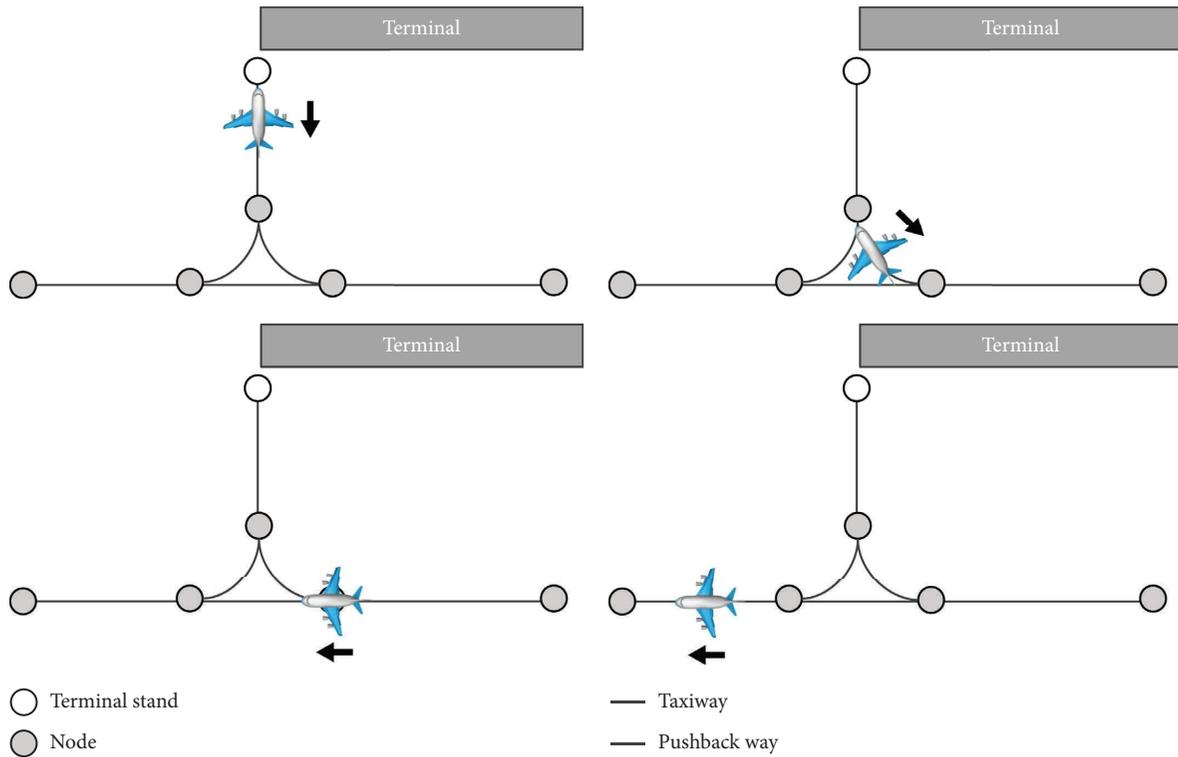


FIGURE 3: Airport pushback taxiway diagram.

The taxiing speed of an aircraft is influenced by acceleration, which is determined by the engine thrust setting. In Table 1, analyzing experimental data for aircraft models, the Airbus A320 and Boeing B738 are the main aircraft models, accounting for about 83%, and both are equipped with CFM56 aviation engines. According to ICAO statistics for the CFM56 engine model, its thrust level during ground taxiing ranges between 7% and 2%, with corresponding fuel consumption rates of 0.124 kg/s to 0.0355 kg/s [17]. Since ICAO does not specify the exact relationship between thrust level and taxiing speed, this study assumes a 7% thrust level corresponds to the maximum taxiing speed of 10 m/s, while a 2% thrust level corresponds to the lowest speed taxiing and idle fuel consumption. Therefore, this study establishes a linear relationship between fuel consumption from 7% to 2% and taxiing speeds from 10 m/s to 3 m/s, thereby determining the relationship between speed and fuel consumption as shown in Table 2, indicating that the setting of taxiing speed has a significant impact on the solution results.

Based on the relationship between the maximum taxiing speed allowed on the taxiway and fuel consumption, a cost vector matrix $C_{nm} \in R$ is formulated for each taxiway, representing the connection from point n to point m , with dimensions $p \times q$. In this matrix, each row p represents a specific thrust level, while each column q represents different optimization objectives [13]. In this study, these optimization objectives primarily focus on taxi time and fuel consumption, thus forming a matrix with 2 columns. The permissible thrust levels for each taxiway differ due to their restrictions, leading to variations in the dimensions of the cost vector matrix. In straight taxiways, when maximum

thrust is allowed, the dimensions are $p = 7, q = 2$; whereas, for turn taxiways, due to thrust level limitations, there may be up to six different dimensional matrices.

$$C_{n,m} = \begin{bmatrix} c_{n,m,1,1} & c_{n,m,1,2} & \cdots & c_{n,m,1,q} \\ c_{n,m,2,1} & c_{n,m,2,2} & \cdots & c_{n,m,2,q} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,m,p,1} & c_{n,m,p,2} & \cdots & c_{n,m,p,q} \end{bmatrix}, \quad (1)$$

4. Resolution Methods

This section shows in detail the advantages and reasons for the methods used in this article, and represents what adjustments are made when applying these methods to airport path planning.

4.1. CEPO. The CEPO algorithm, innovatively proposed based on the RSA, is designed specifically for dynamic routing. Its core advantage is to use the “ripple effect” to simulate the expansion of the path from the starting point to the end point, similar to the ripple diffusion phenomenon in nature. This simulation process covers the process of stimulating ripples from a diffusion starting point to spread outward, touching other nodes and reinitiating them, thus forming a chain of spreading effects. The algorithm takes into account the time, speed, and positions of other aircraft to dynamically adjust paths to avoid conflicts, and optimizes taxiing time by evaluating each possible path branch, prioritizing those that will get to the destination the fastest and safest.

