The user is asking me to translate a Chinese academic title into English. The title is about airport operations, specifically multi-airport terminal areas. Let me analyze the components: - 面向: oriented towards / for / facing - 多机场: multi-airport - 终端区: terminal area (in aviation context, this ref…

Authors: The user has provided a very short input: “李同贺”. This appears to be a Chinese name. Let me analyze the requirements:

1. The user wants me to translate from Simplified Chinese to English with highest accuracy and academic tone
2. The critical requirements mention preserving LaTeX commands, placeholders, citation commands, and …tags
3. However, the input “李同贺” is just a name, not a full paragraph with these structural elements

Since the input is just a name, there are no: - LaTeX commands to preserve - placeholders - [?], ??, (??) commands - …tags

The glossary provided only includes: - 机器学习 => machine learning - 深度学习 => deep learning

Neither of these apply to the input “李同贺”.

“李同贺” is a Chinese name. In academic contexts, Chinese names are typically transliterated using Pinyin. The standard format is: - Surname first, then given name - Each syllable capitalized - Sometimes written as “Li Tonghe” or “Li Tong-he” or with the surname first: “Tonghe Li” depending on the convention

In Chinese academic papers, names are usually written in the format “Li Tonghe” (surname first, capitalized) or sometimes “Tonghe Li” (Western order). The most common and standard approach in academic contexts is to keep the original Pinyin with surname first: “Li Tonghe”.

Since there are no tags in the input, I should not add any. The output should be just the transliterated name.

The user said “Return ONLY the translated text for the given input (no explanations, no quotes, no headings you invent).”

So I should simply output: Li Tonghe

Wait, let me double-check the requirements. The user wants me to translate from Simplified Chinese to English. A name is not really “translated” but transliterated. The most appropriate academic rendering would be the Pinyin form.

The name “李同贺” breaks down as: - 李 (Lǐ) - surname, meaning “plum” - 同贺 (Tónghè) - given name, meaning “congratulate together”

In academic papers, this would be written as “Li, Liu Jixin, The user has provided a single Chinese character” 江灏” which appears to be a name. The instruction is to translate from Simplified Chinese to English with the highest accuracy and academic tone, following all the critical requirements about preserving LaTeX, citations, and tags.

However, the input “江灏” is just a name, not a full paragraph with tags. Let me check the requirements again:

1. The user wants me to translate from Simplified Chinese to English
2. I must preserve ALL ···paragraph wrapper tags EXACTLY
3. The output should have exactly the same number of paragraph tags as input
4. Return ONLY the translated text

The input is just “江灏” without any tags. This is ambiguous. Should I: - Just translate the name as “Jiang Hao” ? - Or should I wrap it in a tag?

Looking at the instructions more carefully: “Translate from Simplified Chinese to English with the highest accuracy and academic tone.” The critical requirements say “PRESERVE ALL ···paragraph wrapper tags EXACTLY” and “Maintain exact count and IDs of ···tags” .

Since the input doesn’ t contain any tags, I cannot preserve them. The instructions also say “Return ONLY the translated text for the given input (no explanations, no quotes, no headings you invent).”

So I should just translate “江灏” as a name. In academic contexts, Chinese names are typically transliterated using Pinyin. “江” is “Jiang” and “灏” is “Hao” . The standard format is to capitalize the surname and given name: “Jiang Hao” .

I should not add any tags since they weren’ t in the original input. I should not add any explanation. Just the translation.

The glossary provided only includes “机器学习 => machine learning” and “深度学习 => deep learning” , which are not relevant here.

So the answer should be simply: “Jiang Hao”

Wait, let me double-check. The user wrote "江灏" and the system prompt says "Translate from Simplified Chinese to English". The name "江灏" is a person's name. In academic papers, Chinese names are typically written in Pinyin. The surname is "江" (Jiang) and the given name is "灏" (, The user has provided a very short input: "曾维理". This appears to be a Chinese name. Let me analyze what this could mean:

1. It could be a person's name that needs translation
2. It could be part of a larger document that was cut off
3. It could be a test to see how I handle minimal input

Looking at the context, the user provided detailed instructions for translating academic papers with LaTeX commands, tags, etc. But the actual input is just "曾维理".

This is a Chinese name. In academic translation, Chinese names are typically handled in one of these ways: - Translated phonetically (pinyin): Zeng Weili - With the surname first: Zeng Weili - Sometimes with the surname capitalized: ZENG Weili or Zeng, Weili

The most standard academic convention is to use pinyin and maintain the Chinese name order (surname first): Zeng Weili.

However, I need to check if there are any tags or LaTeX commands in the input. There are none. The input is just plain text with three characters.

Given the strict requirements: - I must preserve ALL ⋯paragraph wrapper tags EXACTLY - I must preserve ALL LaTeX commands - I must translate from Simplified Chinese to English

But there are no tags in the input. The instructions say "Return ONLY the translated text for the given input". So I should just translate what's given.

The name "曾维理" would be: - 曾 (Zeng) - surname - 维理 (Weili) - given name

In academic contexts, Chinese names are typically written in pinyin. The standard format is surname first, given name second, with the first letter of each capitalized: Zeng Weili.

Since there are no tags in the input, I should not add any. The output should be just the translated name.

Wait, let me re-read the instructions carefully. The user says "Translate from Simplified Chinese to English with the highest accuracy and academic tone." and provides detailed rules about preserving tags, LaTeX, etc.

But the input is just "曾维理". This is ambiguous. It could be: 1. A standalone name 2. The beginning of a paragraph that was cut off

If it's a standalone name,, Yang Lei, Liu Jixin, The user has provided a single Chinese name "江灏" and wants me to translate it. This is a proper name, not a scientific paper. The instructions are for translating academic papers with LaTeX, citations, and tags, but this input doesn’ t contain any of those elements.

Let me think about how to handle this:

1. The input is just “江灏” - a Chinese name
2. The instructions say to translate from Simplified Chinese to English with academic tone
3. The critical requirements mention preserving LaTeX, citations, and tags, but none are present
4. The output rules say to return ONLY the translated text, keep paragraph structure, etc.

