aerospace

MDPI

*Article*

# Multiobjective Optimization Framework for Complex Interdependent Airspace Conflict Resolution

**Yichuan Huang [1,*], Qing Cheng [1,2], Zhihao Luo [1] and Yan Song [2]**

[1] Laboratory for Big Data and Decision, College of Systems Engineering, National University of Defense Technology, Changsha 410073, China; sgggps@163.com (Q.C.); luozhihao15@nudt.edu.cn (Z.L.)
[2] Hunan Advanced Technology Research Institute, Changsha 410073, China; 15111344839@163.com
* Correspondence: huangyichuan20@nudt.edu.cn

**Abstract**

As diverse aircraft increasingly operate in shared airspace, resolving airspace conflicts under complex interdependent requirements has become a core challenge in air traffic management (ATM). Conventional methods struggle to achieve efficient high-quality solutions in high-dimensional multiobjective optimization (MOO) with multiple constraints and strong temporal correlations across airspace groups. This study introduces an optimized airspace conflict resolution method constructed using an improved nondominated sorting genetic algorithm III (NSGA-III) within a three-layer progressive framework. First, the conflict detection layer integrates interval coincidence detection with the separating axis theorem to achieve precise identification of 3D airspace attribute conflicts. Second, the conflict resolution layer establishes a constrained high-dimensional MOO model that minimizes airspace priority, types, quantities, and magnitudes of airspace adjustments. Finally, the airspace group optimization layer introduces neighboring airspace concepts and a dynamic airspace rejection strategy, which are implemented through improved NSGA-III to address strong inter-airspace constraints and domino effects. For NSGA-III, we propose an airspace-group-oriented encoding–decoding mechanism, a conflict-pair-based penalty function design, and a dynamic rejection strategy that jointly enhance convergence and solution diversity in high-dimensional objective spaces. Simulation experiments using real-world airspace demand data show that the proposed method achieves complete conflict resolution among retained demands (100%; mean rejection rate: 7.15%), with lower rejection costs, higher mean HV (1.14 versus 0.49, 0.51, and 0.96), and mean runtime reductions of 78.6%, 35.4%, and 48.5% relative to MOCOA, MORIME, and MOEA/D, respectively. Overall, the proposed method provides a robust and efficient paradigm for intelligent optimization of structured demand-based airspace systems.

**Keywords:** air traffic management; conflict detection and resolution; multiobjective optimization; NSGA-III

Check for updates

Received: 31 August 2026
Revised: 29 September 2026
Accepted: 1 October 2026
Published: 5 October 2026

**Copyright:** © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

## 1. Introduction

The widespread integration of diverse aerial vehicles into modern aviation systems has increased the research value and strategic importance of airspace as the operational environment for aircraft [1–3]. In air traffic management (ATM), proactive airspace adjustment to ensure conflict-free aircraft operations has attracted growing attention [4,5]. In large-scale dense airspace scenarios, spatiotemporal aircraft conflicts pose catastrophic

risks to ground safety, necessitating strict separation of concurrent airspace demands to prevent overlapping deployment [6].

In mission-critical environments, airspace utilization encounters inherent challenges stemming from high demand intensity, limited resource capacity, and rapid temporal evolution [7,8]. These factors collectively contribute to multidimensional conflicts across temporal, vertical, and horizontal dimensions. Airspace conflicts occur when the 3D spatiotemporal separation between airspaces drops below predefined safety thresholds [9]. Airspace conflict resolution (ACR) is envisioned as a pivotal component of future ATM systems, serving as a prerequisite for safe aircraft navigation and ensuring operational safety [10].

ACR tasks involve two fundamental stages—conflict detection and resolution—aimed at generating feasible adjustment strategies for airspace demands [6]. Under varying task-driven scenarios, the required shapes of airspace exhibit heterogeneity among users. Therefore, it is essential to develop a method that is capable of accurately detecting conflicts among heterogeneous airspaces. Moreover, ACR solutions are required to concurrently eliminate conflicts and satisfy multidimensional optimization requirements [11]. However, traditional single-objective approaches employing exhaustive pairwise resolution mechanisms suffer from inefficiency and suboptimal solutions [11]. More critically, strong temporal dependencies may exist among airspaces within the same group, characterized by strict sequential relationships. These constraints significantly complicate the optimization process, leading to challenges such as restricted search space and the emergence of local conflicts during resolution. Based on the above analysis, the ACR problem can be summarized into three key challenges: (a) accurate detection of conflicts among heterogeneous airspaces; (b) efficient airspace conflict resolution while satisfying users' multidimensional optimization requirements; (c) effective handling of strong temporal dependency constraints among interrelated airspaces. This motivates our research toward designing an optimized ACR framework tailored for multidimensional objectives under complex interdependent constraints that is capable of rapidly resolving airspace conflicts while generating appropriate adjustment strategies for users under stringent temporal correlation constraints.

This study explicitly focuses on conflict resolution within demand-based segregated airspace systems. While conventional ATM and ICAO separation standards [12] inform our safety criteria, the core problem centers on managing demanded 3D airspace rather than continuous trajectories. This paradigm applies to: (a) airspace corridor scheduling in urban air mobility (UAM) and unmanned aircraft system traffic management (UTM) within low-altitude operations and (b) military or special missions utilizing temporary segregated areas (TSAs) or temporary reserved areas (TRAs), representing conflict resolution in temporary airspace zones.

This study thus proposes an improved nondominated sorting genetic algorithm III (NSGA-III)-based ACR optimization method. Herein, airspace groups formalize interdependent requirements, representing strong temporal correlations among co-grouped airspaces. The study methodology follows a three-layer progressive architecture:

1. Conflict detection layer: This integrates interval coincidence analysis with the separating axis theorem (SAT) for precise 3D conflict identification.
2. Conflict resolution layer: This formulates an airspace resolution strategy targeting multidimensional adjustment objectives and constructs a multiobjective conflict resolution model.
3. Airspace group optimization layer: This designs a strategy for generating targeted conflict airspace groups (T-CAGs) under complex interdependent requirements and optimizes airspace groups using an improved NSGA-III, enabling efficient conflict resolution of airspace.

Through experimental evaluations, we demonstrate that the proposed approach outperforms existing methods across diverse airspace scenarios. Our key contributions are as follows:

1. Theoretical innovation: This study investigates the ACR problem under large-scale airspace demands. It innovatively proposes complex interdependent requirements between airspace groups and multidimensional adjustment objectives, transforming the problem into a constrained high-dimensional multiobjective optimization (MOO) task based on the conflict resolution strategy. To address the computational challenges in large-scale airspace networks, the concepts of neighboring airspaces and T-CAGs are introduced. These mechanisms prevent secondary conflicts during resolution, reduce the solution space, and enhance the probability of obtaining feasible solutions, thereby accelerating the computation speed.
2. Algorithmic advancements: To tackle this problem, the NSGA-III framework is enhanced to handle high-dimensional multiobjective and multi-constraint challenges in airspace group optimization. The key improvements include (a) encoding and decoding mechanisms tailored for airspace groups, (b) a penalty function design based on conflict pairs, and (c) dynamic airspace rejection strategies that eliminate unresolvable conflicts during iterations. This framework also offers a reference approach for solving airspace demand optimization problems characterized by multiple local optima and complex nonlinearities.

The remainder of this paper is organized as follows:

Section 2 reviews the related work; Section 3 formulates the problem; Section 4 constructs the ACR optimization framework; Section 5 details the NSGA-III enhancement; Section 6 presents an experimental validation through comparative case studies; and Section 7 ends with a discussion and conclusions.

## 2. Related Work

With the continuous growth in air traffic demand, the ATM field has attracted increasing attention [13,14]. Conflict detection and resolution serve as core applications in ATM, specifically within air traffic flow management (ATFM) and airspace management (ASM) [15]. ATFM has become an active research topic in recent years, while research on ASM remains relatively limited.

ATFM aims to balance airspace capacity and traffic demand through flight scheduling, thereby ensuring aviation safety [16]. An ATFM framework with dual conflict objectives was proposed [17] using an MOO algorithm to effectively reduce lateral collision risks in structured airspace while minimizing flight altitude adjustments. To address the common assumption of constant flight speed in existing studies, researchers have introduced speed control variables and knee-guided evolutionary algorithms [18]. Extensive real-world air traffic dataset experiments demonstrated reduced flight delays and conflicts, yet these approaches failed to achieve complete conflict resolution [17,18].

From a trajectory-based operational perspective, a mixed-integer linear programming model was developed [5] by focusing on key airspace waypoints and applying spatial discretization techniques to resolve en-route conflicts. A novel heuristic algorithm was later designed to rapidly identify conflict-free trajectories, resolving aircraft conflicts across 450 traffic scenarios within one minute. Nevertheless, the discretization approach may discard superior trajectory solutions and neglect the temporal dimensions of trajectory operations. In contrast, a collaborative 4D trajectory planning method was proposed [19] using a hybrid NSGA3-SA algorithm (nondominated sorting genetic algorithm and simulated annealing), enhancing global search capabilities for continuous variables. An airspace congestion prediction model and multiobjective trajectory planning framework were es-

tablished based on historical traffic data, with simulations confirming that optimized trajectories could effectively avoid congested zones and achieve complete ACR. Nevertheless, these studies primarily depend on inter-airspace trajectory operational environments. Trajectory discretization may eliminate potentially superior adjustment options, whereas maintaining continuous trajectories greatly increases the complexity of conflict resolution. Consequently, it becomes challenging to prevent the emergence of new conflicts across different airspace sectors, which poses a significant obstacle to achieving rapid and fully conflict-free airspace operations.

The aforementioned studies focus on reactive ACR following conflict emergence, while some researchers have explored conflict preprocessing strategies. To address preflight demand imbalance, a multilevel full-time ACR algorithm was designed [20] incorporating time-based and resource-use conflict strategies. Real-world airport data validated its ability to eliminate conflicts within predefined time slots while balancing capacity and demand. However, the proposed framework did not account for the multiobjective requirements of different airspace users, which may hinder its adaptability in heterogeneous operational scenarios. A multiagent reinforcement learning (RL) algorithm was proposed [21] for preflight planning in which airspace users learned optimal flight strategies through training, markedly improving overall operational efficiency compared to traditional methods. However, the method relies on RL, requiring extensive prior training of flight behaviors. This complicates data acquisition and limits its adaptability to diverse mission scenarios. While ATFM enables real-time short-term ACR via flow control, ASM ensures efficient airspace control through resource optimization, with proper airspace design directly boosting flow management effectiveness [22]. Airspace sectorization was formulated as a multiobjective complex network clustering problem and solved using a decomposition-based discrete particle swarm optimization algorithm [23]. This approach boosted air traffic monitoring efficiency and reduced conflict probabilities, although potential sector boundary overlaps remain. A 4D multiobjective sectorization method was innovatively developed [24] using NSGA-II, integrating prior knowledge and external archives to accelerate convergence and enhance solution diversity. An experimental analysis on Singapore airspace data demonstrated its capability to provide decision makers with optimized sectorization schemes for conflict mitigation. Reasonable sector design supports conflict-free route network efficiency, whereas macro-level airspace structure design reduces ASM complexity. RL was first applied to drone airspace optimization [25] by introducing hierarchically structured airspace dynamically adjusted across mission scenarios, markedly improving airspace utilization efficiency and safety. However, it fails to meet the predefined multiobjective requirements of different airspace users.

Beyond sectorization, novel ASM paradigms have emerged. Free route airspace (FRA) allows aircraft to select flight paths based on user preferences [26]. Unlike fixed-route systems, FRA enhances airspace capacity and efficiency but increases conflict risks. To address these conflicts, the triple heading change (3HC) method uses three successive heading changes in the horizontal plane to bypass a potential collision point while maintaining the required separation [26]. Additionally, a spatial grid-based conflict detection and resolution method was developed [27] to tackle the domino effect caused by spatiotemporal correlations in low-altitude urban airspaces, achieving significant improvements in resolution efficiency and robustness through spatiotemporal strategies. However, it faces challenges in balancing grid granularity with computational complexity and exhibits limited adaptability to diverse airspace environments. Further, a graph RL framework was introduced [28] to mitigate collision risks by addressing information redundancy in existing learning methods and adapting efficiently to high-density airspace environments. However, the GRL approach is constrained by its reliance on predefined graph structures,

which limits its adaptability to dynamic variations in airspace demand under fixed topologies. In summary, these studies ignore complex interdependent requirements between airspaces, particularly strong temporal correlations across sectors. Moreover, an end-to-end framework for comprehensive airspace conflict resolution has yet to be established in the literature.

Synthesizing the above, the current ATM conflict resolution research falls into two categories:

1. ATFM-based approaches (e.g., trajectory optimization and flight scheduling) face limitations such as airspace structure dependency and unstable resolution outcomes.
2. ASM-based approaches (e.g., sectorization and structure optimization) suffer from single-strategy resolution and MOO challenges.

These gaps motivate a new ACR paradigm that simultaneously addresses resolution robustness and multiobjective demands. Consequently, this study proposes a three-layer progressive optimization framework comprising conflict detection, resolution, and airspace group optimization layers. This framework is specifically designed for demand-based 3D airspace reservation management—contrasting with the trajectory-centric approach of conventional airspace systems—to address the specific challenges of this paradigm under complex operational constraints. It operates independently of historical data, supports efficient conflict resolution across heterogeneous scenarios, and balances multiobjective requirements. Within the airspace group optimization layer, enhancements to the NSGA-III—through a novel encoding–decoding mechanism and fitness computation that incorporate airspace group constraints—are designed to address the optimization challenges posed by strong temporal correlations. A side-by-side comparison between the existing literature and the proposed method is presented in Table 1.

**Table 1.** Comparison of existing approaches and the proposed method.

| Method | Resolution Paradigm | Interdependency | Objective Dimension | Conflict Detection | Resolution Completeness | Applicable Domain |
|---|---|---|---|---|---|---|
| EA [17,18] | Trajectory-centric | No | Multiobjective | Geometric analysis | Partial | Conventional ATM |
| MILP-based [5] | Trajectory-centric (discrete) | No | Single-objective | Waypoint discretization | Partial | En-route airspace |
| NSGA3-SA [19] | Trajectory-centric (continuous) | No | Multiobjective | Continuous trajectory | Complete | Conventional ATM |
| Nondominated GA [20] | Resource reservation | No | Single-objective | Time–resource conflict | Complete | Airport airspace |
| Multiagent RL [21] | Trajectory-centric | No | Single-objective | Learning-based | Partial | General airspace |
| Sectorization [23,24] | Airspace structure | No | Multiobjective | Sector boundary | Partial | Airspace sectors |
| 3HC [26] | Free routing | No | Single-objective | Geometric analysis | Partial | FRA |
| Grid-based [27] | Spatial–temporal volumes | Limited (domino effect) | Multiobjective | Grid occupancy | Complete | Low-altitude urban |
| Graph RL [28] | Graph-based | No | Single-objective | Graph analysis | Partial | High-density airspace |
| **Proposed method** | **Demand-based 3D airspace reservation** | **Yes (temporal correlations)** | **High-dimensional MOO** | **SAT + interval analysis** | **Complete** | **TSA/TRA, potential UAM/UTM** |

## 3. Problem Description

This study addresses the conflict resolution problem within a structured demand-based airspace operational environment. In this paradigm, users submit demands for the exclusive use of a 3D airspace to conduct their operations. This model finds application in airspace safety management within UTM/UAM systems and TSA/TRA operations. The following subsections formalize the key entities within this context. To begin with, the parameters and notations used in this section are summarized in Table 2, whereas several essential parameters are elaborated in detail in the subsequent discussion.

**Table 2.** Definition of parameters and notations in Section 3.

| Symbol | Meaning |
|---|---|
| $Airspace_i$ | The $i$-th airspace in the airspace demand set. |
| $(Airspace_i, Airspace_j)$ | The airspace conflict pair composed of the $i$-th and $j$-th airspaces. |
| $H$ | The vertical bounds of the airspace. |
| $S$ | The set of spatial positions of airspace demand. |
| $T$ | The temporal windows of the airspace. |
| $P$ | The priority level of the airspace demand. |
| $\Delta_{\max}$ | The maximum allowable adjustment magnitudes of the airspace. |
| $\Delta_{\min}$ | The safety separation thresholds of the airspace. |
| Distance$(\cdot)$ | The function that calculates the minimum horizontal projection distance between two airspaces. |
| $AG_i$ | The $i$-th airspace group in the airspace group set $AG$. |

### 3.1. Airspace

An airspace refers to an available 3D spatial region for aircraft operations [29]. For any airspace demand submitted by users, its key attributes can be formally characterized using six parameters,

$$Airspace =< H, S, T, P, \Delta_{\max}, \Delta_{\min} > \quad (1)$$

where:

- $H = [h_1, h_2]$: Vertical bounds of the airspace, with $h_1$ and $h_2$ denoting the minimum and maximum altitudes, respectively.
- $S = \{(x_i, y_i)\}$ $(i = 1, 2, ..., n)$: Planar coordinate data representing the airspace's horizontal footprint, interpreted according to its shape type. To account for heterogeneous structural features while maintaining algorithmic expressiveness, this study assumes that all airspaces project as convex polygons or circles. (In practice, for non-convex airspaces, convex decomposition can be employed to partition them into multiple convex polygons.) For convex polygons, $S$ contains the polygon vertices. For circles, $S$ contains the four corners of the tight axis-aligned bounding square. Here, $x_i$ and $y_i$ denote the east–west and north–south coordinates of the $i$-th vertex.
- $T = [t_1, t_2]$: Temporal windows of the airspace application, where $t_1$ and $t_2$ represent the start and end times of the demanded usage period.
- $P$: Priority level of the airspace demand (lower values indicate higher priority).
- $\Delta_{\max} = \{h_{\max}, s_{\max}, t_{\max}\}$: Maximum allowable adjustment magnitudes for vertical ($h_{\max}$), horizontal ($s_{\max}$), and temporal dimensions ($t_{\max}$).
- $\Delta_{\min} = \{h_{\min}, s_{\min}, t_{\min}\}$: Safety separation thresholds are established based on operational safety standards for vertical ($h_{\min}$), horizontal ($s_{\min}$), and temporal separations ($t_{\min}$), where lateral separation specifically denotes the horizontal projection distance between airspaces [30].