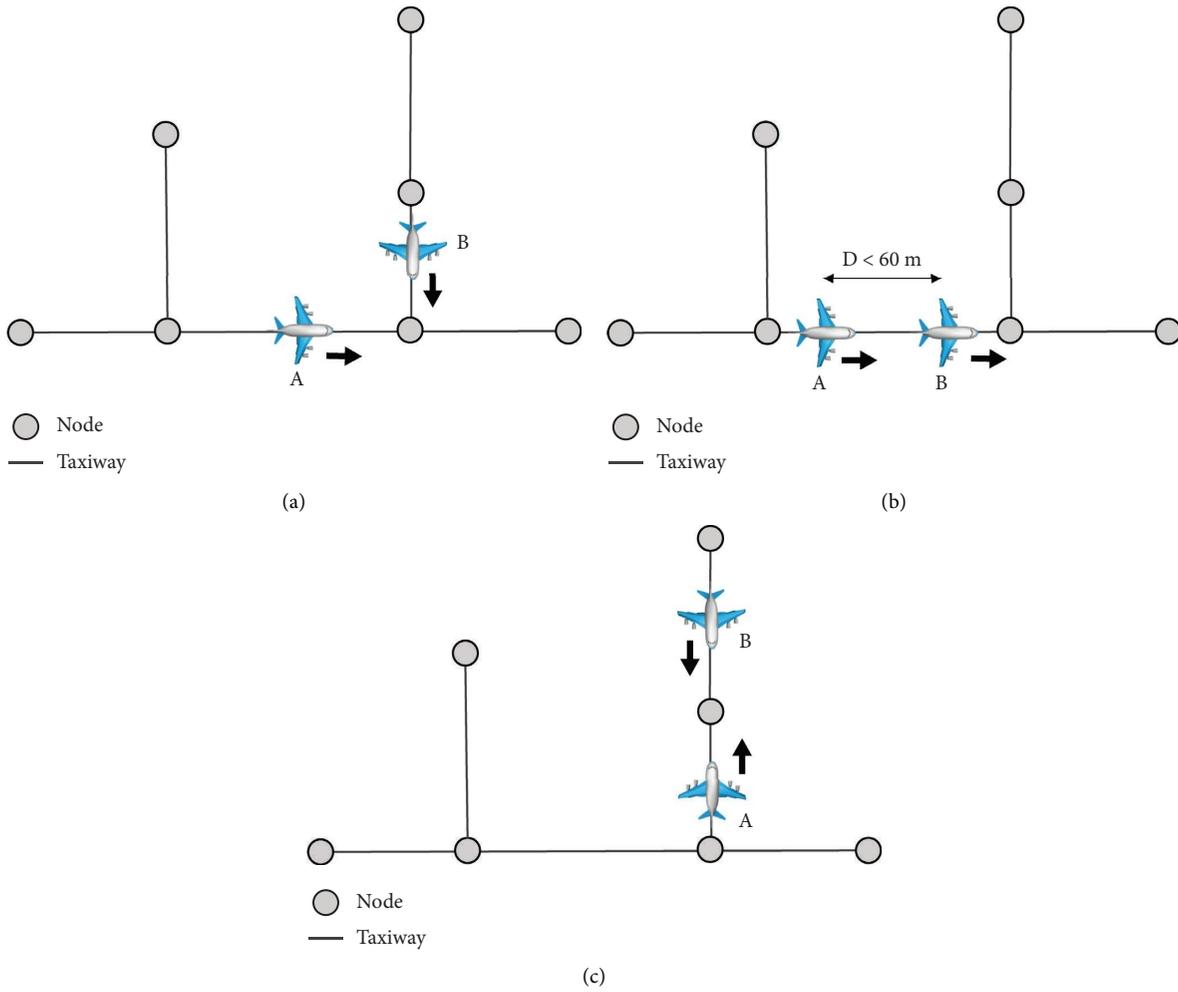


FIGURE 4: Example of three common types of conflicts.

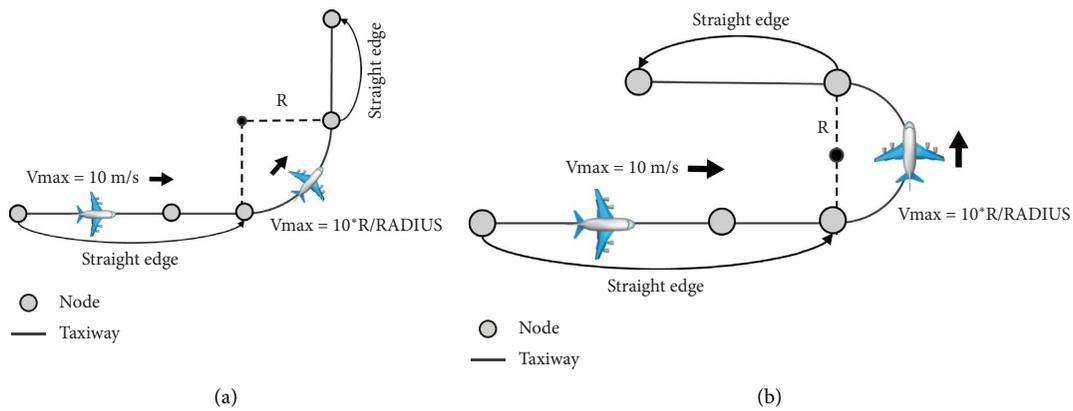


FIGURE 5: Aircraft turning taxiing speed and turning radius relationship diagram.

A distinctive feature of the algorithm is the detailed consideration of node state. In Algorithm 1, the node is changed from the “waiting” state to the “active” state to simulate the taxiing possibility of the aircraft at each node. In line 22 of Algorithm 1, when the target node is activated, the algorithm can trace back from the end point to the starting

point through reverse search to determine an optimal route for each aircraft that takes into account all necessary constraints. This ensures safe and efficient taxiing of the aircraft.

Another significant advantage of the CEPO algorithm is its routing capability that is dynamically synchronized with time. Line 9 of Algorithm 1 demonstrates how the algorithm

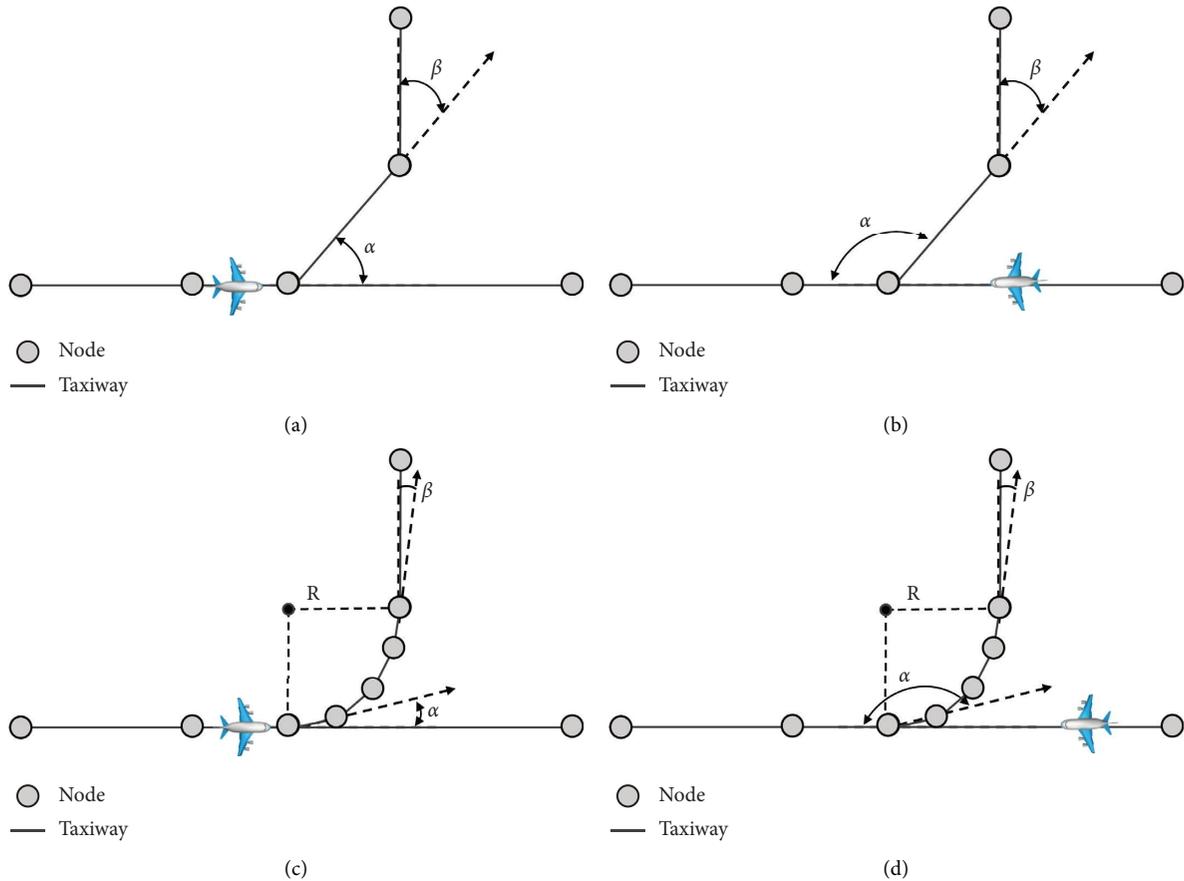


FIGURE 6: Aircraft taxiway turning angle diagram. (a) Angle between straight taxiway. (b) Angle between straight taxiway. (c) Angle between turning taxiway. (d) Angle between turning taxiway.