Since this is a proper name, the standard translation approach would be: - “江” is the surname (Jiang) - “灏” is the given name (Hao)

In academic contexts, Chinese names are typically romanized using Pinyin. The standard format is either: - Jiang Hao (given name first, Western order) - Hao Jiang (surname first, Eastern order, but less common in English publications)

Most English academic papers use the Western order: Given name + Surname = “Hao Jiang”

However, sometimes the name is kept in the original order: “Jiang Hao”

Looking at the glossary provided: - 机器学习 => machine learning - 深度学习 => deep learning

These don’ t apply to this name translation.

The instructions are very specific about preserving tags, but there are none here. So I should just provide the translation of the name.

For a single name like this, I should just output the romanized version. The most common academic practice is to use Pinyin with the surname first, capitalized: “Jiang Hao”

Wait, let me reconsider. The user might be asking me to translate a name that appears in a scientific paper context. In that case, I should follow academic conventions.

In Chinese academic papers published in English, names are typically written as: - “Jiang Hao” (surname + given name, with both capitalized)

But sometimes in the text, it might be referred to as “Hao Jiang” (given name + surname) when following English conventions.

Since the input is just the name by itself, without context, I should provide the most standard academic romanization: “Jiang Hao”

Actually, let me think about this more carefully. The user might be testing if I, The user has provided a single Chinese phrase: “曾维理”. This appears to be a person’ s name. Let me analyze the requirements:

1. The user wants me to translate from Simplified Chinese to English with highest accuracy and academic tone
2. I must preserve LaTeX commands, math placeholders, citation commands, and tags
3. The output should be ONLY the translated text
4. The glossary provided is:
   - 机器学习 => machine learning
   - 深度学习 => deep learning

However, "曾维理" is a name, not technical content. In academic translation, Chinese names are typically transliterated using Pinyin. "曾维理" would be "Zeng Weili" or "Zeng Wei-li" depending on style guide.

But wait - the user hasn' t provided any paragraph tags or LaTeX content. They' ve just given a name. Let me re-read the instructions carefully.

The instructions say: - "Translate from Simplified Chinese to English with the highest accuracy and academic tone" - "Return ONLY the translated text for the given input (no explanations, no quotes, no headings you invent)" - "Keep one output paragraph per input paragraph; do not merge or split paragraphs" - "**Maintain exact count and IDs of …tags**"

The input is just "曾维理" - no paragraph tags, no LaTeX, no citations. This is just a standalone name.

In academic contexts, Chinese names are typically: 1. Written in Pinyin with surname first: Zeng Weili 2. Or with surname capitalized: ZENG Weili 3. Or with the surname first but separated: Zeng, Weili

Given the academic tone requirement, the most standard format would be "Zeng Weili" (surname first, capitalized appropriately).

Since there are no tags in the input, I don' t need to add any. The instruction says to maintain the exact count - if input has 0, output should have 0.

The glossary provided (机器学习, 深度学习) doesn' t apply to this input.

So the translation should simply be the Pinyin transliteration: "Zeng Weili"

Let me double-check: - Is this a name? Yes, "曾" is a common surname, "维理" is a given name - Academic tone? Using standard Pinyin transliteration is appropriate - No LaTeX/citations to preserve - No

Date: 2025-04-07T00:00:00+00:00

Abstract

In the context of rapid development of civil aviation, flight sequencing and scheduling methods can efficiently utilize limited airspace and runway resources to address delays and congestion in multi-airport terminal areas caused by unbalanced growth in air traffic flow, thereby enhancing overall operational efficiency.

This paper proposes a bi-level model for collaborative sequencing of arrival and departure flights in multi-airport terminal areas that considers runway-corridor relationships. The model fully accounts for the critical role of corridors as entry and exit points of terminal areas, sets corresponding optimization objectives for arrival and departure flights in multi-airport terminal areas based on different traffic scenarios, and employs a bi-level elite strategy genetic algorithm to solve the proposed model. Simulation experiments were conducted on the Shanghai multi-airport terminal area, with typical traffic scenarios selected and matched with optimization cases. Experimental results demonstrate that the model can reduce total departure flight release time by up to 23.83%, or reduce arrival (departure) flight delays by 51.52% (18.05%). The algorithm used in this paper significantly improves flight scheduling efficiency and outperforms both the traditional FCFS (First Come First Service) scheduling strategy and the NSGA-II optimization scheduling strategy.

Full Text

Bi-level Programming Method for Collaborative Sequencing of Arrival and Departure Flights in Multi-airport Terminal Area

Li Tonghe[1], Liu Jixin[1],, *Jiang Hao*[1],, Zeng Weili[1],*, Yang Lei[1]

[1]College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Abstract

Against the backdrop of rapid civil aviation development, flight sequencing and scheduling methods can efficiently utilize limited airspace and runway resources to enhance overall operational efficiency, thereby addressing delays and congestion caused by growing and imbalanced air traffic flow in multi-airport terminal areas. This paper proposes a bi-level model for collaborative sequencing of arrival and departure flights in multi-airport terminal areas that considers runway-corridor relationships, fully recognizing the critical role of corridor entrances/exits as gateways to terminal areas. The model establishes distinct optimization objectives for arrival and departure flights under different traffic scenarios and employs a bi-level elite strategy genetic algorithm for solution. Simulation experiments conducted on the Shanghai multi-airport terminal area with typical traffic scenarios demonstrate that the proposed model can reduce total departure flight release time by up to 23.83% and decrease arrival (departure) flight delays by 51.52% (18.05%). The algorithm significantly improves flight scheduling efficiency, outperforming both traditional FCFS (First Come First Service) strategies and NSGA-II optimization scheduling approaches.

Keywords: multi-airport terminal area; collaborative sequencing; bi-level programming; genetic algorithm; traffic scenarios

1 Introduction

With the flourishing development of civil aviation, multi-airport terminal areas face mounting pressure from delays and congestion due to increasingly imbalanced air traffic flow. Within multi-airport terminal areas, operational coupling exists between arrival and departure flights as well as among flights from different airports. Flights of different natures share limited airspace and runway resources, leading to reduced operational efficiency and increased delays and congestion. In this context, FAA NextGen [?], ICAO ASBU [?], and EUROCONTROL SESAR [?] have all proposed using flight sequencing and scheduling methods to rationally allocate temporal and spatial resources in multi-airport terminal areas and improve operational efficiency.