### 3.2. Airspace Conflict

An airspace conflict arises when two or more airspace demands overlap spatially or maintain insufficient separation during the same time period, indicating simultaneous violations of safety thresholds in the vertical, horizontal, and temporal dimensions. If conflicts exist between airspaces $Airspace_1$ and $Airspace_2$, they form a conflict pair $(Airspace_1, Airspace_2)$, determined by three criteria:

(a) The vertical intervals overlap or the vertical separation is less than the safety threshold $h_{\min}$;

(b) The temporal intervals overlap or the temporal interval is less than the safety threshold $t_{\min}$;

(c) The horizontal projections overlap or the lateral separation is less than the safety threshold $s_{\min}$.

Let Distance$(Airspace_1, Airspace_2)$ denote the minimum lateral separation distance between $Airspace_1$ and $Airspace_2$. The mathematical formulation for conflict detection is as follows:

$$\begin{cases} \max(h_1^1, h_1^2) - \min(h_2^1, h_2^2) & < h_{\min} \\ \max(t_1^1, t_1^2) - \min(t_2^1, t_2^2) & < t_{\min} \\ \text{Distance}(Airspace_1, Airspace_2) & < s_{\min} \end{cases} \tag{2}$$

As illustrated in Figure 1, the temporal, horizontal, and vertical separations ($\Delta t$, $\Delta s$, $\Delta h$) between $Airspace_1$ and $Airspace_2$ simultaneously satisfy the conflict conditions as (3):

$$\begin{cases} \Delta t < t_{\min} \\ \Delta s < s_{\min} \\ \Delta h < h_{\min} \end{cases} \tag{3}$$

Thus, $Airspace_1$ and $Airspace_2$ form a conflict pair $(Airspace_1, Airspace_2)$.

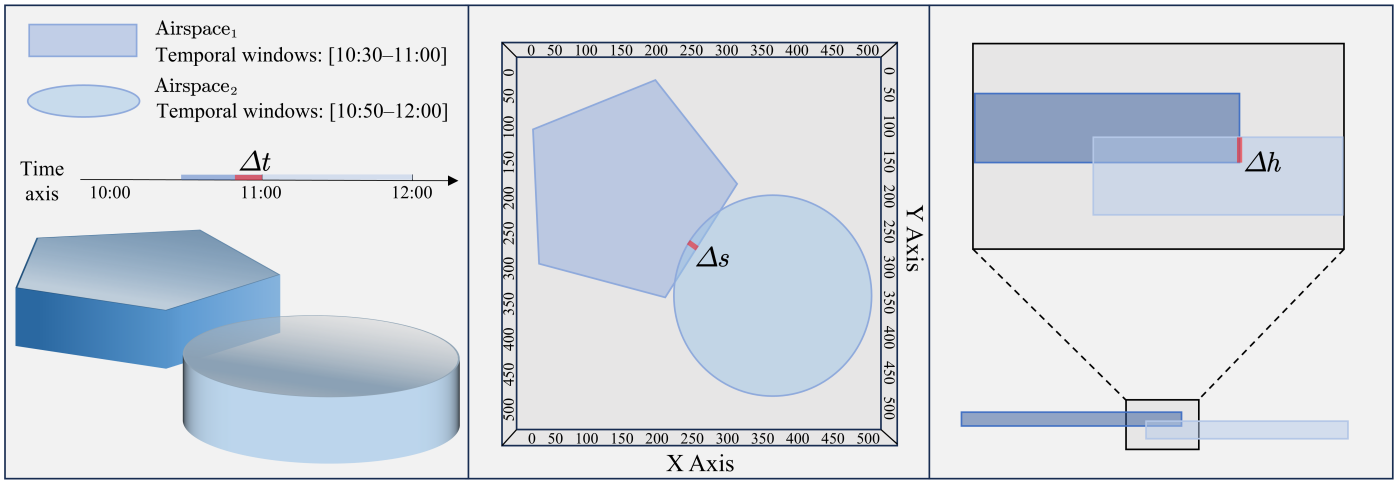



**Figure 1.** The conflict between two airspaces is visualized from three perspectives: the 3D view, the top-down view (XOY plane), and the side view (YOZ plane).

### *3.3. Airspace Group*

After conflict detection, airspace conflicts manifest as conflict pairs. Resolving these conflicts involves adjusting the specific parameters (vertical, horizontal, or temporal attributes) of one airspace demand within a pair. Consequently, ACR strategies are categorized as vertical, horizontal, or temporal adjustments.

Further, certain airspace demands exhibit strong temporal correlations and are organized into airspace groups based on mission-level temporal coupling. Let the airspace group set be $AG = \{AG_1, AG_2, \ldots, AG_n\}$. For any airspace group $AG_i$, its constituent demands serve the same operational mission and maintain prescribed temporal sequences and fixed time intervals. Group membership is determined by these mission-level dependencies, while the fixed intervals specify the relative temporal offsets to be preserved during ACR.

Whenever a retained airspace demand undergoes a temporal adjustment, all the other retained demands within the same group must undergo the same temporal shift to preserve task coordination requirements. This shift is applied equally to the start and end times of each retained demand. Therefore, relative temporal relationships within the same airspace group remain invariant during ACR.

Formally, for any airspace group $AG_i$, let $AG_i^{\mathrm{a}} \subseteq AG_i$ denote its retained demands. Consider arbitrary airspaces $\mathrm{Airspace}_p$ and $\mathrm{Airspace}_q$ in $AG_i^{\mathrm{a}}$, with signed temporal adjustments $\Delta_p$ and $\Delta_q$, respectively. Their temporal adjustments must satisfy:

$$\Delta_p = \Delta_q, \qquad \forall\, \mathrm{Airspace}_p, \mathrm{Airspace}_q \in AG_i^{\mathrm{a}}, \quad i = 1, \ldots, n. \tag{4}$$

## 4. Airspace Conflict Resolution Optimization Framework

This study focuses on developing an ACR optimization methodology tailored for complex interdependent requirements. The research framework is structured into three sequential components:

First, a formal mathematical formulation of airspaces, airspace conflicts, and airspace groups is established. This includes defining airspace attributes, conflict criteria based on simultaneous violations of safety thresholds in vertical, horizontal, and temporal dimensions, and strong temporal interdependent requirements between airspace groups.

Second, the core research systematically addresses complex interdependent requirements in airspace group optimization through a three-layer hierarchical framework. The conflict detection layer establishes an algorithm based on determination criteria by applying interval overlap comparison for vertical and temporal dimensions and the SAT for lateral separation distance computation. Subsequently, the conflict resolution layer formulates a constrained MOO model targeting user-defined adjustment objectives, transforming the problem into a high-dimensional MOO task with explicit safety thresholds ($h_{\min}, s_{\min}, t_{\min}$). Finally, within the airspace group optimization layer, an improved NSGA-III-based framework is introduced to tackle large-scale airspace networks with complex interdependent requirements and multi-constraint challenges. The key improvements of this airspace group optimization method are as follows:

1. To mitigate domino effects [31] and resolution complexity in large-scale airspace demands, this study proposes neighboring airspace identification and T-CAG generation. These strategies prevent secondary conflicts, reduce computational load, and define the workflow for T-CAG formation.
2. To address high-dimensional multiobjective problems in the conflict resolution model, an improved NSGA-III framework is proposed through airspace-group-oriented encoding/decoding designs, penalty function construction (converting constrained problems to unconstrained formulations), and dynamic rejection strategies for unresolvable conflicts during evolutionary iterations.

Third, airspace conflict optimization case experiments are conducted to validate the proposed method. The framework is benchmarked against multiobjective algorithms (MOCOA, MORIME and MOEA/D) under varying maximum adjustment magnitude settings ($\Delta_{\max}$). Comparative analyses focus on ACR completeness, multiobjective metrics (HV and NDS), and computational efficiency. Figure 2 illustrates the overall research framework, highlighting the proposed method's role in balancing computational efficiency and robustness within complex nonlinear airspace systems.

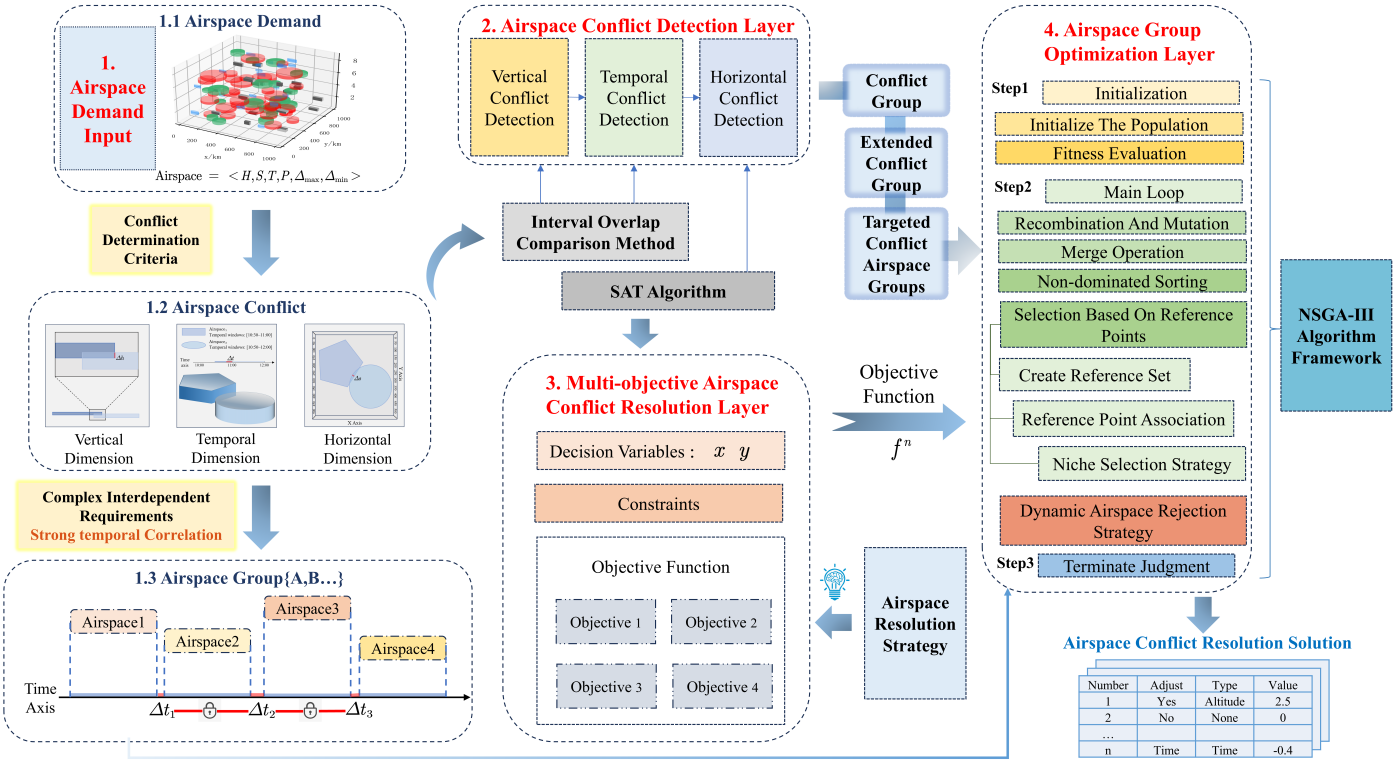



**Figure 2.** Overview of the proposed three-layer ACR framework: conflict detection, multiobjective conflict resolution, and airspace group optimization. Conflicts are identified using interval checks and the separating axis theorem, while resolution schemes are evaluated using four adjustment objectives. Airspace group optimization integrates targeted conflict airspace groups (T-CAGs) and dynamic rejection into an NSGA-III-based search to address interdependent demands.

## 5. ACR Optimization Method Based on Improved NSGA-III

To address complex interdependent requirements in airspace group optimization, this section presents an airspace conflict detection workflow and a multiobjective ACR model followed by an enhanced NSGA-III-based airspace group optimization methodology. In this section, most of the variables appearing in the equations are defined in their immediate context, while the key parameters and the principal symbols used in the pseudocode are declared and summarized in Table 3.

**Table 3.** Definition of parameters and notations.

| Symbol | Meaning |
|---|---|
| $F_M$ | Penalty function combining conflict magnitude and count |
| $M$ | Penalty coefficient |
| $n_c$ | Number of airspace conflicts |
| $K_i$ | Dynamic rejection strategy parameters, where $K_1$, $K_2$, and $K_3$ correspond to the maximum number of rejections, local maximum number of iterations, and the early-stopping threshold, respectively |
| $R_t$ | Combined population generated by merging the parent population $P_t$ and the offspring population $Q_t$ |
| $N_{pop}$ | Population size in NSGA-III evolution |
| $PF_i$ | Pareto front of rank $i$ |
| $P_{t+1}$ | New population generated after evolutionary operations |
| $Z^r$, $Z^s$ | Ideal point set and normalized reference point set |
| $\pi(s)$ | Mapping from individual $s$ to nearest reference point |
| $d(s)$ | Euclidean distance from $s$ to its reference point |
| $\rho_j$ | Niche count of reference point $j$ |

### *5.1. Airspace Conflict Detection Model*

Accurate and efficient conflict detection serves as a prerequisite for ACR [28]. According to the conflict criteria, identifying airspace conflicts requires pairwise checks of 3D attributes. A conflict pair is formed if violations of safety thresholds are detected. This study first detects vertical and temporal attributes using the interval overlap comparison method (2), yielding a set $AG_{ht}$ of airspaces conflicting in vertical and temporal attributes. The critical step lies in horizontal attribute detection, which employs the SAT algorithm.

In the domain of high-precision collision detection, the SAT [32] and GJK algorithms [33] represent the most widely adopted methodologies. For the problem at hand, where horizontal projections of airspaces are either convex polygons or circles, the GJK algorithm fundamentally relies on Minkowski difference construction, which poses challenges in representing circular airspaces owing to the inherent difficulty of approximating continuous curvature with discrete geometric operations. Hence, this study adopts the SAT algorithm.

Based on the SAT, airspace conflict detection in the horizontal dimension involves computing the minimum horizontal projection distance $d_{\min}$ between airspaces. As illustrated in Figure 3, this process includes three steps:

(a) Projection axis calculation. For convex polygons, all edge normals form projection axes; for circles, the line connecting their centers defines the axis; for combinations of convex polygons and circles, the axis includes the closest vertex-center line and polygon edges.

(b) Interval projection. Taking $\text{Airspace}_\text{A}$ and $\text{Airspace}_\text{B}$ as examples, their projected intervals $[PA_{\min}, PA_{\max}]$ and $[PB_{\min}, PB_{\max}]$ yield $d_{\min}$ as the smallest interval difference (5). A negative value indicates penetration depth, implying spatial overlap.

$$d_{\min} = Min[Max(PA_{\min}, PB_{\min}) - Min(PA_{\max}, PB_{\max})] \tag{5}$$

(c) Conflict determination. The computed minimum horizontal distance $d_{\min}$ is compared against the safety threshold $s_{\min}$. A conflict is declared if $d_{\min} < s_{\min}$.

The overall conflict detection workflow is summarized in Algorithm 1, integrating interval overlap comparison for vertical and temporal dimensions with SAT-based horizontal projection distance computation.

Furthermore, for two non-convex polygonal airspaces $Airspace_1$ and $Airspace_2$, convex decomposition techniques can be employed to partition $Airspace_1$ and $Airspace_2$ into several convex polygonal airspaces. Conflict detection is then performed pairwise between these decomposed convex components. If any conflict is detected between any pair of components, $Airspace_1$ and $Airspace_2$ are deemed to be in conflict.

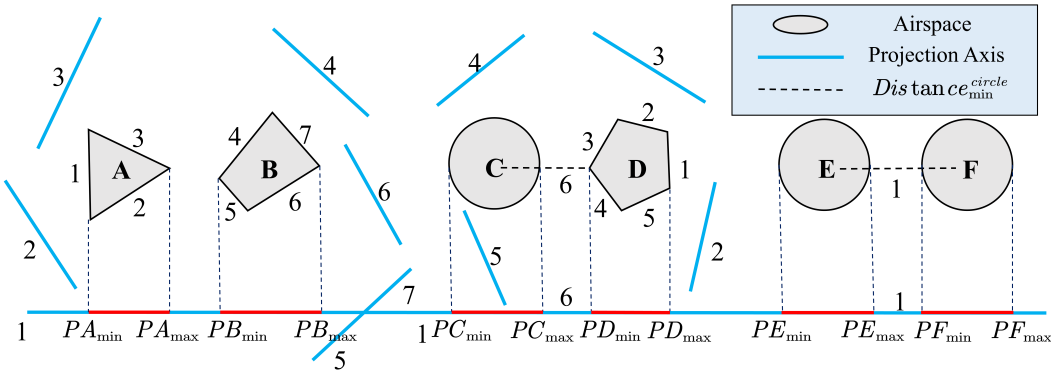



**Figure 3.** Diagram of airspace conflict detection based on the SAT algorithm (airspace conflict detection based on SAT is classified into three cases: two convex polygons, one convex and one circular airspace, and two circular airspaces. Gray shapes represent the horizontal airspace footprints, solid blue lines indicate candidate projection axes, and red segments show the projected intervals. Each case follows a projection-axis workflow for minimum separation computation).

**Algorithm 1:** *Airspace* conflict detection

**Input:** The airspace set to be detected $AG$

**Output:** Conflict pair set $Conflicts$

Initialize the conflict pair set $Conflicts \leftarrow [\,]$
**for** $i \leftarrow 1$ **to** $Len(AG)$ **do**
  **for** $j \leftarrow i+1$ **to** $Len(AG)$ **do**
  $A_1 \leftarrow AG[i]$, $A_2 \leftarrow AG[j]$
  $\Delta_t \leftarrow t_{\min} - \left(\max(A_1^{time}[0], A_2^{time}[0]) - \min(A_1^{time}[1], A_2^{time}[1])\right)$ *// Temporal attribute*
  $\Delta_h \leftarrow h_{\min} - \left(\max(A_1^{altitude}[0], A_2^{altitude}[0]) - \min(A_1^{altitude}[1], A_2^{altitude}[1])\right)$ *// Vertical attribute*
  **if** $\Delta_t > 0$ **and** $\Delta_h > 0$ **then**
  $d_{\min} \leftarrow \text{Distance}(A_1, A_2)$ *// Horizontal distance*
  **if** $s_{\min} - d_{\min} > 0$ **then**
  $Conflicts$.append$((A_1, A_2))$
  **end if**
  **end if**
  **end for**
**end for**

## *5.2. Multiobjective ACR*

### 5.2.1. ACR Strategy

As primary entities in ACR, airspaces vary in importance due to heterogeneous user demands [34]. High-priority airspaces require minimal adjustment, whereas lower-priority airspaces permit greater flexibility. Consequently, the latter take precedence in adjustment. Additionally, for airspace adjustments, the three methods (vertical, horizontal, and temporal) follow a prioritized hierarchy.