TABLE 1: Number of different aircraft types used in the data.

Aircraft	Quantity
A319	10
A320	114
A321	22
B734	3
B737	25
B738	232
B739	11

TABLE 2: Relationship between taxiing speed and fuel consumption rate of aircraft.

Speed (m/s)	Fuel consumption rate (kg/s)
10	0.124
9	0.1116
8	0.0992
7	0.0868
6	0.0744
5	0.062
4	0.0496
3	0.0355
0	0.0355

updates the state of nodes. In a time-based network, the algorithm only needs to predict the location of obstacles once. This can realize obstacle avoidance routing that co-evolves with dynamic obstacles. This feature enhances the practicality and efficiency of the algorithm.

In order to improve the adaptability of the algorithm in the airport road network, this paper makes the following improvements in the algorithm:

- Refinement of network initialization: By constructing the airport road network in detail, the connection and distance information between each node and its neighboring nodes are optimized and represented at the expense of taxiing time. For instance, lines 14 to 17 of the Algorithm 1 demonstrate path selection under the constraints of aircraft turning angle limitations. This design not only improves the accuracy of routing, but also enhances the algorithm's adaptability to complex scenarios.
- Deep integration of flight information: When the algorithm inputs parameters, detailed information such as flight take-off and landing sequence, aircraft model, etc. are taken into account. Line 4 of the algorithm considers determining the taxiing endpoints of take-off and landing flights based on the model of each

flight, achieving a more comprehensive personalized and efficient routing. This approach ensures that the route for each aircraft not only avoids potential conflicts but can also be optimally adjusted based on the specific characteristics of the flight.

- Dynamic adjustment and enhancement of conflict avoidance mechanism: The ripple spread process in Algorithm 2 simulates the taxiing process of aircraft. Line 1 of Algorithm 2 determines whether a node n can be activated based on a preassessment in lines 8–10 of Algorithm 1, which identifies time periods D during which activation is permissible. Moreover, line 9 of Algorithm 1 adjusts the activation time D by considering the safety distance between taxiing aircraft, ensuring that the algorithm can dynamically adjust the path in the event of potential conflicts. The introduction of this mechanism significantly improves the practicality and safety of the algorithm.

4.2. TMOA* Algorithm. TMOA* algorithm, as a path search algorithm with time window characteristics, compared with the CEPO algorithm, its core is to use “edge” as the smallest unit of search, and set a start and end time window for each edge. Such a design makes the algorithm more accurate and flexible in the process of road network construction and search. When initializing the algorithm, a graph $G = (N, E)$ is input, which reflects in detail the nodes N of the airport road network and their connection relationships, including the coordinates of the starting and ending points of the taxiway and runway, taxiing speed cost information costs, Time window information and angle information, etc., are similar to the CEPO algorithm, and also include flight sequence information *Flight*.

In the implementation of the TMOA* algorithm, particular emphasis is placed on the initialization of acyclic SG information and the use of the cost matrix *Costs* using nondominated cost vectors and heuristic estimates, the algorithm can effectively narrow the search scope and accelerates the decision-making process. Furthermore, the algorithm can consider the time window in routing and set the start and end time for each “edge,” which enhances the adaptability to dynamic obstacles. This method uses time window management and path selection strategies to prevent potential path conflicts from the source and improve taxiing safety.

To enhance the performance of the TMOA* algorithm in airport network routing, we have made several key improvements:

- Personalized handling of flight information: To more accurately simulate the actual operations of flights, the TMOA* algorithm can customize the starting point and end point of the aircraft’s taxiing based on the specific information of the flight (such as aircraft model). This improvement allows algorithms to plan the most suitable taxi paths for different types of aircraft needs, thereby optimizing the overall efficiency of airport ground traffic management.

- Customized cost function for turning paths and speeds: Taking into account the performance differences at various engine thrust levels (different speeds) when aircraft are turning, we have customized the cost function “costs” for the input in the algorithm. From lines 19 to 24 of Algorithm 3, we can see the restriction of turning angle on path selection. Particularly, in lines 25 to 27 of Algorithm 3, by iterating over possible speed changes for each taxiway, more options are provided for potential taxi paths in the path expansion of Algorithm 4 as Open possibilities. These adjustments make the cost function more aligned with the actual operational characteristics of aircraft, thereby improving the practicality and accuracy of routing.
- Fusion output of path and time window information: In lines 13 to 14 of Algorithm 3, after determining the final route, it combines the start taxiing time of the aircraft with the path’s taxiing cost, converting it into a path with time window information (label path). This obtains the precise moments when aircraft reaching each node and taxiway during ground taxiing. This design facilitates updates and management of the airport network’s time windows, further enhancing the algorithm’s adaptability to the airport’s dynamic environment.
- Innovation in time window management strategy: At line 15 of Algorithm 3, we have redesigned the time window update strategy to ensure that each taxiway has a continuous time window, avoiding the complexity and potential taxiing conflicts brought by previously multiple discontinuous time windows. This improvement simplifies the routing process and increases the accuracy and safety of path selection. This improvement simplifies the routing process and increases the accuracy and safety of path selection.
- Selection of heuristic functions: In terms of heuristic functions, we adopted two approaches: H1 and H2. H1 is based on the ideal cost estimation of the shortest path between nodes, considering the maximum taxiing speed for the design of taxiways, providing a lower bound estimate of the cost; while H2 uses the Manhattan distance as the basis for estimation. Through experimental comparisons of these two heuristic functions, H1 has shown better performance in most cases, thus it has been selected as the primary heuristic function to accelerate the search process and improve the efficiency of routing. Selection of heuristic functions: In terms of heuristic functions, we adopted two approaches: H1 and H2.

These improvements not only increase the efficiency and accuracy of the algorithm but also enhance its adaptability to specific aircraft models and flight requirements.

5. Results

This section compares the computational efficiency and solution performance differences between the CEPO

Input: Network graph $G = (N, E)$ with edge $e \in E$, node $n \in N$, the set of flight information *Flight*, the set of pushback points P
Output: Paths: Complete point-in-time accurate path information *path* for each flight

1. **for** *flight* in *Flight* **do**
2. $(s, e, t) \leftarrow \text{GetFlightInformation}(\text{flight})$ \triangleright start point s , end point e , and start taxiing time t
3. Initialize R, s as $ep \triangleright$ the set of ripple R , epicenter ep
4. At start point $r_i(V, L, S) \leftarrow (V, 0, 2), R.add(r_i)$
5. /* Ripples spread speed, spreading distance L , and $S = 1, 2, 3$ the node's state is waiting, active, dead */
6. **for** $n_i \in N$ **do** \triangleright Update activation time D and S
7. $n_i(S, D) \leftarrow \text{UPDATE_STATE_AND_TIME}(S, D)$
8. **end for**
9. **while** $e(S) \neq 2$ **do** \triangleright the state of end node isn't activated
10. **for** r_i in R **do**
11. $ep \leftarrow \text{GET_EPICENTER}(r_i)$
12. **for** n in N_e **do** $\triangleright N_e$ is the set of epicenter's neighbor node
 $ang_rad \leftarrow out_angles_{last_n}^{ep} - in_angles_{ep}^n$
 /* $last_n$ is last node of ep */
13. **if** $ang_rad \geq 90$ **then** \triangleright Angle checking
14. **go to** Line 11
15. **end if**
16. $R \leftarrow \text{RIPPLE_SPREADING}(P, n, R)$
17. **end for**
18. **end for**
19. **end while**
20. $path \leftarrow \text{REVERSE_SEARCH}(R, N)$
21. $Paths.add(path)$
22. **end for**
23. **return** $Paths$

ALGORITHM 1: Dynamic routing for airport based on the CEPO algorithm.

