Integrated Arrival Solution

Compatible Arrival Sequencing and Trajectory Solution

Richard Louie
Department of Mechanical and Aerospace Engineering
Hong Kong University of Science and Technology
Hong Kong, Hong Kong
ryhlouie@connect.ust.hk

Rhea P. Liem
Department of Aeronautics
Imperial College London
London, United Kingdom
r.liem@imperial.ac.uk

***Abstract*—Extended air traffic congestion within the terminal maneuvering area has resulted in frequent flight delays and degraded arrival efficiency. Accumulated delays increase fuel consumption and traffic guidance workload, thereby posing potential risks to flight safety. To mitigate the impact, modern air traffic management systems combine manual and computer-based solutions in the decision-making process. The manual component, however, is typically less consistent and vulnerable to traffic conditions; this may result in a suboptimal performance that fails to fulfill the targets projected by the computational tools. Thus, an integrated arrival solution is necessary to reduce workload and alleviate delays during peak traffic periods. We propose an integrated arrival solution with the trajectory grafting method, which leverages historical trajectory solutions to synthesize high-fidelity new trajectories that retain embedded features. Without prior information on the arrival entry time, the arrival sequencing solution is trajectory-driven. Using fuel consumption reduction as the main objective, the proposed integrated arrival solution is demonstrated and tested using Hong Kong arrival schedules from 2018. Results show that arrival sequencing can be fully driven by trajectory solutions without pre-sequencing or prior knowledge of future arrival information. Our integrated arrival solution with trajectory grafting can reduce arrival fuel consumption while maintaining safety distances. Compared to the current decoupled processes, the resulting trajectories are more predictable and require less reactive decision-making. To a certain extent, this attempt aligns with the principles of trajectory-based operations and demonstrates the capability and potential of the trajectory grafting method.**

***Keywords*—Integrated arrival solution, Arrival sequencing, Trajectory solution, Trajectory based operations**

I. Introduction

A. Terminal maneuvering area congestion and increased fuel consumption

The terminal maneuvering area (TMA) is a critical airspace preceding flight landing. Within this area, arrival flights from different entry points descend and converge along a set of standard arrival routes (STARs), composing dynamic and complex traffic interactions. Air traffic controllers (ATCOs) monitor and guide the traffic toward the runway while providing sufficient separation between flights. During peak traffic periods, as traffic accumulates inside the TMA, the queueing and maneuvering of arrival flights inside TMA at lower altitudes leads to extra fuel consumption [1]. Estimates suggest that terminal area inefficiencies account for approximately 6% of overall fuel consumption [2]. Beyond the extra fuel consumption, the prolonged delay and accumulated traffic within the TMA can overload the air traffic system and subsequently trigger more frequent go-around and flight diversion [3].

As a prominent aviation hub in the Asia-Pacific region, the Hong Kong International Airport (HKG) has faced persistent TMA congestion. The associated extra operating costs and flight safety risks undermine the competitiveness of the air traffic system and local carriers. These challenges necessitate the implementation of operational solutions to mitigate the negative impact of TMA traffic congestion.

B. Decoupled arrival sequencing and flight trajectory solutions

Research has identified ineffective air traffic management (ATM) as a contributing factor to arrival delays [2; 4; 5]. Assisted by computational tools, the ATM decision-making processes are still primarily performed by human operators. Modern ATM tools such as Arrival Manager (AMAN) provide key arrival projections which include estimated landing time, required delay, and landing sequence. Subsequently, ATCOs synthesize arrival trajectories and the associated verbal commands to achieve these projected targets. However, during peak periods, heightened traffic management workload can lead to degradation in control efficiency and trajectory quality. Ultimately, the degraded performance can lead to further arrival delays, increased fuel consumption, and intensified traffic congestion [4]. The decoupled arrival solutions derivation process is insufficient to relieve the workload of ATCOs during peak traffic periods.

Similarly, in studies on arrival solutions, arrival sequencing and arrival trajectory solutions are often examined separately, despite their interdependent nature. The constraints present in each problem can impose limitations on the other. This calls for a simultaneous investigation of arrival sequencing and flight trajectory solutions. Despite the decoupled nature of these studies, their proposed methods and concepts are crucial in the formulation of an integrated arrival solution (IAS).

For arrival sequencing problem, the goal is to arrange the arrival order of the flights that optimizes an objective (e.g., arrival transit time [ATT]) while satisfying a range of constraints. A common challenge in this problem is the computational complexity, in which the exploration of all possible sequences would be computationally costly. To this end, Balakrishnan and Chandran [6] presented an arrival sequencing optimization with constrained position shifting (CPS). Using a first-come-first-serve (FCFS) schedule as the baseline, CPS explores a smaller set of alternate sequences by bounding the allowable shift in landing position. With CPS, a solution better than FCFS, though not necessarily optimal, can be found in a short time. However, this type of decoupled arrival sequencing solution often neglects the interference between flights (e.g., radar separation). A similar work presented by Eun *et al.* [7] acknowledged that the radar separation requirement could cause arrival delay and render the solution infeasible. Airports with arrival restrictions such as airspace limitations can have concentrated arrival routes and limited space for maneuvering. Therefore, air traffic factors should be considered in arrival sequencing or IAS.