Temporal adjustments are generally deprioritized for airspace demands within interdependent airspace groups as temporal adjustments can significantly disrupt mission timing. Specifically, any non-zero temporal adjustment of a retained demand requires the same temporal shift for all the other retained demands in its group to preserve mission-level coordination; otherwise, feasibility cannot be guaranteed. Resolving conflicts using a single parameter that meets safety thresholds suffices, but spatiotemporal adjustments should minimize deviation. Airspace plan optimization requires strict control of the modification count and magnitude. Furthermore, for conflicting pairs in airspace demands that cannot achieve conflict resolution, a rejection operation is permitted, meaning the airspace demand can be canceled, but there are certain limitations on the number of rejected airspace

demands. By integrating these principles, this study formulates an ACR strategy with four prioritized objectives:

- Prioritize adjustments based on ascending airspace priority (from lowest to highest).
- Prefer adjustment types with higher operational priority.
- Minimize the count of adjusted airspaces.
- Minimize adjustment magnitudes.

#### 5.2.2. Multiobjective ACR Model

The construction of decision variables, objective functions, and constraint conditions is carried out sequentially.

(a) Defining decision variables

Decision variables $x_i$, $y_i$, and $z_i$ represent the adjustment type, adjustment magnitude, and rejection status of Airspace$_i$, respectively:

$$x_i = \begin{cases} 0, & \text{no adjustment,} \\ 1, & \text{vertical adjustment,} \\ 2, & \text{horizontal adjustment,} \\ 3, & \text{temporal adjustment.} \end{cases} \tag{6}$$

Here, $x_i = 0$ indicates no intervention, while $y_i$ quantifies the adjustment magnitude for non-zero $x_i$. Additionally, $z_i = 1$ indicates that Airspace$_i$ is rejected, while $z_i = 0$ signifies that it is retained. The values of $x_i$ identify adjustment categories; their associated preference weights are defined separately in the objective function below.

(b) Defining the objective functions

To align with the ACR strategy, objective functions are established across four dimensions: airspace priority, adjustment-type cost, the number of adjusted airspaces, and adjustment magnitude.

Optimization Objective 1:

$$O_1 = \min F_1 = \sum_{i=1}^{n} \mathrm{sgn}(x_i)(P_{\max} - P_i) \tag{7}$$

$F_1$ quantifies the priority-related cost of airspace adjustments. The priority code $P_i \in \{1, 2, 3, 4, 5\}$ follows an inverse numbering convention: 1 denotes the highest priority and 5 the lowest. Here, $P_{\max} = 5$ denotes the fixed maximum numerical priority code. The term $P_{\max} - P_i$ converts the priority code into an adjustment weight, assigning larger penalties to adjustments of higher-priority airspaces. For example, adjusting an airspace with priority code 1, 3, or 5 contributes 4, 2, or 0, respectively, to $F_1$. The lowest-priority class therefore serves as the zero-cost reference for this objective (sgn: sign function).

Optimization Objective 2:

$$O_2 = \min F_2 = \sum_{i=1}^{n} c(x_i) \tag{8}$$

$F_2$ measures the cumulative adjustment-type cost, where $c(x_i)$ maps each adjustment category to its corresponding preference penalty: $c(0) = 0$, $c(1) = w_{\mathrm{v}}$, $c(2) = w_{\mathrm{h}}$, and $c(3) = w_{\mathrm{t}}$.

Here, $w_{\mathrm{v}}$, $w_{\mathrm{h}}$, and $w_{\mathrm{t}}$ are positive dimensionless preference weights for vertical, horizontal, and temporal adjustments, respectively. The baseline configuration adopts $(w_{\mathrm{v}}, w_{\mathrm{h}}, w_{\mathrm{t}}) = (1, 2, 3)$. Temporal adjustments receive the largest penalty because they may require synchronized changes to associated airspace demands. Sensitivity to the

adjustment-type preference ordering is evaluated by exchanging the vertical and horizontal weights while retaining the adjustment-category definitions.

Optimization Objective 3:

$$O_3 = \min F_3 = \sum_{i=1}^{n} \mathrm{sgn}(x_i) \tag{9}$$

The $F_3$ function quantifies the number of adjusted airspaces, and $O_3$ aims to minimize this quantity through evolutionary iterations. Unlike $F_2$, $F_3$ assigns the same contribution to every adjusted airspace irrespective of its adjustment type. Let $N_\mathrm{v}$, $N_\mathrm{h}$, and $N_\mathrm{t}$ denote the numbers of vertical, horizontal, and temporal adjustments. Under the baseline weights, $F_3 = N_\mathrm{v} + N_\mathrm{h} + N_\mathrm{t}$ and $F_2 = F_3 + N_\mathrm{h} + 2N_\mathrm{t}$. Thus, the objectives are related but not equivalent: $F_3$ controls the number of affected airspaces, whereas $F_2$ additionally distinguishes their adjustment-type composition.

Optimization Objective 4:

$$O_4 = \min F_4 = \sum_{i=1}^{n} f(x_i, y_i),$$
$$f(x_i, y_i) = \begin{cases} 0, & x_i = 0, \\ \dfrac{|y_i|}{h_{\max}}, & x_i = 1, \\ \dfrac{|y_i|}{s_{\max}}, & x_i = 2, \\ \dfrac{|y_i|}{t_{\max}}, & x_i = 3. \end{cases} \tag{10}$$

The $F_4$ function quantifies airspace adjustment magnitudes, with $f(x_i, y_i)$ normalizing deviations by the maximum allowable ranges $h_{\max}$, $s_{\max}$, and $t_{\max}$. In this function, $f(x_i, y_i)$ takes the airspace adjustment type and the corresponding adjustment magnitude as inputs and outputs the normalized adjustment magnitude, thereby mapping adjustment values with different physical dimensions into a common dimensionless interval $[0, 1]$. This formulation ensures that the optimization objective $O_4$ minimizes the total normalized deviation from the original airspace configurations.

The number of rejected airspaces is controlled by the rejection-budget constraint and the dynamic rejection mechanism described below and is reported separately as an operational performance metric.

The overall objective $O$ of this multiobjective optimization problem is formulated in Equation (11). The four adjustment-related objectives are minimized jointly subject to the feasibility constraints that require the elimination of conflicts among retained airspace demands.

$$O = \min \left\{ F_1 = \sum_{i=1}^{n} \mathrm{sgn}(x_i)(P_{\max} - P_i), \quad F_2 = \sum_{i=1}^{n} c(x_i), \quad F_3 = \sum_{i=1}^{n} \mathrm{sgn}(x_i), \quad F_4 = \sum_{i=1}^{n} f(x_i, y_i) \right\} \tag{11}$$

(c) Defining the constraints

Adjustment bound constraints $C_1$: For any airspace, adjustments must not exceed predefined limits:

$$\begin{cases} |y_i| \le h_{\max}, & x_i = 1 \\ |y_i| \le s_{\max}, & x_i = 2 \\ |y_i| \le t_{\max}, & x_i = 3 \end{cases} \tag{12}$$

Variable coupling constraints $C_2$: If $x_i = 0$, then $y_i = 0$, enforcing zero adjustments for unselected airspaces.

Airspace operational constraint $C_3$: The number of rejected airspace demands is bounded above (with the maximum allowable number of rejections denoted as $K_1$).

$$\sum_{i=1}^{n} z_i \leq K_1 \tag{13}$$

### *5.3. Airspace Group Optimization Problem*

To address the strong temporal interdependencies in airspace group optimization, this study introduces the concepts of neighboring airspaces and T-CAGs. These mechanisms prevent secondary conflicts during resolution, reduce the computational scale, and enhance feasibility probability while accelerating convergence. To tackle high-dimensional multiobjective tasks in ACR, the NSGA-III framework is enhanced with domain-specific modifications, including encoding and decoding design, penalty function construction, and a dynamic airspace rejection strategy. The generation of T-CAGs is summarized in Algorithm 2.

**Algorithm 2:** Generation of T-CAGs

**Input:** $Airspaces$, $Conflicts$, airspace group set $AG$
**Output:** $T-CAGs$

Initialize neighboring airspace list $Neighbors$ with $Len(Airspaces)$ empty rows
Conflict Group $\leftarrow$ Combine($Conflicts$) *// Generate conflict group*
**for** $i \leftarrow 1$ **to**$Len(Airspaces)$ **do**
  $A_i \leftarrow Airspaces[i]$
  **for each** $A_j \in Airspaces$ **do**
  **if** $\max(A_i^{altitude}[0] - A_j^{altitude}[1],\ A_j^{altitude}[0] - A_i^{altitude}[1]) < h_{\min} + 2h_{\max}$ **and**
  $\max(A_i^{time}[0] - A_j^{time}[1],\ A_j^{time}[0] - A_i^{time}[1]) < t_{\min} + 2t_{\max}$ **and**
  $\text{Distance}(A_i, A_j) < s_{\min} + 2s_{\max}$ **and**
  $A_i \neq A_j$ **then**
  $Neighbors[i].\text{append}(A_j)$ *// Generate neighboring airspaces*
  **end if**
  **end for**
**end for**
ECG $\leftarrow$ Combine($Neighbors,\ ConflictGroup$) *// Extended conflict group*
$T-CAGs \leftarrow$ Combine($AG,\ ECG$) *// Targeted conflict airspace groups*

#### 5.3.1. T-CAG Generation Strategy

Before ACR, conflict pairs are processed to form T-CAGs through the following workflow:

First, conflicting pairs that share common airspaces are aggregated into conflict clusters via traversal.

Second, neighboring airspaces are defined to avoid new conflicts arising from adjustments. For $\text{Airspace}_\text{i}$, $\text{Airspace}_\text{j}$ qualifies as its neighbor if it satisfies (14):

$$\begin{cases} \max(h_1^i - h_2^j, h_1^j - h_2^i) < h_{\min} + 2h_{\max} \\ \max(t_1^i - t_2^j, t_1^j - t_2^i) < t_{\min} + 2t_{\max} \\ \text{Distance}(\text{Airspace}_\text{i}, \text{Airspace}_\text{j}) < s_{\min} + 2s_{\max} \end{cases} \tag{14}$$

Taking the vertical attribute as an example, consider simultaneous adjustments to $\text{Airspace}_\text{i}$ and $\text{Airspace}_\text{j}$, where the maximum adjustment magnitude is $h_{\max}$. If the vertical separation exceeds $h_{\min} + 2h_{\max}$, the probability of vertical conflicts becomes zero;

otherwise, the two airspaces are deemed adjacent. When adjacencies persist across vertical, temporal, and horizontal dimensions, the airspaces qualify as neighboring entities, enabling conflict propagation analysis. Constructing neighboring airspaces reduces the computational scale of the ACR, facilitating the expansion of conflict clusters into extended conflict groups.

Third, given strict temporal synchronization within airspace groups, extended conflict groups are integrated with co-group airspaces to form T-CAGs—the final set requiring resolution. This ensures temporal consistency while maintaining inter-group constraints. Figure 4 illustrates the workflow.

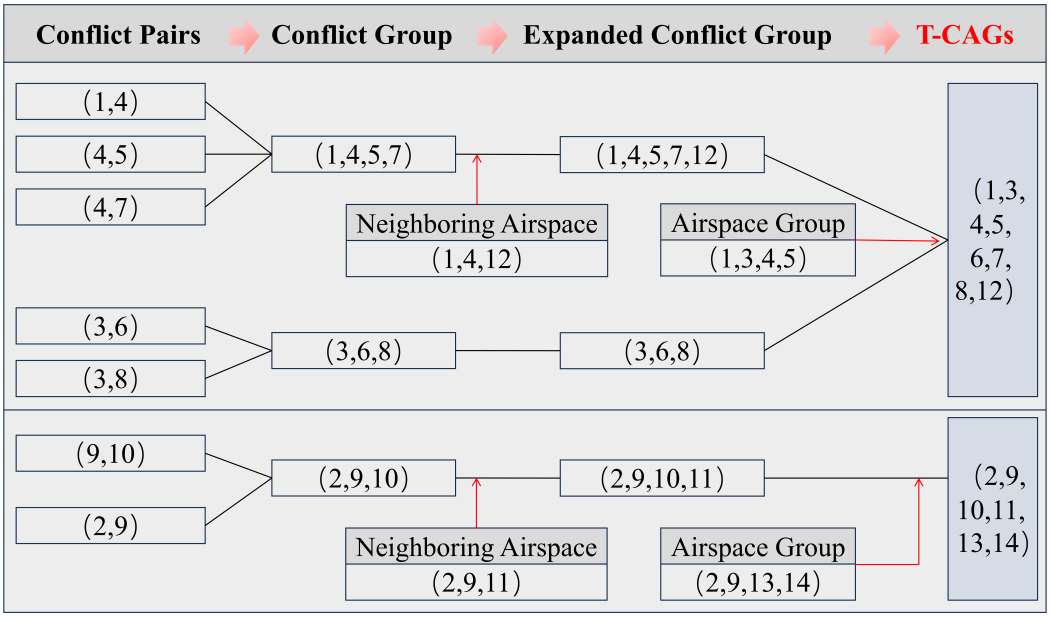



**Figure 4.** Diagram for generating T-CAGs (merge conflict pairs to generate conflict groups, then combine with neighboring airspace to generate expanded conflict groups, and finally combine with airspace groups to generate T-CAGs).

#### 5.3.2. Airspace Group Optimization Method Based on Improved NSGA-III

The airspace group optimization problem is formulated as a high-dimensional multiobjective task based on the multiobjective ACR model. NSGA-III [35] is selected for its ability to balance solution diversity and convergence via nondominated sorting and reference-point mechanisms. The proposed method extends NSGA-III with domain-specific enhancements, supporting the search for feasible nondominated solutions and the generation of complementary trade-off alternatives.

(a) Encoding and Decoding Design Based on Airspace Groups

A real-number encoding strategy is employed for ACR schemes. Each airspace adjustment parameter $\alpha_i$ spans $[0, 8\Delta]$, where $\Delta$ denotes the base encoding block interval. As illustrated in Figure 5, the encoding and decoding scheme comprehensively represents the adjustment options:

- No Adjustment: $\alpha_i \in [0, 2\Delta]$.
- Vertical Adjustment: $\alpha_i \in (2\Delta, 4\Delta]$ with deviation $d_i = \frac{\alpha_i - 3\Delta}{h_{\max}}$.
- Temporal Adjustment: $\alpha_i \in (4\Delta, 6\Delta]$ with deviation $d_i = \frac{\alpha_i - 5\Delta}{t_{\max}}$.
- Horizontal Adjustment: $\alpha_i \in (6\Delta, 7\Delta]$ with deviation $d_i = \frac{\alpha_i - 6.5\Delta}{s_{\max}}$ in the east–west direction; $\alpha_i \in (7\Delta, 8\Delta]$ with deviation $d_i = \frac{\alpha_i - 7.5\Delta}{s_{\max}}$ in the north–south direction.

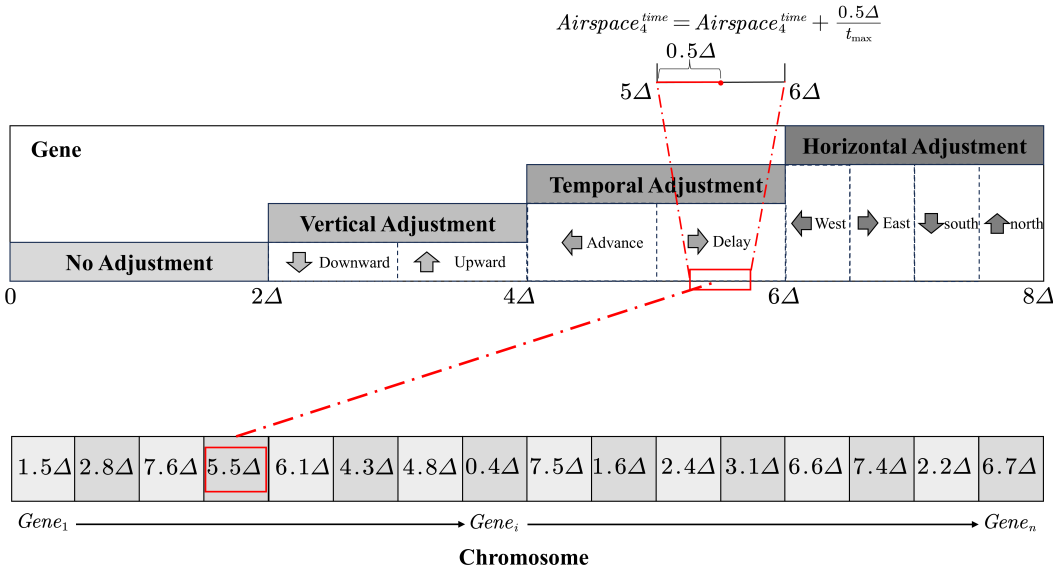



**Figure 5.** Coding diagram of the ACR scheme. For the real-valued encoding of genes, the basic unit on the horizontal axis is Δ, with distinct adjustment types assigned to specific subintervals within the range $[0, 8\Delta]$. For example, when a gene in the chromosome is encoded as $5.5\Delta$, it maps to a temporal adjustment operation with a corresponding time delay of $\frac{0.5\Delta}{t_{\max}}$. Meanwhile, the chromosome consists of $n$ genes, where $n$ represents the number of T-CAGs.

