Adaptive Conflict Resolution for Multi-UAV 4D Routes Optimization Using Stochastic Fractal Search Algorithm

Bizhao Pang [a, b], Kin Huat Low [a,*], Chen Lv [a]

a. School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798

b. Air Traffic Management Research Institute, Nanyang Technological University, Singapore 637460

Abstract: The increasing unmanned aircraft system (UAS) applications in urban environments pose challenges for safe and efficient low altitude air traffic management. As an essential enabler to meet these challenges, pre-flight 4D routes optimization is required to conduct conflict detection and resolution (CD&R) and to generate conflict-free flight routes before departure. Existing studies on strategic deconfliction cover several types of strategies such as scheduling or rerouting. However, a single type of strategy used to solve different types of conflicts may lead to an unsafe and inefficient way of conflict resolution. This paper proposes an adaptive decision-making framework to optimize the resolution strategies used for different types of conflicts with explainable mechanisms. The proposed framework is formulated as a double-layer optimization problem with the considerations of scheduling, speed adjustment, and rerouting strategies for conflict resolution. The first layer of the framework is established as a probabilistic selection model to make decisions on which strategy should be selected for what type of conflict. The second layer is developed as a mixed-integer nonlinear programming (MINLP) model to optimize the decision variables of the strategies selected by the first layer. To solve the proposed double-layer optimization problem, we introduce and improve a novel meta-heuristic stochastic fractal search (SFS) algorithm with two major improvements of a penalty-guided fitness function and an exploitation-exploration balancing scheme. Simulation results demonstrate that the proposed adaptive conflict resolution framework successfully optimizes the strategies used for each type of flight conflict, which subsequently optimizes the 4D routes with significant reductions in total operational cost, number of flight conflicts, and flight delays. The improved stochastic fractal search (ISFS) algorithm is also proved effective and reliable in solving the proposed optimization problem in different traffic density scenarios.

Keywords: unmanned aircraft system, conflict resolution, mixed-integer nonlinear programming, third-party risk, urban environments

Acronyms

ADM	Adaptive decision-making	ISFS	Improved stochastic fractal search
CD&R	Conflict detection and resolution	MINLP	Mixed-integer nonlinear programming
CWP	Crossing waypoint	OD	Origin and destination
ETA	Estimated time of arrival	PSO	Particle swarm optimization
ETD	Estimated time of departure	SFS	Stochastic fractal search
FDB	Fitness-distance balance	UAS	Unmanned aircraft system
IA	Immune algorithm	UAV	Unmanned aerial vehicle

* Corresponding author: K.H. Low (mkhlow@ntu.edu.sg)

Notations[2]

N_{uav}	Total number of UAVs	P_{best}	Particle with the best solution in the current group
N_{node}	Total number of nodes in the route network	ε	Weightage factor to adjust the degree of exploitation
N_{fc}	Total number of flight conflicts	ε'	Weightage factor to adjust the degree of exploration
C	Set of flight conflicts	B_{lower}	Lower bound of the constraint vector
c_i	ith flight conflict	B_{upper}	Upper bound of the constraint vector
S	Set of strategies for conflict resolution	D_P	Distance vector to evaluate the distance from current solution point to the best solution point
P_i	ith root particle in the population	F_P	Fitness vector to evaluate the fitness value of solutions
P_i^{η}	ηth new particle created by central particle P_i	S_P	Score vector to evaluate the overall score of the solutions considering D_P and F_P

1. Introduction

The rapid booming of UAS operations in and around urban airspace brings huge opportunities to unlock the potential of the sky (FAA & NASA, 2020; Hately & et al, 2019). For instance, during the Covid-19 pandemic, unmanned aerial vehicles (UAVs) were employed for various missions such as transporting vaccines and blood bags, delivering foods, conducting security patrol and monitoring, etc., which save the life of patients and prevent people from the risks of exposure to the virus (Patchou et al., 2021; Saeed et al., 2021). Besides, UAS operations also bring economic benefits, as drone delivery and urban air passenger transport could reshape the traditional way of people and cargo flows with fast speed and better experiences (Sacramento et al., 2019; Yang & Wei, 2021). The pollution and noise issues could also be reduced by leveraging electric UAVs. In addition, UAVs liberate us from labor-consuming tasks like security patrol, disaster inspection, and traffic monitoring, which enables us to do some more creative works.

However, we still rarely see large-scale operations happening in urban environments because several challenges are yet to be solved. First, UAVs operating in urban environments bring safety risks to third parties, such as fatality risks to people and property damage risks to important infrastructures (Blom et al., 2021; X. Hu et al., 2020). Flight conflict risk is another major challenge. The various UAV applications always require operating at the same period of time in the same airspace, as flight demands (e.g. food delivery and passenger transport) have peaks (Kleinbekman et al., 2020; Rigas et al., 2021). That increases the potential flight conflicts. To solve these challenges, the risk-based UAV 4D route optimization with adaptive conflict resolution problem is investigated. By generating the risk-based routes, UAVs can avoid not only explicit obstacles but the implicit high-risk areas where the probability of causing fatalities and damages is high. With the obtained risk-based routes, an effective conflict resolution method is required to optimize the pre-planned 4D routes by solving the conflicts using different types of strategies.

1.1. Related works

This review focuses on the decision-making strategies of conflict resolution for 4D routes optimization. The strategies consist of two aspects. The first aspect is from the temporal dimension including the scheduling and speed adjustment strategies. The scheduling is to change the estimated time of departure (ETD) of each flight while the speed adjustment will change the estimated time of arrival (ETA) of each waypoint. The second aspect is from the

[2] (Notations listed do not include those of mathematical formulations to be presented in Section 3.2.2.)

spatial dimension and the strategy is rerouting, which is to adjust the flight route by changing the coordinates (x, y, z). The commonly used uncertainty models in conflict resolution studies are also discussed in this section.

1.1.1. Strategies in temporal dimension: scheduling and speed adjustment

A. Scheduling strategy

Scheduling is an effective way to tackle pre-flight conflicts. By adjusting the departure time, potential conflicts or congestions can be solved. In civil aviation fields, the scheduling problems have been studied to optimize the air traffic flow management system. A 4D strategic deconfliction problem at a continental scale was developed as a mixed-integer programming model to optimize the trajectory interactions (Chaimatanan et al., 2014). In this study, the departure time of flights can be shifted to solve the potential trajectory interactions. In a follow-up study, the wind-optimal flight trajectories were considered in the strategic deconfliction problem for the North Atlantic oceanic airspace (Rodionova et al., 2016). The objective of this study is to minimize the conflicts by changing the departure time and trajectory, while constraints are given to minimize the deviations of the flight plan from the initial one. Authors also developed mathematical models to maximize the runway throughput (Prakash et al., 2018) as well as to utilize en-route capabilities (Zhang et al., 2019). A follow-up work (Birolini et al., 2021) considered the demand and supply balance and proposed an integrated flight scheduling framework to efficiently optimize the traffic flow.

Compared with the commercial manned aircraft, the scheduling of UAS operations has more challenges, as the UAS traffic network is more complex in low-altitude urban airspace. In recent years, researchers have studied the UAS scheduling problems with various optimization objectives, ranging from minimizing flight conflicts (Dai et al., 2021; Hao et al., 2018; Liu et al., 2019) and flight delays (Q. Tan et al., 2019; Wu, Low, & Hu, 2021), to maximizing coverage rate of surveillance missions (F. Cheng et al., 2018; M. Hu et al., 2019). To achieve these objectives, various models have been proposed. Mathematical programming is the most used one. Based on different modelling requirements of decision variables, authors (Q. Tan et al., 2019) developed the mixed-integer linear programming model to optimize the pre-planned routes in discrete environments where the departure time was taken as an integer. In other studies (Cabreira et al., 2018; C. Cheng et al., 2020; Rigas et al., 2021), the models involve energy constraint or payload constraint, which are nonlinear. The optimization problems were developed as mixed-integer nonlinear programming to generate feasible flight routes. To solve the proposed optimization models, both exact methods such as branch-and-cut algorithm (C. Cheng et al., 2020), dynamic programming algorithm (Peng et al., 2021), and heuristic algorithms like auction algorithm (Yao et al., 2019), genetic algorithm (Q. Tan et al., 2019) were developed. In some cases, the feasible region of scheduling problems is nonconvex due to the complex objective or constraint functions (Da Silva Arantes et al., 2019). To solve the nonconvex optimization problems for drone scheduling, authors investigated an approach to transform a nonconvex problem into two convex subproblems without sacrificing the optimality of the solution (F. Cheng et al., 2018). They also proposed an exact effective algorithm to solve the transformed convex problems. What is more, some authors paid attention to developing more robust methods and algorithms for scheduling and task assignment (M. Hu et al., 2019; Yao et al., 2019). They introduced the iterative strategies to improve the performance of the task assignment algorithm and their strategies improved the stability of proposed methods while reduced the computational complexity.

B. Speed adjustment strategy

Another conflict resolution strategy is to adjust the flight speed, which subsequently adjusts the ETA for the succeeding waypoints of a flight. The adjustment of ETA changes the time difference of two consecutive flights passing a waypoint. For instance, accelerating the preceding flight while decelerating the succeeding one widens the time difference, which solves a conflict. Speed adjustment is also an effective and commonly used strategy for flight conflict resolution in air traffic management fields. To solve the aircraft conflict resolution problem, a geometric optimization method was developed (Bilimoria, 2000). The velocity vector strategy was used to change the heading and speed of the aircraft, and the optimized 4D trajectories, therefore, have minimum deviations from the initial ones. In a follow-up study, a subliminal speed control method was proposed to solve the CD&R problems with only minor adjustments of speed (Rey et al., 2016). That generated the minimum deviations for the initially-planned trajectories and therefore reduced the air traffic controllers' workload on handling the conflicts (Pang et al., 2021). Some research works also studied the robust conflict resolution problems with uncertainties that wind components and velocities of aircraft were randomly generated in a reasonable range (Dias et al., 2022; Dias & Rey, 2020). To overcome the uncertainty impact on the pre-planned 4D routes, one study (González-Arribas et al., 2018) proposed a dynamic adjustment method to adjust the airspeed in response to the leads or lags of flight, which increased adherence to the estimated time of arrival at each waypoint in the pre-planned route. Another work (Berdonosov et al., 2018) studied the speed approach for UAV conflict resolution, while the critical speed range for potential conflicts was considered to reduce the computational cost. To solve the potential conflicts in UAV formation flight, researchers (Geng et al., 2019; Seo et al., 2017) investigated the speed change strategy based on potential field vectors. The idea of these studies is to generate a conflict-free envelope by adjusting the flight speed of the UAV. A recent study (Dai et al., 2021) has the same mind to generate conflict-free UAV 4D routes. In that study, the airspace was divided into standardized air blocks where the airspace occupation time and rate were considered. The occupied air blocks were disabled for other routes, which prevents flight conflicts.

1.1.2. Strategy in spatial dimension: rerouting

Rerouting is another effective way to tackle the flight conflict. By adjusting the 3D geometric trajectory, potential conflicts can be resolved in the spatial dimension. Compared with the scheduling strategy, the rerouting strategy changes the flight trajectory to avoid conflicts, convective weather, and other risky areas (Lim & Zhong, 2018; Xie et al., 2017). For manned aircraft, rerouting is not a preferred conflict resolution strategy considering safety and efficiency issues. Commercial manned aircraft is only allowed to fly within the airway boundary, and lateral rerouting may deviate the trajectory out of the boundary resulting in risk issues. The altitude change is also a fuel-consuming and unsafe way, which is justified by a number of research works that aim to minimize the trajectory deviations from the initial one when using rerouting strategy (Chaimatanan et al., 2014; Courchelle et al., 2019; Hernández-Romero et al., 2020; Pelegrín & D'Ambrosio, 2022).

From an efficient point of view, minimizing the flight distance is always one of the objectives for solving both manned and unmanned conflict resolution problems using the rerouting strategy. However, it would be more challenging for UAS operations, as rerouting may generate extra flight distance and that may make the mission

infeasible for battery-powered UAVs due to its limited flight duration. To solve this problem, some authors (C. Cheng et al., 2020; Kim & Lim, 2020) proposed the energy-constraint rerouting method to ensure the UAV is able to safely return home. Come to the application of rerouting strategy, it was conducted by using both discrete and continuous methods. In discrete environments, the airspace is meshed into units and the UAV operates from one waypoint to another. Rerouting in that environment is to search an alternative route point-by-point using methods such as A* (Dai et al., 2021) and ant colony algorithm (Wu, Low, Pang, et al., 2021). The advantage of discrete rerouting is that the traffic flow can be better organized via the waypoint-based network, which is good for high-density traffic management (Pang, Dai, et al., 2020; Wu et al., 2020). On the other hand, researchers also investigated the rerouting problems using continuous approaches such as vector fields (Marchidan & Bakolas, 2020; Wilhelm & Clem, 2019). Their proposed vector field methods considered UAV kinematics and approximate speed modulation, which enabled smooth and conflict-free routes for multi-UAV operations. B-spline curves have also been used to connect each route waypoints and produced continuous conflict-free routes for several application uses (Wenchao Ding et al., 2019; Nikolos et al., 2003). Compared with the discrete method, the continuous methods perform better in terms of local route optimization and route smoothing. Instead of changing the coordinates of the route in three dimensions, some authors investigated the flight-level adjustment method for deconfliction (Tang et al., 2021). They proposed a flight-level assignment strategy to solve conflicts for pre-planned 4D routes, and the operational information was considered to determine the assigned flight level.

1.1.3. Uncertainties in flight conflict resolution

The common uncertainties considered in conflict resolution models for manned aircraft are weather uncertainties (Courchelle et al., 2019; González-Arribas et al., 2018; Hernández-Romero et al., 2020; Rodionova et al., 2016), and trajectory prediction uncertainties (Chaimatanan et al., 2014; Dias & Rey, 2020; Rey et al., 2016). The weather uncertainty models are always developed with a probability density function that is derived from the data of weather forecasts for the prevailing wind in large-scale space. While the aircraft trajectory prediction uncertainties are always presented as random variables of velocities and positions in reasonable ranges. However, in the small-scale low-altitude urban airspace, weather conditions are difficult to predict (e.g., wind shear in the landing phase for manned aircraft), which makes it difficult to develop the weather uncertainty model for UAS operations. The trajectory uncertainties for UAS operations are also preferred to be solved in the tactical phase at robotic levels (Kahn et al., 2017; Yang & Wei, 2020, 2021).

As discussed above, existing works have investigated the conflict resolution problems with various strategies considered. However, there are still several research gaps and challenges as the UAS operations introduce into urban environments with larger scale and higher requirements for conflict resolution capabilities. First, existing studies have fewer considerations on the third-party risks when UAS operates in urban environments. The third-party risk is defined as the people and property not associated with and not directly benefit from the UAS operations, such as fatality risk to people on the ground (Blom et al., 2021; Melnyk et al., 2014). These risks should be considered when conducting specific conflict resolution strategies. Because existing conflict resolution methods may not perform well in a risk environment, as some solution points may turn out to be high-risk areas and therefore invalid for planning. Second, the conflict resolution strategies in works of literature are separately modelled and used such as only conducting

scheduling or rerouting. However, some types of conflicts cannot be solved by scheduling (or speed adjustment) such as the head-to-head conflict (Pallottino et al., 2002). Rerouting also has its issue. It pays a high price for downstream impact (Wenzhe Ding et al., 2018), as replanned routes may intersect with other initial routes causing new conflicts. What is more, there are very few studies that have combined the different strategies to solve flight conflicts. A decision-making method is still lacking to adaptively select the best suitable resolution strategy for different types of flight conflicts.

This paper will focus on the strategic deconfliction for multiple UAVs in risk-based environments. An adaptive decision-making method is developed to solve the different types of flight conflicts with different strategies. To make the scope of this paper clearer, the following assumptions are made:

(1) The risk map and the pre-planned route for each UAV flight are generated by an existing study (Pang et al., 2022). The focus of this paper is on strategic deconfliction among multiple UAVs.

(2) The UAV refers to both quadrotor and fixed-wing unmanned aerial vehicles. As fixed-wing UAVs are unable to hover with the speed of zero, hovering is not made as an option in the problem formulation section.

1.2. Research contributions

This paper proposes a double-layer decision-making framework to adaptively solve flight conflicts and optimize pre-flight 4D routes. The main contributions are concluded as follows:

(1) We propose an adaptive decision-making framework for multi-UAV conflicts resolution considering scheduling, speed adjustment, and rerouting strategies with elaborations of their inner mechanisms for conflict resolution. The adaptive framework contributes to the conflict resolution for air traffic control and management of emerging UAV operations in low-altitude urban airspace.

(2) We model the adaptive decision-making as a double-layer optimization problem. The first layer is established as a probabilistic selection model, which is used to optimize the strategies for different types of flight conflicts. The second layer is developed as a mixed-integer nonlinear programming (MINLP) model to optimize the decision variables of the strategies selected by the first layer, which consequently optimizes the 4D routes.