For the arrival trajectory problem, the goal is to devise individual flight trajectory that optimizes an objective (e.g., minimizing fuel consumption) while satisfying a range of constraints. Considerations of flight physics, often encapsulated in equations of motion, are often included in the study to ensure flight feasibility of the trajectory [8; 9]. However, trajectory solution is often devised independently without considering air traffic constraints (e.g., radar separation). Wickramasinghe *et al.* [8] pointed out that while the inclusion of traffic constraints in arrival trajectory solution is possible, it can lead to substantial increases in computation time for some methods (in their case, dynamic programming). Furthermore, some traffic constraints can be difficult to quantify and model into constraints in trajectory optimization problems. Examples of such constraints include ATCO workload and traffic control effectiveness. Machine-learning (ML) has emerged as a promising direction for flight trajectory studies. ML offers high computational efficiency and high output accuracy. Trained on flight data, the method can generate realistic and compliant trajectories [10]. Addressing the lack of air traffic factor problem, Zhang *et al.* [11] presented an attention-based reinforcement learning approach for flight trajectory and conflict resolution. Although ML approaches have demonstrated effective and reliable performance, they lack interpretability that is critical in high-stake live decision-making operations [12]. Although the underlying constraints and training data are based on established knowledge, the black-box process remains obscure in nature, thus the output interpretability concern is not fully cleared.

C. Contemporary and emerging methods

The industry—in collaboration with the authorities and research institutions—has been advocating a range of modern computer-assisted solutions to strengthen the capability and enhance the efficiency of the global aviation industry. For arrival operations, notable solutions such as the previously mentioned AMAN, Extended Arrival Manager (EAMAN), Point-merge, and trajectory-based operations (TBO), have demonstrated superior performance during simulations and trials [1; 13]. However, most of them have yet to address the decoupled nature of arrival sequencing and trajectory, as well as the associated ATCO workload which could undermine the effectiveness of these solutions in areas with higher traffic load [14].

Among these solutions, TBO stands out as an approach that can fulfill majority of the requirement. TBO can enhance the arrival operation efficiency by enabling collaborative trajectory adjustments that optimize arrival path for superior fuel efficiency [15]. This approach also provides trajectory solution to the ATCO prior to the flight entry which enables better traffic predictability and alleviate the corresponding workload.

D. Solution fidelity and computation cost

A few notable progress in the arrival solution integration has been made in recent years. These attempts summarized some key challenges in combining arrival sequencing and trajectory solutions. As an example, Sáez *et al.* [16] formulated an arrival solution encompassing arrival sequencing and compatible trajectories. Using mixed integer programming, pre-sequencing, and a grid-based trajectory solution approach, the work demonstrated the feasibility of simultaneous derivation of arrival sequence and trajectory under optimization objective (i.e., minimize ATT). Constrained by the computation cost, the trajectories were processed and expressed in a grid-based format, which might have limited accuracy and feasibility in actual operations. In addition, in high-traffic cases, the method required long computational time. Kamo *et al.* [17] demonstrated an IAS that accounts for weather uncertainty. Using nonlinear programming problem formulation, the method can reduce the computational time substantially. However, the proposed method measures the flight separation only at the path merging point and assumes the flights have sufficient separation prior to the point, which may not be practical in restrictive airspace. Furthermore, the study was limited to a small number of flights due to the high computational cost associated with the increased problem size, which may not adequately represent the complexity of high-traffic environments, limiting the comprehensiveness of the proposed solution.

The interdependence between arrival sequencing and trajectory solutions presents challenges in ensuring compatibility when these components are derived separately. Previous approaches have typically considered a single batch of arrival traffic and assumed the sequencing was performed before the arrival. This approach highly depends on accurate entrance time prediction, which may be impractical in continuous dynamic operations. The solution fidelity and computational cost present challenges in solution feasibility to be implemented in live operation. Achieving solutions with high fidelity, which are necessary to ensure practical feasibility, often requires increased computational resources and processing times. This presents challenges for live operational implementation, where rapid decision-making and real-time responsiveness are critical. The trade-off between solution fidelity and computational efficiency remains a key subject in developing IAS.

In this paper, we propose an IAS with trajectory grafting that derives arrival sequence and trajectory solution simultaneously. The proposed method can provide more efficient arrival solutions and alleviate ATCOs' workload. We aim to improve the solution fidelity and computation performance with a new trajectory generation method—trajectory grafting. The proposed approach is evaluated under realistic traffic conditions and operational constraints.

II. Methodology

To reinforce ATCOs capabilities, a more comprehensive computer-assisted arrival solution that can simultaneously determine the arrival sequencing and trajectory solutions is needed. The primary goal of this study is to integrate arrival sequencing with compatible trajectories within relevant air traffic constraints. Major challenges observed in previous studies are high computational cost, limited traffic load, and limited solution fidelity. In addition, limited consideration of human factors in arrival operations can undermine the solution feasibility in live operations. We propose an IAS with trajectory grafting method to address these challenges. We first discuss the arrival problem formulation in Section II-A, then the method evaluation setup and metrics in Section II-B. The proposed IAS will be elaborated in Section II-E, whereas the trajectory grafting method—for realistic flight trajectory generation—will be described in Section II-D.

A. Problem Formulation

The integrated arrival problem formulation is presented in Table I, which is aimed to find the compatible set of arrival trajectories and sequence that can reduce TMA fuel consumption $\sum F^*$. The landing sequence is indirectly represented by flight landing times $t_Z^{i^*}$ while the corresponding trajectories are denoted as $J_S^{i^*}$, for every flight $i = \{1, ..., n\}$, where n is the total number of flights awaiting to land. This integrated arrival problem is subjected to three major constraints, namely wake separation, radar separation, and airspace restrictions (including altitude, latitude, and longitude bounds that are denoted as h_X, φ_X, and λ_X, respectively).