For airspaces within the same group exhibiting strong temporal interdependencies, if any $\alpha_i \in (4\Delta, 6\Delta]$, the entire group's adjustments are mapped to the minimal temporal deviation to preserve synchronization.

(b) Fitness Evaluation with Penalty Function

In general, the fitness value of each individual corresponds to the objective function of the ACR model. To enforce mandatory conflict resolution and transform the conflict-constrained optimization problem into an unconstrained conflict-free formulation, this study introduces a penalty function. The design is as follows:

Let the total feasibility deviation of the airspace be denoted by $d$, where $h_d$, $t_d$, and $s_d$ represent the feasibility deviations in the vertical, temporal, and horizontal dimensions, respectively, satisfying

$$d = h_d + t_d + s_d \tag{15}$$

where:

$$\begin{cases} h_d = \sum\limits_{(i,j)\in Conflicts} \dfrac{h_{\min} - \max(h_1^i - h_2^j, h_1^j - h_2^i)}{h_{\min}} \\ t_d = \sum\limits_{(i,j)\in Conflicts} \dfrac{t_{\min} - \max(t_1^i - t_2^j, t_1^j - t_2^i)}{t_{\min}} \\ s_d = \sum\limits_{(i,j)\in Conflicts} \dfrac{s_{\min} - \text{Distance}(\text{Airspace}_i, \text{Airspace}_j)}{s_{\min}} \end{cases} \tag{16}$$

In (16), the feasibility deviation values of each conflict pair under different dimensions were calculated, and the feasibility deviation characterizes the degree of deviation associated with airspace conflicts. Taking the vertical feasibility deviation $h_d$ as an example, $h_d$ is defined as the ratio between the exceeded portion of the vertical separation—namely the amount by which the current airspace pair's vertical separation surpasses the prescribed vertical safety separation $h_{\min}$—and $h_{\min}$ itself. This ratio represents the normalized vertical conflict intensity of the airspace pair. Larger values of $h_d$ therefore indicate a greater aggregate violation of the minimum vertical separation requirement.

Furthermore, the conflict potential $p$ is defined to quantify the residual conflict level after applying ACR. Accordingly, the penalty function $F_M = M(d + p)$ integrates conflict magnitude ($d$) and count ($p$), where $M$ is a penalty coefficient. The final fitness function combines the original objectives $F = \{F_1, F_2, F_3, F_4\}$:

$$F^n(\alpha_i) = F + F_M \tag{17}$$

(c) Dynamic Airspace Rejection Strategy

Given the presence of large-scale airspaces, interdependent airspace group constraints, and maximum adjustment magnitude limits in the problem domain, unresolved conflict pairs are likely to persist in complex T-CAGs. Achieving complete conflict elimination requires integrating an airspace rejection mechanism into the optimization framework. Inspired by dynamic grouping strategies [36], this study proposes a dynamic airspace rejection strategy that prioritizes unresolvable conflicts while maintaining temporal synchronization.

For airspace rejection, airspaces with the highest occurrence frequency in unresolved conflict pairs are dynamically rejected. If ties exist (equal frequencies), lower-priority airspaces are prioritized to minimize operational impact. The rejection process terminates when all conflicts are resolved ($n_c = 0$) or iteration limits are reached. Based on the above analysis, the ACR workflow defines:

- Maximum rejection attempts $K_1$: Limits the total number of rejections.
- Local iteration cap $K_2$: Restricts iterations per rejection cycle.
- Early stopping threshold $K_3$: Triggers rejection if no improvement occurs over consecutive iterations, enhancing computational efficiency.

(d) Improved NSGA-III

Step 1: Initialization

- Initializing the population: Set $t_2 = 0$ (with $t_1 = 0$ initialized beforehand) and define the population size $N_{pop}$. Each individual in the initial population $P_t$ corresponds to a conflict resolution scheme, with genes randomly generated within $[0, 8\Delta]$.
- Individual fitness evaluation: Evaluate individual fitness by mapping each solution to an adjusted airspace plan using the designed decoding method. Compute the objective function $f^n$ using (11), followed by the penalty function $F_M$ via (15). The final fitness value for all the solutions in $P_t$ is calculated using $F^n(\alpha_i) = F + F_M$.

Step 2: Main Loop

- Crossover and mutation: Simulated binary crossover (SBX) and polynomial mutation are applied to the parent population $P_t$ to generate the offspring $Q_t$.
- Population merging: Merge $P_t$ and $Q_t$ into a combined population $R_t$ of size $2N_{pop}$.
- Nondominated sorting: The combined population $R_t$ is ranked into Pareto fronts $PF_1, PF_2, \ldots$ using nondominated sorting. Individuals from the first fronts are sequentially included into $P_{t+1}$ until the population size reaches $N$. If the boundary front exceeds $N$, only $K = N - |P_{t+1}|$ individuals from the last front $PF_l$ are selected.
- Selection based on reference points [19]: To maintain diversity among the nondominated solutions, the fitness values are first normalized to the interval $[0, 1]$. A structured set of reference points $Z^r$ is then generated from the ideal point set $Z^s$ through a normalization procedure that scales the objective vectors and projects them onto a uniformly distributed hyperplane. Each individual $s \in S_t$ is associated with its nearest reference point $\pi(s)$, and the Euclidean distance $d(s)$ is computed. A niching procedure is then performed to select $K$ individuals based on the niche count $\rho_j$ and the

minimum distance $d(s)$, promoting coverage of different reference directions across the objective space.

- Dynamic Airspace Rejection Strategy: Compute the minimal conflict count $n_c$ in $P_{t+1}$. If $n_c$ shows no improvement over $K_3$ consecutive iterations, reject the airspace with the highest occurrence in unresolved conflict pairs. If ties exist, prioritize lower-priority airspaces for rejection. Update $t_1 = t_1 + 1$ and return to Step 1. Additionally, if $t_2 = K_2$ but $n_c$ remains non-zero, trigger dynamic rejection immediately.

Step 3: Termination criterion

The algorithm terminates when $n_c = 0$ is achieved with $t_2 = K_2$ or $t_1 = K_1$, outputting the final nondominated front $F_1$. Otherwise, increment $t_2$ by 1 and repeat Step 2.

Finally, integrating airspace conflict detection, multiobjective resolution, and airspace group optimization, this study establishes an improved NSGA-III-based ACR workflow (detailed in Algorithm 3 and Figure 6).

**Algorithm 3:** ACR method based on improved NSGA-III

**Input:** $Airspaces$, airspace group set $AG$

**Output:** Final ACR solutions

**Step 1: Airspace data preparation**
Perform conflict detection (Algorithm 1) to obtain $Conflicts$
Generate $T$-CAGs (Algorithm 2)
Initialize rejection counter $t_1 \leftarrow 0$
**Step 2: Initialization**
Set iteration counter $t_2 \leftarrow 0$
Define population size $N_{\text{pop}}$
Generate initial population $P_t \in [0, 8\Delta]$
Compute fitness values $F^n = F(\alpha_i) + F_M$
**Step 3: Evolutionary optimization main loop**
**for** $i \leftarrow 1$ **to** $N_{\text{pop}}$ **do**
  Generate offspring population $Q_t$ using SBX and PM operators
  Merge populations $R_t \leftarrow P_t \cup Q_t$
**end for**
Nondominated sorting: $PF = \{PF_1, \ldots, PF_n\}$
Update population with reference point selection
  | Normalize fitness values: $Normalize(F^n, S_t, Z^r, Z^s)$
  | Associate individuals with reference points: $Associate(S_t, Z^r)$
  | Calculate niche counts $\rho_j$ and select survivors using
  $Niching(K, \rho_j, \pi, d, Z^r, PF_l, P_{t+1})$
  | Update $P_{t+1}$ with $K = N_{pop} - |P_{t+1}|$ candidates
Dynamic airspace rejection: calculate the minimum number of airspace conflicts $n_c$
**if** $n_c$ is unimproved for $K_3$ iterations **or** $(t_2 = K_2 \wedge n_c \neq 0)$ **then**
  Reject frequent conflict airspaces
  $t_1 \leftarrow t_1 + 1$
  **Return to Step 2**
**end if**
**Step 4: Termination criterion**
**if** $(n_c = 0 \wedge t_2 = K_2) \vee t_1 = K_1$ **then**
  Output final Pareto front $PF_1$ and decode ACR solutions
**end if**
**else**
  Increment iteration counter $t_2 \leftarrow t_2 + 1$
  **Return to Step 3**
**end if**

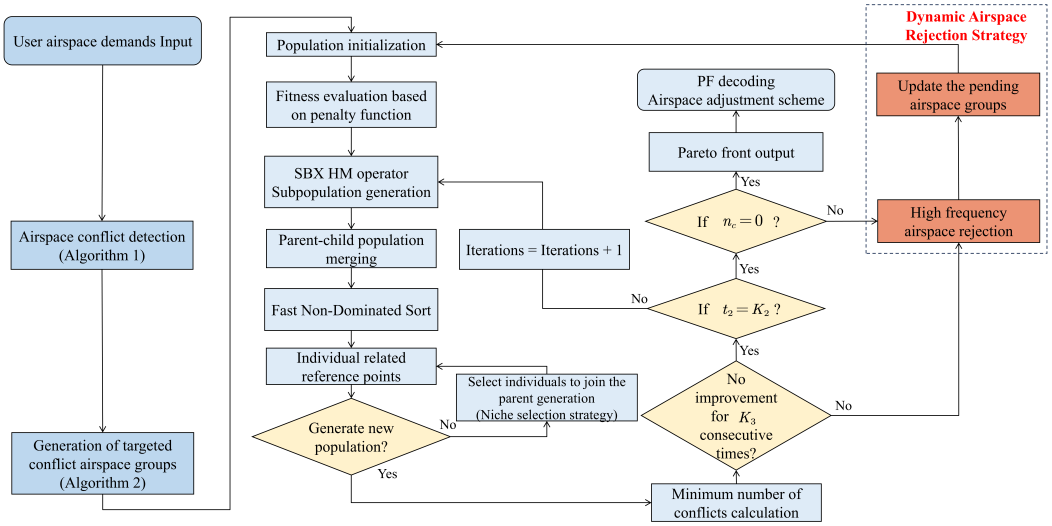



**Figure 6.** Flowchart of ACR based on improved NSGA-III. The left side, highlighted in dark blue, comprises the airspace data preparation phase, which includes user airspace demand input, airspace conflict detection, and generation of T-CAGs. The right side, highlighted in light blue, represents the main workflow of the improved NSGA-III-based algorithm wherein the dynamic airspace conflict rejection strategy is emphasized in orange.

## 6. Airspace Conflict Optimization Case Experiments

This section presents simulation experiments to validate the superiority and robustness of the proposed improved NSGA-III-based ACR method in handling complex interdependent requirements by optimizing airspace plans and comparing the results with state-of-the-art algorithms. To ensure reproducibility and fairness of the experiments, all the experiments in this paper were run in a unified environment where the software used was Pycharm (professional) 2024.3.1 and the core hardware was an Intel(R) Core(TM) i9-14900HX (2.20 GHz).

### *6.1. Airspace Scenario and Parameter Settings*

#### 6.1.1. Airspace Scenario Construction

Before simulation, an airspace scenario was constructed with a horizontal extent of 1000 × 1000 km, an altitude range of 0.3–9.0 km, and a time horizon of 0–600 min. The maximum adjustment magnitudes $\Delta_{max}$ were initialized as $s_{max} = 2.5$ km, $h_{max} = 0.3$ km, and $t_{max} = 5$ min. Since $\Delta_{max}$ directly impacts resolution outcomes, parameter analysis with varying $\Delta_{max}$ was conducted later. The conflict criteria $\Delta_{min}$ were defined using scenario-specific spatial and temporal separation thresholds for airspace reservation demands: horizontal separation $s_{min} = 10$ km, vertical separation $h_{min} = 0.3$ km, and temporal separation $t_{min} = 1.5$ min.

The spatial settings use ICAO Doc 4444, *Procedures for Air Navigation Services—Air Traffic Management*, 16th edition (2016), incorporating Amendment 12 (2024) as a numerical reference [12]. Specifically, the horizontal buffer rounds up the 9.3 km (5 NM) ATS surveillance separation specified in Chapter 8, Section 8.7.3.1; the vertical buffer follows the nominal 300 m (1000 ft) separation specified for operations below FL 290 in Chapter 5, Section 5.3.2(a). These values are adopted as modeling parameters for reservation-level coordination. The temporal buffer of 1.5 min is a scenario assumption that provides a coordination margin between successive occupancy windows. Sensitivity to these thresholds is evaluated in Section 6.6.3.

The airspace demands used in the experiments are constructed based on real-world airspace demand data. To ensure sufficient diversity and coverage of representative operational scenarios, Monte Carlo sampling is employed to generate 100 airspace demands

by perturbing the spatial, temporal, and priority attributes within realistic ranges derived from the real data. A perturbed companion dataset containing 285 airspace demands is provided as Supplementary Dataset S1. Each demand is characterized by horizontal geometry (circular or convex polygonal shapes), vertical bounds, temporal windows, priorities, and airspace group assignments. Figure 7 illustrates a sample construction, and Table 4 lists five example demands.

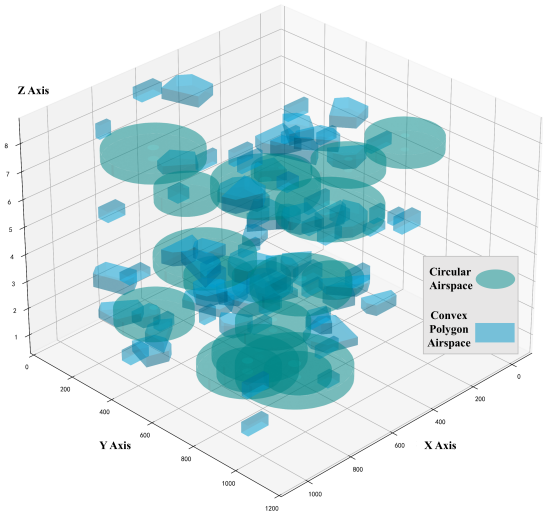



**Figure 7.** Three-dimensional rendering of airspace demands.

**Table 4.** Partial airspace demand data. Coordinates represent polygon vertices or, for circular footprints, the four corners of their tight axis-aligned bounding squares.

| ID | Shape | Horizontal Geometry Coordinates | Vertical Bounds | Temporal Windows | Priority | *AG* |
|---|---|---|---|---|---|---|
| 1 | Circle | (190,861)(190,725)(54,725)(54,861) | [7.6, 8.1] | [314, 429] | 1 | A |
| 20 | Circle | (613,779)(613,663)(497,663)(497,779) | [0.6, 0.9] | [260, 340] | 4 | / |
| 40 | Convex Polygon | (293,678)(259,703)(182,678)(293,599)(306,639) | [6.6, 7.1] | [157, 273] | 2 | B |
| 60 | Convex Polygon | (404,758)(365,729)(351,684)(365,638)(453,609)(492,638) | [7.9, 8.3] | [231, 338] | 2 | / |
| 80 | Convex Polygon | (736,641)(736,618)(641,618)(641,641) | [7.9, 8.4] | [208, 260] | 1 | C |

#### 6.1.2. Algorithm Parameter Settings

To evaluate the proposed method's effectiveness, two state-of-the-art multiobjective algorithms, MOCOA [37] and MORIME [38], together with the classical MOEA/D algorithm [39], are selected for comparison. All the algorithms are evaluated under a common ACR framework using the same airspace demand data, T-CAG generation strategy, encoding–decoding mechanism, conflict-pair penalty function, and dynamic rejection mechanism to ensure a fair comparison.

The same dynamic rejection procedure is applied to NSGA-III, MOCOA, MORIME, and MOEA/D. Rejection is triggered only when conflicts remain and either the minimum residual conflict count has not improved for $K_3$ consecutive iterations or the local iteration limit $K_2$ is reached. The demand involved in the largest number of remaining conflict pairs is rejected; ties are resolved by prioritizing lower-priority demands, corresponding to larger priority codes. The retained-demand set is then updated, and local optimization is reinitialized. The cumulative rejection budget is $K_1$ for every algorithm. Runs terminating with residual conflicts are classified as incomplete resolution. Although the rejection rules are shared, the rejected demands may differ because the algorithms produce different residual conflict configurations.

The core parameters are summarized in Table 5, where the population size, iteration settings, rejection budget, and penalty coefficient remain consistent across methods.

The core parameters of the baseline algorithms are configured based on empirical values [37–41].

After scenario initialization, each algorithm is evaluated over 20 independent simulation runs using the same airspace demand data for the corresponding experimental configuration.

**Table 5.** Core algorithm parameter settings.

| Algorithm | Parameter | Value |
|---|---|---|
| Shared settings | $N_{\text{pop}}$ (Population size) | 150 |
| | $K_1$ (Maximum number of rejected demands) | 10 |
| | $K_2$ (Maximum number of local iterations) | 150 |
| | $K_3$ (Iteration parameter) | 20 |
| | $M$ (Penalty coefficient) | 1000 |
| NSGA-III | $p_{\text{c}}$ (Crossover probability) | 0.9 |
| | $p_{\text{m}}$ (Mutation probability) | 0.05 |
| | $\eta_{\text{c}}$ (Crossover distribution index) | 30 |
| | $\eta_{\text{m}}$ (Mutation distribution index) | 20 |
| MOCOA | $\epsilon$ (Control parameter of Bernoulli mapping) | 0.3 |
| | $L$ (Position update control constant) | $\{1, 2\}$ |
| MORIME | $\omega$ (Soft-rime cycle parameter) | 5 |
| MOEA/D | $N_{\text{w}}$ (Number of weight vectors) | 150 |
| | $T$ (Neighborhood size) | 20 |
| | $p_{\text{nb}}$ (Neighborhood mating probability) | 0.9 |
| | Decomposition method | Tchebycheff |

### 6.2. Algorithmic Time Complexity and Computational Cost Statement

For time complexity formalization, we define: $N$ (the number of objectives), $N_{rv}$ (the number of reference vectors), and $D$ (the number of T-CAGs). The algorithm's time complexity is shown in (18).

$$T_{\text{total}} = \mathcal{O}\Big(K_1 K_2 \Big(N_{pop}^2 N + N N_{pop} N_{rv} + N_{pop} D^2 + N_{pop} D\Big)\Big) \approx \mathcal{O}\Big(K_1 K_2 \Big(N_{pop}^2 + N_{pop} N_{rv} + N_{pop} D^2\Big)\Big) \tag{18}$$

The computational time complexity of the key iteration stage—including nondominated sorting, niching assignment, penalty evaluation, and crossover/mutation operations—along with the computational overhead (based on the average of 20 experimental runs, yielding a mean computation time of 51.76 s) are summarized in Table 6.