(3) We introduce and improve a stochastic fractal search (SFS) algorithm to solve the developed double-layer optimization problem with two main improvements of a penalty-guided fitness function and an exploitation-exploration balancing scheme.

The rest of the paper is structured as follows. Section 2 presents the modelling of the risk-based 4D route and flight conflict. Section 3 formulates the conflict resolution problem with a double-layer decision-making framework. Section 4 develops the solution approach using the base SFS algorithm with two improvements. Simulations and discussions are conducted in Section 5 and the main research findings are concluded in Section 6.

2. Modelling of the risk-based 4D route and flight conflict

In this section, the concept of the risk-based 4D route optimization problem is introduced. The conflict detection based on the 4D routes is defined and the decision variables for solving the detected conflicts are also discussed.

The risk-based 4D route is a sequence of waypoints in discrete space with 3D coordinates and time, which considers the operational risks to third parties (Blom et al., 2021; Pang, Tan, et al., 2020). The 3D coordinates include (x, y, z), while the two primary temporal information are estimated time of departure (ETD) and estimated time of arrival (ETA). The ETD is only influenced by the scheduled departure time, while the ETA can be influenced by both the ETD and flight speed. For instance, rescheduling the ETD will subsequently change the ETA, given the same flight speed. While the ETA can also be influenced if the speed is changed for a flight route, given the same ETD.

We define the 4D route on a directed graph $G = (W, A)$, where $W = \{0,1,2, \dots, w+1\}$ is a set of waypoints. Note that waypoint 0 represents the origin location while waypoint $w+1$ presents the destination. Here $A = \{A_{ij}(w_i, w_j), \forall i, j \in N_{\text{node}}\}$ is the set of arcs that connect two adjacent waypoints, where $w_i = (x_i, y_i, z_i)$ and $w_j = (x_j, y_j, z_j)$. Note that N_{node} is the total number of nodes in the graph. We further define crossing waypoint (CWP) as $W_{\text{CP}} = \{\text{CWP}_1, \text{CWP}_2, \dots, \text{CWP}_k\}, \forall \text{CWP}_k \in W$. The CWP is defined as at least two arcs intersect with each other.

Let $T = \{t_i : \forall t_i \in [0, \infty), \forall i \in N_{\text{node}}\}$ be the set of ETAs. Note that t_i is the ETA of UAV passing waypoint w_i. Let $U = (\{U^1, U^2, \dots, U^a, \dots, U^{N_{\text{uav}}}\}, a \in N_{\text{uav}})$ be the set of UAVs and $R_{4\text{D}} = \{R_{4D}^1, R_{4D}^2, \dots, R_{4D}^a, \dots, R_{4D}^{N_{\text{uav}}}\}$ be the set of 4D routes. Here N_{uav} is the total number of UAVs. A 4D route of U^a is denoted as $R_{4D}^a = (\{W_i^a\}, i \in N_{\text{node}})$, where $W_i^a = (\{\boldsymbol{X}_i^a, t_i^a\}, \boldsymbol{X}_i^a \in \mathbb{R}^3)$.

As shown in Fig. 1, four 4D routes corresponding to the respective UAVs are illustrated with different origins and destinations (ODs). An example of a 4D route of UAV U^a is denoted as $\{w_1, w_2, \text{CWP}_1, w_3, \text{CWP}_3, w_4, \text{CWP}_5, w_5\}$, which has three crossing waypoints $(\text{CWP}_1, \text{CWP}_3, \text{CWP}_5)$.

Based on the definition of the 4D route, flight conflict is defined as two or more UAVs arriving at the same waypoint while the difference of their ETAs is smaller than the minimum safe separation. Formally, a flight conflict is detected if the following condition holds

$$\begin{cases} R_{3D}^1 \cap R_{3D}^2, \dots, \cap R_{3D}^a, \dots, \cap R_{3D}^{N_{\text{uav}}} = \text{CWP}_u \ (a \geq 2, u \geq 1, \forall a, u \in N_{\text{uav}}) \\ \left| t_{\text{CWP}_u}^a - t_{\text{CWP}_u}^b \right| < \Delta t_{\text{conflict}} \ (a, b \in N_{\text{uav}}, a \neq b) \end{cases} \tag{1}$$

where R_{3D}^a is the set of waypoints for the three-dimensional route U^a, while CWP_u is the uth crossing waypoint. Note that $t_{\text{CWP}_u}^a$ and $t_{\text{CWP}_u}^b$ are ETAs of U^a and U^b at CWP_u, respectively. Here, $\Delta t_{\text{conflict}}$ is the separation minima.

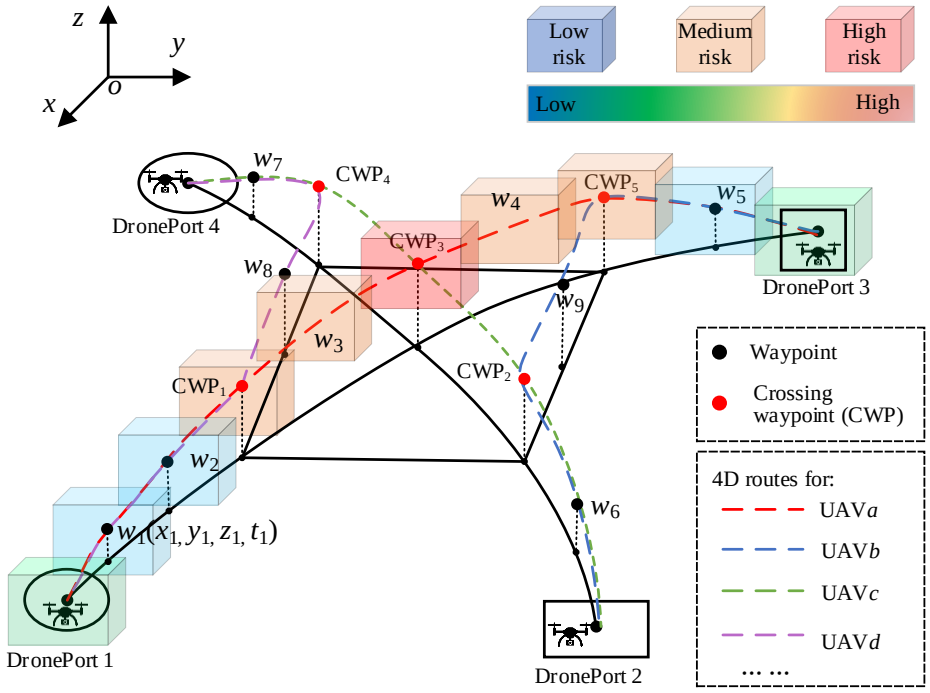


Fig. 1. Illustration of multi-UAV 4D routes in the risk-based airspace

To solve the flight conflict, this paper proposes an adaptive decision-making method for conflict resolution to optimize pre-planned 4D routes. The known information is initial 4D routes including (x, y, z, t) at each waypoint. Our goal is to optimize the 4D routes with the objectives of minimizing the number of flight conflicts, third-party risk, and operational cost. The optimization variables of the 4D route include spatial variables and temporal variables. The spatial variables are 3D coordinates (x, y, z) and they are discretized by position. The temporal variables include ETD and ETA. The resolution strategies are conducted based on these variables to solve flight conflicts.

3. Problem formulation for conflict resolution and 4D routes optimization

The strategies for conflict resolution are formulated and a decision-making framework is proposed for 4D routes optimization. Two groups of resolution strategies are developed from temporal and spatial dimensions. The inner mechanisms of the strategies are discussed with considerations of different flight conflict types. In the decision-making framework, the strategy allocation and 4D routes optimization problems are formulated.

3.1. Strategies for conflict resolution

The strategies for flight conflict resolution can be classified into two aspects from temporal and spatial dimensions. From a temporal aspect, the strategies are to change the ETD and ETA. The ETD can be directly adjusted by changing the departure time of the corresponding UAV. The ETA of each waypoint has two ways to adjust. One is to change the ETD, which will change the global ETA for every single waypoint. The other is to adjust the flight speed (accelerate or decelerate) of UAV in each route segment, and the ETA of all succeeding waypoints will be altered. On

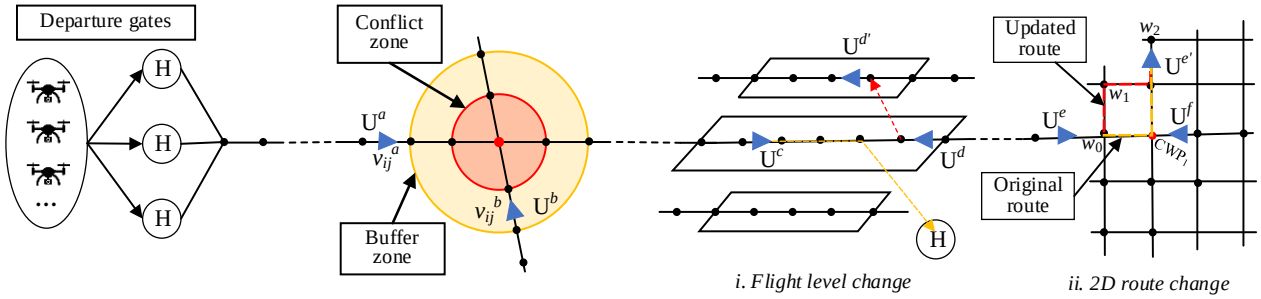
the other hand, the spatial strategy is rerouting, which is to adjust the 3D coordinates of the route. The illustration of scheduling, speed adjustment, and rerouting strategies are presented in Fig. 2.

Scheduling is to adjust the estimated time of departure illustrated in Fig. 2(a). The ETD for a particular U^a can be presented as t^a_{Dep}. Here, $T_{\mathrm{Dep}} = \{t^a_{\mathrm{Dep}}, a \in N_{\mathrm{uav}}\}$ is the set of pre-planned departure times for all UAVs. Due to the capacity limitations of airspace, the preferred departure time proposed by UAV users may not be approved. To obtain the feasible or even optimal departure time for all missions, scheduling is required. By adjusting departure time, the ETA for each waypoint is changed accordingly, and that helps to globally resolve flight conflicts and congestions.

Speed adjustment is to solve flight conflict by increasing (decreasing) the speed, and the ETA is brought forward (backward) to avoid flight conflict from the temporal dimension. Let $V^a = \{v^a_{ij}, i, j \in N_{\mathrm{node}}\}$ be the set of speed for the 4D route of U^a, and the UAV speed at the arc A^a_{ij} is presented as v^a_{ij}. As shown in Fig. 2(b), U^a has speed of v^a_{ij}, and U^b has speed of v^b_{ij}, assuming $v^a_{ij} = v^b_{ij}$. The U^a and U^b are at same flight altitude and approaching the same waypoint. There is a chance that the two UAVs will have conflict at the crossing waypoint. In this scenario, the speed adjustment can be used to solve the conflict. As U^b has entered the buffer zone while U^a is outside the zone, the strategy is to increase the v^b_{ij} and decrease the v^a_{ij}, and subsequently the t^b_j at w_j is brought forward and t^a_j is brought backward. Thus, the time difference of the two UAVs is widened, and U^b can pass w_j and exit conflict zone before U^a enters.

Rerouting is another strategy for conflict resolution, and there are two ways to do so. One is only to change the flight altitude z_i and the other is to adjust the 2D coordinates (x_i, y_i) of waypoint w_i. Flight altitude change is one of the most effective strategies to solve conflict for manned and unmanned aircraft (Challita et al., 2019). This strategy considers the remaining flight range of the flight to determine which one is climbing up or descending. For instance, a UAV approaching its destination will soon descend to land, and climbing is not considered unless there have no other solutions. As shown in Fig. 2(c), the 4D routes of U^c and U^d have one segment in common with the same flight altitude while they are operating in an opposite direction. In this case, scheduling and speed strategies fail to work as they can only postpone the flight conflict but cannot solve it. In that case, the rerouting is the effective way, and the flight altitude of U^d is changed to a higher flight level, as U^c is going to land.

Another way of rerouting is to change the 2D coordinates of the route. In certain situations, the change of 2D coordinates solves flight conflict in a more energy-efficient way as it diverts UAVs in planar space without the need for vertical movement. For instance, two UAVs operating at the same altitude have a flight conflict in a crossing waypoint. As there are a few more alternative waypoints that can be chosen to fly, the quickest strategy, in this case, is to divert one of the UAVs to avoid conflict. The principle of diverting which UAV is relied on their remaining routes and destinations to maximize the smooth transiting and minimize the impact on the current traffic network. One example is given in Fig. 2(c). The UAVs U^e and U^f have conflict in CWP_1. The U^e is heading to w_2 with original route of passing w_0, CWP_1 and w_2, while U^f is moving straightly forward passing CWP_1 and w_0 in sequence. In this case, we divert the U^e from original route (yellow line) to the updated one (red line), which solves the conflict and does not increase the distance-related cost.



(a) Scheduling (t) (b) Speed adjustment (v) (c) Rerouting (x, y, z)

Fig. 2. Decision-making strategies for UAV conflict resolutions

As discussed above, different strategies have their unique advantages and weaknesses. Temporal strategies (scheduling and speed adjustment) solve flight conflict and congestion in a global view, but they are unable to solve certain types of conflicts such as head-to-head ones. Spatial strategy (rerouting) is capable of solving conflicts more locally with immediate resolution effects, but it always has a global impact on the overall flight plans. That is because the change of certain routes may influence other initial ones causing new conflicts. In summary, one strategy cannot fit all flight conflict situations. Different types of conflicts should be solved by different strategies that suit the conflicts. However, when it comes to several resolution strategies with their unique pros and cons, it is challenging to choose the best suitable one for a specific type of conflict. To meet this challenge, we propose a double-layer optimization framework, presented in the following sections.

3.2. A decision-making framework for 4D routes optimization

In this section, a double-layer decision-making framework is proposed to optimize UAV 4D routes and the overall workflow is presented in Fig. 3. The first layer of the framework is constructed as an adaptive decision-making method, which is used to optimize the strategy assigned for different types of flight conflicts. The output of the first layer is the specific resolution strategies for different types of flight conflicts. The second layer of the framework is developed as a mixed-integer nonlinear programming (MINLP) model to optimize the decision variables of the strategies selected from the first layer. While an improved stochastic fractal search algorithm is used to solve the developed optimization problem. The final optimization output is the conflict-free 4D routes with a minimized number of flight conflicts, operational costs, and delays.

3.2.1. First layer: An adaptive method for strategy allocation

The adaptive decision-making (ADM) method is to determine which strategy should be assigned to solve what type of flight conflict. The initial multiple 4D routes are checked to detect flight conflicts, which are, if any, processed to find the best suitable resolution strategies. Let $C = \{c_1, c_2, \dots, c_i\}$ be the set of flight conflicts. Let $S = \{s_1, s_2, s_3\}$ be the set of strategies for conflict resolution. If flight conflicts are detected, the ADM method is used to select the suitable resolution strategy from S. Note that N_{fc} is the total number of conflicts detected. Here n_{s1} is the number of the scheduling strategy being allocated for conflict resolution, while n_{s2} and n_{s3} are the number of speed adjustment and rerouting strategies being used. The ith conflict is denoted as $c_i, i \in N_{\text{fc}}$, and each conflict is allocated with a resolution strategy.