Wake separation is a mandated safety distance between a leader-follower (indicated by superscripts $f-1$ and f, respectively) flight pair during landing. This distance depends on the aircraft weight category combination of the leader-follower pair. The required wake separation ω is compared to the actual separation γ with a small allowance (ω_b). Radar separation is a mandated safety distance between all airborne flights, applicable throughout the entire flight. The radar separation includes horizontal separation R^H and vertical separation R^V. A trajectory is deemed infeasible only if both horizontal and vertical separation are insufficient simultaneously. Lastly, airspace restriction encompasses all no-fly zones, airspace classes, and arrival procedures requirements within TMA. The arrival solution has to comply with the operation specifications stated in the Aeronautical Information Publications (AIP). In addition, no prior information of the arrival flight is provided to the algorithm until the flight enters the TMA. This constraint disables any arrival pre-sequencing but also reduces reliance on accurate TMA entry time predictions. The TMA entry time is also fixed.

TABLE I. Integrated arrival problem formulation

	Symbol	Description
Reduce	ΣF^*	Total TMA fuel consumption
With respect to	$J_S^{i^*}, \quad i = \{1, ..., n\}$	TMA flight trajectories
	$t_Z^{i^*}, \quad i = \{1, ..., n\}$	Landing time
Subject to	$\gamma^f \geq \omega^{f-1,f} - \omega_b$	Wake separation requirement
	$\{\alpha(\cdot), \beta(\cdot)\} \geq \{R^H, R^V\}$	Radar separation requirement
	$\{\varphi^K, \lambda^K, h^K\} \neq \{\varphi_X, \lambda_X, h_X\}$	Restricted areas

B. Evaluation setup and metrics

The solution is evaluated across multiple flight schedules (with 68 flights each), with at least 80% traffic loading—this corresponds to the maximum arrival load of 34 flights per hour at HKG. The test traffic environment is constructed using the original traffic feed, as the ATT is dependent on the traffic density within the TMA. To isolate the results from the original air traffic management procedure, the simulation begins with a *warm-up* and ends with a *wrap-up* period, with the core test segment in between. The warm-up period provides sufficient air traffic inside the TMA when the core test begins. The core simulation contains 34 flights. During the wrap-up period, new arrival flights continue to be fed into the simulation to maintain the airborne traffic density, where they can still induce changes to the core flights' trajectory. The traffic and method performance during these periods are excluded from the evaluation as their traffic densities are deemed insufficient.

The method performance is evaluated in terms of fuel consumption and computation time. The total fuel consumption of the IAS trajectories is compared to the original fuel consumption, where lower overall fuel consumption is desired. The fuel consumption estimation method is described in Section II-C. For live operation consideration, the overall computation time is also evaluated. In particular, the computation time of each flight is counted individually from flight entry until a solution is found. Both metrics are evaluated based solely on the core test flights, excluding the flights from the warm-up and wrap-up periods.

C. Trajectory data and embedded features

The flight trajectories and arrival schedules are sourced from the Automatic Dependent Surveillance-Broadcast (ADS-B). ADS-B data provide aircraft position (longitude, latitude, and altitude), speed, heading, time, and identification. Multiple flights form a traffic environment. In particular, HKG arrival data from 2018 (during peak periods) are used in this study. In this study, a consecutive 15-day arrival solution set (8,286 trajectories in total) is created to provide the historical trajectory set for trajectory solution derivation. Meanwhile, 33 arrival schedules (with 68 flights each) are selected for the evaluation. Our ADS-B data have uneven time intervals averaged at 60 seconds between messages. For more accurate measurements, the data are linearly interpolated to 1-second interval. The trajectory set is denoted as $J = \{J^1, \cdots, J^n\}$, where n is the total number of trajectories. Each trajectory J^i is formed by a sequence of trajectory points $J^i = \{K^i_1, \cdots, K^i_Z\}$, where i, K, and Z refer to the trajectory index, trajectory point, and the number of recorded points in the trajectory, respectively. Each trajectory point consists of six attributes denoted as $K^i_q = (\varphi^i_q, \lambda^i_q, h^i_q, V^i_q, \theta^i_q, T^i_q)$, where q is the trajectory point index, φ, λ, h, V, θ, and T represent the latitude, longitude, altitude, ground speed, track angle, and time, respectively. To facilitate the trajectory grafting, trajectories are indexed by their trajectory features. Features such as STARs can cluster similar trajectories into groups for faster matching. These feature extractions can be performed before the test.

The key trajectory features used in this study include fuel consumption, STARs, and go-around detection. Fuel consumption is an embedded feature that can be extracted from the flight trajectory. The Base of Aircraft Data (BADA) model is selected to estimate the fuel consumption of each trajectory. BADA is an aircraft performance model developed by EUROCONTROL for simulating aircraft performance under various operational parameters [18]. Using BADA 3.6, each trajectory's fuel consumption is calculated twice, first in their original aircraft type, then in a common aircraft type (B772 is selected for this study). This approach is necessary as fuel consumption is dependent on the aircraft type, and using a common aircraft type allows for a fair comparison of the fuel efficiency of each historical trajectory. STARs are three-dimensional paths inside the TMA that guide arrival flights from the TMA boundary to the airport. The identification of STAR can categorize similar shape trajectories into groups. Flights that used the same path are expected to have similar trajectories and, therefore, could be a potential solution in trajectory grafting. Go-around detection is a feature that indicates if a trajectory involves a go-around maneuver during landing. Go-around is a procedure where the aircraft aborts the landing and regains altitude. Trajectories with go-around usually have longer ATT and more than one landing attempt. In this study, we assume perfect arrival control according to the generated solution, therefore, go-around trajectories are excluded from the solution search.