Based on the time complexity across iteration stages and considering $D \gg N$, the proposed improved NSGA-III is identified as penalty-function-dominated. The results demonstrate that penalty evaluation accounts for 91.1% of the total computational overhead, which aligns with the theoretical analysis.

**Table 6.** Computational cost breakdown per iteration stage of the improved NSGA-III.

| Iteration Stage | Time Complexity | Computational Cost (%) |
|---|---|---|
| Nondominated sorting | $\mathcal{O}(N_{\text{pop}}^2 N)$ | 4.7 |
| Niching assignment | $\mathcal{O}(N N_{\text{pop}} N_{\text{rv}})$ | 3.4 |
| Penalty evaluation | $\mathcal{O}(N_{\text{pop}} D^2)$ | 91.1 |
| Others (crossover, mutation, etc.) | $\mathcal{O}(N_{\text{pop}} D)$ | 0.8 |

### 6.3. ACR Performance Comparison

Prior to ACR, a conflict detection protocol was applied to 100 airspace demands. Based on Algorithm 1 (conflict detection workflow), 14 conflict pairs were identified:

[(11,13), (13,61), (17,62), (17,69), (25,89), (40,94), (42,47), (44,65), (45,78), (48,64), (48,75), (51,72), (69,84), (80,92)].

Subsequently, Algorithm 2 generated 47 T-CAGs, significantly reducing computational complexity by shrinking the solution space. The generated T-CAGs are shown in Figure 8.

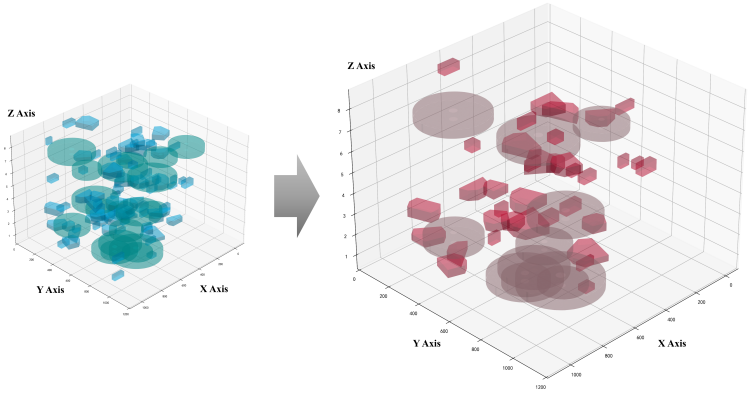



**Figure 8.** Schematic diagram of the generation of T-CAGs. The left panel shows the original set of airspace demands, while the right panel shows the demands included in the T-CAGs, formed by expanding conflict groups to include neighboring airspaces and then incorporating temporally coupled demands from the same airspace groups.

Finally, the NSGA-III-based method (Algorithm 3) was applied to conflict resolution. Figure 9 illustrates a representative run in which seven airspace demands were rejected and no conflicts remained among the retained demands. Complete conflict resolution is defined as a feasible allocation with no remaining conflict pairs among retained demands after rejection subject to the prescribed rejection budget. The operational cost of rejection is quantified by the rejection rate, i.e., the ratio of the number of rejected demands to the initial number of demands. Mean rejection rates are computed across all runs and reported alongside resolution frequencies.

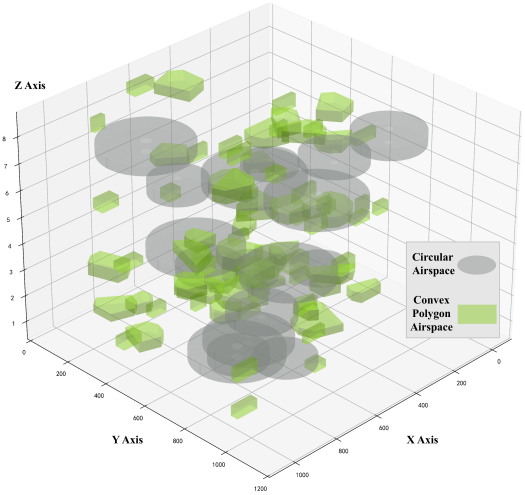
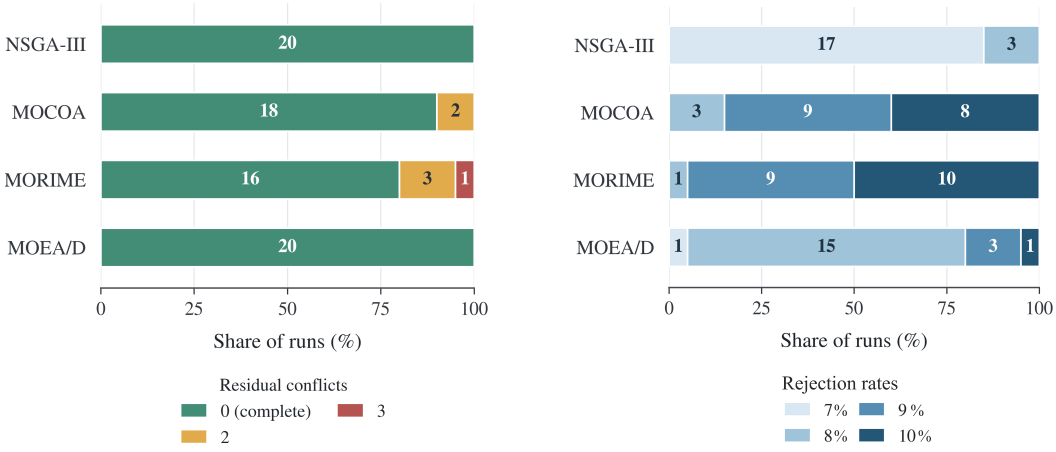



**Figure 9.** Representative ACR result with complete conflict resolution among 93 retained demands after rejecting seven of the initial 100 demands (7%).

Figure 10 presents the distributions of remaining conflict pairs and rejection rate, while Table 7 summarizes the resolution frequencies, rejection rate and descriptive statistics. NSGA-III and MOEA/D achieved complete conflict resolution in all 20 runs, with $0.00 \pm 0.00$ remaining conflict pairs, supporting the applicability of the proposed framework to both multiobjective optimization algorithms in the tested scenario. Their rejection rates were $(7.15 \pm 0.37)\%$ and $(8.20 \pm 0.62)\%$, respectively. At the same observed

frequency of complete resolution, the proposed method reduced the mean rejection rate by approximately 12.8% relative to MOEA/D and exhibited lower variability, demonstrating advantages in demand retention and run-to-run stability.

MOCOA and MORIME achieved complete resolution in 18 and 16 runs, respectively, with mean residual conflict counts of $0.20 \pm 0.62$ and $0.45 \pm 0.94$ across all 20 runs. Their mean rejection rates were $(9.25 \pm 0.72)\%$ and $(9.45 \pm 0.60)\%$, respectively. Compared with these two algorithms, the proposed method achieved a higher observed frequency of complete resolution with fewer rejected demands, providing a more favorable balance between conflict elimination and demand retention. Furthermore, the proposed method achieved zero residual conflicts across its 20 independent runs, with rejection counts ranging from 7 to 8, indicating consistent and robust ACR performance on this instance.




(**a**) Remaining conflict pairs (**b**) Rejection rate

**Figure 10.** Distributions of ACR outcomes over 20 independent runs: (**a**) remaining conflict pairs; (**b**) rejection rate.

**Table 7.** ACR performance over 20 independent runs. The numbers of remaining conflict pairs and rejection rates are reported as mean ± sample standard deviation.

| Algorithm | Complete Resolution (Observed Frequency) | Remaining Conflict Pairs | Rejection Rate (%) | Rejection Count Range |
|---|---|---|---|---|
| NSGA-III | 20/20 (100%) | $0.00 \pm 0.00$ | $7.15 \pm 0.37$ | 7–8 |
| MOCOA | 18/20 (90%) | $0.20 \pm 0.62$ | $9.25 \pm 0.72$ | 8–10 |
| MORIME | 16/20 (80%) | $0.45 \pm 0.94$ | $9.45 \pm 0.60$ | 8–10 |
| MOEA/D | 20/20 (100%) | $0.00 \pm 0.00$ | $8.20 \pm 0.62$ | 7–10 |

Table 8 presents three representative nondominated solutions evaluated using the four adjustment objectives. Solution 3 yields $(F_1, F_2, F_3, F_4) = (13.000, 9.000, 5.000, 1.265)$ with the lowest adjustment-type cost, adjustment count, and adjustment magnitude among the listed solutions. In contrast, Solution 2 achieves the lowest priority-related cost, $F_1 = 11.000$, while requiring six adjustments and a larger normalized adjustment magnitude. These alternatives illustrate the trade-off between priority preservation and adjustment intervention, providing different feasible choices according to operational preferences.

**Table 8.** Representative nondominated solutions for the four adjustment objectives.

| No. | $F_1$ | $F_2$ | $F_3$ | $F_4$ |
|---|---|---|---|---|
| 1 | 12.000 | 11.000 | 6.000 | 1.884 |
| 2 | 11.000 | 10.000 | 6.000 | 1.964 |
| 3 | 13.000 | 9.000 | 5.000 | 1.265 |

### *6.4. MOO Performance Metric Comparison*

To evaluate the proposed method for complex interdependent ACR, HV and NDS were used to assess the obtained solution sets. HV measures the dominated volume relative to a common reference point, whereas NDS counts the obtained feasible nondominated solutions. The results are summarized in Table 9 and Figure 11.

All four objectives are minimized. To account for their different scales, the objective values were normalized as

$$z_j = \frac{F_j - L_j}{U_j - L_j}, \qquad j = 1, \ldots, 4, \tag{19}$$

where $L_j$ and $U_j$ are the minimum and maximum values of objective $j$, pooled from the feasible nondominated sets obtained across all the algorithms, runs, and configurations. The resulting common bounds were $\boldsymbol{L} = (8, 5, 3, 0.884)$ and $\boldsymbol{U} = (16, 13, 7, 5.377)$. These empirical bounds were fixed and applied consistently to all the HV calculations.

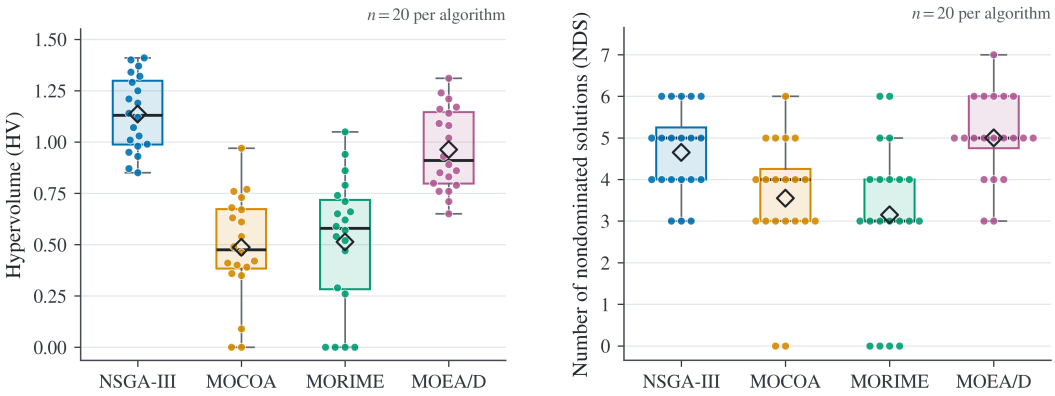
The reference point was fixed at $\boldsymbol{r} = (1.1, 1.1, 1.1, 1.1)$ in the normalized objective space, placing each coordinate 0.1 beyond the corresponding empirical upper bound. For a normalized feasible nondominated set $\mathcal{Z}$, HV was computed as

$$\mathrm{HV}(\mathcal{Z}; \boldsymbol{r}) = \lambda_4 \left( \bigcup_{z \in \mathcal{Z}} \prod_{j=1}^{4} [z_j, r_j] \right), \tag{20}$$

where $\lambda_4$ denotes four-dimensional Lebesgue measure and $\mathrm{HV}(\varnothing; \boldsymbol{r}) = 0$. Under these common settings, a larger HV indicates a greater dominated volume, jointly reflecting the location and spread of the solution set. All the reported HV values were computed using this normalization.

**Table 9.** MOO performance metrics for NSGA-III, MOCOA, MORIME, and MOEA/D.

| Test No. | HV↑ | | | | NDS↑ | | | |
|---|---|---|---|---|---|---|---|---|
| | **NSGA-III** | **MOCOA** | **MORIME** | **MOEA/D** | **NSGA-III** | **MOCOA** | **MORIME** | **MOEA/D** |
| 1 | 1.34 | 0.42 | 0.74 | 1.21 | 6.00 | 5.00 | 4.00 | 7.00 |
| 2 | 1.21 | 0.35 | 0.86 | 1.14 | 5.00 | 4.00 | 4.00 | 5.00 |
| 3 | 0.98 | 0.54 | 0.52 | 1.08 | 5.00 | 4.00 | 3.00 | 5.00 |
| 4 | 0.93 | 0.68 | 0.00 | 0.85 | 4.00 | 4.00 | 0.00 | 5.00 |
| 5 | 1.19 | 0.49 | 0.00 | 0.93 | 3.00 | 3.00 | 0.00 | 6.00 |
| 6 | 1.12 | 0.36 | 0.47 | 1.02 | 6.00 | 5.00 | 5.00 | 5.00 |
| 7 | 1.07 | 0.00 | 0.65 | 1.16 | 5.00 | 0.00 | 3.00 | 5.00 |
| 8 | 1.41 | 0.41 | 0.26 | 1.17 | 3.00 | 3.00 | 6.00 | 4.00 |
| 9 | 1.25 | 0.46 | 0.54 | 0.83 | 4.00 | 3.00 | 6.00 | 3.00 |
| 10 | 1.01 | 0.76 | 0.71 | 0.89 | 4.00 | 6.00 | 4.00 | 6.00 |
| 11 | 0.85 | 0.97 | 0.62 | 0.76 | 4.00 | 3.00 | 3.00 | 6.00 |
| 12 | 1.14 | 0.77 | 0.59 | 1.31 | 4.00 | 5.00 | 3.00 | 6.00 |
| 13 | 1.03 | 0.63 | 0.00 | 0.80 | 6.00 | 4.00 | 0.00 | 5.00 |
| 14 | 0.95 | 0.09 | 0.29 | 0.76 | 6.00 | 4.00 | 3.00 | 5.00 |
| 15 | 0.87 | 0.61 | 0.79 | 0.71 | 6.00 | 4.00 | 3.00 | 5.00 |
| 16 | 0.99 | 0.73 | 0.00 | 0.65 | 4.00 | 5.00 | 0.00 | 4.00 |
| 17 | 1.40 | 0.40 | 0.57 | 0.86 | 5.00 | 3.00 | 3.00 | 4.00 |
| 18 | 1.37 | 0.39 | 0.66 | 0.79 | 5.00 | 3.00 | 4.00 | 6.00 |
| 19 | 1.32 | 0.00 | 0.94 | 1.24 | 5.00 | 0.00 | 4.00 | 3.00 |
| 20 | 1.29 | 0.67 | 1.05 | 1.09 | 3.00 | 3.00 | 5.00 | 5.00 |
| Mean | 1.14 | 0.49 | 0.51 | 0.96 | 4.65 | 3.55 | 3.15 | 5.00 |
| Std. dev. | 0.18 | 0.26 | 0.32 | 0.20 | 1.04 | 1.50 | 1.87 | 1.03 |




(**a**) HV (**b**) NDS

**Figure 11.** Boxplots comparing MOO performance metrics across algorithms under a common ACR framework. (**a**) HV; (**b**) NDS.

#### 6.4.1. Convergence Analysis

The proposed method achieved the highest mean HV, $1.14 \pm 0.18$, compared with $0.49 \pm 0.26$ for MOCOA, $0.51 \pm 0.32$ for MORIME, and $0.96 \pm 0.20$ for MOEA/D. Its mean HV exceeded that of MOEA/D by approximately 18.0%, indicating greater dominated objective-space volume under the common evaluation settings. Its HV standard deviation was also the smallest among the four algorithms, supporting more stable performance across independent runs on this instance.

NSGA-III combines nondominated sorting with reference-direction-based niching, which prioritizes underrepresented directions during environmental selection and helps to retain complementary trade-off solutions, validating its superior convergence in high-dimensional ACR.

MOCOA and MORIME produced no feasible nondominated solutions in two and four runs, respectively; these runs were retained with HV and NDS set to zero. NSGA-III and MOEA/D returned feasible nondominated solutions in every run. The all-run HV results therefore reflect both the ability to obtain feasible solutions and the quality of the resulting sets.

#### 6.4.2. Diversity Analysis

NDS is used here to describe the number of feasible trade-off alternatives. The mean NDS values were $4.65 \pm 1.04$ for NSGA-III, $3.55 \pm 1.50$ for MOCOA, $3.15 \pm 1.87$ for MORIME, and $5.00 \pm 1.03$ for MOEA/D. NSGA-III returned more solutions on average than MOCOA and MORIME, while MOEA/D achieved the highest mean count. Nevertheless, NSGA-III attained a higher HV with slightly fewer solutions than MOEA/D, indicating that its advantage lies in dominated-volume performance rather than solution count alone.

NSGA-III consistently returned 3–6 feasible nondominated solutions, with less variation in NDS than MOCOA and MORIME. This indicates reliable availability of alternative resolution plans across repeated runs.