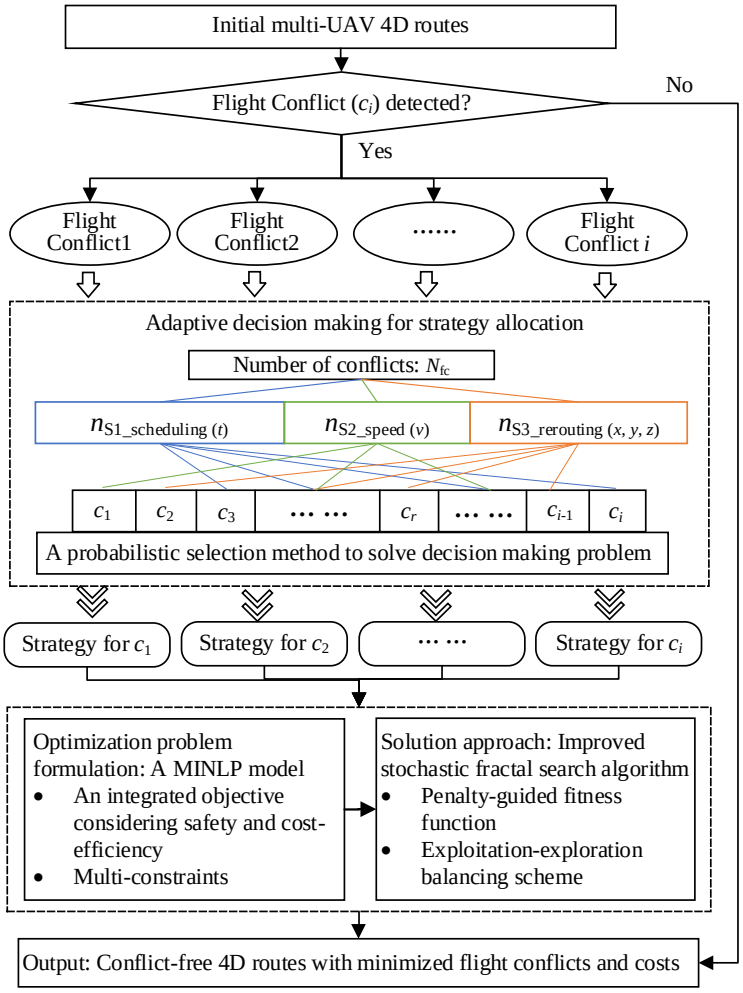


Fig. 3. Overall workflow of the decision-making framework

We further define the objective function of optimizing ith flight conflict is $f_i(\mathbf{s}_i)$, where $\mathbf{s}_i$ is a feasible solution in searching space of set S. Note that $\mathbf{s}_i = g_i(\mathbf{x}_i, t_i, v_i)$, where $\mathbf{x}_i, t_i, v_i$ are decision variables for resolution strategies. So, the objective function for conflict resolution can be denoted as $f_i\big(g_i(\mathbf{x}_i, t_i, v_i)\big)$. The objective of the ADM method is to solve $f_1, f_2, \dots, f_q$ to find better strategies $\mathbf{s}_i$. Instead of solving the flight conflicts one after another, the ADM process all flight conflicts concurrently to obtain the best possible strategies for each type of conflict, and to generate the globally optimized multiple 4D routes.

Instead of using a combination of strategies for each UAV, a single strategy will be used for a particular UAV. For one thing, any changes on the initial flight plan (departure time, flight speed, and 3D route) will have negative impacts on either user's side or traffic management side, which should be minimized (Bilimoria, 2000; Dias & Rey, 2020; Rey et al., 2016). For another, all conflicts can be solved by a single strategy if the suitable one can be found. The ADM method is proposed to find the suitable strategy for every single UAV.

3.2.2. Second layer: A MINLP model for 4D routes optimization

To optimize the decision variables $(\mathbf{x_i}, t_i, v_i)$ of the obtained strategies $\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_i$, the mathematical formulation of the second layer is developed. Parameters, decision variables, objective function, and constraints used in the formulation are given as follows:

Sets

$W = \{0,1,2, \dots, w+1\}$ is the set of waypoints.

$W_{\mathrm{CP}} = \{\mathrm{CWP}_1, \mathrm{CWP}_2, \dots, \mathrm{CWP}_k\}, \forall \mathrm{CWP}_k \in W$, is the set of crossing waypoints.

$A = \{A_{ij}(w_i, w_j), \forall i, j \in N_{\mathrm{node}}\}$ is the set of arcs.

$U = \{U^1, U^2, \dots, U^a, \dots, U^{N_{\mathrm{uav}}}\}$ is the set of UAVs, and $a \in N_{\mathrm{uav}}$.

$R_{4\mathrm{D}} = \{R_{4D}^1, R_{4D}^2, \dots, R_{4D}^a, \dots, R_{4D}^{N_{\mathrm{uav}}}\}$ is the set of 4D routes.

$V^a = \{v_{ij}^a\}$ is the set of speed for the 4D route of U^a.

$T_{\mathrm{Dep}} = \{t_{\mathrm{Dep}}\}$ is the set of pre-planned departure times for all UAVs.

$T_0 = \{t_0\}$ is the set of actual departure times for all UAVs.

$T_i^a = \{t_i^a : \forall t_i^a \in [0, \infty), \forall i \in N_{\mathrm{node}}, a \in N_{\mathrm{uav}}\}$ is the set of ETAs of U^a passing waypoint w_i.

Parameters

$w_i(x_i, y_i, z_i)$: coordinates of waypoint i, $\forall i \in N_{\mathrm{node}}$.

$A_{ij}(w_i, w_j)$: arc connects two adjacent waypoints w_i and w_j.

r_i: risk index value of waypoint w_i.

U^a: ath UAV.

CWP_k: kth crossing waypoint. Here $k \in K$, and K is the number of crossing waypoints.

v_{ij}^a: speed of ath UAV at the arc A_{ij}.

t_{Dep}^a: pre-planned departure time of U^a.

t_0^a: actual departure time of U^a.

$t_{w_i}^a$: ETA of U^a passing waypoint w_i.

R_{4D}^a: 4D route of U^a. $R_{4D}^a = (\{W_i^a\}, i \in N_{\mathrm{node}}, a \in N_{\mathrm{uav}})$.

Decision variables

$w_i^a \in \{0,1\}$: equals to 1 if U^a passes waypoint w_i; equals to 0 otherwise.

v_{ij}^a: the speed of U^a passes arc A_{ij}, which is taken as the integer variable with unit of km. Here $v_{\min}^a \le v_{ij}^a \le v_{\max}^a$.

t_0^a: the actual departure time of U^a, which is taken as the integer variable with the unit of second.

Objective function (MINLP)

The objective of the proposed model is to minimize the third-part risks, flight delays, and airborne time-related costs. The overall objective function is denoted as

$$\min: f_{\mathrm{obj}} = \omega_{\mathrm{risk}} R_{\mathrm{TP}} + \omega_{\mathrm{t\_delay}} T_{\mathrm{delay}} + \omega_{\mathrm{t\_air}} T_{\mathrm{air}} \quad (2)$$

where ω_{risk}, $\omega_{\mathrm{t\_delay}}$, and $\omega_{\mathrm{t\_air}}$ are weightage factors for risk, delay, and airborne time, respectively. Note that $\{\omega_{\mathrm{risk}}, \omega_{\mathrm{t\_delay}}, \omega_{\mathrm{t\_air}}\} \in [0,1]$, and $\omega_{\mathrm{risk}} + \omega_{\mathrm{t\_delay}} + \omega_{\mathrm{t\_air}} = 100\%$.

Safety is the top priority for UAV operations in urban environments. The first sub-objective of the model is to minimize the third-party risks caused by UAV, presented as

$$R_{\mathrm{TP}} = \sum_{a=1}^{N_{\mathrm{uav}}} \sum_{i \in \mathrm{W_p}} r_i^a w_i^a \tag{3}$$

where R_{TP} is the total risk of UAV routes, and r_i^a is the risk value pertaining to waypoint w_i^a of U^a. Note that $\mathrm{W_p}$ is the set of waypoints that are passed by UAV.

The second sub-objective of the model is to minimize the overall flight delays

$$T_{\mathrm{delay}} = \sum_{a=1}^{N_{\mathrm{uav}}} \left| t_0^a - t_{\mathrm{Dep}}^a \right| \tag{4}$$

where T_{delay} is the total delay of all UAVs. Here t_0^a is actual departure time and t_{Dep}^a is the pre-planned departure time of UAV U^a. Note that the T_{delay} is a nonlinear function, as it is the summation of the absolute value of time difference between actual departure and pre-planned departure,

Compared with flight distance, airborne time is more crucial for UAV operations in terms of operational safety (limited battery life) and efficiency (time-related flight costs). The third sub-objective is to minimize the total airborne time, denoted as

$$T_{\mathrm{air}} = \sum_{a=1}^{N_{\mathrm{uav}}} \sum_{w_i,\, w_j \in \mathrm{W_r}} \left(\frac{d_{ij}}{v_{ij}^a} \right) = \sum_{a=1}^{N_{\mathrm{uav}}} \sum_{w_i,\, w_j \in \mathrm{W_r}} \left(\sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2} / v_{ij}^a \right) \tag{5}$$

in which T_{air} is the total airborne time consumed by all UAVs, and the total number of UAVs is denoted as N_{uav}. Here, w_i and w_j are neighboring waypoints, which belong to a set of waypoints $\mathrm{W_r}$ for a route. The neighboring correlation is defined by the absolute difference of their index i and j, denoted as: $|i - j| = 1$. Note that d_{ij} is the Euclidean distance of the arc $A_{ij}(w_i,\, w_j)$, and v_{ij}^a is the speed of U^a in A_{ij}. The function of T_{air} is nonlinear, as the decision variable v_{ij}^a is the divisor in the function. Both Eq. (4) and Eq. (5) are both nonlinear functions, therefore the optimization problem is developed as an MINLP model.

Constraints

$$\begin{cases} x_{i+1} = x_i + \xi_x^i, x_i \in \{0, l, 2l, \ldots, X_{\max}\} \\ y_{i+1} = y_i + \xi_y^i, y_i \in \{0, l, 2l, \ldots, Y_{\max}\}, \forall\, \xi_x^i, \xi_y^i, \xi_z^i \in \{-l, 0, l\}, \forall i \in N_{\mathrm{node}} \\ z_{i+1} = z_i + \xi_z^i, z_i \in \{0, l, 2l, \ldots, Z_{\max}\} \end{cases} \tag{6}$$

$$\left|\xi_x^i\right| + \left|\xi_y^i\right| + \left|\xi_z^i\right| \neq 0, \forall\, \xi_x^i, \xi_y^i, \xi_z^i \in \{-l, 0, l\} \tag{7}$$

$$\left| \left(t_0^a + T_{\mathrm{air}}^a\right) - \left(t_{\mathrm{Dep}}^a + \mathrm{T_{air\_planned}}\right) \right| \leq T_{\mathrm{delay}}^a, \forall a \in N_{\mathrm{uav}} \tag{8}$$

$$\sum_{i,j \in N_{\mathrm{p}}} \left(\frac{d_{ij}}{v_{ij}^a} \right) \leq T_{\mathrm{BTRY}}^a, \forall a \in N_{\mathrm{uav}} \tag{9}$$

$$\sum_{a \in N_{\text{uav}}} U_{\text{O}}^{a} = \sum_{b \in N_{\text{uav}}} U_{\text{D}}^{b} \tag{10}$$

$$v_{\min}^{a} \le v_{ij}^{a} \le v_{\max}^{a}, \forall a \in N_{\text{uav}} \tag{11}$$

$$0 < t_{\text{Dep}}^{a} \le T_{\text{slot}}, t_{ij}^{a}, t_{0}^{a} > 0, \forall i, j \in N_{\text{node}}, \forall a \in N_{\text{uav}} \tag{12}$$

Constraint (6) ensures the consistency of flight motion in discrete space, and 26 candidate waypoints can be chosen as the next waypoint, including planar diagonal and cubical diagonal. Constraint (7) restricts that hovering is not allowed during flight, as the UAV refers to both the quadrotor and the fixed-wing unmanned aerial vehicles in this paper. Constraint (8) restricts the arrival delay of the flight. The flight arrival time to a destination should not exceed the threshold T_{delay}^{a}, otherwise the corresponding flight will be regarded as a delayed flight. Where t_{0}^{a} is the actual departure time and T_{air}^{a} is the actual airborne time, while t_{Dep}^{a} is pre-planned departure and $T_{\text{air\_planned}}$ is the pre-planned airborne time. Constraint (9) ensures the flight time of the planned route does not exceed the battery life of the UAV. Constraint (10) ensures that traffic flow from origins to destinations is balanced. If a flight cannot reach its destination, that flight will be rejected. Constraint (11) ensures the assigned flight speed is within the range of UAV speed capability. Constraint (12) ensures the assigned departure time for each flight is within the given departure time window.

4. Solution approach using the stochastic fractal search algorithm

In Section 3, the adaptive conflict resolution problem is developed as an MINLP optimization model. The model has several decision variables that act under various constraints. The bounds of the constraints include both simple forms such as the range of flight speed (Constraint (11)) and nonlinear relationships such as the flight duration (Constraint (9)). The MINLP has been proven as a Non-Deterministic Polynomial (NP-hard) problem (Burer & Letchford, 2012) with the complexity of O(n!) (Arora, 2001). With the increase of the problem size n, the search space for the high-dimensional optimization problem increases dramatically. The solution quality and computational efficiency for such type of problem become more unacceptable if using classical optimization methods such as exhaustive methods (Courchelle et al., 2019). Furthermore, as the conflict resolution problem is highly combinatorial, the classical methods are only applicable to solve the problem with a small number of aircraft (Courchelle et al., 2019; Durand & Alliot, 2009). That motivates the use of metaheuristic algorithms.

Metaheuristic algorithms, as approximate methods, are not sensitive to the size of the search space (Amr et al., 2013; Loubière et al., 2018) and have become the primary alternative to solve large-scale global optimization problems. The stochastic fractal search (SFS) algorithm is one of the novel and powerful metaheuristic algorithms for solving both discreet and continuous optimization problems (Salimi, 2015). The SFS is a stochastic algorithm, and it performs well in global optimization problems. That makes it a suitable algorithm to solve the proposed constrained CD&R optimization problem with objectives to globally minimize the total number of flight conflicts and total costs. Besides, the SFS algorithm has been proven the better performance in terms of solution quality and convergence time in various engineering and optimization problems (Aras et al., 2021; Mellal & Zio, 2016; Salimi, 2015), which enables the strategic deconfliction to be completed in a shorter time with a better solution. That subsequently improves the UAV

traffic performance and accelerates the process for strategic flight planning. The flexibility of the base SFS algorithm also allows it to be extended and implemented in solving the proposed MINLP problem in this paper.

The SFS algorithm has two key processes, diffusion process and update process, to provide exploitation and exploration capabilities in solving optimization problems. The fundamental concept of SFS is to mimic the diffusion property in fractals, which can be widely seen in the self-similarity of patterns in nature like trees, crystals, snowflakes, etc. Based on the diffusion property, each particle (point or individual solution) tries to simulate the branching property of a dielectric breakdown and propagate from the central point. This property allows the central point creates a series of new particles adjacent to each other with the capability of moving in different directions with various distances. With the diffusion property, the initial energy at a central point can be effectively exploited to other objects that have reactions to this energy. The exploration of solutions is performed in the selection and update process based on the fitness function.

In this paper, we improve the SFS algorithm from two aspects. First, to minimize the number of flight conflicts, we propose a conflict penalty-guided fitness function to evaluate the solution candidates. Solutions with more flight conflicts will have a higher probability of being discarded in the next optimization process. After optimization, the obtained solution achieves conflict-free routes with a minimum number of flight delays and total cost. Second, we introduce an exploitation and exploration balancing scheme to improve the effectiveness of conflict resolution methods in a risk-based environment. The scheme improves the searching diversity of the SFS algorithm by considering the fitness value and solution point distance to the best point. That prevents new solution points from being too close to the existing best point, which avoids the solutions being trapped in local minima. With the improved SFS, we develop the adaptive decision-making method to allocate the best suitable strategies for different types of flight conflicts.

4.1. Penalty-guided fitness function in exploitation and exploration processes

A. Exploitation and exploration processes

In the SFS algorithm, the diffusion process performs exploitation tasks while the update process provides exploration capabilities. The diffusion process is to search new solutions by creating particles adjacent to the central point. The position of new particles in the search space is distributed based on Gaussian random walks, denoted as

$$P_i^{\eta} = P_i + \text{Gaussian}(\mu_P, \sigma) = P_i + \frac{1}{\sqrt{2\pi}\sigma} e^{-(\mu_P-\sigma)^2/2\sigma^2} \tag{13}$$

$$P_i^{\eta} = P_i + \text{Gaussian}(\mu_P, \sigma) + (\varepsilon P_{\text{best}} - \varepsilon' P_i) \tag{14}$$

where P_i^{η} is the ηth new particle created by central particle P_i, and η is the number of new particles generated. Here μ_P is the mean of Gaussian distribution of new particle positions, which is equal to the absolute value of ith solution particle denoted as $|P_i|$. Here, P_{best} is the particle with the best solution in the current group. Note that ε and ε' are weightage factors to adjust the degrees of exploitation and exploration. They are random values following the uniform distribution and are in the range of [0,1]. Here, σ is the standard deviation, computed by

$$\sigma = \left| \frac{\log(N_{\text{Iter}})}{N_{\text{Iter}}} P_i - P_{\text{best}} \right| \tag{15}$$

Note that, N_{Iter} is the number of iterations.

After the diffusion process, the particles are evaluated by the fitness function. The update process is to select particles that have high fitness value while ensuring the high diversity of solution particles. The first update process is to select the particles with good fitness values. The jth diffused seed particle of P_i will be updated, only if the following condition holds

$$\begin{cases} P_{a_i} < \varepsilon, \varepsilon \in [0,1] \\ P_{a_i} = \dfrac{rank(P_i)}{n_{\text{p}}} \end{cases} \tag{16}$$

in which P_{a_i} is the probability of a particle being updated, and n_{p} is the number of particles. Particles with good solutions are ranked higher in the diffusion group and are given a smaller value of $rank(P_i)$, so that these particles are secured and exploited. On the contrary, particles with bad solutions will be updated to search for new positions.

