



REGULAR PAPER

A framework for optimising flight efficiency of a crossing waypoint by balancing flight conflict frequency and flight-level usage benefits

D. Sui¹ and K. Liu^{1,2}

¹Nanjing University of Aeronautics and Astronautics, College of Civil Aviation, Nanjing, China and ²Anhui Civil Aviation Airport Group, Hefei, China

Corresponding author: K. Liu; Email: liukechen@nuaa.edu.cn

Received: 28 April 2022; **Revised:** 31 March 2023; **Accepted:** 2 May 2023

Keywords: air traffic management; cross waypoint optimization; flight conflict resolution; flight level optimization

Abstract

With the increase of air transportation, some crossing waypoints (CWPs) are becoming bottlenecks in the operation of air traffic networks. This paper presents a CWP operation optimisation framework based on a two-stage optimisation method. First, we considered the interests of airlines and air traffic controllers and established a flight-level dynamic allocation model for the CWP to minimise the flight-level deviation and the number of flight conflicts. A multi-objective, self-adaptive differential evolution-local search hybrid algorithm was used to solve the model in a parallel computing manner. Subsequently, a flight conflict resolution algorithm based on the Monte-Carlo tree search was designed for flight conflicts that existed after the optimisation. Finally, based on real operation data, four experimental scenarios were constructed, and the air traffic operation simulation system was used for experimental validation. For daily traffic and 1.2 times peak traffic scenarios, the average flight-level deviation reduction rates after optimisation were 53% and 39%, and the successful flight conflict resolution rates reached 89% and 75%, respectively. The experimental results showed that this optimisation framework can effectively balance the number of flight conflicts with the efficiency of flight-level usage and directly improve the capacity of the CWP, which can be used as a reference for air traffic control auxiliary decision support systems.

Nomenclature

AFL_i^{min}	the lowest available flight-level for flight i
AFL_i^{max}	the highest available flight-level for flight i
C_1	the number of flight conflicts in the CWP core airspace
C_2	the number of flight conflicts in the CWP altitude-adjustment airspace
D	the flight-level deviation for all flights
FL_i	the flight-level allocated to flight i , $i = 1, 2, 3, \dots, N$
FPL_i	the flight plan of flight i , $i = 1, 2, 3, \dots, N$
N	number of flights in the scenario to be optimised
OFL_i	the optimal flight-level for flight i , $i = 1, 2, 3, \dots, N$
RFL_i	the flight-level of flight i before optimisation
R_i	the flight path of flight i , $i = 1, 2, 3, \dots, N$
$T(h)$	the start time of the time horizon h , $h = 1, 2, 3, \dots, N_{horizon}$
$\Delta T_{horizon}$	the length of each time horizon
$\Delta T_{interval}$	the length of each time interval
τ_i	the flight path of flight i , $i = 1, 2, 3, \dots, N$

Abbreviations

<i>ATCO</i>	air traffic controller
<i>ATOSS</i>	air traffic operation simulation system
<i>BCWP</i>	busy crossing waypoint
<i>COOF</i>	CWP operation optimisation framework
<i>CWP</i>	crossing waypoint
<i>DE</i>	differential evolution
<i>DOCM</i>	dynamic optimisation control module
<i>FCRM</i>	flight conflict resolution module
<i>FLAM</i>	flight-level allocation module
<i>MCT</i>	Monte-Carlo tree
<i>MCTS</i>	Monte-Carlo tree search
<i>MILP</i>	mixed integer linear programming
<i>MINLP</i>	mixed-integer nonlinear programming
<i>MOEAs</i>	multi-objective evolutionary algorithms
<i>RHC</i>	receding horizontal control
<i>SDEMO</i>	self-adaptive differential evolution multi-objective optimisation
<i>SDELS</i>	self-adaptive differential evolution-local search hybrid algorithm
<i>UCT</i>	upper confidence bound apply to tree

1.0 Introduction

The scale of civil aviation traffic has shown a rapidly increasing trend in recent years, with a report from Eurocontrol [1] indicating that daily flights in Europe increased by 8% in 2019 compared to 2018. In 2019, US and foreign airlines serving the United States set a new record by carrying 1.053 billion scheduled service passengers, representing a 3.9% increase from the previous annual record of 1.014 billion passengers in 2018 [2]. China's civil aviation completed 4.96 million flight operations during the same period, with an increase of 5.8% every year [3]. As traffic volumes continue to increase, congestions at crossing waypoints (CWPs) due to the convergence of traffic flow have become increasingly severe, and some busy crossing waypoints (BCWPs) have become bottlenecks in the operation of the routes network [4]. Although congestion has eased during the Coronavirus outbreak, the alleviation due to this crisis is temporary, and improving the efficiency of the traffic at route intersections remains an issue that must be addressed urgently. At BCWPs, owing to the convergence of traffic flow in multiple directions, the surrounding airspace has a high-traffic density and a high frequency of flight conflicts. To meet the required vertical and horizontal separation and to ensure safety due to work pressure, air traffic controllers (ATCOs) are often forced to adopt control strategies that sacrifice efficiency. Therefore, the International Civil Aviation Organization (ICAO) has put forward the optimisation goal of reducing congestion and flight fuel consumption at BCWPs in the Aviation System Block Upgrade (ASBU) and has developed modules, namely, B0-FRTO, B0-OPFL and B1-FRTO, with the core of trajectory planning and flight-level optimisation to achieve the above optimisation objectives.

Current research mainly uses trajectory planning to optimise air traffic resources at the level of supply and demand. However, it is to improve the operational efficiency of the air traffic networks by controlling the traffic volume at the bottlenecks, which essentially does not improve the capacity of CWPs. Furthermore, controllers at busy CWPs often prioritise safety over efficiency, which can significantly increase flight operation costs. Given the continuous growth of air traffic, operational bottlenecks will always affect the operational efficiency and safety of the network. As such, it is imperative to investigate optimisation methods that can improve the capacity of CWPs, thereby mitigating congestion and reducing operational costs.

The primary contribution of this paper is developing a simulation-based optimisation approach to propose a CWP operational optimisation framework (COOF). The COOF provides an economic flight-level allocation strategy that considers the frequency of flight conflicts and also provides flight conflict resolution strategies to ensure the safe crossing of BCWPs airspace. The main objective of the COOF is

to provide decision support to air traffic controllers for improving the current inefficient flight operation patterns at BCWPs, reducing congestion and minimising flight fuel consumption.

The structure of the paper is as follows: Section 2.0 presents a review of relevant research work. Section 3.0 outlines the current status of BCWPs operation, followed by an analysis of the problem of CWP operational optimisation. Next, we present the COOF developed in this paper. Technical approaches and details adopted for the flight-level allocation module and the flight conflict resolution module in COOF are discussed in Sections 4.0 and 5.0, respectively. In Section 6.0, we evaluate the effectiveness of the COOF using various examples. Finally, Section 7.0 summarises the research and discusses future research directions.

2.0 Literature review

There is not much research on improving CWP operational efficiency. Mostly, it is improved indirectly through flight trajectory planning. The few studies focused on improving CWP efficiency by planning conflict-free flight paths or combining flight conflict resolution with advanced trajectory planning. This paper will review past research on flight trajectory planning and conflict resolution to find the best ways to optimise CWP operational efficiency.

2.1 Flight trajectory planning

The related research can be divided into strategic planning (pre-departure planning) and tactical planning (in-flight planning) according to the flight phase. Strategic planning is more concerned with the overall operation of the air traffic network, and its core research focus is to use multiple means to pre-separate flight trajectories, thereby reducing the interactions between flights.

In pre-departure planning, flight trajectory separation can be achieved by adjusting the departure time [5, 6]. Although the implementation of ground delay programs can reduce the additional fuel consumption while separating flight trajectories, this will lead to longer ground waiting times and significantly increased delay levels as the flight volume increases; there are also studies that achieve trajectory separation by adjusting the flight speed [7], which essentially separates the flight trajectories in time. However, the limitation of this method is that the adjustable time-range is small.

Reassigning flight paths to flights can also separate flight trajectories. Two heuristic algorithms were used in Ref. (8) to reassign flight paths to flights, which could balance the distribution of traffic flows at the sector level but did not manage flight conflicts. Trajectory separation can also be achieved by optimising the flight altitude. In Ref. (9), Barnier et al. applied graph colouring techniques to reallocate the flight-levels for flights. In Ref. (10), time uncertainty was considered, and the Tabucol graph colouring algorithm was used to achieve flight-level allocation for large-scale flights. In Ref. (11), based on the spectral graph colouring model, a flight-level assignment method was proposed that can balance flight safety and operational costs.

Recent pre-departure trajectory planning studies typically combine multiple trajectory separation means and are evolving into robust trajectory planning methods. For example, Chaimatanan [12] considered aircraft uncertainty in the horizontal position and designed a hybrid meta-heuristic algorithm to reallocate takeoff times and flight paths for flights to find planning solutions with minimal interaction between trajectories. In Ref. (13), Chaimatanan extended the work in Ref. (12) by considering uncertainty in horizontal and vertical aircraft positions and enhancing the hybrid metaheuristic optimisation algorithm proposed in Ref. (12) for solving large-scale mixed-variable optimisation problems. A novel trajectory planning method is proposed in Ref. (14) to optimise flight costs, which divides the trajectory planning process into two flight plan determination processes: horizontal flight planning (determining the waypoints of the flight plan path) and vertical flight planning (determining the flight profile and the altitude/speed change position of the flight). Furthermore, a method was introduced to consider the uncertainty in the time domain. The Ref. (15) presents a mixed-variable optimisation model for the robust trajectory planning problem, which considers uncertainty and unpredictable events in trajectory

prediction. This model integrates ground delay, flight path adjustment, and flight altitude adjustment as means of adjustment.

Strategic-level trajectory planning can reduce the number of flight conflicts, reduce the workload of ATCOs, and realise the overall operational planning of the air traffic network, but its actual implementation effect is vulnerable to uncertainties and is very likely deviate from the planning objectives. For CWP, pre-departure trajectory planning reduces the operational stress and controls the amount of traffic passing through CWPs; however, it suppresses the demand for flights (e.g., by making them fly away from their optimal cruising altitude or making them wait on the ground) and does not improve the capacity of the CWPs.

In-flight planning uses flight-level, speed and heading adjustments to achieve local optimisation of flight trajectories. Because in-flight planning focuses more on the operation of the local area in the route network, the planning objectives are different from the pre-departure planning. Reference (16) prioritises the safety of flight trajectories and proposes a stochastic storm model coupled with an optimal control algorithm. The model maximises the probability of the flight reaching the designated waypoint while avoiding hazardous weather areas. Part of the research focuses on the conflict resolution problem in the local area, and in Ref. (17), a generic solution framework is developed for the problem of chain-of-flight-conflicts that may arise from the cross convergence of multi-directional traffic flows in a two-dimensional airspace by reserving conflict zones for traffic flows. Ref. (18) uses spatial discretisation technology to flexibly adjust the airspace entry point of flights, and combined with flight speed adjustment can avoid flight conflicts in real-time. It effectively reduces the monitoring time of ATCOs. There are also studies that focus on the target trajectory control of flights, such as the research in Ref. (19), in which a real-time 4D trajectory guidance method for flights to reduce the deviation between the actual flight trajectory and the planned 4D trajectory, with the expected arrival time of the flight as a constraint, was investigated. Pasini [20] proposed a CWP-optimisation operation, in which as many straight flight paths as possible are provided for the aircraft to cross the CWP while allowing flights to follow the original flight path. The flight-level and speed are adjusted to avoid flight conflicts, and a conflict-free trajectory with minimal flight-level and speed adjustments is planned for all flights expected to pass through the CWP.