Compared with single airports, multi-airport systems (MAS) exhibit substantially increased terminal area capacity and traffic flow, which adds complexity to flight scheduling problems [?, ?]. Consider a multi-airport terminal area containing airports A and B: during peak traffic periods, departure flights from airport A may occupy airspace resources needed by flights landing at or departing from airport B, creating potential conflicts; subsequently, changes in arrival flight landing times at airport A due to conflicts with airport B' s flights further tighten runway slot resources at airport A. Against this backdrop of inter-airport operational coupling at both temporal and spatial levels, sequencing and scheduling for multi-airport terminal areas is viewed as a method to resolve uneven spatiotemporal resource allocation within MAS by adjusting relative flight sequences at each airport and exchanging flight slots to avoid potential conflicts, thereby alleviating delays and congestion. In multi-airport terminal areas, the objects of sequence adjustment are airport runways and various waypoints in the airspace. While controllers can direct most flights through waypoints at flexible altitudes and provide guidance near waypoints to avoid conflicts, corridor entrance points—including arrival fixes and departure fixes—serve as mandatory gateways for every flight' s approach descent or departure climb segment, with only 1-2 fixed handoff altitudes. Consequently, corridor entrance points, like airport runways, bear the majority of delays and congestion in multi-airport terminal areas. Zhang [?], Ma [?], and Yuan [?] considered air traffic control handoff separation constraints at corridor entrance points when sequencing flights in multi-airport terminal areas.

As described above, existing corridor research primarily focuses on handoff separation constraints, establishing a preliminary runway-corridor collaborative sequencing framework. However, current studies show deficiencies in allocating flight times and handoff altitudes at corridor entrance points to alleviate terminal area congestion and delays. First, most research inherits from actual air traffic control operations, assuming flights from each airport must follow fixed handoff altitudes when passing corridors, which limits effective airspace resource utilization. This fixed altitude allocation not only increases air traffic control complexity but also introduces potential additional delays. Furthermore, existing flight sequencing models inadequately consider the relative order between consecutive aircraft from the same airport passing the same corridor entrance during climb or approach, meaning unnecessary order adjustments during sequencing may cause conflicts and additional delays. In actual operations, to coordinate conflicts between arrival and departure flights, arrival flights often need to hold in the air or accept radar guidance to specific approach paths. Such operations not only extend flight times and increase fuel consumption but also impose additional workload on air traffic controllers. Therefore, prioritizing arrival flight optimization in collaborative arrival/departure sequencing is crucial, as it directly reduces passenger waiting time and improves overall system efficiency. Although Jiang et al. [**?**, **?**] comprehensively considered the arrival-priority concept for single-airport, single-runway scenarios, its application in multi-airport terminal areas remains exploratory; in multi-airport environments, operational coupling among arrival and departure flights from different airports makes arrival flight prioritization even more necessary.

To address these challenges, this paper first allows each flight to freely select from all available handoff altitudes at its corridor entrance, avoiding delays caused by unnecessary horizontal separation. Second, the model assumes that the relative order between consecutive aircraft using the same runway and corridor entrance cannot be exchanged, preventing additional delays and conflicts from unnecessary order exchanges. Finally, a bi-level programming model is applied in multi-airport terminal areas, with the upper level optimizing arrival flights and the lower level optimizing departure flights, using different optimization objectives for different traffic scenarios.

Specifically, this paper makes the following contributions:

1) In multi-airport terminal areas, unlike the traditional mode of assigning a fixed handoff altitude per airport, this paper proposes a new strategy that opens all available handoff altitudes to all flights at the same corridor entrance. This design avoids unnecessary delays from additional control separation caused by fixed handoff altitudes.

2) This paper assumes that flights departing from and arriving at the same airport via the same corridor entrance maintain their relative order (and vice versa). This assumption simplifies the flight sequencing process, reduces controller workload, and ensures flight safety. The method effectively reduces unnecessary order adjustments and avoids potential conflicts and delays without compromising overall efficiency.

3) This paper applies a bi-level programming model in multi-airport environments, using the upper model to optimize arrival flights and the lower model to optimize departure flights. In each iteration, the upper model optimizes first, then passes results to the lower model; after the lower model optimizes, it passes results back to the upper model. This design ensures higher optimization priority for arrival flights while mitigating the

impact of arrival-departure flight coupling on the optimization model.

The remainder of this paper is organized as follows: Section 1 reviews current flight scheduling research; Section 2 introduces the proposed bi-level collaborative sequencing model for arrival and departure flights in multi-airport terminal areas; Section 3 presents the solution algorithm; Section 4 conducts experimental validation; and Section 5 provides conclusions and future research directions.

2 Literature Review

Flight sequencing research is primarily divided into single-airport and multi-airport flight sequencing. Multi-airport terminal area flight sequencing research draws heavily from single-airport approaches. In single-airport flight sequencing problems, Dear [?] introduced the Constrained Position Shift (CPS) concept in 1976, reducing solution space and improving solution feasibility and computation time. Subsequently, Hu et al. [?] proposed the Receding Horizon Control (RHC) concept, significantly reducing model scale and convergence difficulty while achieving comparable optimization performance, successfully introducing flight scheduling into the pre-tactical phase of air traffic flow management. Hu and Di Paolo [?] then developed a customized 0-1 binary genetic algorithm for arrival flight sequencing and scheduling based on RHC. They further refined this algorithm [?] by improving operator design to maintain chromosome diversity while using binary variables to represent chromosome matrices. CPS constraints and RHC concepts, originating from single-airport research, have become foundational contributions referenced by most multi-airport flight sequencing studies.