In addition to the quantifiable features, flight trajectories also have less-quantifiable embedded features, such as ATCO workload and operation procedures. These embedded features constitute the solution's feasibility for live operation. Given the utilization of historical trajectories in trajectory generation, it is assumed that the resulting trajectory inherently contains embedded features including command rate, arrival procedure, aircraft limits, and geographical restrictions. These inherent embedded features are difficult to extract and quantify, yet they are crucial in the solution feasibility. Nevertheless, by using the trajectory grafting method, some of these constraints can be inherently satisfied.

D. Trajectory grafting

Trajectory grafting is a method that creates a new trajectory or adjusts an existing trajectory by combining segments from historical trajectories under operational constraints. The dimension and frequency of the grafted trajectory matches that of the input data, enables better solution fidelity and resolution control. The grafted trajectory also retains operational and qualitative features embedded in the historical trajectory, which contributes to their feasibility and effectiveness, ensures that the resulting trajectories are operationally viable. It is worth noting that information on ATCO workload, which is often excluded in trajectory studies, can be captured implicitly in this trajectory grafting procedure. In essence, trajectory grafting selects the best feasible trajectory segments from the solution set and grafts them together to form a new continual trajectory.

The computational time and solution availability of trajectory grafting are influenced by the quality and size of the solution set (discussed in Section II-C). While a larger solution set can increase general solution availability, the computational time will be prolonged as the solution compilation and exploration become longer. An example of trajectory grafting is illustrated in Fig. 1, which showcases the possible grafting options (scions[1]) from a single point of a trajectory. Larger solution set will increase the number of scions and, consequently, the associated computation time.

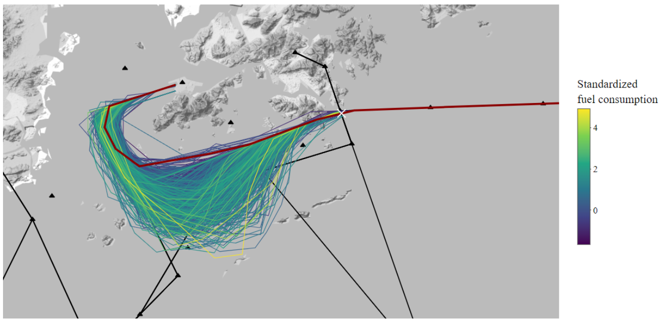


Figure 1. An example of trajectory grafting scion from a single point (marked in white cross). The original trajectory is shown in red, and the scions are colored according to their fuel consumption.

The trajectory grafting process can be summarized in three parts, namely solution aggregation, graft-on, and feasibility test. We denote the target flight as f and the target original trajectory as J^f. In the solution aggregation phase, the algorithm aggregates the historical trajectory points that are spatially and kinematically within a search range r from J^f. The value of r depends on the target flight and speed and is expressed as $r^f_q = \{r^f_{q,\varphi}, r^f_{q,\lambda}, r_h, r^f_{q,V}, r_\theta\}$. $r^f_{q,\varphi}$ and $r^f_{q,\lambda}$ are distance metrics that can be expressed as

$$r^f_{q,\varphi} = r^f_{q,\lambda} = t_s V^f_q, \tag{1}$$

[1] In horticulture, scion refers to the selected upper part of a graft with desired characteristics. In trajectory grafting, scion represents the potential trajectory segment.

where t_s is the search range in time unit and V is as previously defined. While a longer search range can increase the probability of finding graft point, it may reduce the validity of the trajectory. Similarly, the track angle range r_θ is expressed as

$$r_\theta = 3t_s, \tag{2}$$

in which the factor three is selected according to the maximum heading change rate. Points from trajectory solution set J within the search range are aggregated and denoted as $\hat{K}$. These points are then arranged in descending objective value. Each point in $\hat{K}$ serves as a graft point, representing the first trajectory point of a scion, which is denoted as $\overline{K_p^i K_Z^i}, K_p^i \in \hat{K}$. An optional constraint on the scion's estimated time of arrival (ETA) can be added here to retain the landing order.

In the graft-on phase, the scion is grafted onto the targeted trajectory. The spatial difference between the original trajectory and the scion is connected with a linearly interpolated segment $[K_{\text{interp}}]$. Therefore, the grafted trajectory is expressed as

$$\begin{aligned} J^{f^*} &= \left\{K_1^f, \cdots, K_q^f, [K_{\text{interp}}], K_p^i, \cdots, K_Z^i\right\}, \forall i \neq f \\ &= \left\{K_1^{f^*}, \cdots, K_Z^{f^*}\right\}. \end{aligned} \tag{3}$$

In the feasibility test phase, the grafted trajectory is subjected to wake and radar separation tests. The wake separation is measured between the leading flight $f-1$ and the target flight f (the grafted trajectory) when flight $f-1$ lands (i.e., T_Z^{f-1}). The actual separation γ^f is compared with the required wake separation $\omega^{f-1,f}$ minus a tolerance constant ω_b. ω_b is included to compensate for interpolation error. The trajectory passes the test if γ^f is larger. The radar separation test includes horizontal separation $\alpha(\cdot)$ and vertical separation $\beta(\cdot)$. A trajectory fails the test only if both horizontal and vertical separation requirements are simultaneously violated. The test is conducted against all other arrival flights airborne inside the TMA, from entry until landing. The tests are expressed as

$$\alpha\left(K_q^{f^*}, K_p^i \mid T_p^i = T_q^{f^*}\right) = d_h\left(K_q^{f^*}, K_p^i\right); \tag{4}$$

$$\beta\left(K_q^{f^*}, K_p^i \mid T_p^i = T_q^{f^*}\right) = d_v\left(K_q^{f^*}, K_p^i\right), \tag{5}$$

where $\alpha(\cdot)$ and $\beta(\cdot)$ refer to the horizontal and vertical great circle distances $d(\cdot)$ between f and other flights airborne inside TMA, respectively. The distances are calculated at every timestep $T_q, q \in \{1, Z\}$ of J^{f^*}. The position of the flights are matched with the target flight using their timestamps T^i, i.e., $T_p^i = T_q^{f^*}$, in $\alpha(\cdot)$ and $\beta(\cdot)$. The trajectory is rejected if any of the points fail the test. The algorithm repeats the grafting process through the list of possible points $\hat{K}$ until a feasible solution is found. Trajectory grafting serves as the core of the IAS in trajectory generation and trajectory adjustment.