1. **if** $n(S, D) = (1, t)$ **or** n in P **then** \triangleright At time t , n isn't activated
2. **if** all n in N_n state are 2 **then** $\triangleright N_n$ the set of n 's neighbor nodes
3. $R_i(V, L, S) \leftarrow (V, 0, 3)$
4. $R \leftarrow R$ remove $r_i \triangleright r_i$ has triggered all the reachable nodes around it
5. **Else**
6. $t \leftarrow t + 1 \triangleright t + 1$ is equal to all ripples spreading for 1 s
7. **if** $n_j(S, D) = (2, t)$ **then** $\triangleright n_j$ is touched by ripple at time t
8. $r_j(V, L, S) \leftarrow (V, 0, 2) \triangleright$ Active new ripple at node n_j
9. $R.add(r_j)$
10. Record the node n_j was activated by r_j
11. **end if**
12. **end if**
13. **end if**

ALGORITHM 2: Ripple spreading.

algorithm and the TMOA* algorithm by using actual flight data, aiming to evaluate their feasibility in practical applications.

5.1. Comparison of Computational Efficiency. The simulation experiments conducted in this study are based on real data from the flight area of Tianjin Binhai International Airport, including runways, aprons, taxiways, and Pushback routes, covering the flight data of the 10 busiest days from 2015 to 2019. These data are detailed in Table 3 and all originate from the historical records of Tianjin Binhai International

Airport. The experiments were carried out using a single-threaded application written in Python, running on a 2021 MacBook equipped with an M1 Pro chip, 16 GB of memory, and a 512 GB solid-state drive.

Based on the comparison of solution efficiency between the CEPO algorithm and the TMOA* algorithm as shown in Figure 7, the CEPO algorithm exhibits lower efficiency in solving the taxiing paths of flights, with its complexity being $O(N^2)$. In contrast, the complexity of the TMOA* algorithm is $O(N)$. The CEPO algorithm requires at least 3000 s (about 0.8 h) to determine the final routes of one day's flight data. Meanwhile, the resolution time for the TMOA* algorithm is

Input: Graph $G = (N, E)$ with weights w and time windows $time_windows$, the set of flight information $Flight$, the set of cost matrix costs for all $e \in E$

Output: The path p and the path cost $Costs$ for each $flight$

```

1. Initialize  $SG, G\_op, G\_cl, Open, Costs \triangleright SG$  is a noncircle data to record the path
2. for  $flight$  in  $Flight$  do
3.    $(s, e, t) \leftarrow GetFlightInformation(flight)$ 
4.   while  $Open \neq \emptyset$  do
5.      $(n, g_n, f_n) \leftarrow SELECT\_FROM\_OPEN(Open)$ 
6.      $Open \leftarrow Open$  remove  $(n, g_n, f_n)$ 
7.     if  $n = e$  then
8.        $Costs.add(g_n)$ 
9.        $Open \leftarrow Open$  remove dominated alternatives using  $g_n$ 
10.    if  $Open = \emptyset$  then
11.       $p \leftarrow RECONSTRUCT\_PATHS(SG, e, s)$ 
12.       $Label\_p \leftarrow Construct\_label\_paths(time\_windows, p, G, T)$ 
13.       $Time\_windows^* \leftarrow Readjustin\_timewindow(time\_windows, Label\_p)$ 
14.    end if
15.    return  $p, Costs, time\_windows^*$ 
16.  else
17.    for  $m$  in  $N_n$  do  $\triangleright$  The set of  $n$ 's neighbor nodes
18.    /* Check angle between current segment and next segment */
19.     $ang\_rad \leftarrow out\_angles_{SG[n]}^n - in\_angles_n^m$ 
20.    /*  $SG[n]$  is last node of  $n$  */
21.    if  $ang\_rad \geq 90$  then
22.      go to Line 19
23.    end if
24.    for  $c\_n\_m\_l$  in  $costs$  do
25.       $\triangleright c\_n\_m\_l$  is cost vector  $n$  to  $m$ 
26.      Perform PE in Algorithm 4  $\triangleright$  Expand path
27.    end for
28.  end for
29. end while
30. return  $p\_empty, Costs$ 
31. end for

```

ALGORITHM 3: TMOA*.

at most only 10 s, which is nearly three orders of magnitude faster. Table 4 further highlights the difference in resolution time per aircraft, with the CEPO algorithm taking an average of about 8 s to solve a single flight's route. Considering airport taxiing rules, if the resolution time for a route is 8 s, with ground aircraft moving at a speed of 10 m per second, this would result in excessively long taxiing distances while updating the aircraft's taxiing path. There is a risk of missing intersections and deviating from the intended taxiing direction. Moreover, within 8 s, an aircraft would taxi 80 m, exceeding the current safety standards for ground taxiing distances 60 m, potentially increasing the risk of aircraft conflicts and thus failing to ensure the safety of ground taxiing. Therefore, to meet the requirements for immediate planning of ground aircraft routes at airports, the TMOA* algorithm demonstrates higher adaptability and is capable of effectively completing real-time dynamic routing.

5.2. Comparison of Solution Performance. In comparing the efficiency of two algorithms with the airport's established taxiing path rules, we observed that the algorithms

significantly shorten the path length while resolving taxiing conflicts. As demonstrated in Figures 8 and 9, when comparing the fixed taxiing paths of the airport (shown in blue) with the paths generated by the algorithms (shown in red), the latter substantially reduces both path length and taxiing time under the premise of ensuring safety. Therefore, these results indicate that the taxiing paths generated by the algorithms are overall markedly superior to the fixed paths of the airport.

Since the CEPO algorithm is a single-objective optimization algorithm that can only optimize taxiing distance, we exclusively utilize the solutions obtained by the TMOA* algorithm with the weight setting of (1:0), which corresponds to minimizing taxiing time, for comparisons across different dimensions, such as total taxiing time, total taxiing distance, and fuel consumption etc.

According to the simulation results, the two algorithms were compared across multiple dimensions. First, we calculated the conflict numbers and duration during the taxi routing process using 10 days of data for both the CEPO and TMOA* algorithms. Here, a conflict is defined as an

```

1. for window in time_windowsnm do ▷ Check Time Window
2.   tcurrent ← Determine_current_time(Gop, Gcl)
3.   if cnml[0] + tcurrent is outside the window then. ▷ time cost cnml[0]
4.     he ← HOLDING_CHECK(cnml, window, tcurrent)
5.     hc ← CALCULATE_HOLDING_COST(cnml, window, tcurrent)
6.   end if
7.   end for
8.   if he is True then
9.     cnml* ← ADD_HOLDING_COST(cnml, hc)
10.  else
11.    return SG, Gop, Gcl, Open
12.  end if
13. Calculate the cost of the new route found to m: gm = gn + cnml*
14. if m is not in SG then
15.   Calculate fm = gm + h(m, end)
16.   /* h function is getting the shortest path cost from m to end e */
17.   if fm is not dominated by any vector in Costs then
18.     Open.add((m, gm, fm)) ▷ Add (m, gm, fm) to Open
19.     Gopm.add(gm) ▷ Add gm to Gopm
20.     Get edge (m, n) with labels gm and cnml* and create it in SG
21.   end if
22. else if gm equals some cost vector in Gopm ∪ Gclm then
23.   create edge (m, n) in SG with labels gm and cnml*
24. else if gm is not dominated by any vector in Gopm ∪ Gclm then
25.   /* a path to m with new interesting cost has been found */
26.   Eliminate from Gopm and Gclm vectors dominated by gm. For each vector gm' eliminated from Gopm, eliminate the corresponding (m, gm, fm) from Open
27.   Calculate fm = gm + h(m, end)
28.   if fm is not dominated by any vector in Costs then
29.     Open.add((m, gm, fm))
30.     Gopm.add(gm)
31.     Get edge (m, n) with labels gm and cnml* and create it in SG
32.   end if
33. end if
34. return SG, Gop, Gcl, Open

```

ALGORITHM 4: Path expansion (PE).