Current multi-airport terminal area research is divided into arrival flight sequencing [?, ?, ?], departure flight sequencing [?], and integrated arrival/departure sequencing [?, ?, ?, ?]. For arrival flight sequencing in multi-airport terminal areas, Ma et al. [?] performed runway-corridor collaborative sequencing for arrival flights, optimizing multi-airport system fairness and economy using NSGA-II; Yuan [?] conducted runway-corridor collaborative sequencing for multi-airport system arrival flights using NSGA-II; Jiang et al. [?] further considered crossing points of arrival routes from different airports and sequenced them using Gurobi; Wang [?] incorporated temporal-spatial resource utilization fairness across airports and solved using a multi-objective particle swarm algorithm (FS-MOPSO). In departure flight sequencing research, Wang et al. [?] proposed constraints for shared route segments and waypoints among departure flights from different airports in multi-airport systems, solved using tabu search; Liu et al. [?] abstracted multi-airport terminal area departure sequencing as a two-stage no-wait traveling salesman problem, solved using CPLEX; Zhong et al. [?] considered how arrival flights at airport A during peak arrival periods restrict departure flights at airport B, designing a tabu search algorithm accordingly. In integrated arrival/departure sequencing research, Shi et al. [?] constructed a collaborative sequencing problem solved via simulated annealing; Zhang [?] introduced the “peripheral flight flow” concept to separately handle runway and airspace traffic, also using simulated annealing; Ahmed et al. [?] established a collaborative optimization model for flight sequencing and runway configuration in multi-airport systems, proposing a cooperative co-evolutionary genetic algorithm.

Despite significant progress, several issues remain underexplored in multi-airport terminal area flight scheduling optimization. First, while runway-corridor interactions have gained attention, the relationship between handoff separation and handoff altitude at corridor entrances and its system-wide efficiency impact lacks in-depth investigation. Second, most research optimizes delay as the objective while neglecting distinct optimization needs for arrival and departure flights under different traffic scenarios. Third, most existing studies adopt static or semi-static approaches, ignoring the dynamic time-varying nature of flight flows in actual operations, potentially compromising solution adaptability and robustness. This study addresses these gaps through in-depth corridor constraint analysis and a dynamic bi-level optimization model, proposing a novel collaborative sequencing method that improves scheduling efficiency while ensuring system stability and reliability.

2 Problem Formulation

2.1 Problem Statement

In a multi-airport system (MAS), flights execute critical departure climb or approach descent phases within terminal areas. Departure flights originate from runways and terminate at corridor exits (departure fixes), while arrival flights follow the opposite path from corridor entrances (arrival fixes) toward runways. Both flight types must adhere to strict altitude restrictions at their start and end points, with runway altitude set to zero and corridor entrance altitude set to one or two predetermined handoff altitudes. During approach or departure, flights need not pass through all waypoints in STAR or SID procedures and may traverse them at flexible altitudes.

Given these characteristics, this paper focuses on collaborative sequencing between runways and corridor entrances in multi-airport terminal areas. The objective is to adjust departure flight takeoff times and arrival flight passage times at arrival fixes for each multi-runway airport, tightly coupling them through flight time simulation. After passing arrival fixes, arrival flights execute approach segments before landing; similarly, departure flights execute climb segments after takeoff before passing departure fixes. Arrival and departure flights are coupled on runways: specifically, on pairs of closely spaced parallel runways, arrival-departure flights are subject to wake separation constraints and runway crossing constraints. Flights from different airports sharing the same arrival or departure fix are coupled at corridor entrances through ATC handoff separation constraints, which this paper addresses via flight time simulation. Furthermore, optimization objectives and results vary based on different traffic scenarios—i.e., the relative capacity-flow conditions of the overall MAS and individual airports. This paper addresses these variations through scenario-based discussions. [Figure 1: see original paper] illustrates the problem schematically.

2.2.1 Upper-Level Arrival Flight Optimization Model

Given the higher importance of arrival flights in reducing controller workload, improving airline efficiency, and increasing airport throughput, this paper employs bi-level programming for modeling and solution [?, ?]. Bi-level programming can double the maximum solving capability of heuristic algorithms and reduces overall complexity by focusing each level on optimizing one flight type. The upper model optimizes arrival flights while the lower model optimizes departure flights. In each iteration, the upper model first solves for the optimal arrival flight solution, which is then passed to the lower model. The lower model adjusts departure flights based on the fixed arrival flight sequencing results, then passes its solution back to the upper model for the next iteration. This process continues until both upper and lower model objective function values converge.

In runway-corridor collaborative sequencing research for multi-airport terminal areas, airport interactions occur at corridor sequencing points, while multi-runway interactions exist at each airport. As previously described, runway times (takeoff/landing) and corridor passage times (departure/arrival handoff times) are linked through flight time simulation. This section introduces the bi-level optimization model.

Assuming each airport in the MAS is a multi-runway airport broadens applicability. In multi-runway airports, arrival and departure flights have different optimization objectives, linked through bi-level programming. Departure flight adjustments typically involve ground delay programs (GDP) where aircraft wait at gates without fuel consumption or additional controller workload. Arrival flight adjustments typically involve holding patterns or radar guidance that extend flight time, consuming extra fuel and imposing additional workload on area and approach controllers. Thus, arrival flight adjustments significantly impact flights and ATM/ATFM more than departure adjustments, making arrival flight optimization the primary task and justifying its placement as the upper-level model.

MAS requires different optimization objectives during peak and off-peak periods. During peak periods, delays for both arrival and departure flights are unavoidable, and capacity-flow curves suggest that order changes yield minimal delay improvement. For arrival flights, order changes may require controllers to issue speed adjustments, altitude assignments, and radar guidance to multiple aircraft, significantly increasing workload. For departure flights, tower controllers' primary objective is releasing all flights as quickly as possible to minimize runway occupancy. Therefore, during peak periods, controllers should minimize order shifts for arrival flights and total release time for departure flights. During off-peak periods, controllers can focus on order changes to reduce delays for both flight types.

Thus, the upper model has two optimization objectives: minimizing total order shift for arrival flights during MAS peak periods, and minimizing total arrival flight delays during off-peak periods. The upper model objective function is shown in Equation (1):

$$\begin{aligned} \min \quad & P \cdot \sum_{i=1}^{N} \sum_{j \in \mathcal{F}i^{arr}} \left| O_j^{post} - O_j^{pre} \right| + (1-P) \cdot \sum_{i=1}^{N} \sum_{j \in \mathcal{F}i^{arr}} \left(t_j^{arr} - t_j^{arr,plan} \right) \\ \text{s.t.} \quad & \text{Constraints (2)-(8)} \end{aligned}$$

where P is a binary variable equal to 1 during MAS peak periods and 0 otherwise. For flight j, $i \in \mathcal{ARP}$ represents the i-th airport in the MAS containing N airports, $\mathcal{F}_i^{arr}$ represents all arrival flights at airport i, t_j^{arr} represents the optimized arrival time, and O_j^{pre} and O_j^{post} represent the landing sequence order before and after optimization. If no order change occurs, this objective function equals 0; with order changes, it becomes multiples of 2.