Tactical-level trajectory planning usually treats flight conflicts as hard constraints or optimisation objectives, and for flight conflicts that cannot be completely avoided, the model resolves them by fine-tuning the flight profiles of the affected flights or directly handing them over to the ATCO. For direct CWP operation optimisation studies, Ref. (20) is used to solve the congestion problem by generating flight trajectories that avoid the CWP; however, the model has limitations in its application due to the airspace restrictions that usually exist around a CWP.

2.2 *Flight conflict resolution*

For the conflict resolution problem, the current leading solutions include optimal control methods, mathematical planning methods, swarm intelligence optimisation or search methods, geometric optimisation methods, and machine learning methods.

Matsuno [21, 22] developed a stochastic optimal control method for aircraft conflict resolution based on the polynomial chaos expansion and pseudo-spectral methods, considering the uncertainty of spatially correlated winds. Tang [23] introduced a dynamic conflict resolution method that accounts for wind vector perturbation. The method employs heading and ground speed adjustments as conflict resolution strategies, incorporating a receding horizon control mechanism. Liu [24] proposes a new model to describe the stochastic wind dynamic and the convergent weather areas with time-evolving size and adopts the stochastic optimal control method to realise that the aircraft keeps collision-free and has the best trajectory. Optimal control methods typically yield a conflict-free optimised trajectory consisting of successive position points for the aircraft. This trajectory can guide flights to modify their flight plans at conflict resolution's strategic and tactical levels. However, at the pre-tactical level, the conflict resolution scheme obtained differs substantially from the schemes employed by ATCOs in actual operations.

Mathematical planning methods generally use mixed-integer nonlinear programming (MINLP) [25–29] or mixed integer linear programming (MILP) [30, 31] to build mathematical models. There are

also studies that build a general nonlinear model [32] or follow a two-step optimisation approach to solve the conflict resolution problem [33]. Means of flight conflict resolution include adjustment of aircraft altitude, speed, heading, etc. Mathematical planning methods utilise dedicated solvers for efficient computation, enabling them to meet real-time requirements for conflict resolution. However, the many assumptions made in mathematical planning methods often lead to impractical solutions. As the number of aircraft increases, the sharp increase in model complexity poses computational challenges.

The swarm intelligence optimisation or search method uses simulation to build the airspace operating environment. Conflict resolution solutions are obtained in an iterative or search manner. The method can consider multiple nonlinear constraints and easily migrate to different scenarios. Ma [34] abstracted the conflict resolution environment as a grid and used a genetic algorithm to solve the conflict resolution problem under free flight conditions. Emami [35] studied the conflict resolution problem based on a multi-agent approach using the particle swarm optimisation algorithm. Sui [36] addressed the conflict resolution problem in congested airspace by employing a Monte-Carlo tree search (MCTS) algorithm with aircraft speed, altitude and heading adjustment as conflict resolution strategies.

The geometric optimisation method is simple and intuitive and hence, often applied in the engineering field for conflict resolution, such as in the NASA study [37–39]. Recent research has often used machine learning methods to solve conflict resolution problems, such as supervised and deep reinforcement learning methods. Supervised learning methods (e.g., convolutional neural network) use conflict resolution datasets to build and train models to obtain a mapping between conflict scenarios and conflict resolution solutions [40–42]. Deep reinforcement learning methods focus on making models self-learn, so that they can be made to behave like humans [43, 44]. Although the machine learning method has a fast solution speed and high intelligence level, its generality in different airspace operation environments is not strong. Hence, it is less used in practical engineering applications.

3.0 CWP Operation optimisation

This section introduces the current operational status of CWP, specifies the CWP operational optimisation problem to be addressed in this paper, and proposes the COOF along with its relevant implementation details.

3.1 Problem analysis

Owing to the convergence of a large number of flights, flight conflicts are highly likely to arise in BCWPs. The resolution of a large number of flight conflicts leads to an increased workload for the ATCO and congestions at the BCWPs. For example, the waypoint RESMI in Brest airspace, France, which is a convergence node for 14 routes, is often congested at 37,000ft and requires traffic flow sequencing to avoid conflicts [20]. At the waypoint HFE (Luogang VOR) in China, the ATCO reduces the workload by specifying the flight-level used by each direction of traffic flow, as shown in Fig. 1. Although this operation mode reduces flight conflicts, there are often large deviations between the flight-level used by flights and the optimal cruise altitude, which increases the fuel consumption of flights.

An operational mode that allows flexible allocation of flight-levels could lead to a significant increase in flight conflicts. The first core problem to be addressed is minimising the overall deviation between the allocated and desired flight-levels while ensuring minimum conflicts. The second core issue is generating resolution recommendations for the remaining conflicts to reduce the controller's workload from a safety perspective. There is a temporal and interactive relationship between solving the first and second problems for the same batch of flights in a given time period and an interaction between the optimisation of different batches of flights for consecutive flights spanning multiple periods. Therefore, a COOF should be constructed to rationalise the various interactions and influences. The two core issues are then examined in detail.

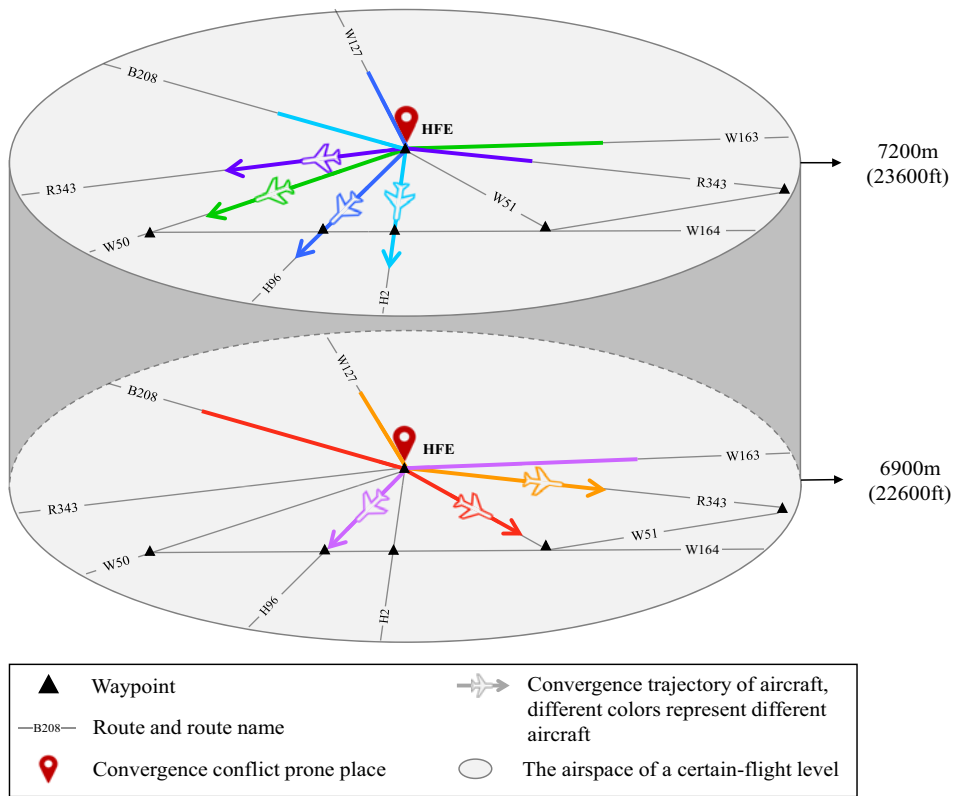


Figure 1. The operation mode of waypoint HFE.

3.2 CWP operation optimisation framework

Due to the convergence of multiple routes in CWP, there are complex configurations and high traffic volumes, which can result in severe operational knock-on effects if the optimisation solution is unrealistic. Traditional mathematical models require necessary assumptions about the execution process, leading to less practical optimisation results. On the other hand, although simulation-based optimisation methods take longer to solve than mathematical modelling methods, simulation-based optimisation methods can recreate optimisation scenarios, simulate flight operating conditions and consider uncertainties in flight operations. Therefore, this paper uses the Air Traffic Operation Simulation System (ATOSS) as a simulation tool and adopts a simulation-based optimisation method to construct the CWP operational optimisation framework.

COOF provides ATCOs with a continuous flight-level allocation and conflict resolution strategy. It comprises three functional modules, namely the dynamic optimisation control module (DOCM), flight-level allocation module (FLAM) and flight conflict resolution module (FCRM), combined based on the application sequence to form the COOF, as shown in Fig. 2.

- DOCM:** The RHC strategy is a finite-time horizon optimisation control strategy that utilises rolling optimisation of time horizons to achieve dynamic optimisation. The optimisation time is divided into $N_{horizon}$ time horizons of length $\Delta T_{horizon}$, with each time horizon comprising $N_{interval}$ time intervals of length $\Delta T_{interval}$. The start time of the time horizon h is $T(h)$. Figure 3 employs $N_{interval} = 3$ as an illustrative example. Only a time horizon is optimised at each optimisation. However, the optimisation outcomes are executed solely during the initial time interval of the optimised horizon, with the optimised time interval serving as an environmental variable for

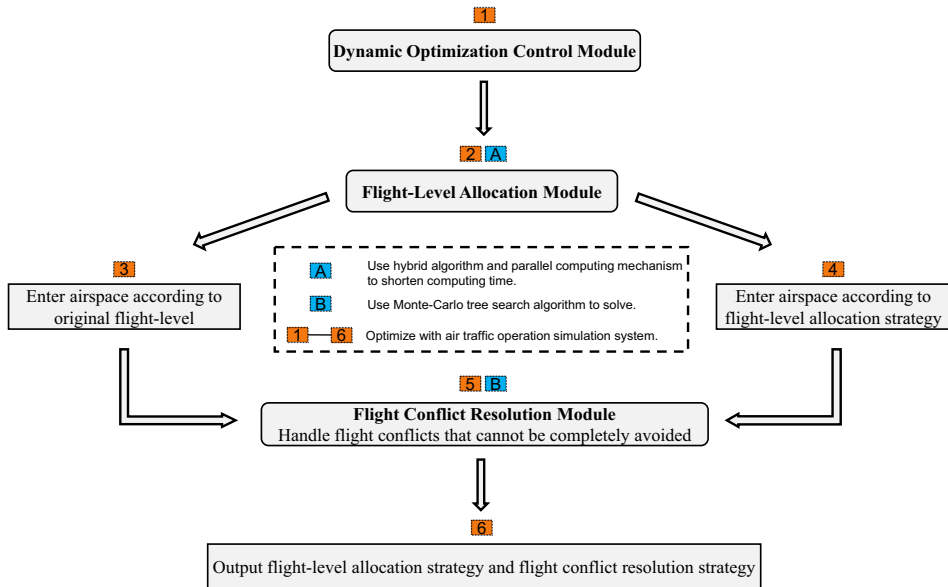


Figure 2. CWP operation optimisation framework.