TABLE 3: Flight data information for 10 days at Tianjin Binhai international airport.

Date	Number of flights
2017-08-03	499
2017-08-06	519
2017-08-14	527
2017-08-17	517
2017-08-19	511
2018-07-18	535
2018-07-29	529
2018-09-29	533
2019-02-01	539
2019-08-07	524

irreconcilable situation between two or more aircraft, and the conflict duration is measured from the onset of the conflict until all aircraft can resume normal taxiing.

The experimental data in Figure 10 and Table 5 show that the TMOA* algorithm performs better in terms of both the number and duration of taxiing conflicts. For example, on

February 1, 2019, the TMOA* algorithm reduced the number of conflicts by 101 and the conflict duration by 267 s. This indicates that the TMOA* algorithm is more inclined to avoid conflicts by rerouting rather than relying on waiting. In contrast, the CEPO algorithm tends to resolve conflicts through waiting, which helps reduce the total path length but often leads to longer conflict durations and a higher frequency of conflicts in dynamic networks. Although the CEPO algorithm aims to find the shortest path to the destination, it does not account for waiting time in the path cost. This leads the CEPO algorithm to minimize the total taxiing distance by waiting to resolve conflicts in dynamic network environments. However, this approach to reducing distance often comes at the cost of longer waiting times and more frequent conflicts.

According to the simulation results displayed in Figure 11, it is presented a comparison of the solution of the two algorithms across four different dimensions. Firstly, Figure 11(c) shows the difference in the total taxiing distances of the paths solved by the two algorithms, with the results indicating that the total taxiing distance obtained by

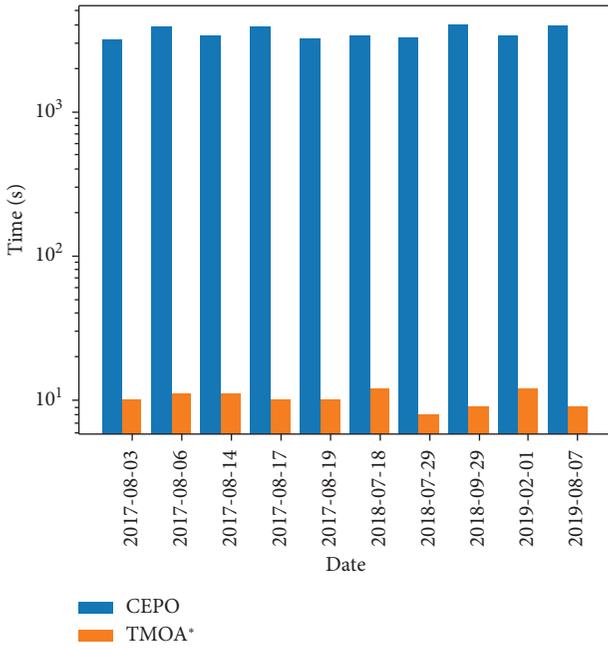


FIGURE 7: CEPO and TMOA* algorithm solution efficiency comparison.

TABLE 4: Average computation time required to compute path for one flight.

Algorithm name	Average computation time per flight (s)
CEPO	7.54
TMOA*	0.01

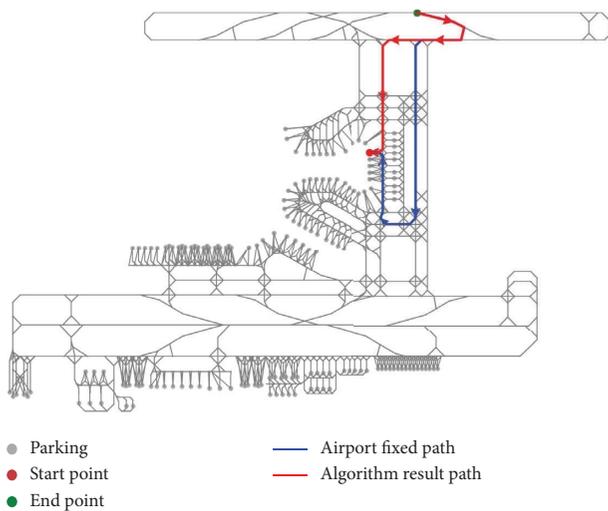


FIGURE 8: Comparison between algorithm solution path and airport fixed path 1.

the TMOA* algorithm is slightly greater than that of the CEPO algorithm, with a deviation of less than 1% between the two algorithms. This is because, when dealing with dynamic networks, the CEPO algorithm's method of traversing the solution space ensures the shortest total taxiing

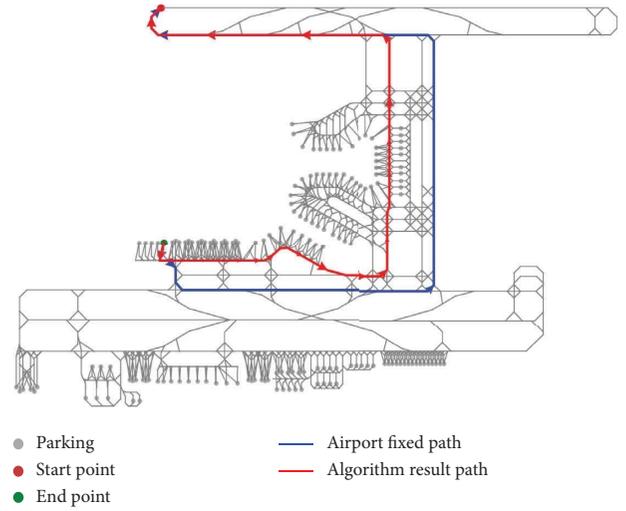


FIGURE 9: Comparison between algorithm solution path and airport fixed path 2.

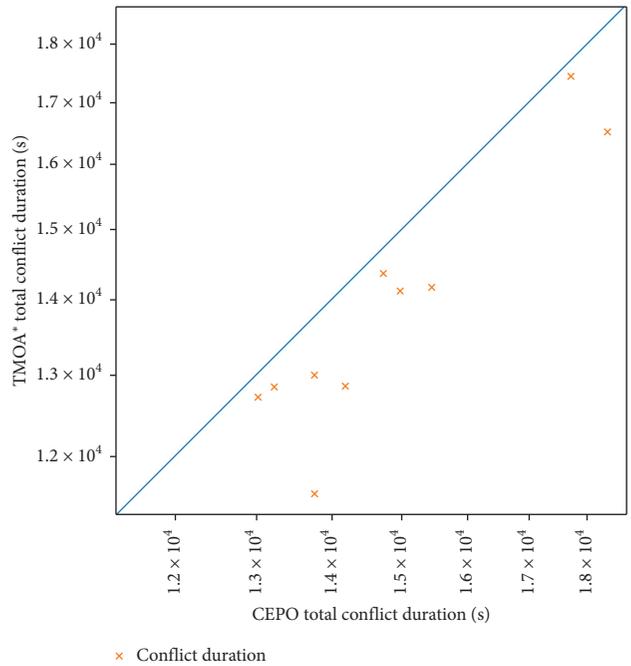


FIGURE 10: Comparison of total conflict duration between CEPO and TMOA* algorithms*.

distance. Therefore, the CEPO algorithm can achieve the minimization of dynamic path taxiing distances, which is consistent with the simulation results. In Figure 11(a), the total taxiing time solved by the TMOA* algorithm is generally shorter, reducing taxiing time by up to 5%. Therefore, the TMOA* algorithm demonstrates a clear advantage in ensuring that aircraft can take off quickly and on schedule during airport ground operations, especially considering that the cost of total taxiing time is more critical. Simultaneously, the simulation results regarding total fuel consumption shown in Figure 11(b) indicate that the TMOA* algorithm also has advantages in reducing fuel consumption

TABLE 5: Comparison of conflict number and conflict duration between CEPO and TMOA algorithms for different dates.