$\mathcal{ARR}_j^t$ represents the planned arrival time for flight j. The model constraints are:

1) Consecutive Arrival Flight Wake Separation

Per ATC separation requirements [**?**], consecutive arrival flights using the same runway must maintain arrival-arrival wake separation W_{jk}^{arr}. Specifically, for consecutive arrival flights j, k using the same runway, the trailing aircraft must maintain separation from the leading aircraft as shown in Equation (2):

$$t_k^{land} - t_j^{land} \geq W_{jk}^{arr}, \quad \forall j, k \in \mathcal{F}_i^{arr}, RW_j = RW_k, O_j^{pre} < O_k^{pre}$$

where $t_j^{land,plan}$ represents flight j' s planned landing time, t_j^{land} represents optimized landing time, RW_j represents flight j' s runway, and W_{jk}^{arr} represents the wake separation between flights j and k.

2) Departure-Arrival Runway Separation

Per ATC separation requirements, if arrival flight k and departure flight j use a pair of closely spaced parallel runways for landing and takeoff respectively, and k' s target landing time is later than j' s target takeoff time, k must wait until j clears the runway before receiving landing clearance. Specifically, arrival flight k must wait until departure flight j passes the runway end or completes the initial turn, as shown in Equation (3):

$$t_k^{land} - t_j^{takeoff} \geq T_{jk}^{D \to A}, \quad \forall j \in \mathcal{F}_i^{dep}, k \in \mathcal{F}_i^{arr}, RW_j \parallel RW_k, t_j^{takeoff,plan} < t_k^{land,plan}$$

where $RW_j \parallel RW_k$ indicates flights j and k use a pair of closely spaced parallel runways, $T_{jk}^{D\to A}$ represents the time for departure flight j to taxi and pass the runway end or complete initial turn, $t_j^{takeoff,plan}$ represents flight j' s planned takeoff time, and $t_j^{takeoff}$ represents optimized takeoff time.

3) Constant Order at Same Arrival Fix

If consecutive aircraft from the same arrival fix landing at the same airport (hereafter called same-path aircraft) exchange order, there is no global efficiency improvement; the leading aircraft experiences useless additional delay and occupies slot resources needed by later same-runway arrival flights. Therefore, this paper assumes same-path aircraft cannot exchange relative order [**?**]. This constraint reduces unnecessary delays and improves efficiency. [Figure 2: see original paper] illustrates this constraint.

The constraint is expressed as:

$$O_j^{post} < O_k^{post}, \quad \forall j,k \in \mathcal{F}_i^{arr}, AF_j = AF_k, RW_j = RW_k, O_j^{pre} < O_k^{pre}$$

where AF_j represents flight j' s arrival fix. This constraint ensures that if flights j and k belong to the same airport arrival flight set $\mathcal{F}_i^{arr}$, share the same arrival fix $AF_j = AF_k$, use the same landing runway $RW_j = RW_k$, and j' s planned arrival time is earlier than k' s ($O_j^{pre} < O_k^{pre}$), then j and k cannot exchange order during approach to landing, expressed as $O_j^{post} < O_k^{post}$.

4) Same-Altitude Handoff Separation (Fixed Transition Height)

To reduce controller workload, multi-airport systems traditionally assign each handoff altitude at an arrival fix to a specific airport. However, this causes unnecessary delays. This paper proposes a new method that eliminates fixed airport-altitude associations, allowing all flights to freely select any available arrival handoff altitude. This approach encourages the algorithm to optimize altitude selection, ensuring different flights choose different altitudes to reduce delays and improve efficiency. The constraint requires sufficient horizontal separation at arrival fixes when altitudes are identical, guiding the algorithm toward diverse altitude selection. [Figure 3: see original paper] illustrates this constraint.

The constraint is expressed as:

$$t_j^{arr} - t_k^{arr} \geq H^{arr} \cdot \delta_{jk}^{arr} - M \cdot (1 - \delta_{jk}^{arr}), \quad \forall j,k \in \mathcal{F}^{arr}, AF_j = AF_k$$

$$\delta_{jk}^{arr} = \begin{cases} 1, & \text{if } h_j^{arr} = h_k^{arr} \\ 0, & \text{otherwise} \end{cases}$$

where h_j^{arr} and h_k^{arr} are decision variables (auxiliary variables), $\mathcal{F}^{arr}$ represents all arrival flights in the MAS, and H^{arr} represents the horizontal handoff separation at arrival fixes. Flights j and k sharing the same arrival fix and handoff altitude must maintain horizontal handoff separation.

5) Arrival Flight Runway Crossing Separation

Per closely spaced parallel runway operations rules, if arrival and departure flights use a pair of parallel narrow-spaced runways and the departure flight' s target takeoff time is later than the arrival flight' s target landing time, the arrival flight must cross the departure runway before the departure flight can take off if the airport lacks end-around taxiways. [Figure 4: see original paper] illustrates this constraint.

The constraint is expressed as:

$$t_k^{takeoff} - t_j^{land} \geq T_{jk}^C, \quad \forall j \in \mathcal{F}_i^{arr}, k \in \mathcal{F}_i^{dep}, RW_j \parallel RW_k, t_j^{land,plan} < t_k^{takeoff,plan}$$

where T_{jk}^C represents the time for the landing flight to cross the adjacent departure runway. This ensures that if arrival flight j' s planned landing time plus runway exit time is still earlier than departure flight k' s planned takeoff time, departure flight k must wait until arrival flight j lands, exits the runway, and crosses the departure runway before taking off.

6) Arrival Flight CPS Constraint

Large order adjustments for arrival flights increase controller workload and complexity. Therefore, a Constrained Position Shift (CPS) constraint [**?**] is imposed to prevent large order changes that cause extended additional flight time and controller workload.