The first update process function is denoted as

$$P_i'(j) = P_r(j) - \varepsilon\big(P_t(j) - P_i(j)\big) \tag{17}$$

where $P_i'(j)$ is the updated position of the ith particle, and j is the dimension of current points. Here, P_r and P_t are other particles randomly selected from the current population.

The second update process is to ensure high diversity of solutions by considering the positions of other solution particles in the whole population, formulated as

$$P_i'' = P_i' - \varepsilon'(P_t' - P_{\text{best}}), \varepsilon' \leq 0.5 \tag{18}$$

$$P_i'' = P_i' + \varepsilon'(P_t' - P_r'), \varepsilon' > 0.5 \tag{19}$$

Three reference points are employed in the update process, and they are the position of the current particle P_i', randomly selected particle P_t' (or P_r') and the position with the best particle P_{best}.

B. Penalty-guided fitness function

Once obtained the initial population of particles, we evaluate the fitness value and select the best individual solution. In this paper, the objective function is to minimize the total operational cost that includes third-party risks, flight delay, and airborne time (as denoted in Eq. (2)). What is more, flight conflict is the most important risk factor, and it should be minimized in the strategic deconfliction phase. As the flight conflict may dominate the objective function, it is considered as a significant penalty in the fitness function of the algorithm. Solution candidates who have flight conflicts will be significantly penalized with an extra value, and the overall fitness function is denoted as

$$f_{\text{value}} = f_{\text{obj}}(w,t,v) + \psi(w,t,v) \tag{20}$$

where $\psi(w,t,v)$ is the penalty function. The total conflict penalty is correlated with the number of conflicts, which can be computed by

$$\psi(w,t,v) = \sum_{i=1}^{N_{\text{fc}}} \omega_{\text{fc}} f_{\text{obj}}(w,t,v)_i \tag{21}$$

in which ω_{fc} is weightage factor of flight conflict and N_{fc} is the total number of flight conflicts in a solution.

Assuming an optimization problem has a solution with dimension N, each individual particle is considered as a potential solution, and it is an N-dimensional vector. The initialization of particles is randomly conducted within the constraints of the problem, which is denoted as

$$P_i = B_{\text{lower}} + \gamma\left(B_{\text{upper}} - B_{\text{lower}}\right) \tag{22}$$

where P_i is the ith root particle in the population. Here B_{lower} and B_{upper} are the lower bound and upper bound of the constraint vectors. Note that γ is a random number and $\gamma \in [0,1]$. When $\gamma = 0$, the ith root particle $P_i = B_{\text{lower}}$, while $P_i = B_{\text{upper}}$ when $\gamma = 1$.

4.2. Exploitation-exploration balancing scheme

Exploitation and exploration are two key requirements for optimization algorithms, as the exploitation is to utilize the current best solution while the exploration is to search for better solutions (K. C. Tan et al., 2009). A good trade-off between them yields a high quality of solution while maintaining a reasonable computational time (Cuevas et al., 2014). In the SFS algorithm, the update processes do not consider the distance between the solution candidate to the best particle in the population. That too many particles close to the best one leads to an ineffective diversity of exploration.

To solve this problem in meta-heuristic search algorithms, researchers proposed a fitness-distance balance (FDB) method (Kahraman et al., 2020). The FDB method proposed a score value to evaluate each particle in selection and update processes. The score value incorporates the fitness value of each particle (computed by Eq. (20)) and the distance value between the solution particle and the best particle. That ensures particles with high fitness values are selected while avoiding selecting the particles that are too close to the best particle.

The score values of each particle in the population are computed, and the score vector S_P is presented as

$$S_P = [s_1 \quad s_2 \quad \cdots \quad s_n]^T \tag{23}$$

where s_1 is the score value of P_1, and n is the number of particles in the population.

To calculate the distance values between solution particles and the best particle, we denote the position of the best particle in the population is X_{best}. The distance value for each solution particle is computed by

$$d_{P_i} = \sqrt{\left(x_{1P_i} - x_{1P_{\text{best}}}\right) + \left(x_{2P_i} - x_{2P_{\text{best}}}\right) + \cdots + \left(x_{\zeta P_i} - x_{\zeta P_{\text{best}}}\right)} \tag{24}$$

where d_{P_i} is the distance between the ith solution position $(x_{1P_i}, x_{2P_i}, \ldots, x_{\zeta P_i})$ of P_i ($P_i \neq P_{\text{best}}$) and the best solution position $(x_{1P_{\text{best}}}, x_{2P_{\text{best}}}, \ldots, x_{\zeta P_{\text{best}}})$ of P_{best}. Here, ζ is the dimension space of the particles ($\forall_{i=1}^{n}, P_i \in \mathbb{R}^{\zeta}$).

Accordingly, we have the distance vector D_P, presented as

$$D_P = [d_1 \quad d_2 \quad \cdots \quad d_n]^T \tag{25}$$

The fitness value is computed by Eq. (20), and the fitness vector F_P is expressed as

$$F_P = [f_1 \quad f_2 \quad \cdots \quad f_n]^T \tag{26}$$

To prevent the dominance of fitness value or distance value in the score function, normalizations are conducted. Specifically, the distance vector and the fitness vector are divided by the maximum value among their vectors, respectively. The normalized distance vector is denoted as $D_{P\_\text{norm}} = D_P / d_{\max}$ ($d_{\max} = \max\{d_1, d_2, \ldots, d_n\}$ and $d_{\max} \neq 0$). While the normalized fitness vector can be denoted as $F_{P\_\text{norm}} = F_P / f_{\max}$ ($f_{\max} = \max\{f_1, f_2, \ldots, f_n\}$ and

$f_{\max} \neq 0$). After the normalizations, all distance values and fitness values fall within the range of [0, 1]. The obtained score vector S_P is then denoted as

$$S_P = \omega_{\text{FDB}} F_{P\_\text{norm}} + (1 - \omega_{\text{FDB}}) D_{P\_\text{norm}} \tag{27}$$

where ω_{FDB} is a parameter to adjust the weightage of fitness value and distance value when conducting selection and update process in FDB method, which is in the range of [0, 1]. When $\omega_{\text{FDB}} = 1$, only the fitness value will involve and the solution searching process is conducted in a greedy manner, meaning that the exploitation is effective. In contrast, when $\omega_{\text{FDB}} = 0$, only the distance value will be counted, which provides a good variation of population and great exploration capability.

The pseudo-code of the ISFS algorithm is presented in Algorithm 1. The input includes all initial 4D routes information and the RiskCost dataset, which is generated by computing the risk index in pertaining environments (Pang et al., 2022). The ISFS algorithm has two key processes. One is the diffusion process (Line 5 to Line 8), which is to exploit the first obtained best solution by searching its neighbors. The other is the update process (Line 9 to Line 16). The first update process is to randomly explore the solutions in the population, while the second update process is to ensure the diversity of the searching process by considering other solution points in the population.

Throughout the whole process, the fitness value is computed by the improved penalty-guided fitness function. That guides the solution to a minimum number of flight conflicts. On the other hand, the improved fitness-distance balancing strategies are used in the update process (Line 11 and Line 14) to avoid the explored points that are too close to the current best point, which improves the diversity of ISFS and avoids the searching being trapped in local minima. The ISFS algorithm is an evolutionary metaheuristic algorithm. As shown in Algorithm 1, the number of executions for each script in the main loop of the ISFS is n. That gives the complexity of ISFS is O(n).

4.3. ISFS-based ADM method for strategy allocation

Based on the ADM framework proposed in Section 3.2 and the ISFS algorithm, we develop the ADM method using a probabilistic scheme to optimize the strategies allocated for conflict resolution. The pseudo-code of the ADM method is presented in Algorithm 2. The core of this method is to sample explicit probabilistic models of the promising solutions, which guides the search for the optimum. The input of the ADM method is the lower bound and upper bound of decision variables. The initialization is to generate a probability matrix for strategy allocation. The main optimization loop (Line 3 to Line 19) is an incremental process of the probability that a strategy with a better fitness value will be chosen for conflicts. In each iteration, the dominant population with a better fitness value is selected and the corresponding probability is updated by the formula shown in Line 18. The learning rate l_{rate} is introduced to adjust the trade-off between the exploitation of the last-step probability matrix and the exploration for the next probability increment. Here the l_{rate} is taken as 0.5 to balance the exploitation and exploration. For the ADM algorithm, the complexity of the outer loop is O(n) while the ISFS algorithm is called in the inner loop. That makes the complexity of the overall ADM algorithm being O(n^2).

Algorithm 1: Improved Stochastic Fractal Search (ISFS) Algorithm

Input: Initial 4D routes *(x, y, z, t),* RiskCost Dataset
Initialization: generate a population (P) of n solution points with N-dimension
Selection: obtain the first best solution by evaluating the fitness value
while $i <= n_{gen}$ **Do** % Optimization loop, n_{gen} is the number of generations
    begin Diffusion process (Exploitation)
        Searching neighbors of the first best solution point (**Eq. (13)** and **Eq. (14)**)
        Update the fitness value and the best solution point
    end
    begin Update process (Exploration)
        First update process
        Update the position of solution based on two random points in P (**Eq. (17)**) and fitness-distance balancing (**Eq. (27)**)
        Update the fitness value and the best solution point
        Second update process
        Update the position of the solution based on other solution points in P (**Eq. (18)** and **Eq. (19)**) and the fitness-distance balancing (**Eq. (27)**)
        Update the fitness value and the best solution point
    end
end while
Return (*BestSolution, fitnessValue*)

Algorithm 2: Adaptive Decision Making (ADM) Algorithm for Strategy Allocation

Input: lower bounds and upper bounds of (x, y, z, t, v)
Initialization: probability update function P_{update} of strategy allocation
while $i <= n_{gen}$ **Do** % main loop
    Generate species: Randomly allocate the strategies (Sj) for each conflict
        for j = 1:3 **Do**
            if j==1 **Do** % use S1: Scheduling strategy
                Solution1=ISFS(species, P_{update});
            else if j==2 **Do** % use S2: Speed adjustment strategy
                Solution2=ISFS(species, P_{update});
            else Do % use S3: Rerouting strategy
                Solution3=ISFS(species, P_{update});
            end if
        end if
    end for
    FitnessValue=FitnessFunction(Solution1, Solution2, Solution3)
    [FitnessValue, index] = sort(FitnessValue);
    dominantPop{:, 1} = species{index(:), 1}; %select dominant population
    P_{update} = (1-l_{rate})*P_{update}+l_{rate}*dominantPop/dominantNo; %update probability matrix
end while
Return (*BestSolution, FitnessValue*)

5. Simulation results and discussions

Simulations are conducted in a real-world urban environment to demonstrate the performance of the proposed adaptive decision-making framework. A comprehensive sensitivity analysis is performed for the multiple parameters used in the proposed model. The proposed algorithm is also applied to an increasing density of air traffic and algorithm comparisons are carried out to demonstrate the effectiveness and reliability of the ISFS algorithm.

5.1. Environment setup and initial 4D routes generation

A typical urban environment is selected for third-party risk modelling and risk-based route planning and decision making. The size of selected low-altitude airspace is 6,000m×6,000m×120m (width, length, and altitude). Based on the AirMatrix concept (Pang, Dai, et al., 2020), the airspace is divided into standard discrete air blocks with a size of 100m×100m×30m. The selected urban area is full of dense high-rise buildings, central business districts, city squares, shopping centers, residential areas, parks, etc., which are representative of modern mega cities. More detailed environmental risk factors and data, such as population density and vehicle density, can be found from our previous study (Pang et al., 2022).

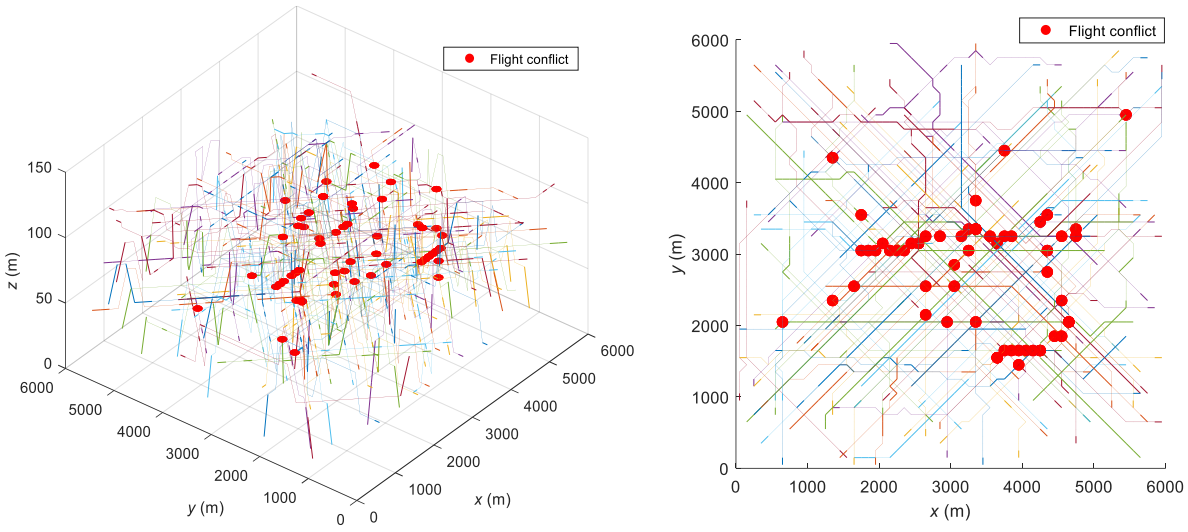
With the risk mapping data, we perform the risk-based 4D route planning for all UAVs. We assign 100 flights for this simulation in the selected airspace. The origin point and destination point of each mission are randomly selected within the boundaries, and the flight distance of each assigned mission is in the range of [2, 8] km. The time window for the departure of the 100 flights is 60 minutes, and the ETD for each flight is randomly assigned within the range. The default speed of the UAV is taken as 8 m/s. We use an exact method, the Dijkstra algorithm, to obtain the optimal pre-planned 4D routes.

Other parameters used in this study are given in Table 1. There are five decision variables (x, y, z, t, v) for each 4D route. Based on the boundary of the simulation environment and air block size, the range of (x, y, z) can be determined. The range of UAV speed is taken as [5, 15] m/s based on one of the most used drone specifications (DJI, 2018). The time window for departure is taken as 60 minutes, so the range of ETD for 100 flights is [1, 3600] s. There are three constraint thresholds. The safe separation for flight conflict is determined by the minimum time difference of consecutive UAVs passing the same waypoint. The succeeding UAV should not reach the conflict zone boundary before the preceding UAV exists the zone (illustrated in Fig. 2(b)). Based on that, the separation minima $\Delta t_{\text{conflict}}$ for flight conflict can be obtained as 30 seconds. The threshold of flight delay T^a_{delay} is taken as 20 minutes (1200 s), and the battery duration T^a_{BTRY} is taken as 25 minutes (1500 s). As flight missions may need to be taken over before the battery of the previous flight runs out, a slot of 5 minutes (300 s) should be allocated for the takeover. As to the weightage factors of the three sub-objectives in the MINLP model, we take the weightage of operational risk ω_{risk} is 0.5, while the flight delay weightage $\omega_{\text{t\_delay}}$ and airborne time weightage $\omega_{\text{t\_air}}$ are taken as 0.25. Here, we take the conflict penalty weightage ω_{fc} as 0.10 times of total cost index in each corresponding iteration.

Table 1. Parameters used in simulation studies

Decision variables	Range of value	Constraint thresholds	Value	Weightage factors	Value
x	[1, 60]	$\Delta t_{\text{conflict}}$ (s)	30	ω_{risk}	0.50
y	[1, 60]	T^a_{delay} (s)	1200	$\omega_{\text{t\_delay}}$	0.25
z	[1, 4]	T^a_{BTRY} (s)	1500	$\omega_{\text{t\_air}}$	0.25
v (m/s)	[5, 15]			ω_{fc}	0.10
ETD (s)	[1, 3600]				

With the environment set up, we generate 100 UAV 4D routes and detect the flight conflicts as shown in Fig. 4. The OD pairs of 100 flights are randomly distributed in the environment map, and 74 flight conflicts are detected. Basically, there are two characteristics of conflict distribution. The most obvious one is in the middle of the simulation map where flight routes densely intersect with each other, and the dense traffic causes more conflicts than other areas. The conflict also occurs at waypoints where more than two routes are intersecting with. For instance, as shown in Fig. 4(b), the conflicts in positions of (1500, 4300), (3900, 4400), and (5500, 5000) are not in a dense area, but they have three routes crossed at the positions, which makes these waypoints vulnerable to have flight conflicts. All these potential flight conflicts threaten the safe operations of UAVs in the urban area and need to be mitigated.