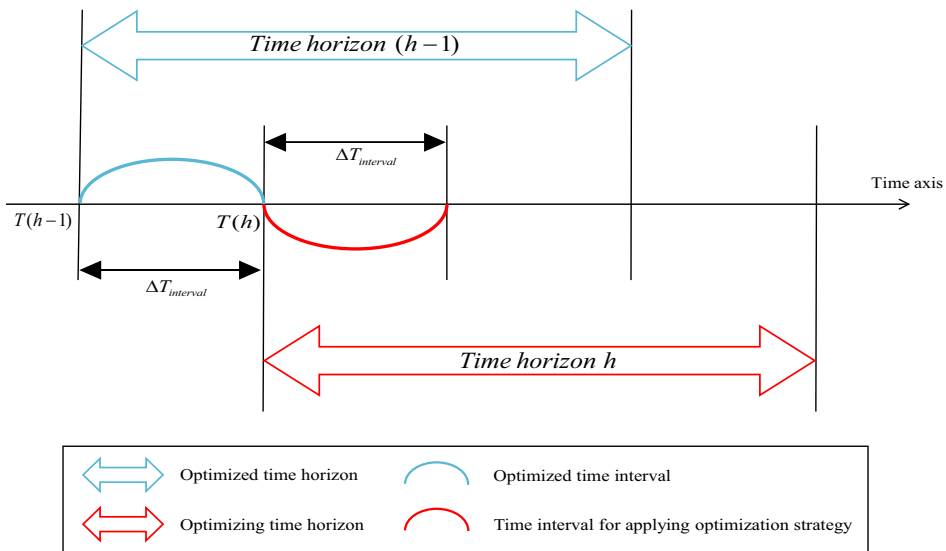


Figure 3. Dynamic optimisation process based on RHC strategy.

subsequent optimisations. The optimisation computation commences earlier than the intended optimisation time horizon to enable a continuous optimisation process. For instance, the optimisation of time horizon h is completed during $[T(h-1), T(h))$. Consequently, the maximum optimisation computation time for each time horizon is $\Delta T_{interval}$.

- **FLAM:** The module calculates a flight-level allocation strategy for upcoming flights passing through the CWP, allowing them to approach their optimal cruise flight-level while minimising potential conflicts. This simulation-based optimisation approach is described in technical detail in Section 4.0.

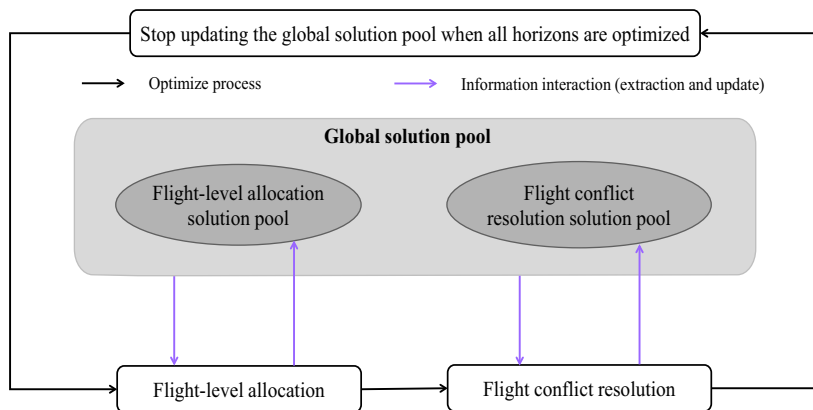


Figure 4. Strategy extraction and update process for global strategy pool.

- **FCRM:** The module manages potential conflicts arising from the flight-level allocation strategy by providing a resolution strategy. If a resolution strategy is not computed within the expected timeframe, the controller takes over and provides a resolution strategy, which is documented. The module utilises the MCTS algorithm and real-time scenario simulation to search for flight conflict resolution strategies, detailed in Section 5.0.

The flight-level allocation and flight conflict resolution processes within an optimised time horizon are sequential, with the former influencing the latter. Optimisation strategies implemented in the optimised time horizon will impact the results in the un-optimised time horizon. Given that COOF employs a simulation-based optimisation approach to implement module functions, a global strategy pool is constructed to recreate optimisation scenarios accurately and rationalise timing and module interaction across different optimisation time horizons. The global policy pool is a memory for optimisation strategies based on time horizon updates. As depicted in Fig. 4, optimisation strategies are first extracted from the global strategy pool before different modules commence optimisation to recreate optimisation scenarios. The global strategy pool is further divided into flight-level allocation and flight conflict resolution strategy pools, facilitating storage and timely updating of optimisation strategies.

To prevent flight conflicts resulting from arbitrary adjustments to flight operations in the complex operational situation at the CWP, the airspace at CWP has been partitioned into distinct functional areas. A circular area with a radius of 80km is designated as the CWP airspace, with the CWP as the centre of the circle. A circular area with a radius of 30km is defined as the CWP core airspace, in which the flight-status should be adjusted as little as possible; a toroid area with a width of 50km between the CWP airspace and its core airspace is the altitude-adjustment airspace.

Figure 5 depicts the optimisation process of COOF. The start time of the optimised time horizon, T_h , is determined by the DOCM based on the earliest time flights enter CWP airspace. Aircraft 1 is expected to enter the CWP airspace during the first time interval of the current optimised time horizon. Therefore, it will perform the optimisation results from that time horizon. The flight-level allocation and flight conflict resolution strategies are then calculated by the FLAM and FCRM modules, respectively. The calculation process is completed within the time range $[T_h - \Delta T_{interval}, T_h)$. Aircraft 1 executes the flight-level allocation and flight conflict resolution strategies as soon as it enters the altitude-adjustment airspace. Suppose the flight-level allocation strategy is unavailable in time. In that case, the flight follows the original flight plan, and the flight conflict resolution module handles any potential conflicts, as illustrated by Aircraft 2 in Fig. 5.

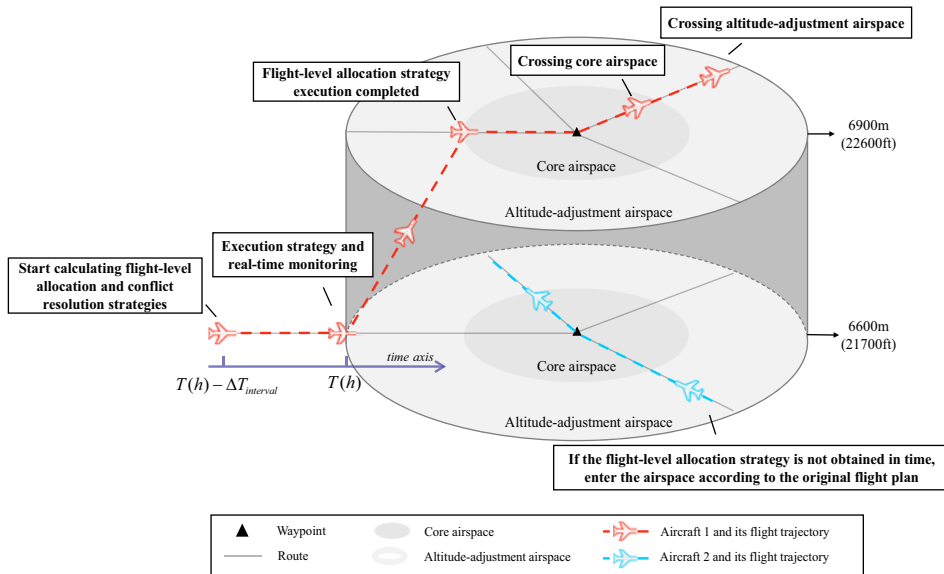


Figure 5. The optimisation process of COOF.

4.0 Flight-level allocation at CWP

This section presents the technical details of the flight-level allocation module in COOF, covering the flight-level allocation model and the hybrid optimisation algorithm employed for its solution.

4.1 Flight-level allocation model

The flight-level allocation module in COOF aims to keep flights as close to their optimal cruising altitude as possible while reducing the number of flight conflicts. Simulation is used to evaluate the objective function, constraints and flight motion state. The subsequent section presents a detailed description of the flight-level allocation model developed in this paper.

4.1.1 Assumption

BCWPs are generally composed of multiple cross-routes. A small number of BCWPs in the air traffic network are often scattered in different busy areas, hence, the optimisation scenario of the model is set to a single representative BCWP. The optimised environment does not consider the effects of bad weather, and the aircraft follows the route and crosses the BCWP airspace at the flight-level allocated for it.

4.1.2 Decision variables

In the optimisation model, the decision variable denoted as FL_i represents the flight-level allocated to flight i . FL_i is selected from the flight-level provisioning table, and the total number of flights optimised is denoted as N , $i = 1, 2, \dots, N$. The flight-level provisioning table is shown in Table 1.

4.1.3 Optimisation objectives

- (i) Minimise the number of flight conflicts generated by the flight in the CWP core airspace.

$$\min C_1 = \sum_{j=i+1}^N \sum_{i=1}^N \text{conflict}_1(FL_i, FL_j, FPL_i, FPL_j) \tag{1}$$

Table 1. *Flight-level provisioning table and its code*

Flight-level (m)	6000	...	$6000 + 300 \times 8$ (8400)	8900	...	$8900 + 300 \times 12$ (12, 500)
Flight-level (ft)	19,700	...	$19,700 + 1000 \times 8$ (27, 700)	29,100	...	$29,100 + 1000 \times 12$ (41, 100)
Serial number	0	...	8	9	...	21

where C_1 denotes the number of flight conflicts in the CWP core airspace. The flight plan of flight i , denoted as $FPL_i = \{RFL_i, R_i, \tau_i\}$, comprises of its planned flight-level RFL_i , flight path R_i , and expected arrival time τ_i in the CWP airspace. A binary variable $conflict_1(FL_i, FL_j, FPL_i, FPL_j)$ is employed to identify potential conflicts between flight i and flight j when adjusting their flight-level to FL_i and FL_j , respectively, based on their original flight plan, within the core airspace of the CWP. A value of 1 is assigned to $conflict_1(FL_i, FL_j, FPL_i, FPL_j)$ if a conflict is detected. Otherwise, $conflict_1(FL_i, FL_j, FPL_i, FPL_j)$ is assigned a value of 0.

The simulation system assesses potential flight conflicts using a simulation-based optimisation method in the model calculation. According to CCAR-93TM-R5 guidelines for aircraft safety intervals, the simulation system establishes a horizontal safety interval of 10km and a vertical safety interval of 300m. Any horizontal or vertical Euclidean distance between two aircraft that falls below these safety intervals is flagged as a flight conflict.