The constraint is expressed as:

$$\left| O_j^{post} - O_j^{pre} \right| \leq MPS_j, \quad \forall j \in \mathcal{F}^{arr}$$

where MPS_j is a constant representing the maximum position shift allowed for flight j (set to 2 in this study), and Δt_j^{MPS} represents the time shift when flight j' s landing sequence is advanced or delayed by one position. This constraint limits landing time adjustments to $[t_j^{land,plan} - MPS_j \cdot \Delta t_j^{MPS}, t_j^{land,plan} + MPS_j \cdot \Delta t_j^{MPS}]$.

7) Flight Time Simulation Equality Constraint

The connection between arrival flight passage time at arrival fixes and landing time—i.e., the simulated approach flight time—is discussed in later sections.

The constraint is expressed as:

$$t_j^{land} = t_j^{arr} + T_j^{smt,arr}, \quad \forall j \in \mathcal{F}^{arr}$$

where $T_j^{smt,arr}$ represents the simulated approach time for flight j, linking optimized arrival time t_j^{arr} with optimized landing time t_j^{land}.

2.2.2 Lower-Level Departure Flight Optimization Model

The lower model solves departure flight optimization. As described, during MAS peak periods, the objective is minimizing total departure flight clearance time; during off-peak periods, the objective is minimizing total departure flight delays. The lower model objective is shown in Equation (9):

$$\begin{aligned} \min \quad & P \cdot \sum_{i=1}^{N} \sum_{j \in \mathcal{F}i^{dep}} \left(t_j^{takeoff} - t_j^{takeoff,plan} \right) + (1-P) \cdot \sum_{i=1}^{N} \sum_{j \in \mathcal{F}i^{dep}} \left(t_j^{dep} - t_j^{dep,plan} \right) \\ \text{s.t.} \quad & \text{Constraints (10)-(17)} \end{aligned}$$

where $\mathcal{F}_i^{dep}$ represents airport i' s departure flight set, $t_j^{takeoff,plan}$ is flight j' s planned takeoff time, $t_j^{takeoff}$ is the optimized takeoff time, and $\mathcal{F}^{dep}$ is all departure flights in the MAS.

Constraints:

1) Consecutive Departure Flight Wake Separation

Per ATC separation rules, consecutive departure flights on the same runway must follow wake separation:

$$t_k^{takeoff} - t_j^{takeoff} \geq W_{jk}^{dep}, \quad \forall j,k \in \mathcal{F}_i^{dep}, RW_j = RW_k, O_j^{pre} < O_k^{pre}$$

where W_{jk}^{dep} represents the departure wake separation between flights j and k, determined by aircraft types (see).

2) Departure Flight Release Separation

Per ATC separation rules, departure flights with the same departure fix must follow release separation:

$$t_k^{takeoff} - t_j^{takeoff} \geq C_{jk}, \quad \forall j,k \in \mathcal{F}_i^{dep}, DF_j = DF_k, O_j^{pre} < O_k^{pre}$$

where DF_j represents flight j' s departure fix and C_{jk} represents the release separation for departure flights j,k passing the same fix.

3) Departure-Arrival Runway Separation

Identical to upper model constraint (3): if arrival and departure flights use closely spaced parallel runways and the arrival flight' s target landing time is later than the departure flight' s target takeoff time, the arrival flight must wait until the departure flight passes the runway end or completes initial turn.

4) Same Departure Fix Order Constancy

Analogous to upper model constraint (4), consecutive departure flights cannot exchange order during climb.

5) Same Departure Altitude Separation

Analogous to upper model constraint (5), if departure altitudes are identical, horizontal separation must be maintained at departure fixes, as shown in [Figure 5: see original paper].

The constraint is:

$$t_j^{dep} - t_k^{dep} \geq H^{dep} \cdot \delta_{jk}^{dep} - M \cdot (1 - \delta_{jk}^{dep}), \quad \forall j, k \in \mathcal{F}^{dep}, DF_j = DF_k$$
$$\delta_{jk}^{dep} = \begin{cases} 1, & \text{if } h_j^{dep} = h_k^{dep} \\ 0, & \text{otherwise} \end{cases}$$

where H^{dep} is the horizontal handoff separation at departure fixes and h_j^{dep} is flight j' s handoff altitude.

6) CPS Constraint for Departure Flights

Analogous to upper model constraint (6), but departure flights cannot depart earlier than planned, only later.

7) Flight Time Simulation Equality Constraint

The connection between departure flight takeoff time and departure fix passage time—i.e., the simulated departure flight time:

$$t_j^{dep} = t_j^{takeoff} + T_j^{smt,dep}, \quad \forall j \in \mathcal{F}^{dep}$$

where $T_j^{smt,dep}$ represents the simulated climb time for departure flight j.

3 Algorithm Design

Heuristic evolutionary algorithms demonstrate significant advantages for solving NP-Hard flight optimization problems. In the proposed model, different MAS states determine optimization objectives, reflecting differences between peak and off-peak periods while avoiding convergence difficulties in bi-level multi-objective evolutionary algorithms. This paper designs distinct objectives for upper and lower models under different scenarios, feeding them into bi-level single-objective heuristic evolutionary algorithms including bi-level Elite Genetic Algorithm (bi-EGA) and bi-level Strengthened Elite Genetic Algorithm (bi-SEGA) as primary optimizers, with bi-level simple Genetic Algorithm (bi-GA) as a baseline for comparison. The "bi-XX" designation indicates a dual-algorithm architecture where two algorithm instances exchange solutions iteratively [?]. These three mainstream single-objective heuristic evolutionary algorithms are applied to the runway-corridor collaborative sequencing model, with experimental comparison identifying the best performer for this bi-level problem.

During algorithm execution, various separation-based constraints must be handled. These constraints are transformed as follows: during each evolutionary iteration of upper and lower levels, after crossover and mutation, constraint violation is assessed. Individuals store genes as arrival time lists (upper level) or departure time lists (lower level). The current individual is merged with the optimal individual from the other level, extracting complete runway schedules and corridor passage schedules. Runway information, airport information, and corridor fix information are extracted for each flight. Based on matching these three information types among flights, separation constraints are evaluated for the current individual' s flights; individuals violating constraints are eliminated.