(a) 3D view (b) Top view

Fig. 4. Pre-planned 4D routes with detected flight conflicts (74 conflicts)

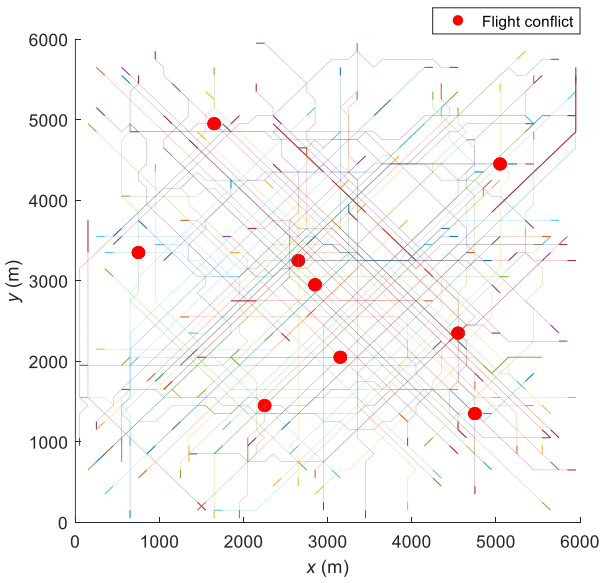
5.2. Results of conflict resolution and 4D routes optimization

We use our proposed adaptive decision-making method to optimize the obtained pre-planned 4D routes. Four aspects of resolution strategies are used: (1) Scheduling, which means only the estimated time of departure (ETD) for UAV flight t_0^a is the decision variable to optimize the routes using proposed MINLP model. (2) Speed adjustment, only the flight speed v_{ij}^a is the decision variable. (3) Rerouting strategy, only the flight altitude, and 2D coordinates can be changed. (4) Adaptive decision-making method, which adaptively employs all the three individual strategies as presented above. The decision variables of ADM strategy are (w_i, v_{ij}^a, t_0^a), and MINLP model is used to optimize the 4D routes. The ISFS algorithm is used to solve the developed optimization problems, and the ADM method is used to find solutions for strategy allocation. The total cost index computed by Eq. (20) is normalized to [1, 100,000] for all four simulations by using four different strategies. The obtained results are presented in Fig. 5. and Table 2. The number of each strategy used in the ADM method is presented in Fig. 7.

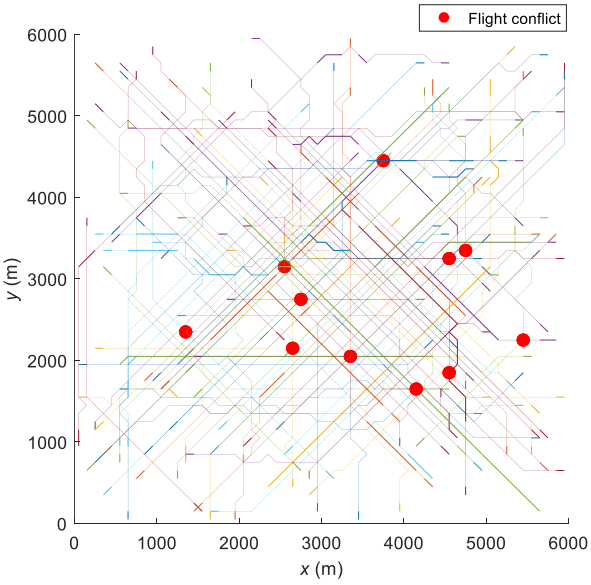
After optimization, the flight conflicts are significantly reduced in all situations where different resolution strategies are used. In the temporal dimension, scheduling and speed adjustment strategies perform well with the reduction of conflict number from 74 (Fig. 4(b)) to 9 (Fig. 5(a)) and to 11 (Fig. 5(b)), respectively. These two strategies

do not change the 3D routes, which has less impact on global flight plans. The conflict distributions are also not congested, which gives more space for further resolution reactions. As to the rerouting strategy, it changes the overall layout of the pre-planned routes. The flight conflicts in optimized routes are reduced from 74 to 12, whereas the conflicts are congested in certain areas, like the location of (4000, 3000) in Fig. 5(c). Our proposed ADM method performs the best, with no flight conflict existing in the optimized routes as shown in Fig. 5(d).

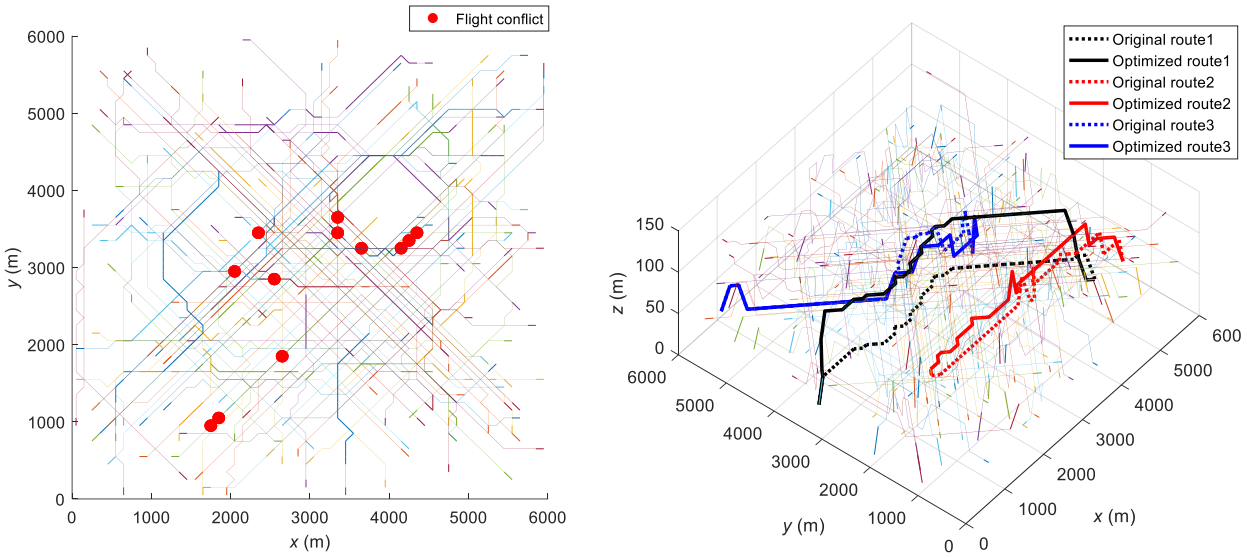
The locations of the conflicts are not the same in the three optimized results by individual strategies. That is because each of the strategies has its unique way to optimize the 4D routes. By adjusting the departure time, the scheduling strategy globally changes the ETA of all waypoints for the pertaining route, and that avoids the traffic getting congested at one area in a short period of time. It subsequently avoids conflicts, especially the conflict clusters. That explains the sparse conflict distribution in Fig. 5(a). The speed adjustment optimizes the 4D routes by changing the flight speed of the UAV. Similar to the scheduling strategy, it also does not change the layout of the routes, which avoids conflict clusters by allocating different speeds for UAVs that are affected by conflicts. For the rerouting strategy, it optimizes the 4D routes in spatial dimension by diverting the flight to avoid conflicts. However, rerouting increases the risk cost, as any change to the pre-planned optimal routes will increase the risk cost index, and there are still conflict clusters as shown in Fig. 5(c). On the other hand, our proposed ADM strategy adaptively employs the scheduling, speed adjustment, and rerouting strategies to select the best suitable strategy for different types of conflicts. For instance, to solve head-to-head flight conflicts, the rerouting strategy is used. As shown in Fig. 5(d), three routes are optimized by rerouting strategy, as the locations of these routes have several conflict clusters and head-to-head conflicts (Fig. 4(b)). The affected routes are rerouted to solve these clusters. Specifically, the flight altitude of the route1 is changed and the 2D coordinates (x, y) of the route2 are completely changed, while a segment of the route3 is diverted to avoid flight conflicts.



(a) Scheduling (9 conflicts)



(b) Speed adjustment (11 conflicts)

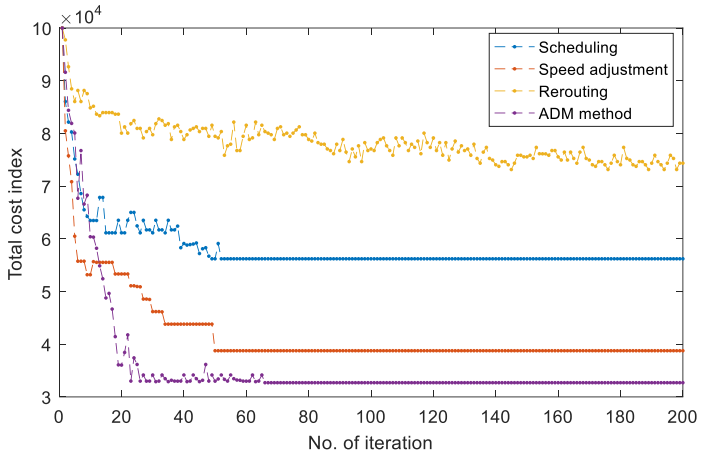


(c) Rerouting (12 conflicts)

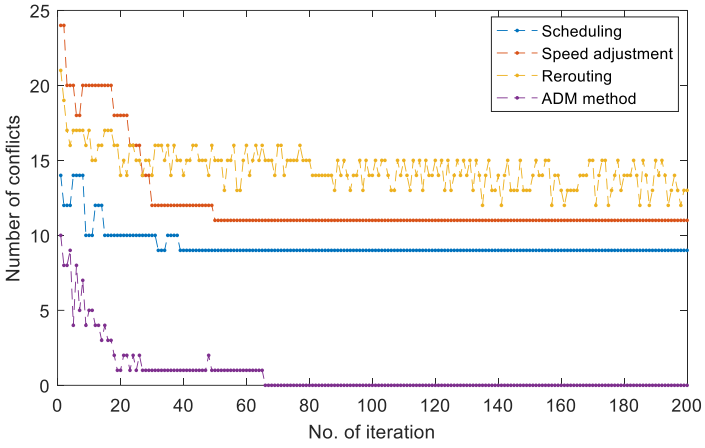
(d) Adaptive dicision-making (zero conflict)

Fig. 5. Optimization results by using different conflict resolution strategies

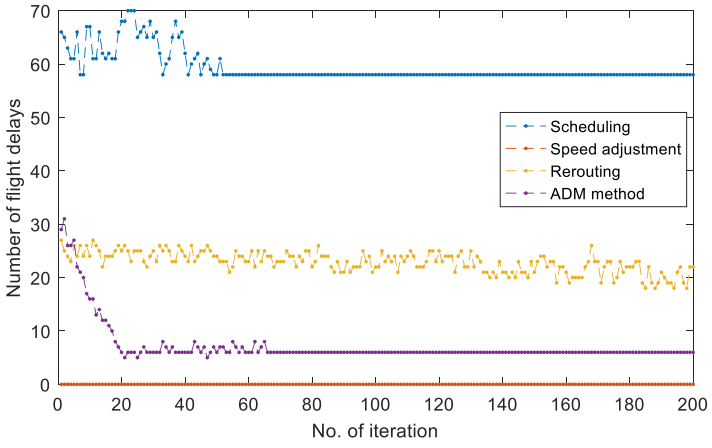
The performance of optimized 4D routes is evaluated by the total cost index, the number of flight conflicts, and the number of flight delays, which are presented in Fig. 6(a), Fig. 6(b), and Fig. 6(c), respectively. As discussed above, the proposed ADM method takes advantage of the other three individual strategies to provide the best suitable solution for each type of conflict. Its performance excels among the other three single strategies, with the lowest cost index of 32697.21 and no flight conflict, shown in Table 2. Followed by the speed adjustment strategy, it has a cost index of 38764.99, and it performs the best in reducing the number of flight delays, as it does not change the departure time and flight route. So, the arrival delay is controllable within the threshold of T^{a}_{delay}, though the number of flight conflicts is large. As to the scheduling strategy, it has the second least of conflict number, but it causes the most flight delays as shown in Table 2. The reason is that the scheduling strategy optimizes the 4D routes only by adjusting the departure time, which in most of the time, postponing departure time and subsequently causes flight delays. Lastly, for the rerouting strategy, it has the worst cost index and a large number of conflicts. As the pre-planned 3D routes are generated by the Dijkstra algorithm, which are optimal. Any change on the pre-planned routes will increase the risk cost and flight distance, which are two key contributors to compute the fitness value. What is more, the change of pre-planned routes resolves some conflicts, however, it also has impacts on other unchanged routes, which may produce new flight conflicts.



(a) Total cost index



(b) Number of flight conflicts



(c) Number of flight delays

Fig. 6. Comparison of different conflict resolution strategies

For the convergency speed, the scheduling and speed strategies have better performance, as their decision variables in the optimization problem are only departure time and flight speed, respectively. The rerouting strategy is the slowest one and even cannot get converged within 500 iterations. The reason is that this strategy has three decision variables

(x, y, z), and each of the variables will have a bigger value range when the airspace size becomes larger. That exponentially increases the algorithm searching space and the optimization process needs more generations to find desirable solutions. The convergency speed of the ADM method is the second fast one, as most of the strategies it employs are the scheduling and speed adjustment (see Fig. 7).

Table 2. Comparison results of different conflict resolution strategies

Strategies	Total cost index	Number of conflicts	Number of flight delays
Scheduling	56208.46	9	58
Speed adjustment	38764.99	11	**0**
Rerouting	74348.46	13	22
ADM method	**32697.21**	**0**	6

As we can see from Fig. 7, the number of speed strategy used notably increases while the number of the other two strategies decreases. After optimization, the number of speed adjustment used in the ADM method is 88, while the scheduling and rerouting strategies are 9 and 3, respectively. This indicates that the best suitable strategy for most conflicts is the speed adjustment strategy. Because changing flight speed is effective to solve conflicts and it brings the least impact on the global flight plan, as it does not change the departure time and pre-planned 3D routes. However, there are still some flight conflicts the speed strategy is unable to handle. For instance, a conflict cluster situation is shown in Fig. 4(b), at the position of (x: 2000-5000, y: 3000). In that case, the scheduling strategy is more suitable to solve the conflict clusters by adjusting the departure time of the pertaining UAVs. In addition, the speed strategy is also invalid in another type of conflict, which is the head-to-head conflict (as shown in Fig. 2(c), the conflict of UAV c and UAV d). The most effective strategy for that case is rerouting, which changes the flight altitude or diverts the route to avoid conflicts (as shown in Fig. 5(d)).

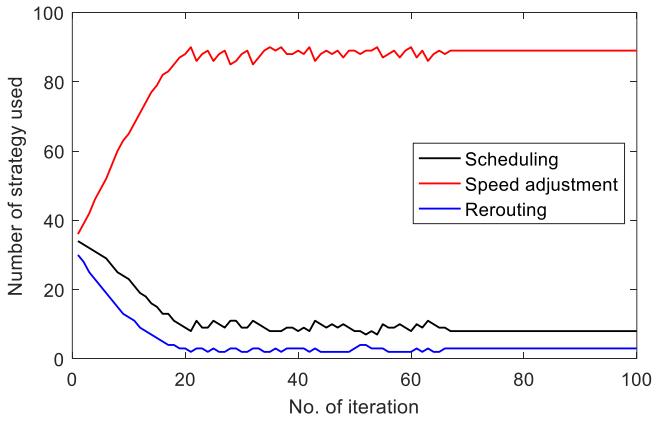


Fig. 7. Number of particular strategies used in the ADM method

5.3. Sensitivity analysis for the parameters used in the proposed model

In this paper, there are multiple parameters used in the algorithm and the mathematical model. They can be classified as algorithm-related parameters and mathematical model-related parameters. The algorithm-related

parameters are basic ones that influence the solution quality and computational efficiency of the algorithm for all application cases. The sensitivity analysis for these parameters has been well conducted in the base SFS algorithm (Salimi, 2015), and will not be covered in this paper. While for the model-related parameters, a sensitivity analysis is needed to investigate the influences of parameters on the optimization results of the multiple 4D flight routes. The model-related parameters include the constraint thresholds and the weightage factors, which are explained and presented in Table 1.

5.3.1. Parameters in the thresholds of constraints

A. Threshold of separation minima

The thresholds of constraints include separation minima, battery duration, and departure delay. The determination of the threshold for separation minima is a compromise between the safety and efficiency of the air traffic. A larger separation between aircraft makes a safer flight operation. However, it reduces the efficiency of air traffic flow and airspace utilization, as a larger separation requires a bigger safety buffer and conflict protection zone for each flight, which occupies more airspace. For the threshold of battery duration, it is determined by the battery capacity of the UAV. A better battery duration enables UAVs to execute a flight path with a longer distance. Lastly, the threshold of departure delay is determined by the user's acceptance of the time of delay. A larger threshold of departure delay can increase the punctuality rate of flights, as it increases the tolerability for the delay.