- (ii) Minimising the deviation of the actual flight-level from the optimal cruise altitude.

$$\min D = \sum_{i=1}^N |FL_i - OFL_i| \quad i = 1, 2, \dots, N \tag{2}$$

D is the flight-level deviation for all flights. OFL_i represents the optimal flight-level for flight i based on the aircraft type and flight distance. Meanwhile, the flight-level allocated to flight i within the CWP airspace is denoted by FL_i .

4.1.4 Constraints

- (i) Flight-level adjustment constraint: the amount of flight-level adjustment is limited by the adjustment space and aircraft performance

$$AFL_i^{\min} \leq FL_i \leq AFL_i^{\max} \quad i = 1, 2, \dots, N \tag{3}$$

AFL_i^{\min} and AFL_i^{\max} denote the lowest and highest available flight-levels, respectively, for flight i . These values are determined by the climb and descent performance of the aircraft within the limited space of the CWP altitude-adjustment airspace. After calculation, the flight altitude adjustment space is established between the upper and lower three available flight-levels in the same direction as the planned flight-level RFL_i (where two flight altitude layers separated by one flight altitude layer are referred to as flight altitude layers in the same direction). By combining the lowest and highest flight altitude layers of the CWP flight, the adjustment boundaries AFL_i^{\min} and AFL_i^{\max} for FL_i can finally be determined.

- (ii) Conflict number constraint in the altitude-adjustment airspace: flights should not create flight conflicts with other flights while implementing the flight-level allocation strategy.

$$C_2 = \sum_{j=i+1}^N \sum_{i=1}^N conflict_2(FL_i, F_j, FPL_i, FPL_j) = 0 \tag{4}$$

where C_2 denotes the number of flight conflicts in the CWP altitude-adjustment airspace. $conflict_2(FL_i, FL_j, FPL_i, FPL_j)$ is a binary variable used to determine the existence of a flight conflict between flight i and flight i with altitude adjustment to FL_i and FL_j under the original flight plan in the CWP altitude-adjustment airspace. The value of $conflict_2(FL_i, FL_j, FPL_i, FPL_j)$ is 1 if there is a conflict and 0 if there is none. The criteria for determining a flight conflict are the same as mentioned above.

- (iii) Constraint on the number of conflicts in the core airspace: to ensure that the calculation of the flight-level allocation and flight conflict resolution strategies is completed within the specified time, the number of flight conflicts in the core airspace is limited to at most 2.

$$C_1 \leq 2 \tag{5}$$

4.2 Self-adaptive differential evolution-local search hybrid multi-objective optimisation algorithm

Because there is no apparent linear relationship between the two objective functions established in this study, it is difficult to transform them into a single-objective form, and the evaluation of the optimisation objectives and the constraints is done by simulation; therefore, an intelligent algorithm is chosen to solve the model. The evolutionary algorithm-based algorithm is the most widely used intelligent algorithm for solving multi-objective optimisation problems, and research shows that multi-objective evolutionary algorithms (MOEAs) that use differential evolution (DE) algorithms as the evolutionary strategy have better results than those based on other search strategies [45]. In this study, an improved multi-objective self-adaptive differential evolution multi-objective optimisation (SDEMO) algorithm based on an elite selection strategy is chosen to solve the flight-level allocation model. To enhance the convergence speed, a local search algorithm is also introduced in this paper. It has been demonstrated in several studies that the hybrid algorithm produced by combining the evolutionary algorithm with the local search algorithm not only has a strong global search capability, but also has a good neighbourhood search capability, both in terms of solution accuracy and computation time relative to individual algorithms [46, 47].

4.2.1 SDEMO algorithm

The SDEMO algorithm proposes a new DE algorithm variation strategy and parameter adaptive control strategy according to the characteristics of multi-objective optimisation problems. The experimental study shows that the second-generation multi-objective evolutionary algorithm represented by Non-dominated Sorting Genetic Algorithm-II (NSGA-II), whose local density estimation method is based on Euclidean distance, cannot accurately reflect the crowding degree of individuals. Therefore, the SDEMO algorithm improves the individual density estimation method and elite selection strategy. After the improvement, the algorithm improves the performance of the DE algorithm based MOEAs in solving optimisation problems with —two to three objectives in an overall way. The improvements in the SDEMO algorithm are described in detail below.

- **Individual density estimation method:**

To enhance population diversity and the uniformity of the non-dominated solution set distribution, SDEMO adopted an improved individual density estimation method that utilises harmonic mean distance to estimate population density.

For an individual i in a population of N_p , assume that the k individuals in the target space with the closest Euclidean distance to i are at distances of $d_{i,1}, d_{i,2}, \dots, d_{i,k}$. The harmonic mean distance d_i of individual i is calculated as shown in Equation (6), where the value of k is set to $N_p - 1$. This approach determines individual crowding degree based on the influence of all individuals in the population except itself.

$$d_i = \frac{k}{\frac{1}{d_{i,1}} + \frac{1}{d_{i,2}} + \dots + \frac{1}{d_{i,k}}} \quad (6)$$

- **Elite selection strategy:**

To apply the elite selection strategy, individuals in the population are ranked based on non-dominance sorting, which compares the fitness values of multiple objective functions in a two-by-two manner. The Pareto-optimal solution among all solutions is assigned dominance rank 1. After eliminating the solutions in dominance rank 1, the Pareto-optimal solution is selected from the remaining population and assigned dominance rank 2. This process continues until all individuals in the population are ranked. The improved elite selection strategy in the SDEMO algorithm considers both the convergence and distribution metrics to enhance the diversity of the elite population. Based on the crowding density information, the strategy proceeds as follows: if the number of non-dominated solutions is less than N_p , all non-dominated solutions are retained to ensure population convergence, and 50% of individuals with smaller crowding density are drawn from each remaining Pareto class, from high to low, until the population size reaches N_p .

If the number of non-dominated solutions exceeds N_p , the pruning strategy in the Ref. (48) is employed to reduce the non-dominated solution individuals to N .

• **DE algorithm variation strategy:**

A new DE algorithm variation strategy is proposed in the SDEMO algorithm, and its effectiveness has been verified experimentally. The specific method is as follows:

$$p'_m = r \cdot p_{best} + (1 - r) \cdot p_m + F \cdot \sum_{k=1}^K (p_{m_f^k} - p_{m_b^k}) \tag{7}$$

where p_{best} represents the current best individual, p_m represents the target individual, and p'_m represents the post-variation individual. p_{best} is determined based on the dominance of p_m in relation to other individuals and the crowding density. If p_m is the dominant solution, p_{best} is chosen randomly from the solutions that dominate p_m . If p_m is the non-dominant solution, p_{best} is chosen randomly from the top five individuals with the lowest crowding density among all non-dominant solutions in the current population. r is the greed factor; K is the number of difference vectors; F is the variation factor, and $p_{m_f^k}$ and $p_{m_b^k}$ are solutions that are selected from the population, which are unidentical to p_m .

• **Parameter adaptive control strategy:**

In multi-objective optimisation, more non-dominated individuals at the early stage indicate poorer population diversity and convergence. To ensure diversity and distributivity of the final solutions, the SDEMO algorithm adapts by increasing individual variability at the early stage to control non-dominated individuals in the population. The adaptive parameter adjustment strategy aims to maintain diversity early and increase the diffusion rate of non-dominated individuals to the sparse region later. The adjustment method is as follows:

$$\begin{cases} F(t) = F_{\min} + (F_{\max} - F_{\min}) * e^{-2 \times \frac{t}{Gen}} \\ CR(t) = CR_{\min} + (CR_{\max} - CR_{\min}) * e^{-2 \times \frac{t}{Gen}} \\ r(t) = r_{\min} + (r_{\max} - r_{\min}) * e^{-2 \times \frac{Gen-t}{Gen}} \end{cases} \tag{8}$$

where t is the number of population evolutionary generations, Gen is the maximum number of evolutionary generations, CR is the crossover factor, F is the variance factor and r is the greed factor.

Combining the above algorithm improvements, the process of the SDEMO algorithm is as follows. The SDEMO algorithm randomly generates multiple individuals to form the initial population, calculates the fitness value of each individual in the population on multiple objectives, generates new child populations using variation and crossover operations, merges the child-parent populations, performs non-dominance ranking and crowding calculations on them, applies the elite selection strategy to form new populations and iterates through this cycle until the end condition is satisfied.

4.2.2 Hybrid algorithm design

To improve the algorithm's search capability and shorten its convergence time, we combine the SDEMO algorithm with the local search algorithm to design a hybrid algorithm for solving the flight-level allocation model.

Since the model is calculated by the simulation-based optimisation method, there are no special requirements for encoding the flight-level, so the relevant calculations involving the flight-level and the chromosome encoding of the hybrid algorithm are encoded in real numbers in the paper. The available flight levels are transformed into real numbers starting from 0, as shown in Table 1, with 19,700ft coded as 0, 20,700ft coded as 1, and so on. Because flights enter the CWP airspace in a sequential order, the order of flights arriving in the CWP airspace is used as the order of chromosome coding in terms of flight-levels. For example, if the flight-levels of four flights are listed as 26,600ft–27,600ft–29,100ft–37,100ft in the order they enter the CWP airspace, the chromosome codes are 7-8-9-17.

To improve the algorithm's search capability and shorten its convergence time, we combine the SDEMO algorithm with the local search algorithm to design a self-adaptive differential evolution-local search hybrid algorithm (SDELS) for solving the flight-level allocation model. The hybrid algorithm procedure is as follows:

- Step 1:** Initialisation settings: set the population size (N_p), iteration times (Gen), current iteration (t), crossover factor (CR), mutation factor (F), greedy factor (r) and difference vector number (K).
- Step 2:** Generate the initial population (P) with N_p individuals, simulate and calculate the fitness value, and perform the non-dominated sorting and congestion calculation.
- Step 3:** Select, cross and mutate P to generate an offspring population (O) containing N_p individuals.
- Step 4:** Combine the parent population P and offspring population O , that is $R = P \cup O$.
- Step 5:** Non-dominated sorting and crowding degree calculations are carried out on the merged population (R), and a new parent population (P) containing N individuals is generated by applying an elite selection strategy.
- Step 6:** When $t \bmod 3 = 0$, select the solution with the highest Pareto level from P , generate a local population (L) with N_p individuals, and calculate its fitness value. Apply the elite selection strategy to $P \cup L$ and update to produce a new parent population P with N_p individuals.
- Step 7:** $t = t + 1$, and calculate F , CR , r according to the parameter update method.
- Step 8:** if $t \leq Gen$, then go to Step 3.
- Step 9:** Output the best individual and decode it to get the flight-level allocation strategy.

5.0 Conflict resolution based on MCTS

The flight conflict resolution module in COOF uses the MCTS method to resolve flight conflicts that cannot be avoided during flight-level allocation. This section details the technical aspects of the module, including the selection of resolution actions and the design of the MCTS algorithm.

Flight conflict resolution refers to the process of avoiding conflict by changing the flight speed, flight altitude and heading after flight conflict is detected, which is essentially a decision-making process. The MCTS algorithm can address the issue of being trapped in a local solution in a vast search space and produce effective solution strategies quickly. As it is an asymmetric decision tree construction process that can be terminated at any point, we employ it in combination with real-time scenario simulation to explore in-flight conflict resolution strategies and ensure the reliability of the resulting strategies in this study.