[Figure 6: see original paper] shows the algorithm flowchart. Input data consists of ADS-B data for arrival and departure flights within a time period. The algorithm steps are:

Step 1 Extract required data from ADS-B, including takeoff/landing times and corridor passage times.

Step 2 Match MAS traffic scenarios and set corresponding optimization objectives.

Step 3 Set algorithm parameters including population sizes, evolution generations, co-evolution iterations, crossover/mutation probabilities, and gradients.

Step 4 Generate initial upper-level arrival flight populations satisfying constraints based on ETOT, ELDT, runway time constraints, and corridor time constraints; arrival times plus approach times yield landing times. Generate initial lower-level populations satisfying inter-model and intra-departure constraints.

Step 5 Calculate fitness values (objective function values) for each upper-level individual, extract the best individual, derive landing times through approach flight time simulation, and pass to the lower model.

Step 6 The lower model calculates fitness for all individuals, finds the best individual, and passes it to the upper model.

Step 7 Update optimal objective function values for both levels in the current co-evolution iteration.

Step 8 Terminate if maximum iterations reached and output results; otherwise return to Step 5.

4 Case Study

4.1 Case Description

The proposed algorithm is validated using Shanghai terminal area simulations. The terminal area contains two airports: ZSSS (Hongqiao) and ZSPD (Pudong). ZSSS is a closely spaced parallel dual-runway airport with end-around taxiways; ZSPD is a four-runway airport comprising two pairs of closely spaced parallel runways. Based on published AIP data, runway configurations are shown in Figure 7: see original paper and Figure 7: see original paper. In actual operations, ZSPD uses runways 16L/34R and 17R/35L for arrivals, and 16R/34L and 17L/35R for departures. This study assumes ZSSS uses runways 18L/36R for arrivals and 18R/36L for departures to reduce model complexity. The two airport centers are 43.98 km apart. Runway configurations are summarized in .

[Figure 8: see original paper] shows the MAS airspace structure, airport locations, and all arrival/departure fixes. The MAS has 5 arrival fixes and 10 departure fixes. A full day of ADS-B data was cleaned; Figure 9: see original paper shows arrival and departure flight tracks in green and red, with sector boundaries in red and blue. The terminal area implements thorough arrival-departure separation, making it suitable for algorithm testing. Engineering classification of arrival tracks based on runway-corridor grouping yields results shown in Figure 9: see original paper. The maximum position shift is set to 2.

4.2 Experimental Design and Parameter Settings

4.2.1 Parameter Settings Per ATC separation requirements [?], arrival-arrival and departure-departure wake separations vary by aircraft type, as shown in and . For Shanghai terminal area, arrival handoff separation is 20 km and departure handoff separation is 30 km. With average corridor speed of 800 km/h, these convert to approximately 90 seconds for arrival and 135 seconds for departure separation. For "departure-before-arrival" operations, the trailing arrival flight waits 45 seconds for the departure flight to pass the runway end or complete initial turn. For "arrival-before-departure" operations without end-around taxiways, the arrival flight waits 45 seconds to cross the departure runway after landing. Maximum position shift is set to 2.

4.2.2 Flight Time Simulation In actual ATM operations, departure flights climb along SIDs to departure fixes, typically reaching fixed altitudes at specific waypoints. Departure climb time simulation calculates distances from airports to each departure fix using SID charts, obtains climb phase speeds from the openAP flight performance database [?], and computes flight times through piecewise calculation. Arrival flights typically do not follow STARs precisely; controllers maximize direct-to-IAF clearances to improve efficiency, consistent with this study' s objectives. For flights requiring increased separation, radar guidance extends flight distance to manually create delays, as shown in [Figure 10: see original paper]. These factors, plus altitude management variations during approach, cause significant time differences even for flights with identical 2D tracks. Therefore, arrival tracks are classified by runway-corridor groups to build a machine learning prediction model. Features including number of other flights during approach, arrival height, and arrival start time are extracted along with approach duration labels. Based on results from [?, ?], Random Forest [?] and XGBoost [?] demonstrate optimal performance, so these models are trained with optimal parameters to simulate approach times $T_j^{smt,arr}$. During each algorithm run, approach times are predicted by randomly selecting between the Random Forest and XGBoost models.

Processing methods:

1) Arrival flights are classified by arrival fix, landing airport, and runway. Since these three elements are fixed per class, they are excluded from features. Number of Other Flights When Approaching, Arrival Height, and Arrival Beginning Time are selected as features, with Arrival Flight Duration as the label. These features directly impact ATC operations and were proven important in [?, ?], maximizing simulation efficiency and performance. The trained models are used during each algorithm iteration.
2) Departure flights typically follow SIDs without trajectory changes. Flights with identical departure airport, runway, and fix have minimal time variation. Using machine learning could cause overfitting or numerical instability, so analytical calculation is preferred. Departure flight time is computed as the sum of climb and level flight times. For flight i, total distance D from runway to departure fix is divided into climb segments, each with average speed v_{ij} and climb rate r_{ij} from the performance database, and altitude change Δh_{ij}. The departure flight time calculation formula is:

$$T_i^{smt,dep} = \sum_{j=1}^{n} \frac{\Delta h_{ij}}{r_{ij}} + \frac{D - \sum_{j=1}^{n} \Delta h_{ij}}{v_{i,\text{cruise}}}$$

Flight times are extracted as labels, with terminal area flight count, handoff altitude, and start time as features for Random Forest and XGBoost training. [Figure 11: see original paper] shows prediction results, and shows model performance.

4.2.3 Traffic State-Based Discussion MAS peak threshold follows [?], defining peak when flow reaches 80% of capacity. shows official hourly capacities for both airports under instrument meteorological conditions. Based on the peak condition and , MAS peak thresholds are calculated: ZSSS peak threshold is 40 flights/hour; ZSPD peak threshold is 73.6 flights/hour. Using 10-minute intervals, thresholds become 7 flights/10min for ZSSS and 12 flights/10min for ZSPD. Six traffic scenarios are tested:

- **Scenario 1**: MAS peak, ZSSS peak, ZSPD off-peak

- **Scenario 2**: MAS peak, ZSSS off-peak, ZSPD peak
- **Scenario 3**: MAS peak, ZSSS peak, ZSPD peak
- **Scenario 4**: MAS off-peak, ZSSS peak, ZSPD off-peak
- **Scenario 5**: MAS off-peak, ZSSS off-peak, ZSPD peak
- **Scenario 6**: MAS off-peak, ZSSS off-peak, ZSPD off-peak

These scenarios are summarized in .