Date	CEPO		TMOA*	
	Conflict number	Conflict duration (s)	Conflict number	Conflict duration (s)
2017/08/03	336	14,189.96	257	12,856.35
2017/08/06	317	13,760.71	217	11,572.13
2017/08/14	330	14,979.94	248	14,119.25
2017/08/17	317	13,024.23	241	12,725.87
2017/08/19	340	15,451.57	254	14,174.64
2018/07/18	391	18,362.44	304	16,515.56
2018/07/29	333	13,766.20	262	13,004.66
2018/09/29	337	14,725.37	273	14,365.16
2019/02/01	448	17,717.67	347	17,450.51
2019/08/07	332	13,234.25	247	12,855.57

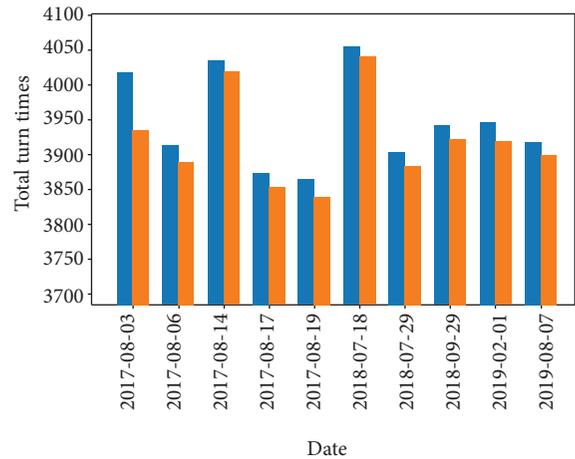
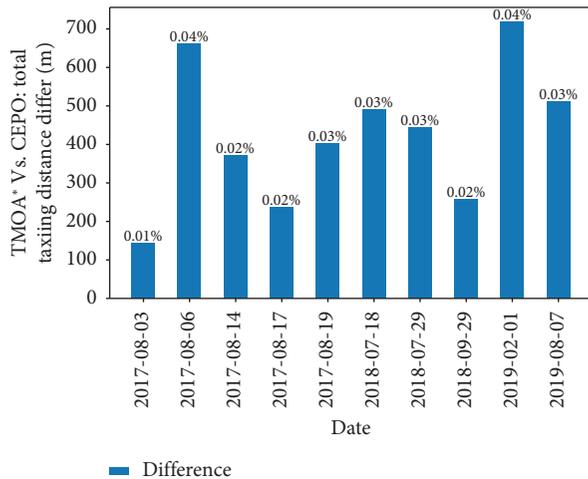
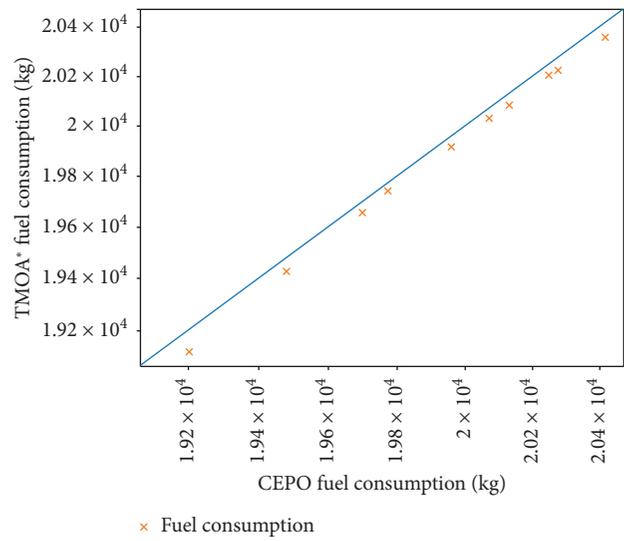
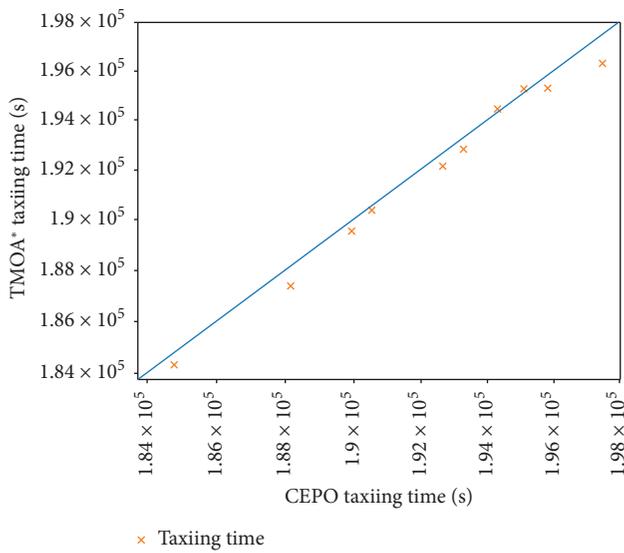


FIGURE 11: Example of four subfigures arrangement. (a) Total taxiing time solved by the two different algorithms. (b) Total fuel consumption solved by the two algorithms. (c) Total taxiing path length solved by the two algorithms. (d) Total number of turn times solved by the two algorithms.

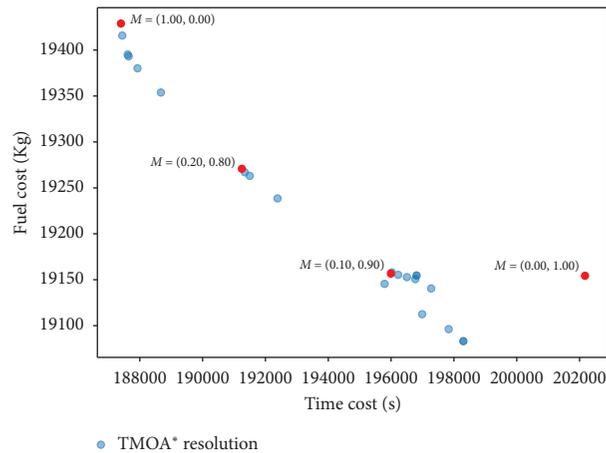


FIGURE 12: Approximate Pareto frontier graph solved by the TMOA* algorithm on 19/08/2017.