In the study, the separation minima is taken from the range of [5, 500] seconds with an interval of five seconds. The thresholds of battery duration and departure delay are taken from the range of [2, 100] minutes with an interval of two minutes. Sensitivity analysis is conducted for all three types of thresholds. The environment setting and initial 4D routes used in this analysis are as same as we presented in Section 5.1.

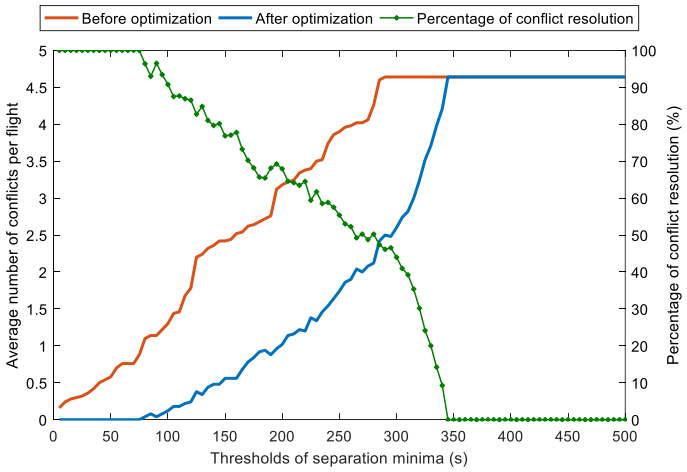
The threshold of separation minima is directly correlated with the number of flight conflicts. As discussed above, increasing the threshold for separation minima will increase the number of conflicts given the same density of air traffic. While the threshold does not correlate with flight delays, as changing the separation minima will not affect the departure time and arrival time. The sensitivity analysis results concerning an average number of conflicts and the total cost index for separation minima are presented in Fig. 8.

Fig. 8(a) shows the average number of conflicts with the changes of the separation minima thresholds before and after optimization. The percentage of conflict resolution after optimization is also presented. Before optimization, with the increase of the separation threshold, the average number of conflicts significantly increase. That is because a larger threshold causes more violations of separation minima, which produces more flight conflicts. While the average number of conflicts ceases to increase once the separation minima exceed a certain threshold. The reason is that the threshold of separation minima is too large, and the time difference between flights that will pass the same crossing waypoints is no longer smaller than the separation minima, as defined in Eq. (1). In this study, the threshold of separation minima is 290 seconds, and the average number of conflicts under that threshold is 4.64. After optimization, the average number of conflicts is significantly reduced. When the minimum separation is below a certain threshold, all conflicts can be solved (100% resolution) by using the adaptive strategies developed in this paper. In this study, the threshold for zero-conflict (no conflict) is 75 seconds with the traffic density being 100 flights/hour in the environment with the size of 36km^2. Passing that threshold, the conflicts cannot be completely solved. With the further

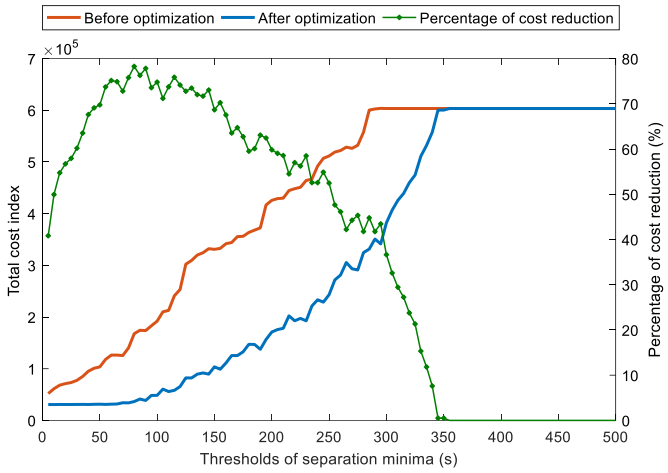
increase of the threshold, the average number of conflicts increases exponentially while the percentage of conflict resolution drastically decreases to zero when exceeding the full-conflict (no more conflicts) threshold 345 seconds in this case.

Fig. 8(b) shows the total cost index with the changes of the separation minima thresholds before and after optimization. The results of the total cost index follow the same trends as the average number of conflicts, as the cost of flight conflict is the only contributor for the total cost index in the analysis of separation minima thresholds. While the percentage of the cost reduction shows a different trend. It increases significantly before exceeding the zero-conflict threshold (75 seconds) and then decreases to aero when exceeding a full-conflict threshold (345 seconds). The percentage of cost reduction is defined as $\left(C_{\text{beforeOpt}} - C_{\text{afterOpt}}\right)/C_{\text{beforeOpt}}$. Note that $C_{\text{beforeOpt}}$ and C_{afterOpt} are total cost before and after optimization, respectively. Before exceeding the zero-conflict threshold, all conflicts can be solved and the C_{afterOpt} does not increase. While the $C_{\text{beforeOpt}}$ continues to rise with the increase of the separation threshold. That makes the percentage of the cost reduction an increasing trend. After exceeding the zero-conflict threshold, both the $C_{\text{beforeOpt}}$ and C_{afterOpt} begin to rise as the number of conflicts increases, which results in a decreasing trend for the percentage.

In summary, the conflict resolution and optimization results are highly sensitive for the threshold of separation minima. Exceeding a certain threshold of separation minima, the algorithm will not be able to solve all conflicts. With the further increase of the threshold, the average number of conflicts will begin to rise. The total cost index follows the same trend as the average number of conflicts showed.



(a) Average number of conflicts per flight



(b) Total cost index

Fig. 8. Sensitivity analysis for the thresholds of separation minima

B. Thresholds of flight delay and battery duration

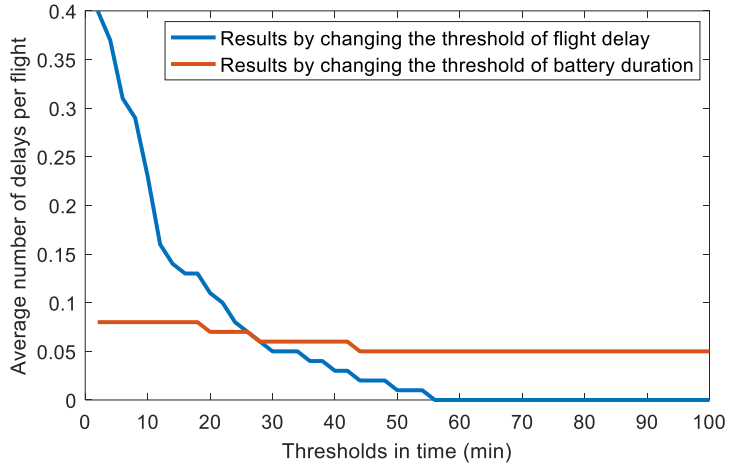
The sensitivity analysis for the thresholds of flight delay and battery duration is conducted. The threshold of flight delay is used to evaluate the number of delays, and the threshold of battery duration is used as a constraint for flight distance. Both of which do not correlate with the number of flight conflicts. These two thresholds will be evaluated by the average number of delays and the total cost index. The obtained results are presented in Fig. 9.

As defined in Eq. (4), a departure delay is determined by the difference between the actual departure time and the pre-planned one. In the optimization process, the actual departure time of some flights could be significantly changed to resolve conflicts. That will yield flight delays if the time difference exceeds the threshold of the departure delay.

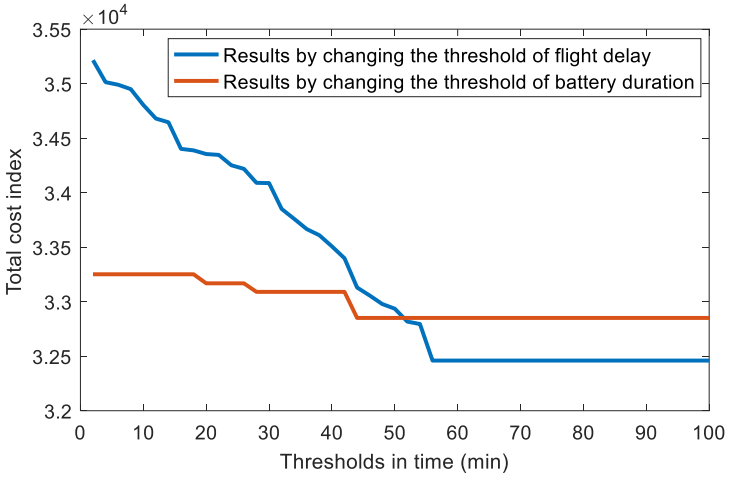
As shown in Fig. 9(a), the average number of delays is sensitive to the threshold of the flight delay. With the increase of the threshold, the number of flight delays exponentially decreases all the way to zero when the threshold exceeds 56 minutes in this case study. However, 88% of the flight delays are within the threshold of 20 minutes (the average number of delays is 0.12), as the overall delay is to be minimized in the optimization process defined in Eq. (4). The total cost index shows a linearly decreasing trend (Fig. 9(b)) with the increase of flight delay threshold, as the number of flight delays is a contributor to the calculation of the total cost index. When the threshold exceeds 56 minutes, the delays are reduced to zero and the total cost index gets converged.

The analysis for the threshold of battery duration shows that the optimization results are not sensitive to that threshold. Both the average number of delays and the total cost index have no significant changes with the increase of the battery duration threshold, as shown in Fig. 9(a) and Fig. 9(b). That is because the battery duration is only considered as a constraint for airborne time, as defined in Eq. (8). The airborne time is correlated with the flight distance and the flight speed of a route. To avoid conflicts, a long flight distance or a very slow flight speed could be assigned, which will increase the airborne time of UAVs for a specific route. In our proposed model, a strategy of generating a long flight route will be discarded in the optimization process, as a longer distance increases both the operational risk (Eq. (3)) and airborne time (Eq. (5)) that are minimized in the objective function. The slow speed (or

even hovering) can also be avoided by the constraint given for the range of the flight speed (Eq. (11)). That all makes the results are not sensitive to the battery duration threshold.



(a) Average number of delays per flight



(b) Total cost index

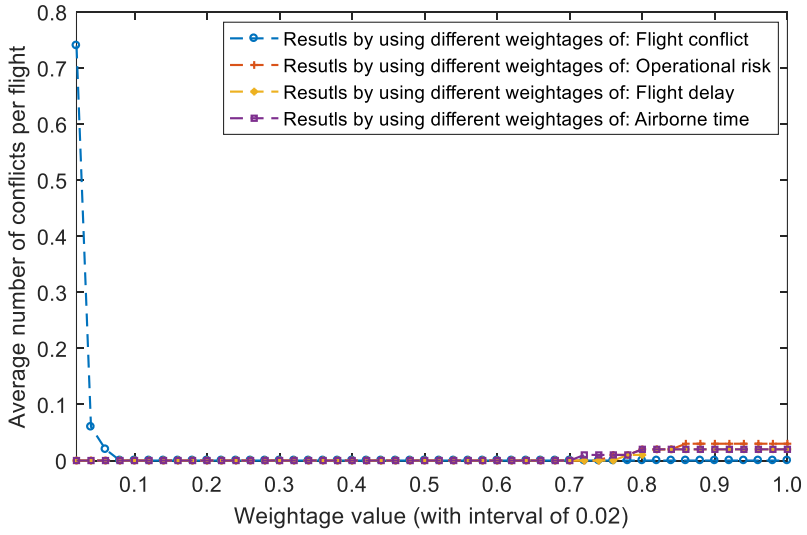
Fig. 9. Sensitivity analysis for the thresholds of flight delay and battery duration

5.3.2. Weightage factors in the objective and penalty functions

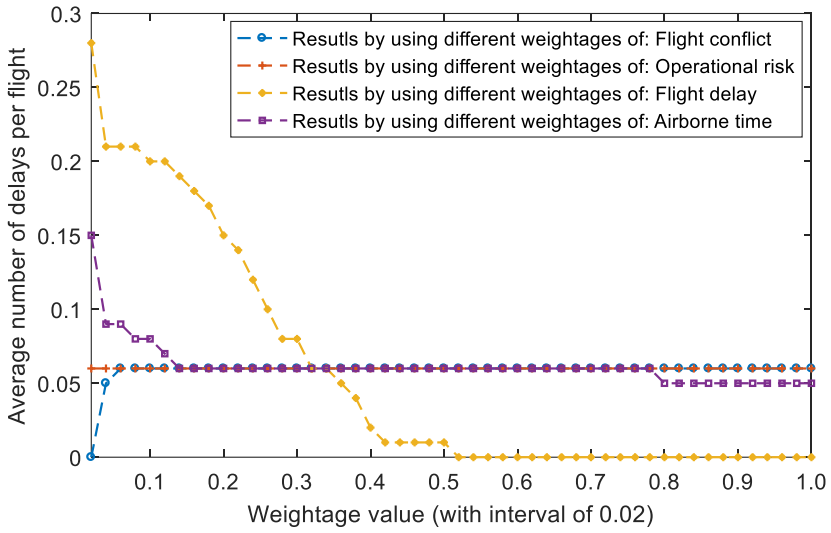
The weightage factors used in the objective function and penalty function may also influence the optimization results. These weightage factors are weightage of flight conflict ω_{fc}, weightage of operational risk ω_{risk}, weightage of flight delay $\omega_{\text{t\_delay}}$, and weightage of airborne time $\omega_{\text{t\_air}}$. In practical applications, these factors can be determined by users' preferences or the significance of the factors. In this analysis, all factors are taken from the range of [0, 1] with an interval of 0.02 to demonstrate the sensitivity of optimization results with the changes of these factors. As the factors of ω_{risk}, $\omega_{\text{t\_delay}}$, and $\omega_{\text{t\_air}}$ are inclusive (Eq. (2)) in the computation of the total cost index, they will be evaluated by the average number of conflicts and the average number of delays. The constraint thresholds are used as the ones given in Table 1, and they are fixed in the analysis. Only the weightage factors will be analyzed in this subsection. The obtained results are presented in Fig. 10.

The average number of conflicts is highly sensitive for the weightage of flight conflict, as shown in Fig. 10(a). There is a giant drop in the average number of conflicts when the weightage of flight conflict comes into play. When the weightage ω_{fc}=0, the number of conflicts is not reduced, as there is no penalty for flight conflicts in the optimization process. While when ω_{fc}=0.02, the average number of conflicts is dramatically reduced from 0.74 to 0.05, as the penalty is introduced to discard solutions with flight conflicts. Though the weightage is small, it is the times of the total cost index, which make the penalty effective. The more conflicts a solution has, the higher probability that the solution will be discarded in the next iteration. On the other hand, the average number of conflicts is not sensitive for the rest of the weightage factors, as they do not contribute to the penalty that is used to penalize the solutions with conflicts. However, these weightage factors have a knock-on effect for conflict resolution when they become large. For instance, when the weightage of the operational risk exceeds 0.86, the average number of conflicts increases from 0 to 0.03, which is equivalent to three conflicts out of 100 flights. The reason is that too high an operational cost will deny the rerouting strategy, as it will increase the cost when changing the route from the original cost-optimal one.

Come to the average number of flight delays, it is highly sensitive to the weightage of the flight delay, as presented in Fig. 10(b). With the increase of this weightage, the average number of delays has significantly decreased. That is because a larger weightage of delay will make the delay cost higher. Solutions that use the scheduling strategy to delay the departure time will be reduced and that reduces the average number of delays. While the analysis results are less sensitive for the weightage of airborne time and flight conflict, compared with the weightage of delay. With the increase of the two weightage factors, the minimization of airborne time will dominate the objective function. Solutions with shorter airborne time will be selected, which increases the punctuality of ETA, and subsequently reduced the number of delays. In contrast, the average number of delays increases with the increase of the flight conflict weightage. The scheduling strategy would be used to solve a certain type of flight conflict when the weightage of conflict comes into play. For the last weightage factor of operational risk, the number of delays is not sensitive to it. As the operational risk is only correlated with the waypoints of the routes and does not influence the delays.