E. Integrated arrival solution with trajectory grafting

IAS with trajectory grafting is a trajectory-driven arrival sequencing method, in which the flight arrival sequence and trajectory solutions are calculated sequentially in the order of their TMA entry. The trajectory solution is derived from historical trajectory solution set through trajectory grafting, the best trajectory solution is selected according to the objective and constraints. In this sequential derivation, a flight's trajectory—from the moment it enters TMA to landing—can be altered to accommodate other arrival flights for better overall objective value. Multiple trajectories together determine an arrival sequence and form the IAS.

IAS can be considered as a method that aligns with trajectory-based operations (TBO). Under IAS, the arrival sequence is trajectory-driven, where the full trajectory is determined upon the flight's entry. The decision-making process is an enhanced collaboration between automation and controllers. As a result, the resulting traffic flow is more predictable than the current human-centric process. Note that while IAS aims to provide a feasible solution, the execution and final decision processes are still delegated to controllers.

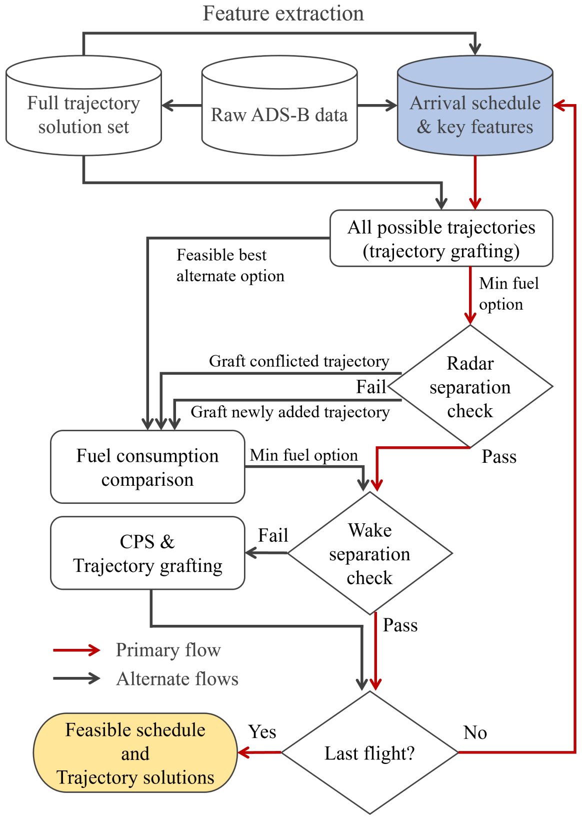

Feature extraction
Full trajectory solution set
Raw ADS-B data
Arrival schedule & key features
All possible trajectories (trajectory grafting)
Feasible best alternate option
Min fuel option
Radar separation check
Graft conflicted trajectory
Fail
Graft newly added trajectory
Pass
Fuel consumption comparison
Min fuel option
CPS & Trajectory grafting
Fail
Wake separation check
Pass
Primary flow
Alternate flows
Feasible schedule and Trajectory solutions
Yes
Last flight?
No

Figure 2. Flow chart of the IAS method.

The mechanism of IAS is illustrated in Fig. 2. Before the arrival operation, the trajectory solution set is derived and key features are extracted (discussed in Section II-C). The workflow begins with the selected arrival schedule (highlighted in blue) extracted from the historical arrival data. An empty simulated TMA is created to host upcoming arrival flights. The simulation advances chronologically, executes IAS calculation at each scheduled flight entry. During a flight entry, a new entry flight f arrives at TMA entrance P. Given the flight's first trajectory point at the TMA boundary as J_1^f, the IAS first searches for all possible trajectories $\hat{J}$ from entrance P with a starting point spatially near J_1^f using trajectory grafting (regardless of radar separation at this stage). The set is then arranged in ascending order of their estimated fuel consumption. Each solution has a new expected landing time, and a new landing leader (in the new arrival sequence). A solution is selected if it satisfies wake separation with the landing leader. The selected minimum fuel option is denoted as $S^f_{\text{min}}(J_{\text{min}}, F_{\text{min}})$ which encompasses trajectory J_{min} and fuel consumption F_{min}. Afterwards, S^f_{min} is subjected to radar separation check $\alpha(\cdot)$ and $\beta(\cdot)$.