The MCTS-based conflict resolution algorithm derives the conflict resolution strategy by performing a large number of resolution action simulations for the current state. In this study, we mainly focus on the two-plane conflict resolution problem, considering the characteristics of a real-time and restricted resolution space, and choose three basic conflict resolution action types, namely, altitude adjustment, speed adjustment and offset (dog-leg), as shown in Fig. 6, in terms of resolution action design, and the action magnitude are designed according to the CWP airspace characteristics, as listed in Table 2.

Considering the execution time of the conflict resolution action, the execution time of the height adjustment and speed adjustment actions is not more than 120s according to simulation experiments, and it is stipulated that at most two resolution actions can be selected for each conflict resolution, but no two offset actions can be selected consecutively. The start execution times is set as 240s and 120s before the conflict occurs. When a pair of aircraft is expected to have a flight conflict, one aircraft is randomly selected to execute the conflict resolution action. If there are multiple flight conflicts to be resolved simultaneously, one aircraft from each pair is selected to perform the conflict resolution action; however, the aircraft selected to perform the conflict resolution actions must be different. When the number of aircraft involved in flight conflicts is less than the number of conflicts, selecting a different

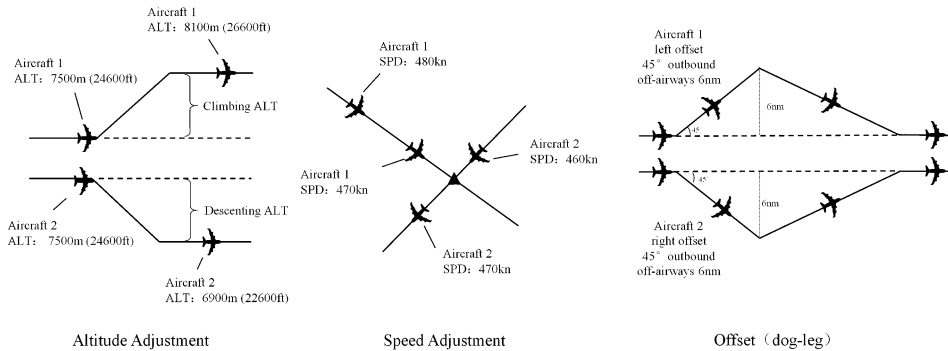


Figure 6. Flight conflict resolution action types.

aircraft for each conflict pair to resolve the conflicts is not possible. This paper limits the number of conflicts to two during flight-level allocation in CWP airspace to address this issue. Suppose no flight-level allocation strategy is found and the number of conflicts generated by the original flight plan upon entering CWP airspace exceeds two. In that case, the controller will handle the conflicts.

The specific conflict resolution algorithm is designed as follows.

Step 1 Define the Monte-Carlo tree (MCT) structure

The MCT contains a root node and multiple layers of child nodes. The root node is the airspace situation when the flight conflict resolution is initiated, and the child nodes are all the flight states after the conflict resolution action is executed. As shown in Fig. 7, if there is only one flight conflict, the first layer of child nodes represents all flight states after the execution of the first resolution action, and the second layer of child nodes represents all flight states after the execution of the second resolution action, with its parent node as the initial state. If there are two flight conflicts to be resolved, the first level sub-node indicates the status of all flights after the execution of the two resolution actions to resolve the first flight conflict, and the second level sub-node indicates the status of all flights after the execution of the two resolution actions to resolve the second flight conflict. The numbers on each node in Fig. 7 correspond to the release action numbers in Table 2.

Step 2 Selection process

The selection process starts from the root node of the MCT and finds a child node that is most appropriate for exploration according to the selection strategy, usually using the upper confidence bound apply to tree (UCT) algorithm as the evaluation function, prioritising the unexplored child nodes and selecting the child node with the largest UCT value if all the children have been explored.

$$UCT = \bar{X}_v + 2C_p \sqrt{\frac{2\ln N(v)}{N(v')}} \tag{9}$$

where \bar{X}_v is the expectation of the reward value obtained from all simulations on a node, $N(v)$ is the number of times the node (V) has been selected, and $N(v')$ is the number of times a child node has been visited. when a child node has never been visited, the UCT of this node tends to positive infinity, ensuring that all children in this layer are traversed before the next layer is expanded. C_p is a constant greater than 0. The former term of the evaluation function, \bar{X}_v , is called the exploitation term (Exploitation), and the latter term is the exploration term (Exploration). The evaluation function uses C_p to adjust the balance between the two terms.

Step 3 Expansion process

The selected node is used as the root node to create a new child node. In this study, we set the maximum number of flight conflicts that need to be resolved simultaneously to 2. Therefore, the expansion node can only be the first layer child node, and the search result must be generated when the second

Table 2. Flight conflict resolution action

Action type	Adjust direction	Action range	Action number
Altitude adjustment	Climb	600m (2000ft)	1
		1200m (4000ft)	2
	Descent	600m (2000ft)	3
		1200m (4000ft)	4
Speed adjustment	Acceleration	10kn	5
	Deceleration	10kn	6
		20kn	7
Offset	Left offset	45° outbound, off-airways 6nm	8
	Right offset	45° outbound, off-airways 6nm	9

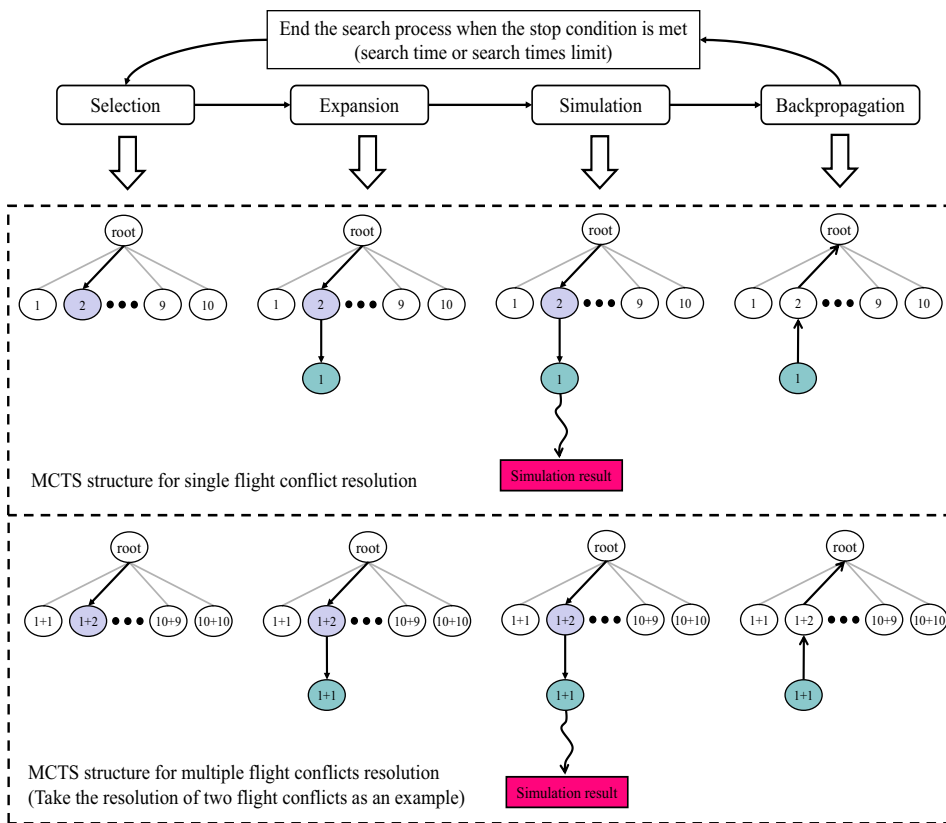


Figure 7. MCTS-based flight conflict resolution process.

layer child node is searched so that the expansion operation is not performed on the second layer child node.

Step 4 Simulation process

Starting from the expanded child node, the simulation is carried out using the random selection strategy to the state where all conflict resolution actions are completed and the conflict resolution result is obtained. When there is only one flight conflict, the simulation process ends if the execution of the

Table 3. Reference table of the best cruising altitude for common aircraft types

Type of aircraft	Itinerary (nm)						
	300	400	500	600	700	800	≥ 850
B737	10,100	10,100	10,700	11,300	11,300	11,300	11,900
A320	10,100	10,100	10,700	10,700	10,700	10,700	10,700
A321	10,100	10,100	10,700	10,700	10,700	10,700	10,700
CRJ9	9500	10,100	10,100	10,100	10,100	10,100	10,100

The optimum cruise altitude in the table is in meters (m), and its corresponding feet altitude can be obtained from the Chinese flight-level configuration table. This table is only the flight altitude in one direction, the optimal cruise altitude for flight in the opposite direction must be increased by 300m (1000ft).

selected first-level child node causes a new conflict to appear and the resolution result is a failure; otherwise, the resolution result is unresolved and the execution of the selected second-level child node continues. If the original conflict is resolved and no new flight conflict is created, the simulation process ends and the resolution result is a success; otherwise, the resolution result is a failure.

When there are two flight conflicts, after executing the action of the selected first-level child node, if one of the flight conflicts is not successfully resolved or a new flight conflict is created, the result of conflict resolution is a failure and the simulation ends. When one of the flight conflicts is successfully resolved and no new flight conflict is generated, the resolution result is not completely resolved, the selected second layer of child node actions is executed, and after the first and second layer of the child node actions are all executed, if both flight conflicts are successfully resolved and no new flight conflict is generated, the conflict resolution result is a success; otherwise, it is a failure.

Step 5 Backpropagation process

The result of the conflict resolution simulation is fed back to the parent node layer by layer, and the number of successes, number of visited nodes, and reward value of all selected nodes are updated and recorded. The reward values for successful and failed deconflictions are 1 and 0, respectively.

After the search starts, the node selection, expansion, simulation, and backpropagation processes are continuously performed until the number of searches reaches the preset number. At the end of the search, the action corresponding to the child node with the largest UCT value at each layer is selected as the conflict resolution strategy obtained from the search.

6.0 Simulation and discussion

In this section, we evaluate and validate the optimisation performance of COOF by designing various experimental scenarios.

6.1 Simulation design

Before evaluating the optimisation effect of COOF, we introduce the experimental data, environment, procedure, parameter settings and evaluation index.

6.1.1 Experimental data

Based on the results of a study by the MITRE corporation [49] on the optimal cruise altitude for flights and referring to the optimal cruise altitude curves provided by airlines for each aircraft type, we determined the optimal cruise altitude for commonly used aircraft types, which are listed in Table 3.

The relevant parameters of the DOCM are set as follows: time horizon size, $\Delta T_{horizon} = 10\text{min}$; time interval size, $\Delta T_{interval} = 5\text{min}$; number of time intervals, $N_{interval} = 2$; the total number of optimised time horizons, $N_{horizon} = 3$; current time horizon, $h = 1$; optimisation start time, $T(h)$, $h = 1, 2, 3 \dots N_{horizon}$.