(a) Average number of conflicts per flight



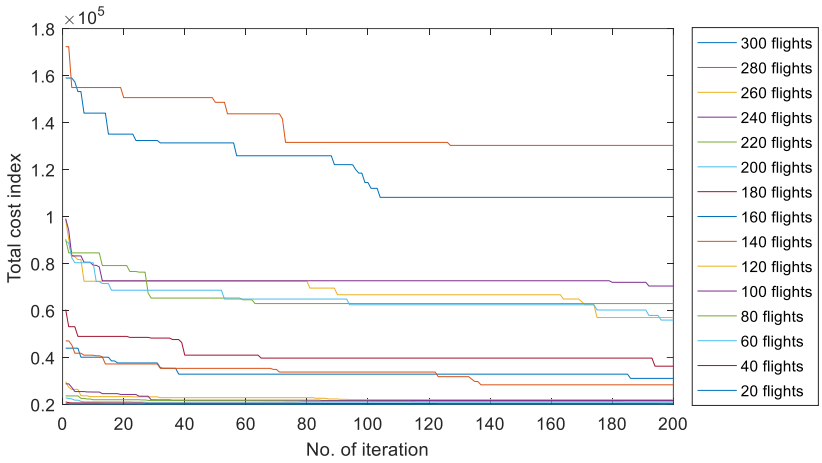
(b) Average number of delays per flight

Fig. 10. Sensitivity analysis for the weightage factors

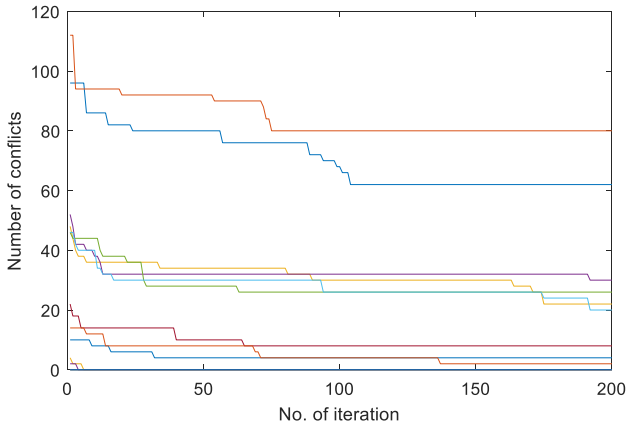
5.4. Algorithm testing in different air traffic densities

Simulations are conducted to investigate the optimization performance of the proposed algorithm in scenarios with different air traffic densities. The highest density in this simulation is 300 drones per hour in an environment with a size of 36 km^2. That is equivalent to 6,000 drone operations per hour (144,000 per day) in Singapore. Simulations are performed in the same environment with an increasing number of flights from 20 to 300 with an interval of 20. The ISFS algorithm is used to optimize the 4D routes, and the obtained results are presented in Fig. 11 and Fig. 12.

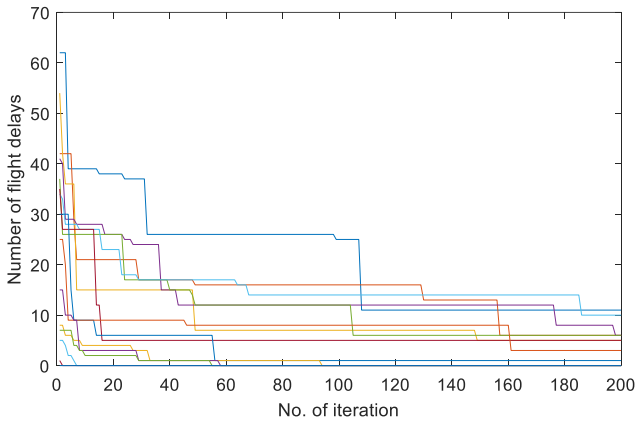
In general, with the increase of flight density, the total cost index (Fig. 11(a)), number of flight conflicts (Fig. 11(b)), and number of flight delays (Fig. 11(c)) increase. It is also notable that within a range of flight numbers, the performance indicators have no significant changes. However, from one range to another, there is a great jump in terms of performance change. For instance, as we can see from the results in Fig. 11, there are four ranges of flight number, which are [20, 120], [140, 180], [200, 260], and [280, 300] with an interval of 20. In the first range, all indicators are convergent. With the increase of traffic density, the algorithm is still effective, but it needs more iterations to get converged. In addition, the case of 300 flights has an even better performance than that of the 280 flights case, which reveals another finding: there has a clear threshold of airspace capacity, which can be evaluated by the operational performance. For instance, the increase of flight numbers within a certain group (e.g. [140, 180]) will not significantly influence traffic performance in terms of operational cost and flight conflicts. That is because, within a certain range, the proposed ADM algorithm can always find suitable strategies to handle flight conflicts and optimize the 4D routes. However, once the flight density exceeds a certain level of capacity threshold, the newly added flights will cause a significant impact on existing flight plans. After that, a new balance is established until the number of flights exceeds the next threshold. This finding can facilitate the determination of the airspace capacity threshold.



(a) Total cost index



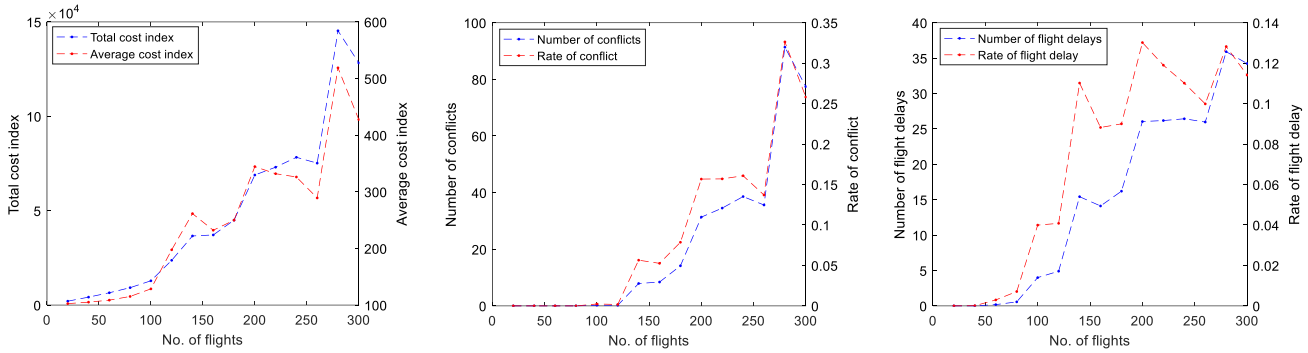
(b) Number of flight conflicts



(c) Number of flight delays

Fig. 11. Optimization performances with different traffic densities. The traffic density is defined as the number of flights per hour in an environment with a size of 36km^2.

Note that Fig. 12 shows the trends of average cost index, rate of flight conflicts, and rate of flight delays. All the indicators increase exponentially with the increase of flight density. In the first range [20, 120], the average cost index and the number of flight delay slightly increases while there is no flight conflict. In this range, the traffic density is low, and the ADM method provides good solutions for all conflicts with only a few flights exceeding the arrival delay threshold. That is because the scheduling strategy is used for a portion of conflicts, which may postpone some flights causing delays. Come to the second range [140, 180], there is a rise in the indicators with an average cost index increases to 250 (Fig. 12(a)), the rate of conflict increases to 0.05 shown in Fig. 12(b), and rate of flight delays to 0.1 shown in Fig. 12(c). A steeper increase is from the second range to the third range, with the average cost index increasing from 200 to 300 and the conflict rate from 0.05 to 0.15. The most significant jump is from the third range [200, 260] to the fourth range [280, 300], with the average cost index and rate of conflict doubled.



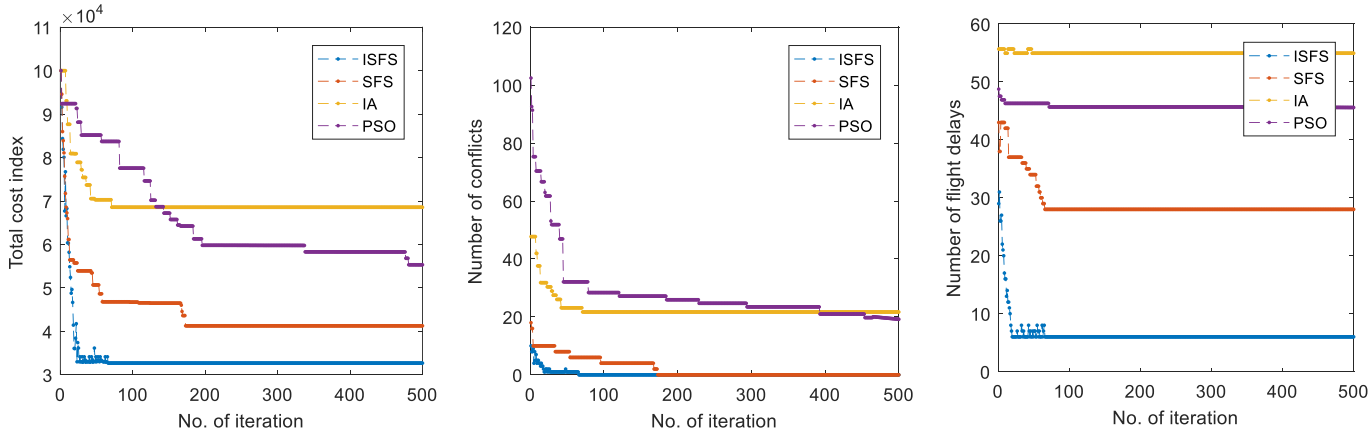
(a) Cost index (b) Flight conflicts (c) Flight delays

Fig. 12. The average performance of indicators decreases with the increase of traffic density. The traffic density is defined as the number of flights per hour in an environment with a size of 36 km^2.

5.5. Comparison of algorithms: ISFS, SFS, IA, and PSO

To verify the performance of the proposed ISFS algorithm, we compare ISFS with other three well-known meta-heuristic search algorithms, which are original stochastic fractal search (SFS), immune algorithm (IA), and particle swarm optimization (PSO) algorithm. The population size is taken as 50 for all algorithms and the number of iterations is taken as 500. The simulation environment is as same as the one we developed in Section 5.1. The number of flights is taken as 100 and the Dijkstra algorithm is used to generate the initial 4D routes as input for the optimization.

Fig. 13 shows the optimization results by using the four algorithms. Our proposed ISFS algorithm performs the best in all indicators and has the fastest convergency rate. It has the lowest cost index (Fig. 13(a)) and has no flight conflict in optimized 4D routes (Fig. 13(b)), while the number of flight delays is also the smallest (Fig. 13(c)). Followed by the SFS method, it has the second lowest cost index and also has no flight conflict after optimization. The PSO algorithm does not converge whereas its performance is good in minimizing the cost index and the number of conflicts. As for the IA method, although it has a fast convergence rate, the cost index of this method is two times more than that of the ISFS algorithm, and it has the largest number of flight delays among the four algorithms.



(a) Total cost index (b) Number of flight conflicts (c) Number of flight delays

Fig. 13. Algorithm comparison results

To test the fairness of the algorithm comparison results and validate the reliability of the proposed ISFS method, we conduct 50 times of independent simulations with the same initial 4D routes in the same environment. The obtained statistical results are shown in Table 3.

On average, the ISFS is the best among the four algorithms in terms of solution quality and convergence time. It has the lowest average cost index (32697.21) with the smallest standard deviation (1594.87). The convergence time of the ISFS is also the lowest (216.59 s). The optimized 4D routes using ISFS has no conflict in all 50 simulations and the average number of flight delays is the smallest (6 delays). The results of the SFS method have an increase of 16.18% in cost index, 0.87% in standard deviation, and one flight conflict, compared with ISFS. The results produced by the PSO algorithm have a dramatic increase in cost index, flight conflicts, and delays with the largest standard deviation (6857.87). The performance of the IA method is down to the bottom among the list, with the average cost index more than doubled compared with ISFS. The number of flight conflicts and delays are the highest with 22 and 55, respectively.

Table 3. Statistical results (on average) of the 50 independent simulations

Algorithms	Cost index	Standard deviation of cost index	Number of conflicts	Number of flight delays	Convergence time (s)
ISFS	**32697.21**	**1594.87**	**0**	**6**	**216.59**
SFS	37988.46	1608.74	1	28	601.92
IA	68557.40	4103.85	22	55	638.92
PSO	55330.20	6857.87	19	46	803.55

6. Concluding remarks

This paper studies the flight conflict resolution problem for risk-based UAV 4D routes optimization in urban environments. A double-layer decision-making framework is proposed to optimize multiple 4D routes, and an improved stochastic fractal search (ISFS) algorithm is developed to solve the problem. Simulations and comparison studies are performed to demonstrate the performance of the proposed model and algorithm. The main findings of this paper are concluded as follows.

First, the adaptive decision-making method can solve flight conflicts and has better performance than individual strategy (e.g., scheduling or rerouting), as it selects the best suitable resolution strategies for different types of conflicts. Among the three individual strategies, the speed adjustment has the best performance in solving conflicts with the least disturbance caused to global planned routes. That is because the speed adjustment does not change the departure time and the initially-planned routes.

Second, to optimize the risk-based 4D routes with assigned conflict resolution strategies, an MINLP model is proposed, and an improved stochastic fractal search algorithm is developed. By introducing the exploitation-exploration balancing scheme, the improved ISFS algorithm performs the best in reducing operational cost, flight conflicts and delays compared with original SFS, IA, and PSO algorithms. Its performance is enabled by the improved fitness-distance balancing scheme that is used in the update processes to avoid exploring points that are too close to the current best point, which improves the searching diversity of ISFS.

Third, the sensitivity analysis on the number of flights indicates that the proposed ISFS algorithm is effective in dealing with conflicts in different traffic density scenarios. There is also a clear trend that the performance indicators of 4D routes have no significant changes within a range of flight numbers. However, from one range to another, there is a great jump in terms of performance decreasing. That could be a useful finding to support the determination of airspace capacity using performance-based thresholds.

Despite the strengths of our proposed approach, there are also some weaknesses requiring further studies. First, the third-party risk index is taken from one of our previous studies and is computed as a static value. In future works, we can study the real-time risk assessment problems considering more types of risk sources to ensure a safer UAV operation. Second, our proposed method is to solve the pre-planned 4D route optimization problem in static environments. In the next steps, we can investigate decision-making approaches for real-time risk-based conflict resolution to handle dynamic risks. In addition, the proposed ISFS algorithm is effective to solve the MINLP model in different traffic density scenarios, but the performance of ISFS is decreasing with the traffic density becoming significantly higher. The scalability problem of the optimization algorithm still has space to improve to handle larger-scale conflict resolution problems with faster computational speed. There are also some other methods such as game theory (Hang et al., 2021) and deep reinforcement learning (Li et al., 2019) have shed lights on solving the real-time conflict resolution problems, much more effects are still required to make these methods robust and implementable for real-world UAV operations.

Acknowledgment

This research is supported by the National Research Foundation (NRF), Singapore, and the Civil Aviation Authority of Singapore (CAAS), under the Aviation Transformation Programme (ATP). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not reflect the views of National Research Foundation, Singapore and the Civil Aviation Authority of Singapore. Research Student Scholarship (RSS) provided by the NTU to the first author is acknowledged. The authors would like to thank Professor Yu Wu and Mr. Wei Dai for their insightful suggestions in the mathematical modelling and simulation design sections. Our gratitude is also expressed to the anonymous reviewers for their detailed reviews, numerous valuable suggestions and comments, which have greatly improved the quality of the final manuscript.