If the candidate trajectory fails the test, three options are available. In the first option, the conflicted trajectories are grafted. A dummy simulated traffic environment is created, in which the conflicted trajectories (except the new entry flight f) are grafted sequentially according to their projected landing time. The trajectory grafting has minimum fuel consumption as the grafting objective, and the grafted trajectory must fulfill the wake and radar separations. If any radar-conflicted flight has no feasible solution, this grafting option is disabled. Each graft incurs a fuel change ΔF_C. The total fuel change to graft all the conflicted trajectories is, therefore, denoted as $\sum \Delta F_C$. In the second option, the newly added trajectory (i.e., J_{min}) is grafted with minimum fuel consumption as the objective. The selected solution of this graft must pass both wake and radar separations and therefore would resolve the radar separation conflict. Compared to the initial fuel cost F_{min}, the graft incurs a fuel change $\Delta F_{\text{graft}} = F_{\text{graft}} - F_{\text{min}}$. Lastly, a best alternate option $S^f_{alt}(J_{\text{alt}}, F_{\text{alt}})$ that passed the radar separation check and wake separation (with leading flights only) is retrieved from the set $\hat{J}$. Compared to the initial fuel cost F_{min}, this alternate option has a fuel change $\Delta F_{\text{alt}} = F_{\text{alt}} - F_{\text{min}}$. The fuel changes of these three options are then compared, and the lowest fuel consumption option is selected.

Once a solution is found, the resulting arrival schedule is subjected to the wake separation check. Wake-conflicted flights are subjected to trajectory grafting (ETA unrestricted). After resolving all the conflicts, the newly inserted flight completes an IAS iteration and is sequenced with a feasible trajectory. The IAS process continues with the next flight inserted until all flights are scheduled.

Although the initial trajectory selection (i.e., S^f_{min}) is a greedy selection based on objective value, the flexible adjustment of landing order through trajectory grafting can improve the overall solution quality. By lifting the ETA constraint in trajectory grafting, the algorithm has no restriction on the resulting landing time. Therefore, the landing order can be changed after grafting which is more flexible in finding better fuel efficiency options. In effect, this relaxation enables trajectory grafting to perform the function of CPS to a certain extent. Instead of exploring all possible landing sequences, unrestricted trajectory grafting examines the ”trajectory-compatible” sequences. Leveraging on this feature, arrival priority can be facilitated at the initial trajectory selection. Selection objective can be set to target arrival time or minimum ATT, regardless of the conflict. The radar and wake separation conflicts are resolved through grafting the non-priority flights and set the landing time to be after that of the prioritized flight.

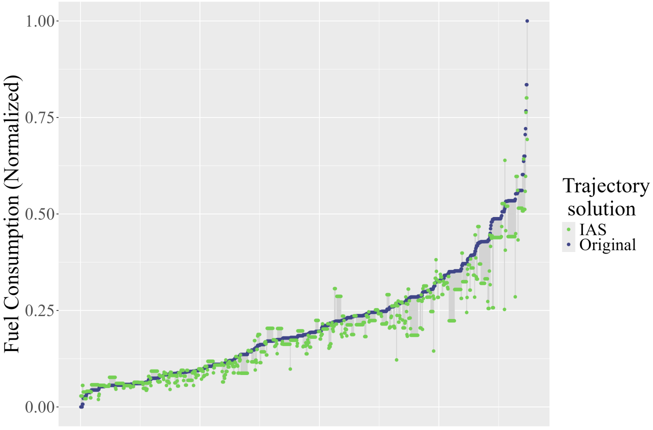


Figure 3. Normalized fuel consumption by flight comparison (in original aircraft type).

III. Result and discussion

A. General performance

In this section, we showcase the general performance of the IAS method with aggregated results from 33 arrival schedules. Note that these results exclude flights from warm-up and wrap-up periods. Only the core 34 flights from each schedule are included. Fig. 3 shows the fuel consumption comparison between the original trajectory (in blue) and the IAS trajectory solution (in green), where more fuel reduction (in IAS trajectory) is observed in trajectories with higher fuel consumption. The IAS achieved an average fuel consumption reduction of 4.84% (calculated based on the original aircraft type). The reduction in fuel consumption can be attributed to the selection of fuel-efficient paths and the reduction in ATT. The fuel-efficient flight path is mainly attributed to the faster arrival speed which can be explained with the fuel model. The BADA total energy model for fuel consumption is expressed as

$$(T - D)V = mg\dot{h} + mV\dot{V}, \tag{6}$$

where T is thrust, D is Drag, V is true airspeed, m is aircraft mass, g is gravitational constant, $\dot{h}$ is vertical speed, and $\dot{V}$ is acceleration. During arrival descent, the change in altitude $\dot{h}$ provides potential energy, thus the aircraft can travel at higher speed without increased thrust or fuel consumption. In contrast, if a flight has to travel at lower speed or level flight with maneuvers (e.g., holding pattern), the fuel consumption increases. This strategy is similar to the continuous descent operations, where the arrival flight descends continuously with minimal engine thrust applied to reduce fuel consumption. This effect is to be exemplified in Section III-B. Furthermore, we observe that the ATT reduction benefits from the cumulative effect, where the latter flights can benefit from the ATT reduction achieved by the previous flights in the arrival sequence.

Fig. 4 illustrates the computation time required per flight in a cumulative distribution function (CDF). The average total computation time required for 34 core flights is 1.71 hours, and the median for individual flights is 111.1 seconds. The step at near 0.25 probability is due to trajectory grafting. The left segment with computation time below 10 seconds represents flights that do not incur any conflict, thus no trajectory grafting is required. The segment right of the step represents flights that incur one or more conflicts to be resolved with trajectory grafting.

While the computational challenge still exists, our approach presents improvement over previous attempts, particularly at this level of solution resolution and fidelity. The primary driver of the calculation time is the sequential trajectory exploration structure. In the IAS trajectory options comparison, the options are derived sequentially, which can require a long calculation time. In each trajectory grafting, the solution trials are conducted in a loop structure which can also bear a long calculation time if the simulated TMA has high traffic density (lower radar separation). Reduced resolution can alleviate computational cost but might also reduce solution feasibility, further investigation is required to determine the impact. Notably, our method generates trajectories by segments or in full length, rather than through point-by-point generation, which has the potential to reduce computational costs, particularly for longer-range trajectories when the method is fully developed.