#### 6.4.3. Statistical Analysis

The independent run distributions were compared using two-sided Mann–Whitney $U$ tests with the asymptotic approximation with tie and continuity corrections. Holm correction was applied jointly to all 12 comparisons (six algorithm pairs for each of the two metrics) with a significance level of $\alpha = 0.05$. Effect sizes were quantified using estimates of the Vargha–Delaney $A_{12}$ measure,

$$A_{12} = \Pr(M_A > M_B) + \tfrac{1}{2}\Pr(M_A = M_B), \tag{21}$$

where $M_A$ and $M_B$ denote metric values from independent runs of algorithms $A$ and $B$, respectively. Values above 0.5 indicate a tendency for algorithm $A$ to yield larger metric values.

As reported in Table 10, NSGA-III achieved significantly higher HV than MOCOA and MORIME, with $p_{\text{Holm}} = 1.480 \times 10^{-6}$ and $4.279 \times 10^{-6}$, and corresponding effect sizes of 0.990 and 0.970. The comparison with MOEA/D also met the significance criterion ($p_{\text{Holm}} = 0.04885$, $A_{12} = 0.740$). Thus, the HV advantage is supported by both the distributional comparisons and effect sizes.

**Table 10.** Pairwise comparisons of MOO performance metrics using two-sided Mann–Whitney $U$ tests with Holm correction.

| **Algorithm *A*** | **Algorithm *B*** | **HV↑** | | **NDS↑** | |
|---|---|---|---|---|---|
| | | $p_{\text{Holm}}$ | $A_{12}$ | $p_{\text{Holm}}$ | $A_{12}$ |
| NSGA-III | MOCOA | $1.480 \times 10^{-6}$ | 0.990 | 0.05830 | 0.720 |
| NSGA-III | MORIME | $4.279 \times 10^{-6}$ | 0.970 | 0.03306 | 0.751 |
| NSGA-III | MOEA/D | 0.04885 | 0.740 | 0.92904 | 0.409 |
| MOCOA | MORIME | 1.00000 | 0.456 | 1.00000 | 0.561 |
| MOCOA | MOEA/D | $1.567 \times 10^{-5}$ | 0.055 | 0.00768 | 0.206 |
| MORIME | MOEA/D | $1.605 \times 10^{-4}$ | 0.103 | 0.00527 | 0.191 |

For NDS, NSGA-III exceeded MORIME significantly ($p_{\text{Holm}} = 0.03306$, $A_{12} = 0.751$). The comparisons with MOCOA ($p_{\text{Holm}} = 0.05830$) and MOEA/D ($p_{\text{Holm}} = 0.92904$) did not reach statistical significance. The results therefore support an HV advantage and stable feasible-solution generation on the tested instance without establishing superiority in solution count over all the competing algorithms.

### *6.5. Ablation Study*

To identify the contribution of each proposed mechanism, the full framework (A0) was compared with four component-ablation variants (A1–A4) and a standard NSGA-III baseline (A5). Each of A1–A4 omitted or replaced one component while retaining the remaining components. A1 (w/o proposed encoding–decoding ) used direct encoding of adjustment type and magnitude. A2 (w/o T-CAG generation) removed the candidate-airspace restriction used in the T-CAG strategy. A3 (w/o conflict-pair penalty) used feasibility-first constraint handling based on the same residual conflict-pair count. A4 (w/o dynamic rejection) retained all the demands throughout the search.

A5 (standard NSGA-III) was applied to the full decision space without the proposed T-CAG generation, encoding–decoding, conflict-pair penalty, or dynamic rejection mechanisms. This baseline provides a reference for evaluating the combined effect of the proposed mechanisms.

All the configurations used NSGA-III and were evaluated using the same four objectives and feasibility criteria with 20 independent runs per configuration. HV and NDS were evaluated on feasible nondominated solutions and assigned zero when no feasible solution was obtained. Table 11 reports conflict resolution performance, rejection rates, multiobjective performance metrics, and runtime.

The full framework achieved complete resolution in all 20 runs, with a mean rejection rate of $(7.15 \pm 0.37)\%$ and a mean runtime of $51.76 \pm 0.68$ s. Without the proposed encoding–decoding mechanism, complete resolution was achieved in 19/20 runs, while the mean rejection rate and runtime increased to $(8.85 \pm 0.49)\%$ and $54.58 \pm 0.87$ s, respectively. The compact representation jointly encodes adjustment type and magnitude, reducing the number of search variables. The observed improvements are consistent with its role in limiting representation size and supporting efficient adjustment search.

**Table 11.** Ablation and baseline comparison results over 20 independent runs. All metrics except the number of runs achieving complete resolution are reported as mean ± sample standard deviation.

| Variant | Complete Resolution (Runs/20) | Remaining Conflict Pairs | Rejection Rate (%) | HV↑ | NDS↑ | Total Runtime (s) |
|---|---|---|---|---|---|---|
| A0: Full | 20/20 | $0.00 \pm 0.00$ | $7.15 \pm 0.37$ | $1.14 \pm 0.18$ | $4.65 \pm 1.04$ | $51.76 \pm 0.68$ |
| A1: w/o proposed encoding–decoding | 19/20 | $0.10 \pm 0.45$ | $8.85 \pm 0.49$ | $1.04 \pm 0.39$ | $4.45 \pm 1.73$ | $54.58 \pm 0.87$ |
| A2: w/o T-CAG generation | 17/20 | $0.50 \pm 1.24$ | $9.05 \pm 0.60$ | $0.97 \pm 0.45$ | $4.50 \pm 2.04$ | $92.47 \pm 1.41$ |
| A3: w/o conflict-pair penalty | 12/20 | $1.45 \pm 1.88$ | $9.85 \pm 0.49$ | $0.68 \pm 0.61$ | $2.80 \pm 2.48$ | $51.54 \pm 0.77$ |
| A4: w/o dynamic rejection | 0/20 | $5.15 \pm 1.98$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $51.97 \pm 0.65$ |
| A5: Standard NSGA-III | 0/20 | $7.30 \pm 2.56$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $0.00 \pm 0.00$ | $48.43 \pm 0.85$ |

Removing T-CAG generation increased mean runtime to $92.47 \pm 1.41$ s, a 78.7% increase over A0, and reduced complete-resolution frequency to 17/20, with a mean residual conflict count of $0.50 \pm 1.24$ and a mean rejection rate of $(9.05 \pm 0.60)\%$. Restricting optimization to conflict-relevant demands reduces both the search dimension and potential pairwise checks. This component produced the largest observed runtime difference among the single-component ablations, identifying T-CAG generation as the principal source of computational savings in this comparison.

Without the conflict-pair penalty, complete-resolution frequency fell to 12/20, with a mean residual conflict count of $1.45 \pm 1.88$ and a mean rejection rate of $(9.85 \pm 0.49)\%$. Mean HV and NDS decreased to $0.68 \pm 0.61$ and $2.80 \pm 2.48$, respectively, while runtime remained close to A0 at $51.54 \pm 0.77$ s. Incorporating residual conflicts into the objective-based selection process guides the search toward feasible adjustments before additional demands are rejected. The results associate the penalty mechanism primarily with improved feasible-solution attainment and overall multiobjective performance.

Disabling dynamic rejection yielded no completely resolved runs, with a mean residual conflict count of $5.15 \pm 1.98$ and a mean runtime of $51.97 \pm 0.65$ s. Dynamic rejection removes selected competing occupancy requirements, facilitating complete resolution within the prescribed search budget. The mean rejection rate of $(7.15 \pm 0.37)\%$ for A0 quantifies the associated demand loss and provides an operational measure of the balance between conflict elimination and demand retention.

The standard NSGA-III baseline (A5) obtained no feasible solution in the 20 runs, leaving $7.30 \pm 2.56$ conflict pairs on average, with a mean runtime of $48.43 \pm 0.85$ s. Its HV and NDS were therefore zero under the stated evaluation convention. With rejection disabled in both A4 and A5, A4 achieved a lower mean residual conflict count of $5.15 \pm 1.98$. This result is consistent with the combined contribution of T-CAG generation, the proposed encoding–decoding, and conflict-pair penalty handling to conflict reduction. Together with the complete resolution achieved by A0, these comparisons illustrate the complementary roles of adjustment search and demand rejection in obtaining feasible allocations in the tested setting.

A0 achieved the highest overall mean HV and NDS at $1.14 \pm 0.18$ and $4.65 \pm 1.04$, respectively. Its HV standard deviation was 0.18 compared with 0.39, 0.45, and 0.61 for A1–A3, indicating more consistent run-level performance. With unsuccessful runs assigned zero, the aggregate metrics jointly reflect feasible-solution attainment and the performance of the returned feasible sets. The component-level comparisons show complementary contributions: compact encoding supports efficient representation, T-CAG generation

lowers computational cost, and penalty handling and dynamic rejection strengthen feasible conflict resolution in the tested setting.

### *6.6. Parameter Sensitivity Analysis of Improved NSGA-III*

This section presents sensitivity analyses of the internal parameters of the proposed algorithm, the shared hyperparameters of the evolutionary framework, and the operational settings of the ACR model. First, a global sensitivity analysis is conducted using the Morris screening method [42] to identify influential parameters. Subsequently, the key internal parameters are systematically examined to assess the performance and robustness of the adopted configuration. Finally, the separation thresholds, adjustment-type preference weights, and maximum adjustment magnitudes are varied to evaluate the framework's robustness under different separation requirements, operational preferences, and levels of adjustment flexibility.

#### 6.6.1. Parameter Sensitivity Analysis Based on Morris Screening Method

The Morris screening method is a global one-at-a-time sensitivity analysis technique that quantifies the influence of individual parameter perturbations in the input space on the variation of the output. Given the high dimensionality of the parameter space in the proposed algorithm, the Morris method is first employed for efficient parameter screening to identify the most influential parameters, thereby guiding subsequent targeted tuning and in-depth analysis. For each parameter $p_i$, its elementary effect (EE) is defined as:

$$EE_i = \frac{f(p_1, ..., p_i + \Delta, ..., p_k) - f(p_1, ..., p_i, ..., p_k)}{\Delta} \tag{22}$$

In Equation (22), $f$ denotes the performance index function, and $\Delta$ represents the perturbation step size. The mean absolute elementary effect, $\mu_i^* = \frac{1}{R}\sum_{r=1}^{R} |EE_i^{(r)}|$, reflects a parameter's overall importance, where $R$ is the number of sampled trajectories and $EE_i^{(r)}$ is the elementary effect of parameter $i$ along trajectory $r$. The standard deviation, $\sigma_i = \mathrm{std}(EE_i)$, characterizes variation in elementary effects associated with nonlinearities or parameter interactions. In this section, parameters are first ranked by $\mu^*$ using MOO performance metrics (HV, NDS and Spacing) to assess their influence on optimization performance, as shown in Figure 12. The number of unresolved conflict pairs is then used to evaluate conflict resolution performance, as shown in Figure 13.

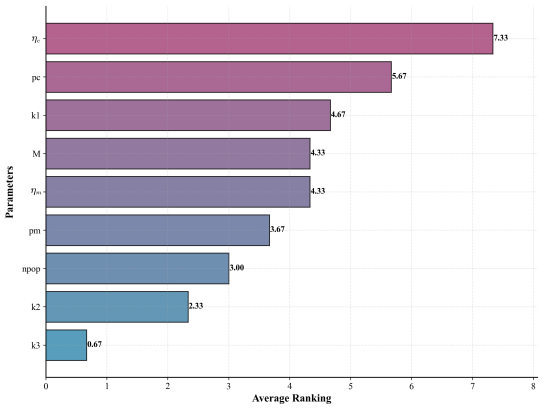



**Figure 12.** Bar chart of parameters ranked by mean elementary effect $\mu^*$ with respect to the MOO performance metric. Lower average ranks indicate greater impact on the algorithm's global optimization performance.

According to Figure 12, parameters $K_3$, $K_2$, and $N_{pop}$ achieve the lowest average ranks (0.67, 2.33, and 3.00, respectively), indicating the strongest influence on optimization

performance and thereby suggesting that they should be prioritized in tuning. Moreover, as shown in Figure 13, $M$ and $K_1$ exhibit the largest $\mu^*$ values and thus have the greatest global impact on conflict resolution, warranting careful calibration. Less influential genetic operator parameters can be fixed to empirical defaults.

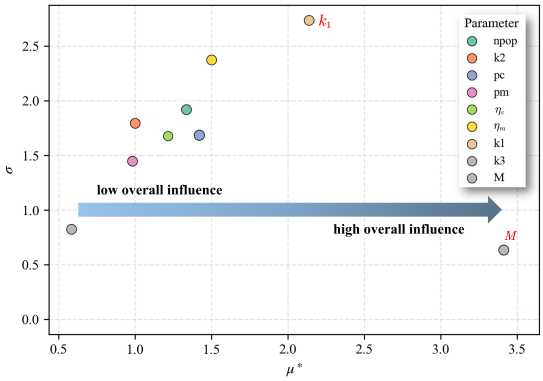



**Figure 13.** Morris screening plot based on unresolved conflict pair count: larger $\mu^*$ implies higher global importance for conflict resolution; $\sigma$ reflects parameter nonlinearity/interactions.

#### 6.6.2. Parameter Sensitivity Analysis of Critical Parameters

To assess the influence of parameters $K_3$, $K_2$, and $N_{pop}$ on multiobjective optimization performance, we conduct a sensitivity analysis over 125 combinations ($K_3 \in \{10, 20, 30, 40, 50\}$, $K_2 \in \{100, 150, 200, 250, 300\}$ and $N_{pop} \in \{50, 100, 150, 200, 250\}$). For fair comparison, the obtained feasible nondominated set for each combination is computed; all the nondominated solutions are then merged and re-sorted to identify the best overall solution set. The corresponding objective values are listed in Table 12.

Table 12 reveals three parameter combinations achieving optimal performance. However, since population size and iteration count—both governed by these parameters—strongly affect computational cost, we further analyze their runtime (Figure 14). The results show that increasing $K_3$, $K_2$, or $N_{pop}$ generally increases execution time, with $K_3$ exhibiting the strongest effect. This supports adaptive parameter selection to balance ACR effectiveness and runtime efficiency.

Given the dominant influence of parameters $M$ and $K_1$ on conflict resolution performance, we evaluate 36 parameter combinations ($M \in \{0, 200, 400, 600, 800, 1000\}$ and $K_1 \in \{0, 2, 4, 6, 8, 10\}$) to analyze their sensitivity. Figure 15 illustrates how varying $M$ and $K_1$ affect the number of unresolved conflict pairs and rejection rate, respectively. The results show that specific settings of $M$ and $K_1$ achieve full conflict resolution while minimizing rejections. This enables adaptive parameter tuning based on operational constraints on airspace rejections.

**Table 12.** Optimal parameter configurations and the corresponding multiobjective values.

| Parameter Set | Multiobjective Values $(F_1, F_2, F_3, F_4)$ |
|---|---|
| $N_{\text{pop}} = 150,\ K_2 = 100,\ K_3 = 20$ | $(10, 8, 5, 1.866)$<br>$(10, 9, 5, 1.799)$ |
| $N_{\text{pop}} = 150,\ K_2 = 150,\ K_3 = 20$ | $(9, 8, 5, 1.931)$ |
| $N_{\text{pop}} = 200,\ K_2 = 300,\ K_3 = 50$ | $(11, 8, 5, 1.873)$<br>$(11, 9, 5, 1.787)$ |

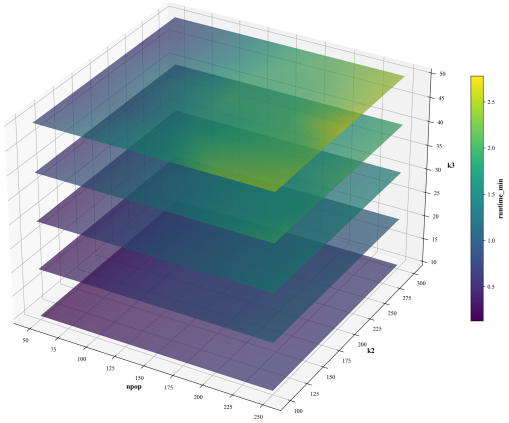
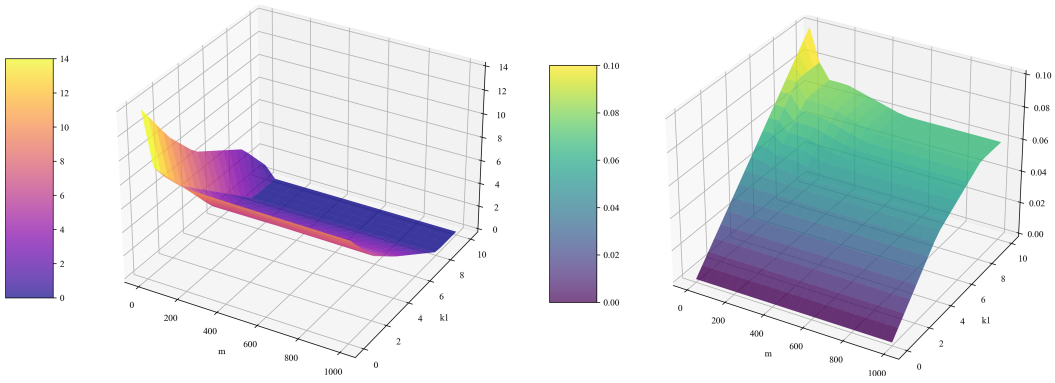



**Figure 14.** Runtime surface over $(K_3, K_2, N_{pop})$ parameter space. Axes denote $N_{pop}$ (x), $K_2$ (y), and $K_3$ (z); execution time is mapped to color via linear interpolation.




(**a**) Number of unresolved conflict pairs (**b**) Rejection rate

**Figure 15.** Sensitivity analysis of different $M$ and $K_1$ parameter combinations: (**a**) number of unresolved conflict pairs and (**b**) rejection rate.

### 6.6.3. Sensitivity to Separation Thresholds

To assess sensitivity to the separation requirements, each threshold was varied individually around the baseline setting $(s_{\min}, h_{\min}, t_{\min}) = (10\ \text{km}, 0.3\ \text{km}, 1.5\ \text{min})$ while the other thresholds and experimental settings were held fixed. The results are summarized in Table 13.