Table 4. Number of flights in the four experimental scenarios

Horizon start time	Horizon end time	Number of flights in Scenario A	Number of flights in Scenario B	Number of flights in Scenario C	Number of flights in Scenario D
8:15	8:20	6	9	9	11
8:20	8:25	6	8	10	10
8:25	8:30	6	9	10	11
8:30	8:35	7	9	9	10

The 2019 Civil Aviation Airspace Development Report indicates that HFE, the busiest waypoint in China, has a peak traffic volume of 106 vehicles/h [4]. Using radar track data and flight plan data from the 13th of August, 2019, a time slice (8:15~8:35) containing three time horizons was randomly intercepted, and a total of 25 flights flew through this time slice. The flight plan (arrival time, flight path, flight altitude level, flight speed, etc.) of the flights crossing the HFE airspace was collected to construct experimental Scenario A, which represents the daily traffic level. To explore the optimisation performance of the COOF in a high flight traffic environment, experimental Scenario B, with a flight volume that reached peak traffic, and experimental Scenarios C and D, with the flight volumes that reached 1.1 and 1.2 times higher than the peak traffic were constructed by adjusting the flight information, and the specific experimental scenario information is listed in Table 4.

Owing to the randomness of the simulation experimental results of a single scenario, 200 simulation inputs were generated for each of the four experimental scenarios for the COOF simulation verification before the simulation experiments.

6.1.2 Experimental environment

- **Simulation platform:** An ATOSS developed by our team is used to cooperate with the optimisation process. The system uses a simulation motion engine based on the BADA database to deduce the state transfer of the environment and aircraft during the simulation, which in turn enables the trajectory projection and prediction, and the predicted trajectory can be used for conflict detection and deconfliction, as shown in Fig. 8.
- **Hardware environment:** HP Z8 G4 Workstation, Intel Xeon(R) Gold 6242 CPU @ 2.8GHz, RAM 64GB.
- **Software environment:** The development tool is IntelliJ PyCharm, and the algorithms were written using Python 3.7.
- **Parallel computing setup:** The multiprocessing module in Python was used to perform multiprocess parallel computing in the hybrid algorithm to calculate the population fitness. Based on the simulated experimental environment and algorithm parameter settings, it was determined that the best acceleration was achieved when the number of concurrent processes was 25, and the average individual computation time was reduced by 68%.

6.1.3 Simulation experiment process

The procedure of the joint simulation experiment is as follows:

Step 1: Initialise the receding horizon window and set $h = 1$;

Step 2: Update the receding horizon information to obtain all flights that are expected to arrive in the HFE airspace within $[T(h), T(h) + \Delta T_{horizon}]$. Flights with expected arrival time in $[T(h), T(h) + \Delta T_{interval}]$ constitute the set of flights to be optimised, that is, $A(h)$. Flights with

6.1.5 Evaluating indicator

- Flight-level deviation

$$\Delta h_{ori} = \sum_{i=1}^N |FL_i - OFL_i| \quad (10)$$

$$\Delta h_{opt} = \sum_{i=1}^N |FL_i - OFL_i| \quad (11)$$

$$\overline{\Delta h_{ori}} = \frac{\sum \Delta h_{ori}}{N_s} \quad (12)$$

$$\overline{\Delta h_{opt}} = \frac{\sum \Delta h_{opt}}{N_s} \quad (13)$$

where Δh_{ori} and Δh_{opt} denote the flight-level deviation of a single simulation before and after optimisation, respectively, $\overline{\Delta h_{ori}}$ and $\overline{\Delta h_{opt}}$ are the average flight-level deviations before optimisation and the average flight-level deviation after optimisation for N_s simulations, respectively.

- Flight-level deviation reduction rate

$$R_{\Delta h} = \frac{\Delta h_{ori} - \Delta h_{opt}}{\Delta h_{ori}} \quad (14)$$

$$\overline{R_{\Delta h}} = \frac{\sum R_{\Delta h}}{N_s} \quad (15)$$

where $R_{\Delta h}$ is the flight-level deviation reduction rate of a single simulation experiment and $\overline{R_{\Delta h}}$ is the average flight-level deviation reduction rate of N_s simulation experiments.

- Percentage of economic flights

$$R_{OFL} = \frac{N_{opt}}{N} \quad (16)$$

$$R'_{OFL} = \frac{N'_{opt}}{N} \quad (17)$$

$$\overline{R_{OFL}} = \frac{\sum R_{OFL}}{N_s} \quad (18)$$

$$\overline{R'_{OFL}} = \frac{\sum R'_{OFL}}{N_s} \quad (19)$$

The economic flights are the flights flying at the optimal cruising altitude, N_{opt} and N'_{opt} denote the number of flights flying at the optimal altitude before and after optimisation, R_{OFL} and R'_{OFL} are the percentage of economic flights in a single simulation experiment before and after optimisation, $\overline{R_{OFL}}$ and $\overline{R'_{OFL}}$ denote the average economic flight percentages before and after the optimisations of N_s simulation experiments, respectively.

- Flight-level adjustment degree

$$R_{deg} = \frac{\Delta h_{opt}}{\Delta H} \quad (20)$$

$$\overline{R_{deg}} = \frac{\sum R_{deg}}{N_s} \quad (21)$$

Table 5. Types of optimisation results for a single time horizon

The flight-level allocation strategy is obtained	Have remaining flight conflicts	The remaining flight conflicts were successfully resolved	Optimisation result type
✓	✓	✓	AO-HS
✓	✓	×	AO-HF
✓	×		AO-NC
×	✓	✓	AN-HS
×	✓	×	AN-HF
×	×		AN-NC

Table 6. Statistics of time horizons optimisation results for Scenario A

Optimisation results	Number of time horizons	Proportion of total time horizon number
AO-HS	295	49.2%
AO-HF	38	6.3%
AO-NC	212	35.3%
AN-HS	14	2.3%
AN-HF	2	0.3%
AN-NC	39	6.5%

where ΔH denotes the maximum flight-level adjustment of the experimental scenario, R_{deg} denotes the flight-level adjustment degree in a single simulation experiment, and $\overline{R_{deg}}$ is the average flight altitude layer adjustment degree of N_s simulation experiments. The adjustable flight-level of the flight is limited by the performance of the aircraft and the range of the altitude-adjustment airspace and cannot be adjusted indefinitely to reduce the flight-level deviation.

6.2 Result discussion

The optimisation results of COOF for a single time horizon can be classified into six categories. Obtaining the flight-level allocation strategy is labelled as AO, while failing to obtain it is labelled as AN. If there is a remaining flight conflict and the conflict resolution module successfully resolves it, it is recorded as HS. If the resolution fails, it is recorded as HF. Finally, if there is no remaining flight conflict, it is labelled as NC. These categories are determined based on the flight-level allocation results and the conflict resolution situation. The detailed categories are presented in Table 5.

The results of 200 simulation experiments for Scenario A show that the percentage of time horizons (AO-HS, AO-NC) that successfully planned flight-levels for flights and completed flight conflict resolution reached 84.5%. Regardless of the completion of flight-level allocation, COOF can resolve flight conflicts, and the percentage of time domains (AO-HS, AO-NC, AN-HS, AN-NC) that did not require additional processing by ATCO reaches 93.4%, as shown in Table 6. This indicates that the COOF can provide stable decision support to the ATCOs and effectively reduces their workload.

In each simulation experiment, flights in the optimised time horizon that do not fly out of the HFE airspace are considered as environmental variables in the subsequent time horizon, which increases the number of flights to be considered for optimisation in the subsequent time horizon. Because the simulation system is called frequently during the optimisation process, the computation time of the subsequent time horizon may increase; therefore, the computation time of each time horizon in

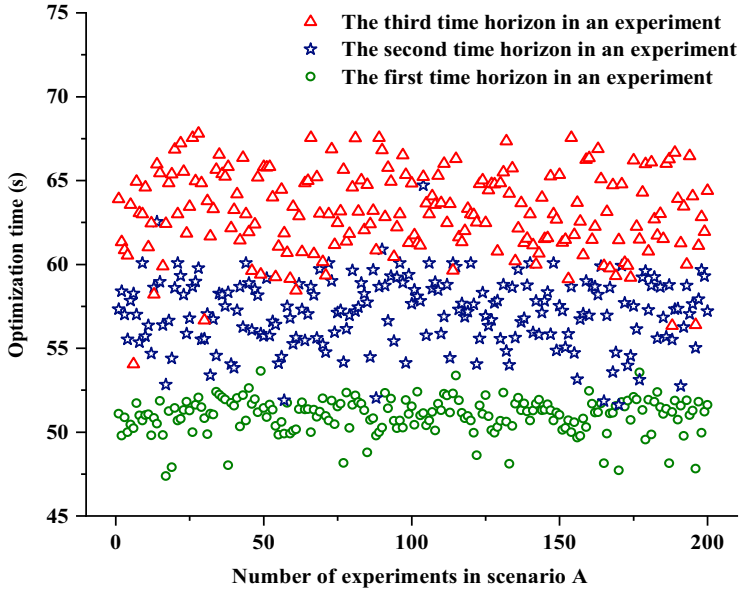


Figure 9. Time horizon operation time statistics for 200 simulation inputs for Scenario A.

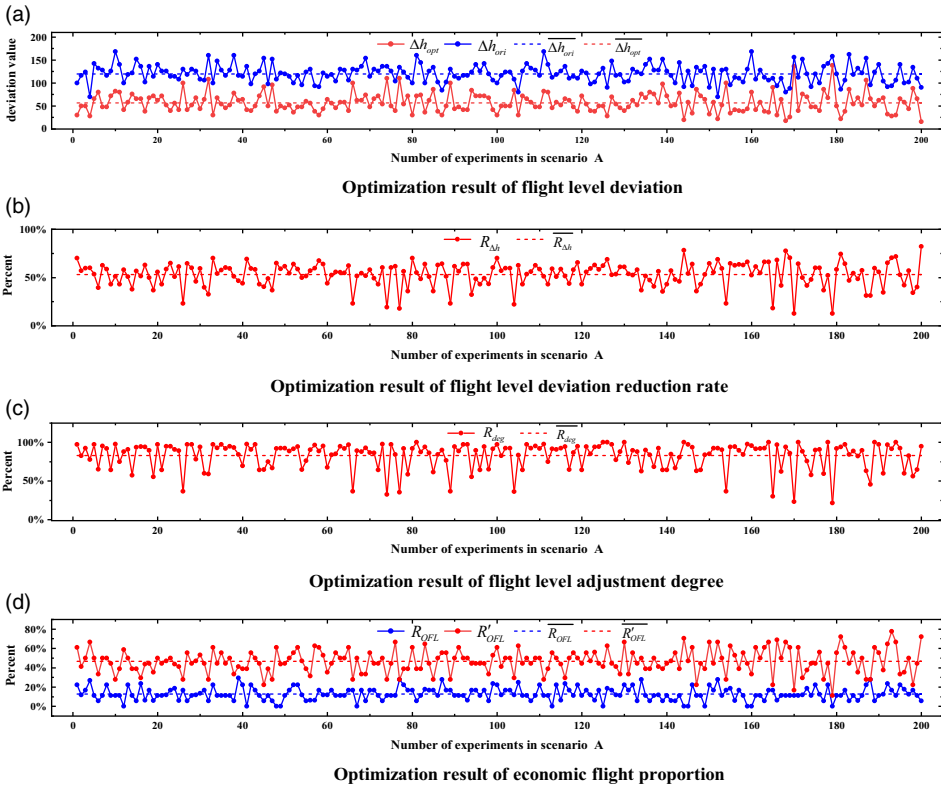


Figure 10. Flight-level optimisation results for Scenario A.