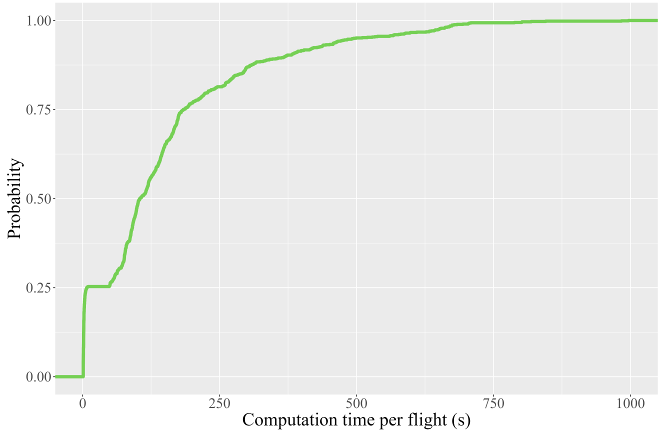
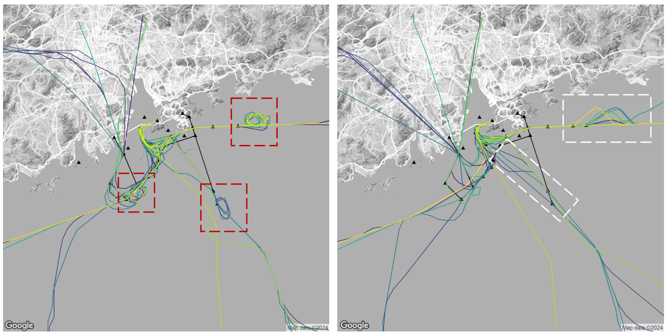


Figure 4. Core schedule computation time per flight cumulative distribution function. All calculations are conducted with AMD Ryzen Threadripper 3990X, 2.9GHz

B. Case study

In this section, we investigate changes in trajectory from the original to the IAS. A schedule with higher fuel consumption reduction is selected for having more noticeable fuel-reduction features.

Fig. 5 shows the original and IAS trajectories, where the trajectories are color coded by flight ID for clarity. The differences can be summarized in three aspects, namely arrival pathing, traffic flow guidance strategy (TFGS[2]), and arrival speed. The arrival pathing of the IAS trajectories is more direct (short-track) and less uniform than the original. The original trajectories are more aligned to the STARs (black lines) which require less attention and guidance in managing. For the use of TFGS, holding patterns (circular trajectory) are often used in the original trajectories while the IAS trajectories have more vectoring. Holding patterns are less flexible than vectoring in terms of arrival time control as it has a fixed holding duration. Although the holding pattern is less flexible, it requires less guidance from the ATCO as the holding location and maneuvers are specified in the AIP. Lastly, the difference in cumulative speed per second distribution inside the TMA is shown in Fig. 6. The IAS trajectories have higher speed at most speed segments. The higher speed arrival trajectories contribute to the lower fuel consumption as discussed in Section III-A.



(a) Original arrival trajectories. Holding patterns are indicated with red dotted boxes.

(b) IAS arrival trajectories. Vectoring are indicated with white dotted boxes.

Figure 5. Arrival trajectories comparison.

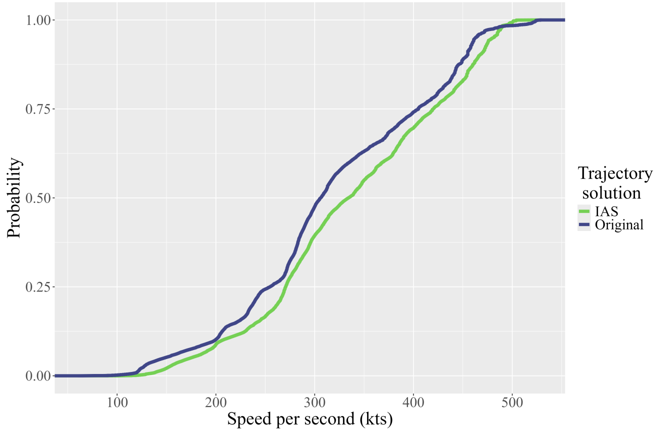


Figure 6. Flight speed cumulative distribution function comparison.

C. Limitations

Based on our results, we observe that fuel consumption feature alone is not sufficient for effective trajectory selection in IAS, for reasons described below. Trajectory with lower fuel consumption does not necessarily have shorter ATT. Sequential greedy selection of minimum fuel trajectory solution can lead

[2] TFGS includes air traffic control techniques used to adjust flight arrival time, such as holding pattern, vectoring, and short-track.

to occasional excessive wake separation. Accumulated excessive separations reduce runway utilization, increase arrival delays, and elevate fuel consumption at a later stage of the arrival schedule.

Note that this IAS structure serves as an improvement upon the current arrival operation but it does not necessarily provide the optimal solution. Optimal arrival solution with both sequencing and trajectories at this level of fidelity would require a substantial computational resources and time (at least in the case of IAS with trajectory grafting). Given the time-sensitive nature of ATM operations, long computational time in solution derivation is undesirable. To this end, the IAS solution quality can be improved with pre-sequencing processes. Assuming that the arrival system has accurate TMA entry time prediction for every arrival flight, the arrival sequence can be arranged based on historical ATT and traffic constraints (e.g., wake separation). Once the optimal sequence is established, compatible trajectory solutions can be derived to fulfill the sequence. Additional adjustments to the sequence and trajectories can be made with trajectory grafting. This pre-sequencing approach has the advantage in solution optimality, runway utilization, and arrival delay.