**Table 13.** Sensitivity to separation thresholds over 20 independent runs per configuration. All metrics except complete-resolution counts are reported as mean ± sample standard deviation.

| Threshold | Value | Complete Resolution (Runs/20) | Remaining Conflict Pairs | Rejection Rate (%) | Total Runtime (s) |
|---|---|---|---|---|---|
| Horizontal threshold ($s_{\min}$) | 8 km | 20/20 | $0.00 \pm 0.00$ | $7.05 \pm 0.22$ | $51.06 \pm 0.64$ |
| | 10 km | 20/20 | $0.00 \pm 0.00$ | $7.15 \pm 0.37$ | $51.76 \pm 0.68$ |
| | 12 km | 19/20 | $0.10 \pm 0.45$ | $7.95 \pm 0.51$ | $51.88 \pm 0.74$ |
| Vertical threshold ($h_{\min}$) | 0.2 km | 20/20 | $0.00 \pm 0.00$ | $7.10 \pm 0.31$ | $51.47 \pm 0.59$ |
| | 0.3 km | 20/20 | $0.00 \pm 0.00$ | $7.15 \pm 0.37$ | $51.76 \pm 0.68$ |
| | 0.4 km | 20/20 | $0.00 \pm 0.00$ | $7.30 \pm 0.57$ | $51.52 \pm 0.71$ |
| Temporal threshold ($t_{\min}$) | 1.0 min | 20/20 | $0.00 \pm 0.00$ | $6.80 \pm 0.41$ | $49.32 \pm 0.35$ |
| | 1.5 min | 20/20 | $0.00 \pm 0.00$ | $7.15 \pm 0.37$ | $51.76 \pm 0.68$ |
| | 2.0 min | 19/20 | $0.15 \pm 0.67$ | $7.95 \pm 0.60$ | $53.15 \pm 0.62$ |

Note: The baseline results are repeated across the three groups for comparison.

Increasing $s_{\min}$ from 8 to 12 km increased the mean rejection rate from $(7.05 \pm 0.22)\%$ to $(7.95 \pm 0.51)\%$ and reduced the complete-resolution count from 20/20 to 19/20. Similarly,

increasing $t_{\min}$ from 1.0 to 2.0 min increased the mean rejection rate from $(6.80 \pm 0.41)\%$ to $(7.95 \pm 0.60)\%$, with complete resolution achieved in 19/20 runs at the upper setting. These trends are consistent with tighter spatial and temporal compatibility constraints: under fixed adjustment bounds, fewer compatible allocations remain available, increasing reliance on demand rejection.

For $h_{\min}$ between 0.2 and 0.4 km, all the configurations achieved complete resolution in 20/20 runs, while the mean rejection rate increased only from $(7.10 \pm 0.31)\%$ to $(7.30 \pm 0.57)\%$. Across all the tested configurations, the observed complete-resolution rate remained between 95% and 100%, and the mean runtimes ranged from $49.32 \pm 0.35$ to $53.15 \pm 0.62$ s. These results support the framework's robustness within the tested threshold ranges, with stricter horizontal and temporal requirements primarily reflected in higher rejection costs.

The consistently high complete-resolution rates within the tested ranges support the framework's adaptability to varying separation requirements. Configurable spatial and temporal buffers provide a basis for adapting the framework to ATM, UTM, and mission-specific airspace reservation scenarios.

#### 6.6.4. Sensitivity to Adjustment-Type Preference Weights

To assess the sensitivity of ACR performance to the preference weights in $F_2$, three configurations are considered: $W_0 = (1, 2, 3)$, $W_1 = (2, 1, 3)$, and $W_2 = (1, 2, 4)$, with weights ordered as $(w_{\mathrm{v}}, w_{\mathrm{h}}, w_{\mathrm{t}})$. Relative to the baseline $W_0$, $W_1$ exchanges the vertical and horizontal preference weights, whereas $W_2$ increases the temporal adjustment penalty while preserving the baseline preference ordering. The adjustment codes identify categories, while $c(x_i)$ assigns explicit dimensionless preference penalties. The weight ratios represent assumed decision preferences whose influence is assessed through these alternative configurations.

As summarized in Table 14, all three configurations achieve complete conflict resolution in 20 out of 20 runs. The mean rejection rates range from $(6.95 \pm 0.39)\%$ to $(7.15 \pm 0.37)\%$, while the mean runtimes range from $51.76 \pm 0.68$ to $52.88 \pm 0.81$ s, with a maximum increase of 2.16% relative to the baseline. These results demonstrate consistent conflict resolution performance and limited variation in computational cost across the tested preference settings.

Figure 16 shows the corresponding adjustment-type composition. The mean counts of vertical, horizontal, and temporal adjustments change from $(2.15 \pm 0.49,\ 3.20 \pm 0.52,\ 1.00 \pm 0.32)$ under $W_0$ to $(3.45 \pm 0.51,\ 1.80 \pm 0.41,\ 2.00 \pm 0.32)$ under $W_1$. Under $W_2$, the temporal adjustment count decreases to $0.00 \pm 0.00$, while the mean vertical and horizontal adjustment counts increase to $3.30 \pm 0.47$ and $3.55 \pm 0.51$, respectively. This pattern is consistent with the higher temporal penalty favoring spatial alternatives. Because temporal adjustments may require synchronized changes to associated demands, the resulting adjustment composition reflects both preference weights and coupled feasibility constraints.

These results clarify the complementary roles of $F_2$ and $F_3$: $F_2$ evaluates cumulative type-dependent penalties, whereas $F_3$ controls the total number of adjusted demands. Although both objectives depend on adjustment counts, solutions with the same $F_3$ can have different $F_2$ values because of their adjustment-type composition. The sensitivity results support using explicit preference weights to influence adjustment composition while maintaining consistent conflict resolution performance.

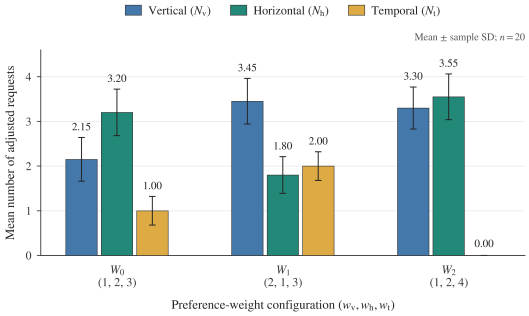



**Figure 16.** Adjustment-type counts under three preference-weight configurations.

**Table 14.** ACR performance under different preference weights over 20 independent runs. All metrics except complete-resolution counts are reported as mean ± sample standard deviation.

| Config. | Weights $(w_v, w_h, w_t)$ | Complete Resolution (Runs/20) | Remaining Conflict Pairs | Rejection Rate (%) | Runtime (s) |
|---|---|---|---|---|---|
| $W_0$ | $(1, 2, 3)$ | 20/20 | $0.00 \pm 0.00$ | $7.15 \pm 0.37$ | $51.76 \pm 0.68$ |
| $W_1$ | $(2, 1, 3)$ | 20/20 | $0.00 \pm 0.00$ | $7.00 \pm 0.32$ | $52.12 \pm 0.74$ |
| $W_2$ | $(1, 2, 4)$ | 20/20 | $0.00 \pm 0.00$ | $6.95 \pm 0.39$ | $52.88 \pm 0.81$ |

#### 6.6.5. Sensitivity Analysis of Maximum Adjustment Magnitudes

To assess sensitivity to the available adjustment flexibility, three configurations of $\Delta_{\max}$ are considered (Table 15), corresponding to two, three, and four times the baseline adjustment bounds.

**Table 15.** Maximum adjustment magnitudes under three configurations.

| Case | $h_{\max}$ (km) | $s_{\max}$ (km) | $t_{\max}$ (min) |
|---|---|---|---|
| Case 1 | 0.60 | 5.00 | 10.00 |
| Case 2 | 0.90 | 7.50 | 15.00 |
| Case 3 | 1.20 | 10.00 | 20.00 |

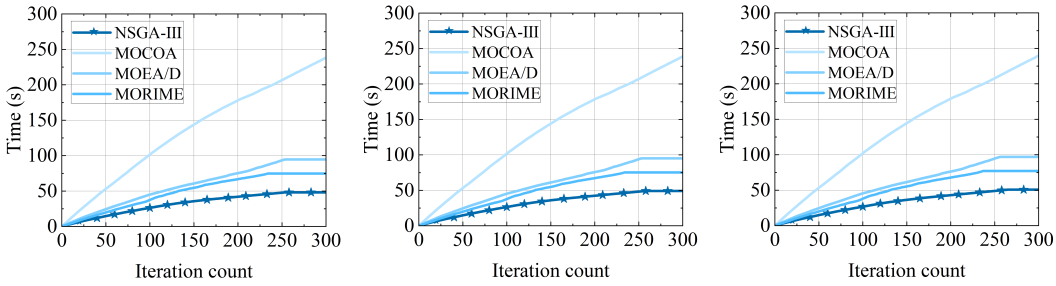
As shown in Table 16, NSGA-III achieves complete conflict resolution in 20/20 runs under each configuration. Its mean rejection rate decreases from $(5.15 \pm 0.59)\%$ in Case 1 to $(0.00 \pm 0.00)\%$ in Cases 2 and 3. MOEA/D also achieves complete resolution in all the runs, with mean rejection rates of $(7.05 \pm 0.69)\%$, $(3.25 \pm 0.44)\%$, and $(2.10 \pm 0.45)\%$, respectively. Increasing the adjustment bounds enlarges the set of admissible spatial and temporal modifications, providing additional opportunities to accommodate demands under the model constraints. The reduction in rejection rates is consistent with this increased coordination flexibility, while the sustained resolution frequency indicates stable feasibility attainment across the tested settings.

HV is calculated from the feasible nondominated solution sets using the same four-objective definition, normalization procedure, and reference-point selection method described above. Within each configuration, all the algorithms use identical evaluation settings. NSGA-III achieves mean HV values of $1.12 \pm 0.14$, $1.06 \pm 0.09$, and $1.10 \pm 0.16$ compared with $0.91 \pm 0.26$, $0.94 \pm 0.24$, and $0.89 \pm 0.07$ for MOEA/D, representing within-case improvements of 23.08%, 12.77%, and 23.60%. Their mean NDS values range from $4.95 \pm 1.15$ to $5.50 \pm 1.88$ and from $5.15 \pm 1.23$ to $5.95 \pm 2.16$, respectively. These results characterize the NSGA-III solution sets by higher dominated objective-space volume and a compact collection of trade-off alternatives across the tested adjustment ranges.

**Table 16.** ACR performance, MOO metrics, and runtime under three adjustment-range configurations. Except for complete-resolution counts, values are reported as mean ± sample standard deviation over 20 independent runs.

| Case | Algorithm | Complete Resolution (Runs/20) | Remaining Conflict Pairs | Rejection Rate (%) | HV↑ | NDS↑ | Total Runtime (s) |
|---|---|---|---|---|---|---|---|
| Case 1 | NSGA-III | 20/20 | 0.00 ± 0.00 | 5.15 ± 0.59 | 1.12 ± 0.14 | 4.95 ± 1.15 | 52.12 ± 0.74 |
| | MOCOA | 19/20 | 0.10 ± 0.45 | 8.35 ± 0.75 | 0.68 ± 0.22 | 4.20 ± 1.28 | 243.78 ± 4.52 |
| | MORIME | 18/20 | 0.25 ± 0.79 | 8.65 ± 0.75 | 0.68 ± 0.28 | 3.85 ± 1.60 | 80.74 ± 2.43 |
| | MOEA/D | 20/20 | 0.00 ± 0.00 | 7.05 ± 0.69 | 0.91 ± 0.26 | 5.15 ± 1.23 | 101.17 ± 3.08 |
| Case 2 | NSGA-III | 20/20 | 0.00 ± 0.00 | 0.00 ± 0.00 | 1.06 ± 0.09 | 5.20 ± 1.36 | 51.45 ± 0.79 |
| | MOCOA | 20/20 | 0.00 ± 0.00 | 5.50 ± 0.83 | 0.79 ± 0.15 | 4.75 ± 0.91 | 240.66 ± 4.92 |
| | MORIME | 19/20 | 0.10 ± 0.45 | 6.30 ± 1.13 | 0.81 ± 0.25 | 4.35 ± 1.39 | 79.71 ± 2.15 |
| | MOEA/D | 20/20 | 0.00 ± 0.00 | 3.25 ± 0.44 | 0.94 ± 0.24 | 5.75 ± 1.97 | 99.87 ± 2.98 |
| Case 3 | NSGA-III | 20/20 | 0.00 ± 0.00 | 0.00 ± 0.00 | 1.10 ± 0.16 | 5.50 ± 1.88 | 53.09 ± 0.81 |
| | MOCOA | 20/20 | 0.00 ± 0.00 | 3.50 ± 0.83 | 0.84 ± 0.13 | 5.05 ± 0.94 | 248.25 ± 4.18 |
| | MORIME | 20/20 | 0.00 ± 0.00 | 4.40 ± 0.88 | 0.87 ± 0.15 | 5.20 ± 1.20 | 82.29 ± 2.57 |
| | MOEA/D | 20/20 | 0.00 ± 0.00 | 2.10 ± 0.45 | 0.89 ± 0.07 | 5.95 ± 2.16 | 103.08 ± 3.26 |

Figure 17 presents the runtime profiles over iterations. The mean total runtime of NSGA-III ranges from 51.45 ± 0.79 to 53.09 ± 0.81 s, approximately 48.5% lower than that of MOEA/D in each configuration. Thus, the expanded adjustment domain is accommodated with limited variation in computational cost. The stable resolution frequency, improved demand retention, and narrow runtime range support the NSGA-III-based method's robustness to the tested adjustment-bound variations. Successful conflict resolution with both algorithms further supports the common ACR framework's compatibility with different multiobjective search methods.




(**a**) Case 1 (**b**) Case 2 (**c**) Case 3

**Figure 17.** Runtime profiles over iterations for NSGA-III, MOCOA, MORIME, and MOEA/D under three adjustment-range configurations: (**a**) Case 1, (**b**) Case 2, and (**c**) Case 3.

### *6.7. ACR Efficiency Analysis*

#### 6.7.1. Comparative Analysis of ACR Efficiency

For ACR, balancing resolution effectiveness with computational efficiency is essential. As shown in Figure 18, the NSGA-III-based method achieves the lowest mean total runtime of 51.76 ± 0.68 s compared with 242.01 ± 4.96 s for MOCOA, 80.18 ± 2.14 s for MORIME, and 100.44 ± 3.21 s for MOEA/D. Relative to these three algorithms, the mean runtime is reduced by 78.61%, 35.45%, and 48.47%, respectively, demonstrating the computational advantage of the NSGA-III-based method in the tested scenario.

The algorithms share the same conflict evaluation and rejection procedures while employing different population-update and selection operations. The observed runtime differences therefore characterize their computational performance within the common ACR framework. The small runtime standard deviation of the NSGA-III-based method further indicates consistent computational performance across repeated runs.

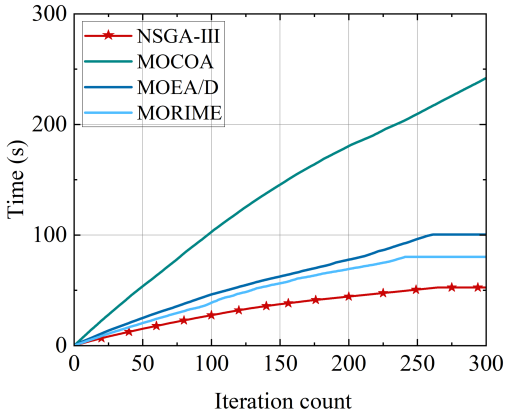



**Figure 18.** Runtime comparison of NSGA-III, MOCOA, MORIME, and MOEA/D under a common ACR framework.

#### 6.7.2. ACR Efficiency Under Varying Airspace Demand Levels

Airspace demand levels vary across planning scenarios; therefore, runtime analysis at different scales provides computational benchmarks for airspace conflict resolution.

As shown in Figure 19, runtime increases nonlinearly with the number of airspace demands. At 90 demands, the mean runtime is 34.67 s, representing a 33.01% reduction relative to the mean runtime of $51.76 \pm 0.68$ s at 100 demands. For instances with fewer than 60 demands, runtimes are on the order of seconds, while larger instances require longer computation.

These results support the method's potential use in computationally efficient pre-tactical airspace reservation planning. Near-real-time decision support is also a potential application where the operational response-time budget accommodates the measured computation times. Operational deployment requires further validation against application-specific response deadlines under representative operating conditions.

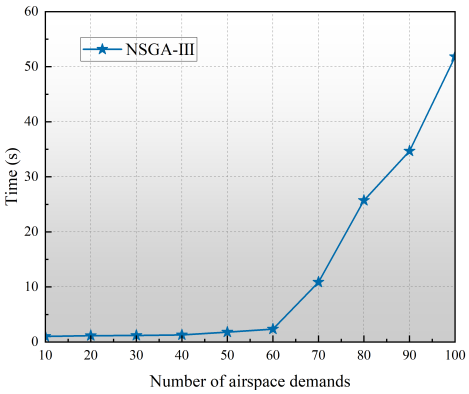
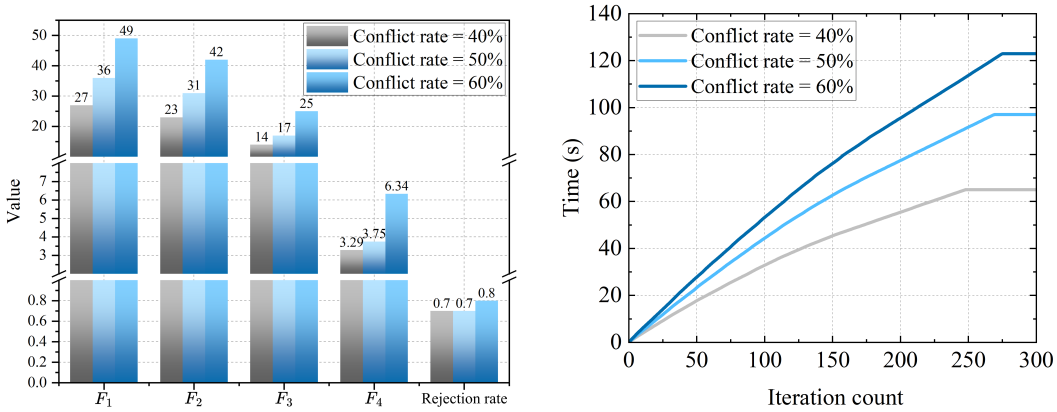



**Figure 19.** Algorithm runtime versus the number of airspace demands.

### *6.8. Airspace Scenario Stress Testing Based on Conflict Rate*

To further validate the conflict resolution capability of the proposed algorithm, stress tests are performed on airspace scenarios generated under parameter constraints derived from real-world airspace demand data with conflict rates of 40%, 50%, and 60%.