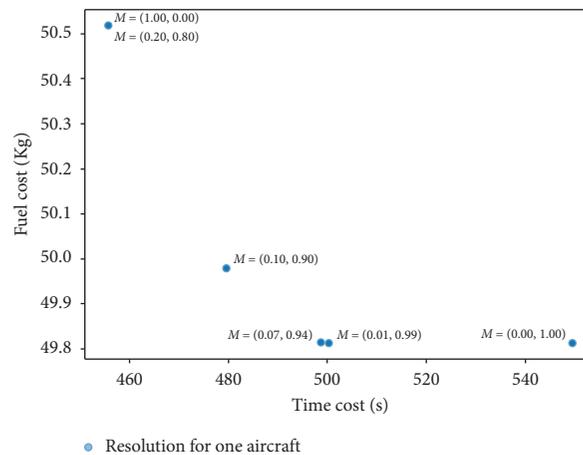


FIGURE 13: Approximate Pareto frontier graph solved by the TMOA* algorithm for one aircraft.

and improving airport operational efficiency. This advantage is partly due to the CEPO algorithm's tendency to resolve conflicts through waiting. Although this method can reduce taxiing time and fuel consumption by minimizing taxiing distances, excessive waiting times mean additional fuel consumption by aircraft engines in idle state. Therefore, the TMOA* algorithm not only has the advantage of reducing taxiing times but also significantly surpasses the CEPO algorithm in saving fuel resources. In the results shown in Figure 11(d), the final route determined by the CEPO algorithm has a higher number of turns than that of the TMOA* algorithm. This phenomenon arises from the fact that, even though turning reduces speed, in some cases, the taxiing distance of the turning route is actually lower than that of the straight route. This leads the CEPO algorithm to favor routes with more turns.

This study not only compared the resolution effects of two algorithms but also conducted simulation experiments based on flight data from a specific day under different weight settings, resulting in the approximate Pareto frontier plot shown in Figure 12. From the figure, as the weight on taxiing time decreases, the total taxiing time gradually

increases, especially when the weight is set to (0.2:0.8), where the increase in time is more pronounced. Hence, the sensitivity of the TMOA* algorithm to taxiing time is primarily evident at lower weight settings. A particular arrival flight from the W3 exit of runway 16L to gate 416L was selected for a focused study on the relationship between fuel consumption and total taxiing time under different weight settings. Through simulation, an approximate Pareto plot for a single aircraft was obtained as shown in Figure 13, which trends similarly to Figure 12. This study specifically focused on four different weight settings: $M(1:0)$, $M(0.2:0.8)$, $M(0.1:0.9)$, and $M(0:1)$, representing minimum taxiing time, two representative weight settings with significant parameter changes, and minimum fuel consumption weight setting, respectively. A visual analysis of these four weight settings was presented in Table 6. The data shows that as the weight of taxiing time decreases, the total taxiing time of the aircraft increases, fuel consumption decreases, and an increase in the number of turns is also noteworthy. Figures 14 and 15 display changes in routes in specific areas. Therefore, we can conclude that routes are primarily influenced by the following two aspects:

TABLE 6: Weight selection criteria.

Weight selection	Taxiing time (s)	Fuel consumption for taxiing (kg)	Number of turns
(1:0)	455	50.51	4
(0.2:0.8)	455	50.51	4
(0.1:0.9)	480	49.98	7
(0:1)	549	49.81	10

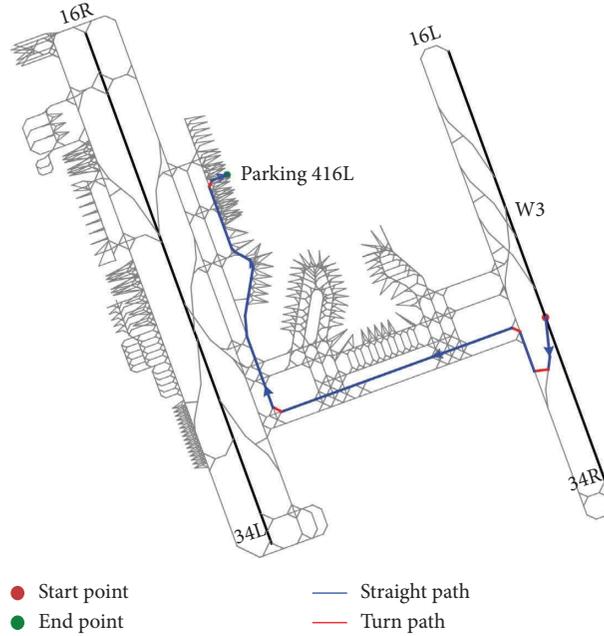


FIGURE 14: Taxiing route of a single aircraft when the weight selection is M (1:0) for the shortest taxiing time.

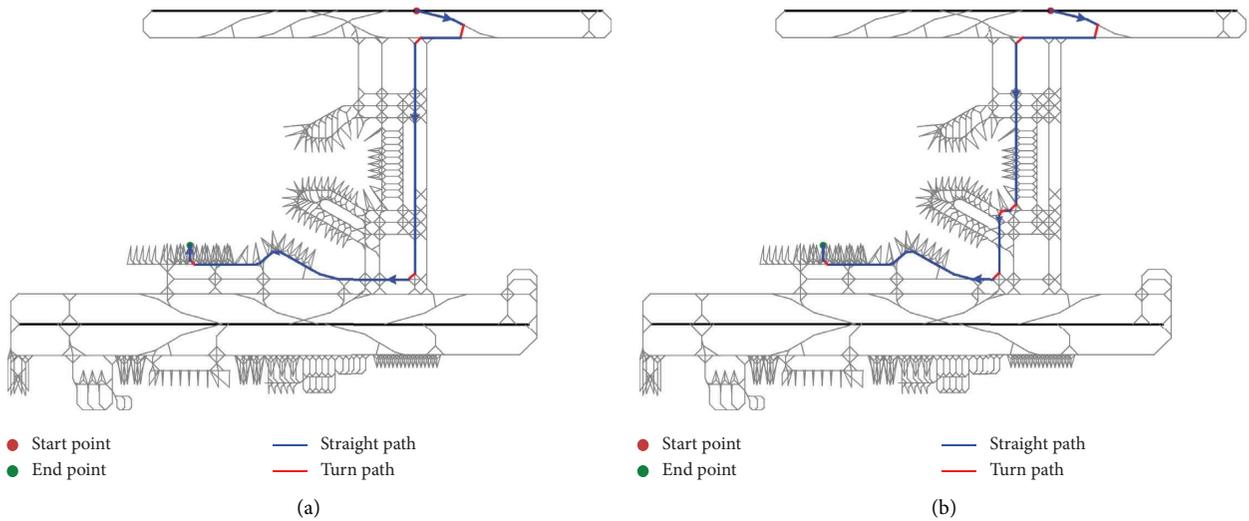


FIGURE 15: Continued.

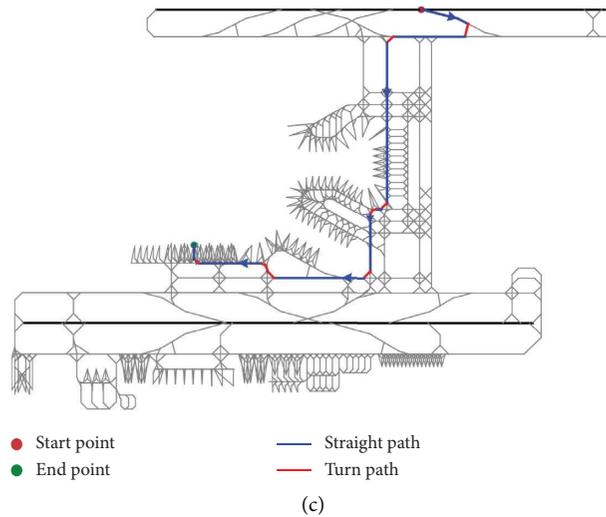


FIGURE 15: Impact of different weight selections on the taxiing route of a single aircraft. (a) $M (0.2:0.8)$. (b) $M (0.1:0.9)$. (c) $M (0:1)$.