IV. Conclusion

In this paper, we proposed a new strategy of integrated arrival solution with trajectory grafting. Leveraging historical best practices, the trajectory grafting method generates new flight trajectories by combining historical trajectory segments. This method can produce feature-embedded high-fidelity trajectory solutions. While the high computational cost still renders live implementation impractical, our approach is still more computationally efficient compared to other solutions with lower level of fidelity and resolution. Lifting the landing time constraints on trajectory grafting, it can function as constrained position shifting in re-sequencing the arrival order. This trajectory-driven sequencing framework is more resilient to inaccurate entry time and more suitable for continuous operation. The test results showed that our method could reduce the average TMA fuel consumption by 4.84%. The fuel reduction was attributed to the selection of fuel-efficient trajectories, accumulated reduction in arrival transit time, and faster arrival speed. The IAS trajectories travel through the TMA at higher speeds along more direct paths, which can reduce fuel consumption. In sequential trajectory derivation, the latter trajectories can benefit from accumulated gain from previous flights, thereby maintaining or even enhancing the improvements. The tests also highlighted the challenges in computational efficiency, solution optimality, and practicality concerns, which calls for further work on sequencing strategy and objective selection. The integrated arrival solution is compatible to the existing air traffic management system and could be implemented without major modification or addition of infrastructure.

References

[1] L. Z. Jun, S. Alam, I. Dhief, and M. Schultz, “Towards a greener extended-arrival manager in air traffic control: A heuristic approach for dynamic speed control using machine-learned delay prediction model,” *Journal of Air Transport Management*, vol. 103, p. 102250, 2022.

[2] M. S. Ryerson, M. Hansen, and J. Bonn, “Time to burn: Flight delay, terminal efficiency, and fuel consumption in the national airspace system,” *Transportation Research Part A: Policy and Practice*, vol. 69, pp. 286–298, 2014.

[3] L. Dai, Y. Liu, and M. Hansen, “Modeling go-around occurrence using principal component logistic regression,” *Transportation Research Part C: Emerging Technologies*, vol. 129, p. 103262, 2021.

[4] H.-H. Yang, Y.-H. Chang, and Y.-H. Chou, “Subjective measures of communication errors between pilots and air traffic controllers,” *Journal of Air Transport Management*, vol. 112, p. 102461, 2023.

[5] R. Louie, G. N. Lui, T. S. Tai, and R. P. Liem, “Data-driven analysis of inefficient arrival separation,” in *AIAA AVIATION 2023 Forum*, 2023, p. 3258.

[6] H. Balakrishnan and B. Chandran, “Scheduling aircraft landings under constrained position shifting,” in *AIAA guidance, navigation, and control conference and exhibit*, 2006, p. 6320.

[7] Y. Eun, I. Hwang, and H. Bang, “Optimal arrival flight sequencing and scheduling using discrete airborne delays,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 359–373, 2010.

[8] N. K. Wickramasinghe, A. Harada, and Y. Miyazawa, “Flight trajectory optimization for an efficient air transportation system,” in *28th International Congress of the Aeronautical Science (ICAS 2012)*, 2012, pp. 4399–4410.

[9] D. Kim and R. P. Liem, “Population-aware sequential flight path optimization for low-noise and low-fuel consumption departure trajectory,” *AIAA Journal*, vol. 60, no. 11, pp. 6116–6132, 2022.

[10] G. Jarry, N. Couellan, and D. Delahaye, “On the use of generative adversarial networks for aircraft trajectory generation and atypical approach detection,” in *Air Traffic Management and Systems IV: Selected Papers of the 6th ENRI International Workshop on ATM/CNS (EIWAC2019) 6*. Springer, 2021, pp. 227–243.

[11] M. Zhang, C. Yan, W. Dai, X. Xiang, and K. H. Low, “Tactical conflict resolution in urban airspace for unmanned aerial vehicles operations using attention-based deep reinforcement learning,” *Green Energy and Intelligent Transportation*, vol. 2, no. 4, p. 100107, 2023.

[12] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature machine intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[13] D. Ivanescu, C. Shaw, C. Tamvaclis, and T. Kettunen, “Models of air traffic merging techniques: evaluating performance of point merge,” in *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS)*, 2009, p. 7013.

[14] H. Hardell, A. Lemetti, T. Polishchuk, L. Smetanová, and K. Zeghal, “Towards a comprehensive characterization of the arrival operations in the terminal area,” in *11th SESAR Innovation Days*, 2021.

[15] ICAO, “PERFORMANCE FRAMEWORK TO ASSESS TRAJECTORY BASED OPERATIONS (TBO) CONCEPT,” 2022. [Online]. Available: https://www.icao.int/Meetings/a41/Documents/WP/wp_131_en.pdf

[16] R. Sáez, T. Polishchuk, C. Schmidt, H. Hardell, L. Smetanová, V. Polishchuk, and X. Prats, “Automated sequencing and merging with dynamic aircraft arrival routes and speed management for continuous descent operations,” *Transportation Research Part C: Emerging Technologies*, vol. 132, p. 103402, 2021.

[17] S. Kamo, J. Rosenow, H. Fricke, and M. Soler, “Robust optimization integrating aircraft trajectory and sequence under weather forecast uncertainty,” *Transportation Research Part C: Emerging Technologies*, vol. 152, p. 104187, 2023.

[18] A. Nuic, D. Poles, and V. Mouillet, “BADA: An advanced aircraft performance model for present and future ATM systems,” *International journal of adaptive control and signal processing*, vol. 24, no. 10, pp. 850–866, 2010.