As shown in Figure 20b the mean runtimes are $65.03 \pm 0.93$ s, $97.03 \pm 1.38$ s, and $122.97 \pm 2.95$ s for the three conflict-rate scenarios, respectively. The maintained conflict resolution capability and limited runtime variability support robust performance across the tested conflict levels. These results further support the method's potential use in pre-tactical airspace reservation planning, consistent with the application scope discussed in Section 6.7.2.




(**a**) Objective values and rejection cost (**b**) Algorithm runtime

**Figure 20.** Stress test results under varying conflict rates: (**a**) objective values and rejection cost; (**b**) algorithm runtime.

## 7. Discussion and Conclusions

This study proposes an ACR optimization framework tailored for complex interdependent requirements that addresses conflicts among airspace reservation demands. The method integrates NSGA-III with a three-layer architecture (conflict detection, resolution, and group-level optimization) to tackle high-dimensional decision spaces and multiple constraints. The key contributions include the formalization of airspace group interdependencies as constrained four-objective optimization problems and the introduction of neighboring airspaces and T-CAGs to reduce the search space and computational cost while supporting feasible-solution search.

By synthesizing interval overlap comparison for vertical and temporal dimensions and the SAT for horizontal conflict detection, the framework accounts for cascading effects [31] by checking for secondary conflicts during resolution. Additionally, airspace-group-oriented encoding–decoding mechanisms, conflict-pair-based penalty function design, and dynamic rejection strategies embedded in evolutionary iterations collectively support feasible-solution search and multiobjective solution quality. These mechanisms coordinate adjustments among interdependent demands while preserving temporal consistency within airspace groups. The ablation study further identifies their complementary contributions to computational efficiency and feasible-solution attainment.

The experimental comparisons under a common ACR framework demonstrate the proposed method's advantages over MOCOA, MORIME, and MOEA/D in hypervolume, computational efficiency, and demand retention. The sensitivity analyses further support robustness within the tested parameter ranges. The dynamic rejection strategy facilitates conflict-free allocation among retained demands within the prescribed rejection budget, with rejection rates explicitly quantifying the associated demand loss. The methodology provides a basis for decision support in ATM systems by respecting scenario-specific separation requirements ($h_{\min}, s_{\min}, t_{\min}$) while balancing multidimensional adjustment objectives.

Furthermore, the method's consistent performance in conflict-rate stress tests supports its ability to handle increased coordination complexity in the tested scenarios. Its capability to manage complex interdependent airspace reservations suggests potential applications in manned–unmanned shared airspace, such as UAM/UTM corridors and large-scale unmanned aerial vehicle (UAV) operational airspace. The measured computation times support a pre-tactical planning role, while potential near-real-time applications require further validation against application-specific response deadlines and operational require-

ments. By supporting coordinated utilization of reserved airspace, this work provides a methodological foundation for further investigation of scalable low-altitude operations.

Although the proposed NSGA-III-based ACR method demonstrates strong performance in handling complex interdependent requirements in the tested scenarios, certain aspects require further investigation. Currently, the framework mainly considers adjustments in vertical, horizontal, and temporal attributes. Future work could couple airspace reservation decisions with trajectory-level planning to incorporate aircraft velocity control and route modification [43–45], thereby enabling a more comprehensive approach to ACR. In addition, the experimental setup is based on static airspace demands; extending the method to dynamically arriving and updated demands would support adaptive replanning under varying operational conditions. Finally, exploring the applicability of the proposed framework in multi-UAV operational airspace [46] could further enhance its flexibility and scalability in heterogeneous traffic environments.

In summary, the proposed framework establishes a robust paradigm for resolving conflicts under complex interdependent requirements, offering theoretical insights and practical tools for efficient airspace resource management. By addressing scalability and adaptability challenges, it lays the foundation for intelligent optimization of next-generation ATM systems.

**Supplementary Materials:** The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/aerospace13100906/s1: Supplementary Dataset S1: Perturbed companion dataset containing 285 airspace demands, including demand attributes, horizontal coordinates, and a README describing the data structure and perturbation procedure.

**Author Contributions:** Conceptualization, Q.C.; methodology, Y.H. and Q.C.; validation, Y.H.; formal analysis, Y.S.; investigation, Y.S.; resources, Q.C.; data curation, Y.H. and Z.L.; writing—review and editing, Y.H. and Z.L.; visualization, Y.H.; supervision, Z.L.; project administration, Z.L.; funding acquisition, Q.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China, grant number 72301290.

**Data Availability Statement:** The original experimental input data remain confidential. A perturbed companion dataset containing 285 airspace demands is provided as Supplementary Dataset S1 (`Supplementary_Data_S1_Airspace_Demands.xlsx`). The workbook includes the demand attributes, coordinate data and a README documenting the data structure and perturbation procedure, supporting independent inspection and further evaluation of the method.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| ACR | Airspace conflict resolution |
| ASM | Airspace management |
| ATM | Air traffic management |
| MOO | Multiobjective optimization |
| NSGA-III | Nondominated sorting genetic algorithm III |
| SAT | Separating axis theorem |
| T-CAG | Targeted conflict airspace group |
| TSA | Temporary segregated area |
| TRA | Temporary reserved area |
| UAM | Urban air mobility |
| UAV | Unmanned aerial vehicle |
| UTM | Unmanned aircraft system traffic management |

## References


1. Xiao, G.; Liu, D.; Liu, X.; He, F.; Ding, D. Study on aircraft generalized conflict resolution and trajectory optimization with multiple constraint in complex airspace environment. In Proceedings of the IEEE/AIAA Digital Avionics Systems Conference (DASC), San Diego, CA, USA, 8–12 September 2019; pp. 1–6.
2. Cai, Q.; Ang, H.; Alam, S. A multiobjective optimization approach for reducing air traffic collision risk. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; pp. 1759–1766.
3. Tang, J.; Lao, S.; Wan, Y. Systematic review of collision-avoidance approaches for unmanned aerial vehicles. *IEEE Syst. J.* **2022**, *16*, 4356–4367. https://doi.org/10.1109/jsyst.2021.3101283.
4. Sergeeva, M. Automated Airspace Sectorization by Genetic Algorithm. Ph.D. Dissertation, Université Paul Sabatier, Toulouse, France, 2017.
5. Cecen, R.K.; Cetek, C. Conflict-free en-route operations with horizontal resolution manoeuvers using a heuristic algorithm. *Aeronaut. J.* **2020**, *124*, 767–785. https://doi.org/10.1017/aer.2020.5.
6. Hao, S.; Cheng, S.; Zhang, Y. A multi-aircraft conflict detection and resolution method for 4-dimensional trajectory-based operation. *Chin. J. Aeronaut.* **2018**, *31*, 1579–1593. https://doi.org/10.1016/j.cja.2018.04.017.
7. Zhou, Y.; Hu, M.; Yang, L.; Zhang, Y. Collaborative conflict-free 4D trajectory planning based on multi-objective hybrid-metaheuristic optimization algorithm. *Trans. Nanjing Univ. Aeronaut. Astronaut.* **2024**, *41*, 372–386. https://doi.org/10.16356/j.1005-1120.2024.03.009.
8. Yan, S.; Cai, K. A multi-objective multi-memetic algorithm for network-wide conflict-free 4D flight trajectories planning. *Chin. J. Aeronaut.* **2017**, *30*, 1161–1173. https://doi.org/10.1016/j.cja.2017.03.008.
9. Ma, L.; Jiang, W.; Tian, Y.; Zhang, Y. Analysis of key technologies for UAV conflict-free four-dimensional trajectory planning in controlled airspace. In Proceedings of the Chinese Control and Decision Conf. (CCDC), Yichang, China, 20–22 May 2023; pp. 2785–2790.
10. Liu, H.; Liu, F.; Zhang, X.; Guan, X.; Chen, J.; Savinaud, P. Aircraft conflict resolution method based on hybrid ant colony optimization and artificial potential field. *Sci. China Inf. Sci.* **2018**, *61*, 129103. https://doi.org/10.1007/s11432-017-9310-5.
11. Hong, Y.; Kim, Y. Two-stage multicriteria decision-making framework for aircraft conflict resolution. *J. Aerosp. Inf. Syst.* **2023**, *20*, 596–604. https://doi.org/10.2514/1.i011152.
12. International Civil Aviation Organization. *Procedures for Air Navigation Services—Air Traffic Management*, 16th ed.; Doc. 444; ICAO: Montréal, QC, Canada, 2016.
13. Alam, S.; Lokan, C.; Abbass, H. What can make an airspace unsafe? characterizing collision risk using multi-objective optimization. In Proceedings of the IEEE Congress on Evolutionary Computation, Brisbane, QLD, Australia, 10–15 June 2012; pp. 1–8.
14. Lü, R.; Guan, X.; Li, X.; Hwang, I. A large-scale flight multi-objective assignment approach based on multi-island parallel evolution algorithm with cooperative coevolutionary. *Sci. China Inf. Sci.* **2016**, *59*, 072201. https://doi.org/10.1007/s11432-015-5495-3.
15. Tang, J.; Liu, G.; Pan, Q. Review on artificial intelligence techniques for improving representative air traffic management capability. *J. Syst. Eng. Electron.* **2022**, *33*, 1123–1134. https://doi.org/10.23919/jsee.2022.000109.
16. Guo, T.; Mei, Y.; Tang, K.; Du, W. Cooperative co-evolution for large-scale multiobjective air traffic flow management. *IEEE Trans. Evol. Comput.* **2024**, *28*, 1644–1658. https://doi.org/10.1109/tevc.2023.3328886.
17. Cai, Q.; Ang, H.; Alam, S. A multiobjective optimization approach for air traffic flow management for airspace safety enhancement. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Padua, Italy, 18–23 July 2022; pp. 1–9.
18. Guo, T.; Mei, Y.; Tang, K.; Du, W. A knee-guided evolutionary algorithm for multi-objective air traffic flow management. *IEEE Trans. Evol. Comput.* **2024**, *28*, 994–1008. https://doi.org/10.1109/TEVC.2023.3281810.
19. Zhou, J.; Zhang, H.; Wang, Y.; Zhong, G.; Liu, H. Multiobjective four-dimensional trajectory synergetic optimization based on congestion prediction and NSGA3-SA. *IEEE Access* **2022**, *10*, 71986–72005. https://doi.org/10.1109/access.2022.3188790.
20. Zhao, Z.; Song, M.; Yuan, J. Pre-flight plan conflict resolution algorithm design and strategy research. In Proceedings of the International Conference on Electronic Information Engineering, Big Data, and Computer Technology (EIBDCT 2023), Xishuangbanna, China, 6–8 January 2023; p. 104.
21. Liu, Y.; Wang, M.; Wang, G. Multi-agent reinforcement learning with attention mechanism for collaborative management of airspace users in the pre-flight phase. *Int. J. Aeronaut. Space Sci.* **2025**, *26*, 2547–2561. https://doi.org/10.1007/s42405-025-00914-w.
22. Oktal, H.; Yaman, K.; Kasımbeyli, R. A mathematical programming approach to optimum airspace sectorisation problem. *J. Navig.* **2020**, *73*, 599–612. https://doi.org/10.1017/s0373463319000833.
23. Chandra, A.; Hazra, S.; Verma, A.; Sooraj, K.P. On the possibilities of efficient air traffic monitoring through complex network clustering based airspace sub-sectorization: A multi-objective discrete particle swarm optimization approach. *Transp. Res. Rec.* **2025**, *2679*, 2127–2143. https://doi.org/10.1177/03611981241263829.
24. Zhang, W.; Hu, M.; Yin, J.; Li, H.; Du, J. Multi-objective 3D airspace sectorization problem using NSGA-II with prior knowledge and external archive. *Aerospace* **2023**, *10*, 216. https://doi.org/10.3390/aerospace10030216.

25. Ribeiro, M.; Ellerbroek, J.; Hoekstra, J. Using reinforcement learning to improve airspace structuring in an urban environment. *Aerospace* **2022**, *9*, 420. https://doi.org/10.3390/aerospace9080420.
26. Dudoit, A.; Rimša, V.; Bogdevičius, M.; Skorupski, J. Effectiveness of conflict resolution methods in air traffic management. *Aerospace* **2022**, *9*, 112. https://doi.org/10.3390/aerospace9020112.
27. Yang, W.; Tang, J.; He, R.; Chen, Y. A medium-term conflict detection and resolution method for open low-altitude city airspace based on temporally and spatially integrated strategies. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 1817–1830. https://doi.org/10.1109/TCST.2019.2925579.
28. Li, Y.; Zhang, Y.; Guo, T.; Liu, Y.; Lv, Y.; Du, W. Graph reinforcement learning for multi-aircraft conflict resolution. *IEEE Trans. Intell. Veh.* **2024**, *9*, 4529–4540. https://doi.org/10.1109/tiv.2024.3364652.
29. Li, B.; Guo, T.; Mei, Y.; Li, Y.; Chen, J.; Zhang, Y.; Tang, K.; Du, W. A multi-objective memetic algorithm with adaptive local search for airspace complexity mitigation. *Swarm Evol. Comput.* **2023**, *83*, 101400. https://doi.org/10.1016/j.swevo.2023.101400.
30. Alonso-Ayuso, A.; Escudero, L.F.; Martín-Campo, F.J. An exact multi-objective mixed integer nonlinear optimization approach for aircraft conflict resolution. *TOP* **2016**, *24*, 381–408. https://doi.org/10.1007/s11750-015-0402-z.
31. Tang, J. Conflict detection and resolution for civil aviation: A literature survey. *IEEE Aerosp. Electron. Syst. Mag.* **2019**, *34*, 20–35. https://doi.org/10.1109/maes.2019.2914986.
32. Melero, F.J.; Aguilera, Á.; Feito, F.R. Fast collision detection between high resolution polygonal models. *Comput. Graph.* **2019**, *83*, 97–106. https://doi.org/10.1016/j.cag.2019.07.006.
33. Montanari, M.; Petrinic, N.; Barbieri, E. Improving the GJK algorithm for faster and more reliable distance queries between convex objects. *ACM Trans. Graph.* **2017**, *36*, 1–17. https://doi.org/10.1145/3083724.
34. Wang, Y.; Cheng, Z.; Wang, Z.; Yuan, J. Complex four-dimensional trajectory conflict detection and resolution based on grid model. In Proceedings of the 4th International Conference Computer Communication and Artificial Intelligence (CCAI), Xi'an, China, 24–26 May 2024; pp. 487–496.
35. Zhang, Z.; Zheng, Y.; Li, C.; Jiang, B.; Li, Y. Designing an urban air mobility corridor network: A multi-objective optimization approach using U-NSGA-III. *Aerospace* **2025**, *12*, 229. https://doi.org/10.3390/aerospace12030229.
36. Guan, X.; Zhang, X.; Wei, J.; Hwang, I.; Zhu, Y.; Cai, K. A strategic conflict avoidance approach based on cooperative coevolutionary with the dynamic grouping strategy. *Int. J. Syst. Sci.* **2016**, *47*, 1995–2008. https://doi.org/10.1080/00207721.2014.966282.
37. Ji, B.-X.; Liu, H.-H.; Cheng, P.; Ren, X.-Y.; Pi, H.-D.; Li, L.-L. Phased optimization of active distribution networks incorporating distributed photovoltaic storage system: A multi-objective coati optimization algorithm. *J. Energy Storage* **2024**, *91*, 112093. https://doi.org/10.1016/j.est.2024.112093.
38. Aljaidi, M.; Mashru, N.; Patel, P.; Adalja, D.; Jangir, P.; Arpita; Pandya, S.B.; Khishe, M. MORIME: A multi-objective RIME optimization framework for efficient truss design. *Results Eng.* **2025**, *25*, 103933. https://doi.org/10.1016/j.rineng.2025.103933.
39. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. https://doi.org/10.1109/TEVC.2007.892759.
40. Dehghani, M.; Montazeri, Z.; Trojovská, E.; Trojovský, P. Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowl.-Based Syst.* **2023**, *259*, 110011. https://doi.org/10.1016/j.knosys.2022.110011.
41. Su, H.; Zhao, D.; Heidari, A.A.; Liu, L.; Zhang, X.; Mafarja, M.; Chen, H. RIME: A physics-based optimization. *Neurocomputing* **2023**, *532*, 183–214. https://doi.org/10.1016/j.neucom.2023.02.010.
42. Ojha, V.; Timmis, J.; Nicosia, G. Assessing ranking and effectiveness of evolutionary algorithm hyperparameters using global sensitivity analysis methodologies. *Swarm Evol. Comput.* **2022**, *74*, 101130. https://doi.org/10.1016/j.swevo.2022.101130.
43. Guo, P.; Liang, B.; Li, S.; Yang, K.; Li, Q. Research on trajectory conflict detection and resolution for TBO. In Proceedings of the International Conference on Guidance, Navigation and Control, Singapore, 16–18 May 2025; pp. 381–392.
44. Liu, Z.; Xiao, G.; Mao, J. An integrated multi-objective model for demand-capacity balancing and strategic de-confliction under autonomous aircraft flight. *Transp. Res. Part C Emerg. Technol.* **2025**, *174*, 105102. https://doi.org/10.1016/j.trc.2025.105102.
45. Ma, X.; Li, Z.; Yang, Z.; Lu, X.; Zheng, L. Flight conflict detection of large fixed-wing UAV in joint airspace. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2024**, *Epub ahead of printing*. https://doi.org/10.1177/09544054241256996.
46. Li, Y.; Du, W.; Yang, P.; Wu, T.; Zhang, J.; Wu, D.; Perc, M. A satisficing conflict resolution approach for multiple UAVs. *IEEE Internet Things J.* **2019**, *6*, 1866–1878. https://doi.org/10.1109/jiot.2018.2885147.


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.