Table 7. Flight conflict resolution results for Scenario A

Evaluation indicators	Experimental results
Number of flights	25
Percentage of simulation inputs with flight conflicts	91%
Average time for a single conflict resolution	24s
Successful conflict resolution rate	89%

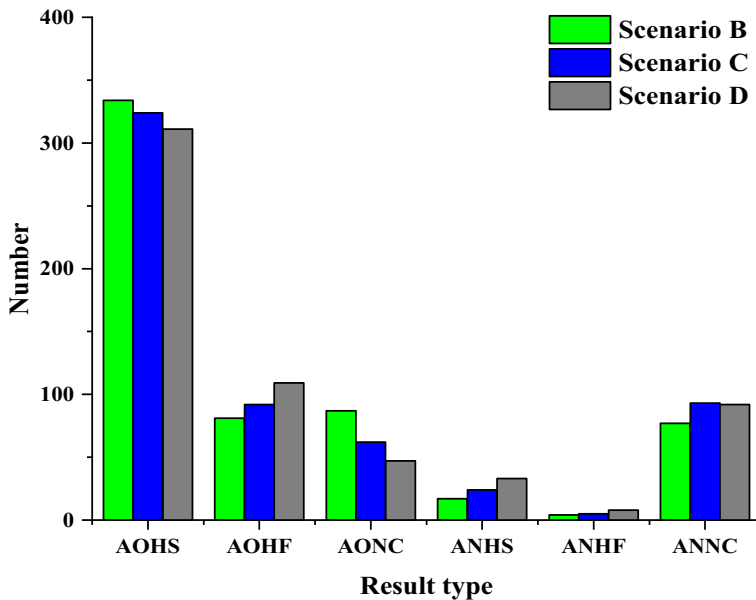


Figure 11. Statistics of time horizon optimisation results for incremental scenes.

Scenario A (the total computation time, including flight conflict resolution) shows a clear stratification phenomenon, as shown in Fig. 9.

For the flight-level allocation, the 200 optimised inputs for Scenario A had 63 less flight-level deviations on average after the COOF was applied, and the average flight-level deviation reduction rate was 53%, as can be observed from Fig. 10(a) and (b). After the optimisation was completed, the average flight-level adjustment degree of Scenario A reached 83%, and the percentage of economic flights was improved by 34% after the optimisation, as can be observed from Fig. 10(c) and (d). It is deduced that the COOF has shown a strong optimisation performance in reducing flight-level deviation.

In terms of flight conflict resolution, 91% of the optimised inputs in Scenario A had flight conflicts, and each flight conflict resolution took 24s. The COOF was able to successfully resolve most of the flight conflicts, with a resolution rate of 89%, as shown in Table 7.

In Scenarios B, C and D, as the number of flights in the experimental scenarios increases, there is a decreasing trend in the number of time horizons capable of generating flight-level allocation strategies, as can be observed in Fig. 11. Because some flights have the same optimal flight cruise altitude, with an increase in flight volume, there will inevitably be flight clustering at some flight altitude layers. Therefore, the possibility of generating flight conflicts increases significantly, and it becomes more difficult to generate flight-level allocation strategies that satisfy the conflict number limit. However, even in the highest traffic for Scenario D, the percentage of time horizons that successfully generate a flight-level allocation strategy reached 77.8%.

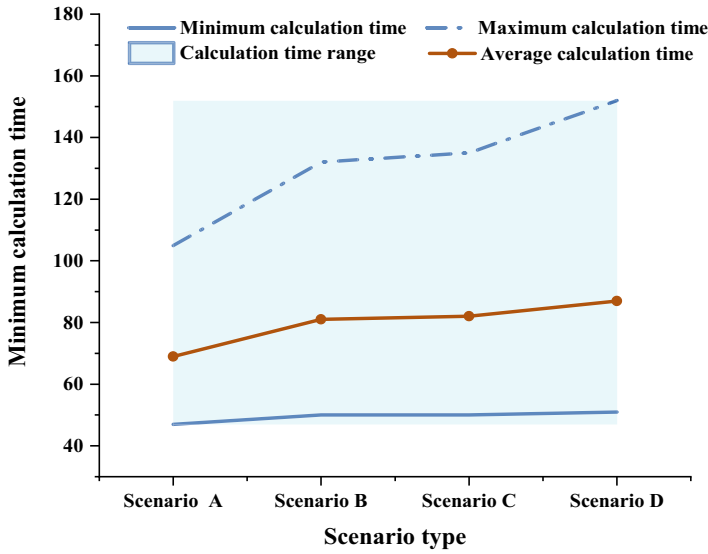


Figure 12. Computation time statistics for a single time horizon in four experimental scenarios.

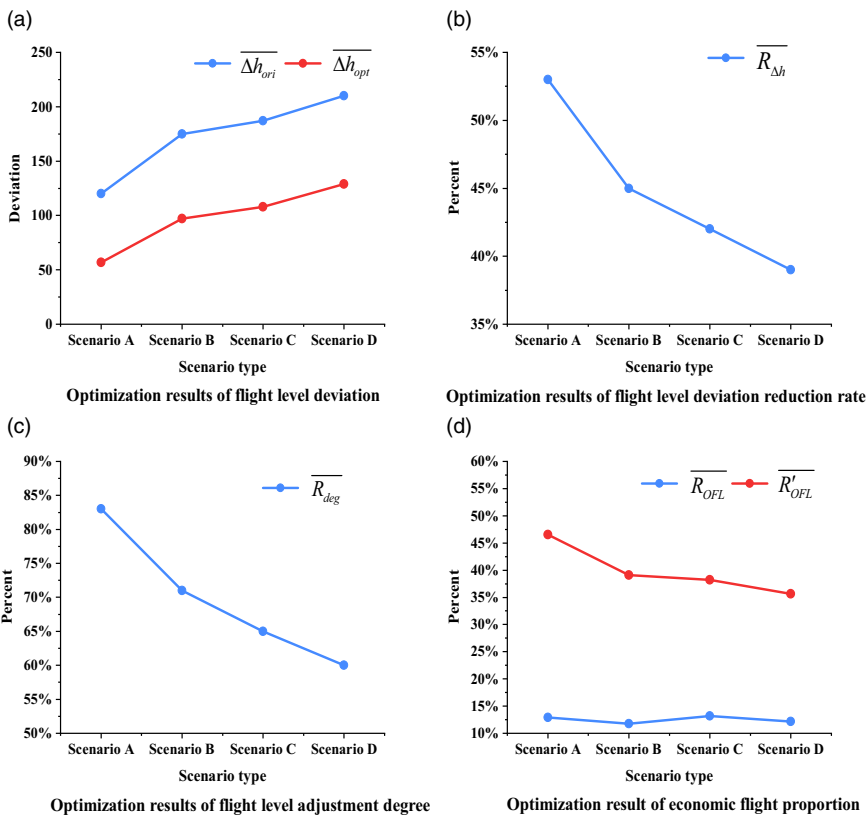


Figure 13. Flight-level optimisation results for incremental scenes.

Table 8. Flight conflict resolution results for incremental scenarios

Evaluation indicators	Scenario B	Scenario C	Scenario D
Number of flights	36	40	44
Percentage of simulation instances with flight conflicts	98.5%	99%	97%
Average time for a single conflict resolution	32s	36s	37s
Successful conflict resolution rate	81%	79%	75%

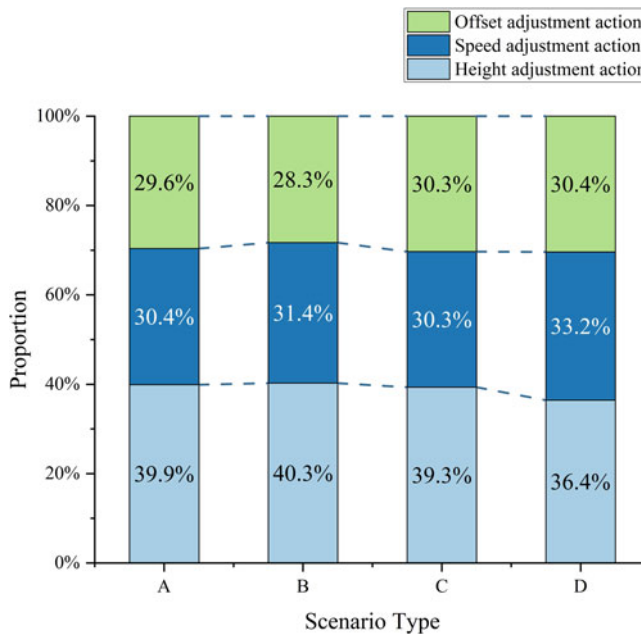


Figure 14. Distribution of conflict resolution actions selected for each scenario (obtained based on 200 experimental data from four types of experimental scenarios).

The computation time of an individual time horizon in a scenario is closely related to the number of flights in that time horizon, and the phenomenon of the computation time of subsequent time horizon gradually increasing as the optimised time horizon advances becomes more pronounced in the incremental scenarios (the difference between the shortest and longest computation times of the time horizon in Scenario D is 101 s), as can be observed from Fig. 12. However, the computation time of individual time horizons in each scenario does not exceed the set upper limit of computation time (as already mentioned, the upper limit of the calculation time for each time horizon is a time interval length).

As the number of flights in the optimised scenario increases, the average flight-level deviation, average flight-level deviation reduction rate, average flight-level adjustment degree, and average economic flight percentage data of each experimental scenario show a decreasing trend, as shown in Fig. 13. However, even in the scenario with 1.2 times higher peak traffic, after the COOF was applied, its average flight-level adjustment degree reached 60%, the average flight-level deviation was reduced by 81, the average flight-level deviation reduction rate reached 40%, and the number of economic flights was improved by 24% after optimisation.

In the three incremental scenarios, the percentage of optimised instances with flight conflict after flight-level allocation reached more than 97%, as shown in Table 8. Considering that the COOF adopts a simulation-based MCTS algorithm to find the conflict resolution strategy, an increase in flight volume leads to an increase in the conflict resolution time and a gradual decrease in the successful flight conflict

resolution rate, but with the successful flight conflict resolution rate of the three incremental scenarios maintained at above 75%. With an increase in traffic density, the solution space of the conflict resolution strategy gradually becomes smaller, which is the main reason for the low success rate of flight conflict resolution. In actual operation, when the traffic density in the airspace exceeds its carrying capacity, controllers generally adopt flow control or air wait to reduce the traffic density and ensure flight safety.