References

Amr, H. A., Khajehzadeh, M., Taha, M., Beheshti, Z., Mariyam, S., & Shamsuddin, H. (2013). A Review of Population-based Meta- Heuristic Algorithm. *Int. J. Advance. Soft Comput. Appl*, *5*(1), 2074–8523. www.i-csrs.org

Aras, S., Gedikli, E., & Kahraman, H. T. (2021). A novel stochastic fractal search algorithm with fitness-Distance balance for global numerical optimization. *Swarm and Evolutionary Computation*, *61*(December 2020), 100821. https://doi.org/10.1016/j.swevo.2020.100821

Arora, S. (2001). Approximation schemes for geometric NP-hard problems: A survey. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *2245*, 16–17. https://doi.org/10.1007/3-540-45294-x_2

Berdonosov, V. D., Zivotova, A. A., Zhuravlev, D. O., & Naing, Z. H. (2018). Implementation of the Speed Approach for UAV Collision Avoidance in Dynamic Environment. *2018 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2018*. https://doi.org/10.1109/FarEastCon.2018.8602815

Bilimoria, K. D. (2000). A geometric optimization approach to aircraft conflict resolution. *AIAA Guidance, Navigation, and Control Conference and Exhibit*, *August*. https://doi.org/10.2514/6.2000-4265

Birolini, S., Antunes, A. P., Cattaneo, M., Malighetti, P., & Paleari, S. (2021). Integrated flight scheduling and fleet assignment with improved supply-demand interactions. *Transportation Research Part B: Methodological*, *149*, 162–180. https://doi.org/10.1016/j.trb.2021.05.001

Blom, H. A. P., Jiang, C., Grimme, W. B. A., Mitici, M., & Cheung, Y. S. (2021). Third party risk modelling of Unmanned Aircraft System operations, with application to parcel delivery service. *Reliability Engineering and System Safety*, *214*(April), 107788. https://doi.org/10.1016/j.ress.2021.107788

Burer, S., & Letchford, A. N. (2012). Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, *17*(2), 97–106. https://doi.org/10.1016/j.sorms.2012.08.001

Cabreira, T. M., Franco, C. Di, Ferreira, P. R., & Buttazzo, G. C. (2018). Energy-Aware spiral coverage path planning for UAV photogrammetric applications. *IEEE Robotics and Automation Letters*, *3*(4), 3662–3668. https://doi.org/10.1109/LRA.2018.2854967

Chaimatanan, S., Delahaye, D., & Mongeau, M. (2014). A hybrid metaheuristic optimization algorithm for strategic planning of 4D aircraft trajectories at the continental scale. *IEEE Computational Intelligence Magazine*, *9*(4), 46–61. https://doi.org/10.1109/MCI.2014.2350951

Challita, U., Saad, W., & Bettstetter, C. (2019). Interference management for cellular-connected UAVs: A deep reinforcement learning approach. *IEEE Transactions on Wireless Communications*, *18*(4), 2125–2140. https://doi.org/10.1109/TWC.2019.2900035

Cheng, C., Adulyasak, Y., & Rousseau, L. M. (2020). Drone routing with energy function: Formulation and exact algorithm. *Transportation Research Part B: Methodological*, *139*, 364–387. https://doi.org/10.1016/j.trb.2020.06.011

Cheng, F., Zhang, S., Li, Z., Chen, Y., Zhao, N., Yu, F. R., & Leung, V. C. M. (2018). UAV Trajectory Optimization for Data Offloading at the Edge of Multiple Cells. *IEEE Transactions on Vehicular Technology*, *67*(7), 6732–6736. https://doi.org/10.1109/TVT.2018.2811942

Courchelle, V., Soler, M., González-Arribas, D., & Delahaye, D. (2019). A simulated annealing approach to 3D strategic aircraft deconfliction based on en-route speed changes under wind and temperature uncertainties. *Transportation Research Part C: Emerging Technologies*, *103*(March), 194–210. https://doi.org/10.1016/j.trc.2019.03.024

Cuevas, E., Echavarría, A., & Ramírez-Ortegón, M. A. (2014). An optimization algorithm inspired by the States of Matter that improves the balance between exploration and exploitation. *Applied Intelligence*, *40*(2), 256–272. https://doi.org/10.1007/s10489-013-0458-0

Da Silva Arantes, M., Toledo, C. F. M., Williams, B. C., & Ono, M. (2019). Collision-Free Encoding for Chance-Constrained Nonconvex Path Planning. *IEEE Transactions on Robotics*, *35*(2), 433–448. https://doi.org/10.1109/TRO.2018.2878996

Dai, W., Pang, B., & Low, K. H. (2021). Conflict-Free Four-Dimensional Path Planning for Urban Air Mobility Considering Airspace Occupations. *Aerospace Science and Technology*, *119*, 107154. https://doi.org/10.1016/j.ast.2021.107154

Dias, F. H. C., Hijazi, H., & Rey, D. (2022). Disjunctive linear separation conditions and mixed-integer formulations for aircraft conflict resolution. *European Journal of Operational Research*, *296*(2), 520–538. https://doi.org/10.1016/j.ejor.2021.03.059

Dias, F. H. C., & Rey, D. (2020). Robust aircraft conflict resolution under trajectory prediction uncertainty. *ArXiv*. http://arxiv.org/abs/2012.08230

Ding, Wenchao, Gao, W., Wang, K., & Shen, S. (2019). An Efficient B-Spline-Based Kinodynamic Replanning Framework for

Quadrotors. *IEEE Transactions on Robotics*, *35*(6), 1287–1306. https://doi.org/10.1109/TRO.2019.2926390

Ding, Wenzhe, Zhang, Y., & Hansen, M. (2018). Downstream impact of flight rerouting. *Transportation Research Part C: Emerging Technologies*, *88*(June 2017), 176–186. https://doi.org/10.1016/j.trc.2018.01.010

DJI. (2018). *User Manual Phantom 4 PRO V1.6*. https://dl.djicdn.com/downloads/phantom_4_pro/20180508/Phantom_4_Pro_Pro_Plus_Series_User_Manual-EN.pdf

Durand, N., & Alliot, J. M. (2009). Ant Colony Optimization for air traffic conflict resolution. *Proceedings of the 8th USA/Europe Air Traffic Management Research and Development Seminar, ATM 2009*, 182–187.

FAA, & NASA. (2020). UAM ConOps v1.0. In *FAA*. https://utm.arc.nasa.gov/docs/2018-UTM-ConOps-v1.0.pdf

Geng, X., Wang, Y., Wang, P., & Zhang, B. (2019). Motion plan of maritime autonomous surface ships by dynamic programming for collision avoidance and speed optimization. *Sensors (Switzerland)*, *19*(2). https://doi.org/10.3390/s19020434

González-Arribas, D., Soler, M., & Sanjurjo-Rivo, M. (2018). Robust aircraft trajectory planning under wind uncertainty using optimal control. *Journal of Guidance, Control, and Dynamics*, *41*(3), 673–688. https://doi.org/10.2514/1.G002928

Hang, P., Lv, C., Huang, C., Xing, Y., & Hu, Z. (2021). Cooperative Decision Making of Connected Automated Vehicles at Multi-Lane Merging Zone: A Coalitional Game Approach. *IEEE Transactions on Intelligent Transportation Systems*, 1–13. https://doi.org/10.1109/TITS.2021.3069463

Hao, S., Cheng, S., & Zhang, Y. (2018). A multi-aircraft conflict detection and resolution method for 4-dimensional trajectory-based operation. *Chinese Journal of Aeronautics*, *31*(7), 1579–1593. https://doi.org/10.1016/j.cja.2018.04.017

Hately, A., & et al. (2019). U-space Concept of Operations. *SESAR Joint Undertaking*, *1*(October 2019), 1–92.

Hernández-Romero, E., Valenzuela, A., & Rivas, D. (2020). Probabilistic multi-aircraft conflict detection and resolution considering wind forecast uncertainty. *Aerospace Science and Technology*, *105*, 105973. https://doi.org/10.1016/j.ast.2020.105973

Hu, M., Liu, W., Peng, K., Ma, X., Cheng, W., Liu, J., & Li, B. (2019). Joint routing and scheduling for vehicle-assisted multidrone surveillance. *IEEE Internet of Things Journal*, *6*(2), 1781–1790. https://doi.org/10.1109/JIOT.2018.2878602

Hu, X., Pang, B., Dai, F., & Low, K. H. (2020). Risk Assessment Model for UAV Cost-Effective Path Planning in Urban Environments. *IEEE Access*, *8*, 150162–150173. https://doi.org/10.1109/ACCESS.2020.3016118

Kahn, G., Villaflor, A., Pong, V., Abbeel, P., & Levine, S. (2017). Uncertainty-Aware Reinforcement Learning for Collision Avoidance. *ArXiv*, 1–12. http://arxiv.org/abs/1702.01182

Kahraman, H. T., Aras, S., & Gedikli, E. (2020). Fitness-distance balance (FDB): A new selection method for meta-heuristic search algorithms. *Knowledge-Based Systems*, *190*, 105169. https://doi.org/10.1016/j.knosys.2019.105169

Kim, S. J., & Lim, G. J. (2020). A Real-Time Rerouting Method for Drone Flights Under Uncertain Flight Time. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *100*(3–4), 1355–1368. https://doi.org/10.1007/s10846-020-01214-z

Kleinbekman, I. C., Mitici, M., & Wei, P. (2020). Rolling-horizon electric vertical takeoff and landing arrival scheduling for on-demand urban air mobility. *Journal of Aerospace Information Systems*, *17*(3), 150–159. https://doi.org/10.2514/1.I010776

Li, S., Egorov, M., & Kochenderfer, M. J. (2019). Optimizing Collision Avoidance in Dense Airspace using Deep Reinforcement Learning. *Thirteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2019)*, *3*, 1–10. http://www.atmseminar.org/seminarContent/seminar13/papers/ATM_Seminar_2019_paper_65.pdf

Lim, W. X., & Zhong, Z. W. (2018). Re-Planning of Flight Routes Avoiding Convective Weather and the "Three Areas." *IEEE Transactions on Intelligent Transportation Systems*, *19*(3), 868–877. https://doi.org/10.1109/TITS.2017.2705098

Liu, Y., Zhang, X., Zhang, Y., & Guan, X. (2019). Collision free 4D path planning for multiple UAVs based on spatial refined voting mechanism and PSO approach. *Chinese Journal of Aeronautics*, *32*(6), 1504–1519. https://doi.org/10.1016/j.cja.2019.03.026

Loubière, P., Jourdan, A., Siarry, P., & Chelouah, R. (2018). A sensitivity analysis method aimed at enhancing the metaheuristics for continuous optimization. *Artificial Intelligence Review*, *50*(4), 625–647. https://doi.org/10.1007/s10462-017-9553-7

Marchidan, A., & Bakolas, E. (2020). Collision avoidance for an unmanned aerial vehicle in the presence of static and moving obstacles. *Journal of Guidance, Control, and Dynamics*, *43*(1), 96–110. https://doi.org/10.2514/1.G004446

Mellal, M. A., & Zio, E. (2016). A penalty guided stochastic fractal search approach for system reliability optimization. *Reliability Engineering and System Safety*, *152*, 213–227. https://doi.org/10.1016/j.ress.2016.03.019

Melnyk, R., Schrage, D., Volovoi, V., & Jimenez, H. (2014). A third-party casualty risk model for unmanned aircraft system operations. *Reliability Engineering and System Safety*, *124*, 105–116. https://doi.org/10.1016/j.ress.2013.11.016

Nikolos, I. K., Valavanis, K. P., Tsourveloudis, N. C., & Kostaras, A. N. (2003). Evolutionary Algorithm Based Offline/Online Path Planner for UAV Navigation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *33*(6), 898–912. https://doi.org/10.1109/TSMCB.2002.804370

Pallottino, L., Feron, E. M., & Bicchi, A. (2002). Conflict Resolution Problems for Air Traffic Management Systems Solved with Mixed Integer Programming. *IEEE Transactions on Intelligent Transportation Systems*, *3*(1), 3–11. https://doi.org/10.1109/6979.994791

Pang, B., Dai, W., Hu, X., Dai, F., & Low, K. H. (2021). Multiple air route crossing waypoints optimization via artificial potential field method. *Chinese Journal of Aeronautics*, *34*(4). https://doi.org/10.1016/j.cja.2020.10.008

Pang, B., Dai, W., Ra, T., & Low, K. H. (2020). A Concept of Airspace Configuration and Operational Rules for UAS in Current Airspace. *2020 IEEE/AIAA 39th Digital Avionics Systems Conference (DASC)*, 1–9.

Pang, B., Hu, X., Dai, W., & Low, K. H. (2022). UAV Path Optimization with An Integrated Cost Assessment Model Considering Third-Party Risks in Metropolitan Environments. *Reliability Engineering and System Safety*, 1–18. https://doi.org/https://doi.org/10.1016/j.ress.2022.108399

Pang, B., Tan, Q., Ra, T., & Low, K. H. (2020). A risk-based uas traffic network model for adaptive urban airspace management. *Aiaa Aviation 2020 Forum*, *1 PartF*, 1–9. https://doi.org/10.2514/6.2020-2900

Patchou, M., Sliwa, B., & Wietfeld, C. (2021). Flying Robots for Safe and Efficient Parcel Delivery Within the COVID-19 Pandemic. *2021 IEEE International Systems Conference (SysCon)*, 1–7. https://doi.org/10.1109/syscon48628.2021.9447142

Pelegrín, M., & D'Ambrosio, C. (2022). Aircraft Deconfliction via Mathematical Programming: Review and Insights. *Transportation Science*, *56*(1), 118–140. https://doi.org/10.1287/trsc.2021.1056

Peng, N., Xi, Y., Rao, J., Ma, X., & Ren, F. (2021). Urban Multiple Route Planning Model Using Dynamic Programming in Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, 1–11. https://doi.org/10.1109/TITS.2021.3075221

Prakash, R., Piplani, R., & Desai, J. (2018). An optimal data-splitting algorithm for aircraft scheduling on a single runway to maximize throughput. *Transportation Research Part C: Emerging Technologies*, *95*(August), 570–581. https://doi.org/10.1016/j.trc.2018.07.031

Rey, D., Rapine, C., Fondacci, R., & El Faouzi, N. E. (2016). Subliminal speed control in air traffic management: Optimization and simulation. *Transportation Science*, *50*(1), 240–262. https://doi.org/10.1287/trsc.2015.0602

Rigas, E. S., Kolios, P., & Ellinas, G. (2021). *Scheduling Aerial Vehicles in an Urban Air Mobility Scheme*. http://arxiv.org/abs/2108.01608

Rodionova, O., Delahaye, D., Sridhar, B., & Ng, H. K. (2016). Deconflicting Wind-Optimal Aircraft Trajectories in North Atlantic Oceanic Airspace. *Advanced Aircraft Efficiency in a Global Air Transport System Conference (AEGATS)*, 1–12.

Sacramento, D., Pisinger, D., & Ropke, S. (2019). An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, *102*(March 2018), 289–315. https://doi.org/10.1016/j.trc.2019.02.018

Saeed, F., Mehmood, A., Majeed, M. F., Maple, C., Saeed, K., Khattak, M. K., Wang, H., & Epiphaniou, G. (2021). Smart delivery and retrieval of swab collection kit for COVID-19 test using autonomous Unmanned Aerial Vehicles. *Physical Communication*, *48*, 101373. https://doi.org/10.1016/j.phycom.2021.101373

Salimi, H. (2015). Stochastic Fractal Search: A powerful metaheuristic algorithm. *Knowledge-Based Systems*, *75*, 1–18. https://doi.org/10.1016/j.knosys.2014.07.025

Seo, J., Kim, Y., Kim, S., & Tsourdos, A. (2017). Collision Avoidance Strategies for Unmanned Aerial Vehicles in Formation Flight. *IEEE Transactions on Aerospace and Electronic Systems*, *53*(6), 2718–2734. https://doi.org/10.1109/TAES.2017.2714898

Tan, K. C., Chiam, S. C., Mamun, A. A., & Goh, C. K. (2009). Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization. *European Journal of Operational Research*, *197*(2), 701–713. https://doi.org/10.1016/j.ejor.2008.07.025

Tan, Q., Wang, Z., Ong, Y. S., & Low, K. H. (2019). Evolutionary optimization-based mission planning for UAS traffic management (UTM). *2019 International Conference on Unmanned Aircraft Systems, ICUAS 2019*, 952–958. https://doi.org/10.1109/ICUAS.2019.8798078

Tang, H., Zhang, Y., Mohmoodian, V., & Charkhgard, H. (2021). Automated flight planning of high-density urban air mobility. *Transportation Research Part C: Emerging Technologies*, *131*(August), 103324. https://doi.org/10.1016/j.trc.2021.103324

Wilhelm, J. P., & Clem, G. (2019). Vector field UAV guidance for path following and obstacle avoidance with minimal deviation. *Journal of Guidance, Control, and Dynamics*, *42*(8), 1848–1856. https://doi.org/10.2514/1.G004053

Wu, Y., Low, K. H., & Hu, X. (2021). Trajectory-based flight scheduling for AirMetro in urban environments by conflict resolution. *Transportation Research Part C: Emerging Technologies*, *131C*, 1–56.

Wu, Y., Low, K. H., & Lv, C. (2020). Cooperative Path Planning for Heterogeneous Unmanned Vehicles in a Search-and-Track Mission Aiming at an Underwater Target. *IEEE Transactions on Vehicular Technology*, *69*(6), 6782–6787. https://doi.org/10.1109/TVT.2020.2991983

Wu, Y., Low, K. H., Pang, B., & Tan, Q. (2021). Swarm-based 4D Path Planning for Drone Operations in Urban Environments. *IEEE Transactions on Vehicular Technology*, *9545*(c). https://doi.org/10.1109/TVT.2021.3093318

Xie, H., Low, K. H., & He, Z. (2017). Adaptive visual servoing of unmanned aerial vehicles in GPS-denied environments. *IEEE/ASME Transactions on Mechatronics*, *22*(6), 2554–2563. https://doi.org/10.1109/TMECH.2017.2755669

Yang, X., & Wei, P. (2020). Scalable multi-agent computational guidance with separation assurance for autonomous urban air mobility. *Journal of Guidance, Control, and Dynamics*, *43*(8), 1473–1486. https://doi.org/10.2514/1.G005000

Yang, X., & Wei, P. (2021). Autonomous Free Flight Operations in Urban Air Mobility With Computational Guidance and Collision Avoidance. *IEEE Transactions on Intelligent Transportation Systems*, 1–14. https://doi.org/10.1109/TITS.2020.3048360

Yao, W., Qi, N., Wan, N., & Liu, Y. (2019). An iterative strategy for task assignment and path planning of distributed multiple unmanned aerial vehicles. *Aerospace Science and Technology*, *86*, 455–464. https://doi.org/10.1016/j.ast.2019.01.061

Zhang, Y., Su, R., Sandamali, G. G. N., Zhang, Y., Cassandras, C. G., & Xie, L. (2019). A Hierarchical Heuristic Approach for Solving Air Traffic Scheduling and Routing Problem with a Novel Air Traffic Model. *IEEE Transactions on Intelligent Transportation Systems*, *20*(9), 3421–3434. https://doi.org/10.1109/TITS.2018.2874235