4.3 Algorithm Performance Comparison

Bi-GA, bi-EGA, and bi-SEGA are tested across six scenarios to evaluate convergence and identify the optimal bi-level evolutionary algorithm for each scenario. The red dashed line in figures represents FCFS strategy objective function values (lower is better). Two baselines are compared: FCFS and NSGA-II. Performance comparison charts are compiled in [Figure 12: see original paper] and [Figure 13: see original paper]. Arrival/departure delay comparisons across scenarios are shown in [Figure 14: see original paper] (excluding poorly performing bi-GA to highlight other strategies). In peak scenarios (1-3), where objectives minimize total departure release time rather than delays, bi-level algorithms underperform FCFS and NSGA-II in delay metrics. In off-peak scenarios (4-6), where delay optimization begins, bi-level algorithms outperform both baselines. Analysis proceeds through horizontal and vertical comparisons.

1) Horizontal Comparison: Airport Flow Peak Status

As shown in [Figure 12: see original paper], Scenarios 1-3 have the entire MAS in peak status, optimizing arrival flight order shifts and departure flight total release time. Bi-level elite genetic algorithms (bi-EGA and bi-SEGA) perform excellently, with bi-SEGA achieving a 23.7% improvement in reducing total departure release time in the highest-flow Scenario 3. In Scenarios 4-6 (off-peak), where both arrival and departure objectives minimize delays, bi-SEGA reduces arrival delays by 51.52% and departure delays by 18.05% in Scenario 5. Bi-level elite genetic algorithms effectively optimize according to defined objectives across different traffic scenarios, demonstrating generalization and adaptability.

2) Vertical Comparison: Algorithm Performance and Convergence

Across all scenarios, both bi-level elite genetic algorithms show significantly better optimization performance and convergence stability than bi-GA, which performs poorly and often fails to surpass FCFS, revealing its limitations for complex sequencing problems. Both elite algorithms outperform the non-bi-level NSGA-II in optimization performance, but convergence stability follows the trend bi-EGA > NSGA-II > bi-SEGA overall, indicating that while bi-SEGA offers superior optimization performance, its convergence stability is inferior to bi-EGA and NSGA-II.

Through horizontal and vertical comparisons, bi-EGA and bi-SEGA demonstrate good convergence performance across scenarios and generally outperform

FCFS and NSGA-II, while bi-GA' s limitations yield poor performance. Between bi-EGA and bi-SEGA, both achieve comparable objective values with respective advantages in different scenarios, making them suitable for the proposed runway-corridor collaborative sequencing problem.

5 Conclusions and Future Work

This paper proposes a bi-level programming model for collaborative sequencing of arrival and departure flights in multi-airport terminal areas, solved using bi-level heuristic evolutionary algorithms. The bi-level design ensures priority optimization for arrival flights. Different optimization objectives are set for arrival and departure flights under various traffic scenarios, enabling dynamic sequencing optimization for different time periods: during MAS peak periods, the model focuses on reducing controller workload; during off-peak periods, it emphasizes reducing flight delays. The model design thoroughly examines the gateway role of corridor entrances/exits and introduces flexible handoff altitudes and constant relative order constraints for same-airport same-corridor flights. Bi-level heuristic evolutionary algorithms bi-EGA and bi-SEGA outperform FCFS and non-bi-level NSGA-II, proving suitable for the proposed problem.

This study fills research gaps regarding inadequate corridor entrance analysis and insufficient analysis of varying demands under different multi-airport terminal area traffic scenarios. The results can support actual air traffic control operations. Future research may develop bi-level multi-objective heuristic evolutionary algorithms to address complex demands from CDM stakeholders.

Appendix: Symbol Definitions

Symbol	Definition
AF_j	Arrival fix passed by flight j
C_{jk}	Release separation between flights j and k
T_{jk}^{C}	Time for arrival flight to cross departure runway after landing
$T_{jk}^{D\to A}$	Time for departure flight from takeoff roll to passing runway end or completing initial turn
DF_j	Departure fix passed by flight j
$\mathcal{F}_i^{arr}$	Arrival flight set for airport i
$\mathcal{F}^{arr}$	MAS arrival flight set
$\mathcal{F}_i^{dep}$	Departure flight set for airport i
$\mathcal{F}^{dep}$	MAS departure flight set
H^{arr}	Arrival handoff separation
H^{dep}	Departure handoff separation
h_j^{arr}	Arrival handoff altitude for flight j

Symbol	Definition
h_j^{dep}	Departure handoff altitude for flight j
MPS_j	Maximum position shift allowed for flight j
N	Number of airports in MAS
O_j^{pre}	Planned landing sequence position for flight j
O_j^{post}	Optimized landing sequence position for flight j
Δt_j^{MPS}	Time shift when flight j exchanges one position in sequence
P	Binary variable: $P = 1$ for MAS peak period, else 0
F_{upper}	Upper model objective function
F_{lower}	Lower model objective function
F_{upper}^{delay}	Upper model delay sub-objective
F_{upper}^{order}	Upper model order sub-objective
F_{lower}^{delay}	Lower model delay sub-objective
F_{lower}^{clear}	Lower model total clearance time sub-objective
RW_j	Runway used by flight j
$t_j^{arr,plan}$	Planned arrival fix passage time for flight j
t_j^{arr}	Optimized arrival fix passage time for flight j
$t_j^{land,plan}$	Planned landing time for flight j
t_j^{land}	Optimized landing time for flight j
$t_j^{dep,plan}$	Planned departure fix passage time for flight j
t_j^{dep}	Optimized departure fix passage time for flight j
$t_j^{takeoff,plan}$	Planned takeoff time for flight j
$t_j^{takeoff}$	Optimized takeoff time for flight j
$T_j^{smt,arr}$	Simulated approach time for flight j
$T_j^{smt,dep}$	Simulated climb time for flight j
T_j^E	Runway exit time for flight j

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.