The distribution of the conflict resolution actions selected for each scenario is shown in Fig. 14. As can be seen in the figure, there is no significant difference in selecting the three conflict resolution actions in the high-density environment. The proportion of choosing two actions, altitude adjustment and speed adjustment, is slightly higher than that of choosing offset action, and the proportion of choosing altitude adjustment action is the highest.

7.0 Conclusion

In this study, a CWP operation optimisation framework was proposed, and a two-stage combined optimisation method was designed to reduce flight conflicts while maximising the demand for flights to fly at the optimal cruising altitude. Four experimental scenarios were constructed through a simulation system using real flight plan data and radar trajectory data. The experimental results showed that among the four experimental scenarios, the computation time of the two-stage optimisation meets the performance requirements, the minimum average flight-level deviation reduction rate was 40%, and the lowest successful flight conflict resolution rate was 75%. This operational optimisation framework is effective in improving ATC operational safety and reducing flight operation costs and can be used as a reference for the ATCO's auxiliary decision support system. The limitation of this paper is that the optimisation scenario does not consider aircraft diversions and detours affected by bad weather. Future work will consider the impact of severe weather conditions and attempt to expand the optimisation environment from a single CWP to a busy sector. Since the main focus of this paper is to improve the traffic efficiency of BCWP, the preferences of different conflict resolution actions are not considered. The preference for flight conflict resolution actions will be added to the future study of large-scale airspace operation optimisation.

Acknowledgements. This research was supported by Civil Aviation Professional Project (No.TM2018-5-1/2).

References

- [1] Eurocontrol. All-Causes Delay and Cancellations to Air Transport in Europe-2019, 2020.
- [2] DOT. 2019 Traffic Data for U.S. Airlines and Foreign Airlines, 2020.
- [3] CAAC. 2019 Civil Aviation Industry Development Statistics Bulletin, 2020.
- [4] Air Traffic Management Bureau, CAAC. 2019 Civil Airspace Development Report, 2020.
- [5] Terrab, M. and Odoni, A.R.O. Strategic flow management for air traffic control, *Oper. Res.*, 1993, **41**, (1), pp 138–152.
- [6] Barnier, N. and Allignol, C. 4D-trajectory deconfliction through departure time adjustment, *Proc. ATM*, Vol. 9, 2009.
- [7] Caferri, S. and Durand, N. Aircraft deconfliction with speed regulation: new models from mixed-integer optimization, *J. Global Optim.*, 2013, **58**, (4), pp 613–629.
- [8] Gianazza, D. and Durand, N. Separating air traffic flows by allocating 3D-trajectories, *The 23rd Digital Avionics Systems Conference (IEEE Cat. No. 04CH37576)*, Vol. 1, 2004.
- [9] Barnier, N. and Brisset, P. Graph coloring for air traffic flow management, *Ann. Oper. Res.*, 2004, **130**, (1), pp 163–178.
- [10] Allignol, C., Barnier, N. and Gondran, A. Optimized flight-level allocation at the continental scale, *5th International Conference for Research in Air Transportation (ICRAT)*, 2012.
- [11] Gimenez-Guzman, J.M., Martínez-Moraian, A., Reyes-Bardales, R.D., Orden, D. and Marsa-Maestre, I. Flight-level assignment using graph coloring, *Appl. Sci.*, 2020, **10**, (18), pp 6157–6172.
- [12] Chaimatanan, S., Delahaye, D. and Mongeau, M. A hybrid metaheuristic optimization algorithm for strategic planning of 4D aircraft trajectories at the continental scale, *IEEE Comput. Intell. Mag.*, 2014, **9**, (4), pp 46–61.
- [13] Chaimatanan, S., Delahaye, D. and Mongeau, M. Aircraft 4D Trajectories Planning Under Uncertainties, *2015 IEEE Symposium Series on Computational Intelligence*, 2015.
- [14] Dancila, R.I. and Botez, R.M. New flight trajectory optimisation method using genetic algorithms, *Aeronaut. J.*, 2021, **125**, (1286), pp 618–671.

- [15] Vidosavljevic, A., Delahaye, D. and Tomic, V. Homotopy route generation model for robust trajectory planning, *Air Traffic Manag. Syst. II*, 2017, pp 69–88.
- [16] Hentzen, D., Kamgarpour, M., Soler, M. and Gonzalez-Arribas, D. On maximizing safety in stochastic aircraft trajectory planning with uncertain thunderstorm development, *Aerosp. Sci. Technol.*, 2018, **79**, pp 543–553.
- [17] Treleven, K. and Mao, Z. Conflict resolution and traffic complexity of multiple intersecting flows of aircraft, *IEEE Trans. Intell. Transp. Syst.*, 2008, **9**, (4), pp 633–643.
- [18] Cecen, R.K. and Cetek, C. Conflict-free en-route operations with horizontal resolution manoeuvres using a heuristic algorithm, *Aeronaut. J.*, 2020, **124**, (1275), pp 767–785.
- [19] Gu, J., Tang, X., Hong, W., Chen, P. and Li, T. Real-time optimization of short-term flight profiles to control time of arrival, *Aerosp. Sci. Technol.*, **84**, 2019, pp 1164–1174.
- [20] Pasini, A., Notry, P. and Delahaye, D. Direct route optimization for air traffic management improvement, *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, 2018.
- [21] Matsuno, Y., Tsuchiya, T. and Matayoshi, N. Near-optimal control for aircraft conflict resolution in the presence of uncertainty, *J. Guid. Cont. Dynam.*, 2016, **39**, (2), pp 326–338.
- [22] Matsuno, Y., Tsuchiya, T. and Wei, J. Stochastic optimal control for aircraft conflict resolution under wind uncertainty, *Aerosp. Sci. Technol.*, 2015, **43**, pp 77–88.
- [23] Xinmin, T., Ping, C. and Bo, L. Optimal air route flight conflict resolution based on receding horizon control, *Aerosp. Sci. Technol.*, 2016, **50**, pp 77–87.
- [24] Wensheng, L. Aircraft trajectory optimization for collision avoidance using stochastic optimal control, *Asian J. Cont.*, 2019, **21**, (5), pp 2308–2320.
- [25] Cafieri, S. and Durand, N. Aircraft deconfliction with speed regulation: New models from mixed-integer optimization, *J. Global Optim.*, 2014, **58**, (4), pp 613–629.
- [26] Cafieri, S. and Rey, D. Maximizing the number of conflict-free aircraft using mixed-integer nonlinear programming, *Comput. Oper. Res.*, 2017, **80**, pp 147–158.
- [27] Cafieri, S. and Omhenni, R. Mixed-integer nonlinear programming for aircraft conflict avoidance by sequentially applying velocity and heading angle changes, *Euro. J. Oper. Res.*, 2017, **260**, (1), pp 283–290.
- [28] Junling, C. and Ning, Z. Mixed integer nonlinear programming for aircraft conflict avoidance by applying velocity and altitude changes, *Arab. J. Sci. Eng.*, 2019, **44**, (10), pp 8893–8903.
- [29] Alonso-Ayuso, A., Escudero, L.F. and Martn-Campo, F.J. Multiobjective optimization for aircraft conflict resolution. A metaheuristic approach, *Euro. J. Oper. Res.*, 2016, **248**, (2), pp 691–702.
- [30] Omer, J. A space-discretized mixed-integer linear model for air-conflict resolution with speed and heading maneuvers, *Comput. Oper. Res.*, 2015, **58**, pp 75–86.
- [31] Cecen, R.K. and Cetek, C. Conflict-free en-route operations with horizontal resolution manoeuvres using a heuristic algorithm, *Aeronaut. J.*, 2020, **124**, (1275), pp 767–785.
- [32] Hong, Y., Choi, B. and Oh, G. Nonlinear conflict resolution and flow management using particle swarm optimization, *IEEE Trans. Intell. Transp. Syst.*, 2017, **18**, (12), pp 3378–3387.
- [33] Cecen, R.K., Sarac, T. and Cetek, C. Meta-heuristic algorithm for aircraft pre-tactical conflict resolution with altitude and heading angle change maneuvers, *Top*, 2021, **29**, (3), pp 629–647.
- [34] Ma, Y., Ni, Y. and Ping, L. Aircraft conflict resolution method based on ADS-B and genetic algorithm, *Sixth International Symposium on Computational Intelligence & Design*. IEEE Computer Society, 2013.
- [35] Emami, H. and Derakhshan, F. Multi-agent based solution for free flight conflict detection and resolution using particle swarm optimization algorithm. *Upb Sci. Bull.*, 2014, **76**, (3).
- [36] Dong, S. and Kai, Z. A tactical conflict detection and resolution method for en route conflicts in trajectory-based operations. *J. Adv. Transport.*, 2022.
- [37] Bilimoria, K. A geometric optimization approach to aircraft conflict resolution, *18th Applied Aerodynamics Conference*, 2000.
- [38] Doweck, G., Munoz, C. and Geser, A. *Tactical Conflict Detection and Resolution in a 3-D Airspace*, Institute for Computer Applications in Science and Engineering, Hampton VA, 2001.
- [39] Carreno, V.A. Evaluation of a pair-wise conflict detection and resolution algorithm in a multiple aircraft scenario (No. NAS 1.15: 211963), 2002.
- [40] Durand, N., Guleria, Y. and Tran, P. A machine learning framework for predicting ATC conflict resolution strategies for conformal automation, *11th SESAR Innovation Days*, 2021.
- [41] Kim, K., Deshmukh, R. and Hwang, I. Development of data-driven conflict resolution generator for en-route airspace, *Aerosp. Sci. Technol.*, 2021, **114**, (1275), pp 106744.
- [42] Rooijen, S., Ellerbroek, J. and Borst, C. Toward individual-sensitive automation for air traffic control using convolutional neural networks, *J. Air Transport.*, 2020, **28**, (3), pp 105–113.
- [43] Li, H., Wang, Z. and Wang, J. Deep reinforcement learning based conflict detection and resolution in air traffic control, *IET Intell. Transp. Syst.*, 2019, **13**, (6), pp 1041–1047.
- [44] Pham, D.T. Reinforcement Learning for Two-Aircraft Conflict Resolution in the Presence of Uncertainty, *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, IEEE, 2019.
- [45] Tusar, T. and Filipic, B. Differential evolution versus genetic algorithms in multiobjective optimization, *International Conference on Evolutionary Multi-Criterion Optimization*, 2007.
- [46] Ponsich, A. and Coello Coello, C.A. A hybrid differential evolution tabu search algorithm for the solution of job-shop scheduling problems, *Appl. Soft Comput.*, **13**, (1), 2013, pp 462–474.

- [47] Li, X. and Gao, L. An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem, *Int. J. Prod. Econom.*, **174**, 2016, pp 93–110.
- [48] Kukkonen, S., Kalyanmoy, D. and Bateman, H. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems, *IEEE Congr. Evol. Comput.*, 2006.
- [49] Foster, J.G., DeArmon, J.S. and Bateman, H. An analysis of flight time at lower-than-optimal cruise altitude, *2018 Aviation Technology, Integration, and Operations Conference*, 2018.