- The number of turns: prioritizing minimizing fuel consumption tends to choose paths with more turns.
- Network availability: The availability of the network is limited, and changing the weight settings does not necessarily change the path, as seen with the setting for $M (0.2:0.8)$ for one aircraft.

Overall, although both algorithms can be applied to the computation of dynamic routing, the CEPO algorithm is less practical in terms of computational efficiency, taxiing time, and fuel consumption. In contrast, the TMOA* algorithm not only effectively reduces taxiing time but also demonstrates superior performance in lowering fuel consumption, providing robust decision support for real-time dynamic routing at airports.

6. Summary and Conclusions

This study introduced and evaluated a new MOA* algorithm with time windows, specifically tailored for optimizing airport ground operations. The research primarily focused on improving the efficiency of taxiing routes for aircraft at large airports, addressing the critical challenges of reducing taxiing times and fuel consumption while ensuring operational safety. Through comprehensive simulations using real-world data from Tianjin Binhai International Airport, the proposed algorithm demonstrated significant improvements in operational efficiency compared to traditional methods.

Our findings indicate that the MOA* algorithm not only minimizes the total taxiing time with minimal deviation from optimal solutions but also excels in optimizing fuel consumption, outperforming existing solutions by operating at a computational speed conducive to real-time application. This efficiency gain underscores the algorithm's potential in providing effective decision support for dynamic routing in airport ground operations, especially during peak operating periods.

Moreover, the algorithm's adaptability to changing airport ground conditions and its ability to accommodate multiobjective optimization criteria particularly fuel consumption and safety considerations make it a robust tool for enhancing airport operational efficiency. The simulation results highlight the practical relevance of integrating dynamic path planning algorithms into AGM strategies, potentially leading to significant cost savings, reduced environmental impact, and improved overall airport capacity.

Future research directions could explore the integration of this algorithm into broader airport operational management systems, such as runway scheduling and apron allocation, to enhance efficiency and sustainability. Additionally, while we currently lack data from other airports, we plan to investigate the algorithm's adaptability in various contexts, particularly at larger airports like Paris Charles de Gaulle (CDG) airport. This will provide insights into its applicability across different airport layouts and traffic conditions. The ongoing development of intelligent decision support systems, leveraging advancements in computational algorithms, including the one proposed in this study, promises to significantly enhance airport ground operations management.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

This research was supported by the National Natural Science Foundation of China under grant number 72301278, the Civil Aviation Airport Engineering Technology Research Center Fund under grant number ERCAOTP20230101, the

Fundamental Research Funds for the Central Universities of Ministry of Education of China under grant number 3122023012, and the Tianjin Applied Basic Research Multi-Input Fund under grant number 21JCQNJC00790.

Acknowledgments

This research was supported by the National Natural Science Foundation of China under grant number 72301278, the Civil Aviation Airport Engineering Technology Research Center Fund under grant number ERCAOTP20230101, the Fundamental Research Funds for the Central Universities of Ministry of Education of China under grant number 3122023012, and the Tianjin Applied Basic Research Multi-input Fund under grant number 21JCQNJC00790. These funds provided valuable financial support and resources for this research. Special thanks to Ye Yu, Ruixin Wang, and Jean-Baptiste Gotteland for their professional guidance, valuable suggestions, and tireless efforts throughout the research process. Their contributions and support have been crucial to the completion of this work.

References

- [1] Eurocontrol, *A-Cdm Impact Assessment* (Brussels, Belgium: European Organisation for the Safety of Air Navigation (EUROCONTROL), 2016).
- [2] J. A. D. Atkin, E. K. Burke, and S. Ravizza, "The Airport Ground Movement Problem: Past and Current Research and Future Directions," in *Proceedings of the 4th International Conference on Research in Air Transportation (ICRAT)* (Budapest, Hungary, May 2010), 131–138.
- [3] E. W. Dijkstra, "A Note on Two Problems in Connexion With Graphs," *Edsger Wybe Dijkstra* 11 (2022): 287–290, <https://doi.org/10.1145/3544585.3544600>.
- [4] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics* 4, no. 2 (1968): 100–107, <https://doi.org/10.1109/tssc.1968.300136>.
- [5] Y. Huang, L. Zhao, T. Van Woensel, and J.-P. Gross, "Timedependent Vehicle Routing Problem With Path Flexibility," *Transportation Research Part B: Methodological* 95 (2017): 169–195, <https://doi.org/10.1016/j.trb.2016.10.013>.
- [6] S. Ravizza, J. A. D. Atkin, and E. K. Burke, "A More Realistic Approach for Airport Ground Movement Optimisation with Stand Holding," *Journal of Scheduling* 17, no. 5 (2014): 507–520, <https://doi.org/10.1007/s10951-013-0323-3>.
- [7] J. C. Namorado Climaco and E. Queirós Vieira Martins, "A Bicriterion Shortest Path Algorithm," *European Journal of Operational Research* 11, no. 4 (1982): 399–404, [https://doi.org/10.1016/0377-2217\(82\)90205-3](https://doi.org/10.1016/0377-2217(82)90205-3).
- [8] S. Ravizza, J. Chen, J. A. D. Atkin, E. K. Burke, and P. Stewart, "The Trade-Off between Taxi Time and Fuel Consumption in Airport Ground Movement," *Public Transport* 5, no. 1-2 (2013): 25–40, <https://doi.org/10.1007/s12469-013-0060-1>.
- [9] B. S. Stewart and C. C. White, "Multiobjective A*," *Journal of the ACM* 38, no. 4 (1991): 775–814, <https://doi.org/10.1145/115234.115368>.
- [10] C. Tung Tung and K. Lin Chew, "A Multicriteria Pareto-Optimal Path Algorithm," *European Journal of Operational Research* 62, no. 2 (1992): 203–209, [https://doi.org/10.1016/0377-2217\(92\)90248-8](https://doi.org/10.1016/0377-2217(92)90248-8).
- [11] L. Mandow and J. L. P. De La Cruz, "Multiobjective A* Search With Consistent Heuristics," *Journal of the ACM* 57, no. 5 (2008): 1–25, <https://doi.org/10.1145/1754399.1754400>.
- [12] L. Mandow, "A New Approach to Multiobjective A* Search," *International Joint Conference on Artificial Intelligence* 8 (2005).
- [13] M. Weiszer, E. K. Burke, and J. Chen, "Multi-Objective Routing and Scheduling for Airport Ground Movement," *Transportation Research Part C: Emerging Technologies* 119 (2020): 102734, <https://doi.org/10.1016/j.trc.2020.102734>.
- [14] Eurocontrol, *A-Cdm Impact Assessment* (Brussels, Belgium: European Organisation for the Safety of Air Navigation (EUROCONTROL), 2016).
- [15] Y.-Q. Gao, T.-Q. Tang, J. Zhang, and F. You, "Which Aircraft Has a Better Fuel Efficiency?—A Case Study in China," *Transportation Business: Transport Dynamics* 10, no. 1 (2022): 1032–1045, <https://doi.org/10.1080/21680566.2021.1997674>.
- [16] S. Zheng, Z. Yang, Z. He, N. Wang, C. Chu, and H. Yu, "Hybrid Simulated Annealing and Reduced Variable Neighbourhood Search for an Aircraft Scheduling and Parking Problem," *International Journal of Production Research* 58, no. 9 (2020): 2626–2646, <https://doi.org/10.1080/00207543.2019.1629663>.
- [17] B. Kim and J. Rachami, "Aircraft Emissions Modeling under Low Power Conditions," in *Proceedings of the A & WMA's 101st Annual Conference and Exhibition* (Portland, OR, June 2008), 24–27.