Point merge system using decentralized control

Yuan Gao[1, *], Arthur Richards[1]

[1] *Department of Aerospace Engineering, University of Bristol, Bristol, BS8 1QU, United Kingdom*

* *Corresponding author, e-mail address: yuan21.gao@bristol.ac.uk*

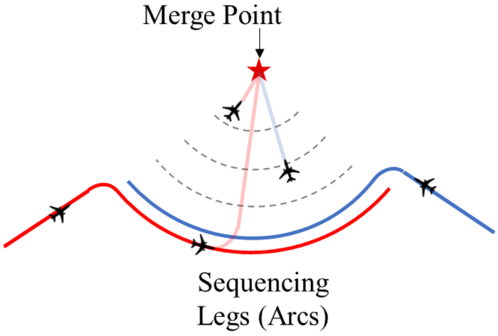
Abstract

Point Merge is a systemised method for sequencing arrival flows of aircraft. A decentralized control method is proposed for the point merge system, which uses message sending schemes to replace centralized controller. Based on the Self-organizing Time Division Multiple Access (STDMA), three different decentralized schemes are proposed, under which all the arriving aircraft should send messages to each other to achieve turn sequencing. Finally, a dynamic scenario with two sequencing legs is used to compare these three schemes. Both sequencing leg capacity and operational results of scheme operation are investigated. It is found that a decentralized scheme using contiguous reservation and a network entry slot can achieve similar capacity to centralized control.

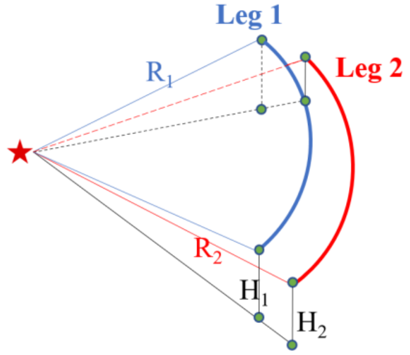
Keywords: Point Merge, Multi-agent system, Decentralized control.

1. Introduction

The past three decades have witnessed a big growth in the global air traffic demand. Since the COVID-19 pandemic, there has been a sharp reduction in the number of daily flights. According to the prediction of the International Civil Aviation Organization (ICAO) in September 2022, most international flights should fully recover to the pre-pandemic levels by either the end of 2022 or early 2023; However, the route groups of East and North Asia could be slower, probably later in 2023 [1]. To address the existed imbalance between the limited capacity and the increasing demand, many projects and studies have been carried out to manage the arrivals in the Terminal Manoeuvring Area (TMA). Among them, Point Merge (PM) was introduced at the EUROCONTROL Experiment Centre to improve the traditional trombone-shaped vectoring patterns in terms of predictability, flight efficiency and environmental impact [2]. PM is a systemised method for sequencing airplane arrivals in the TMA, based on a novel design of Standard Terminal Arrival Route (STAR) [2, 3]. As shown in Fig.1, PM system generally consists of a merge point and the pre-defined sequencing legs (quasi arcs) equidistant from this merge point. The aircraft sequencing in PM system should be achieved with a single "direct-to" instruction from the air traffic controller at the appropriate time [3-5]. PM is an operating method which can rely on the existing technology onboard aircraft and requires no new specific ground tool [2, 5]. Started with the implementation in Oslo (2011), PM has now been applied for 38 airports through 19 countries and 4 continents, such as Dublin (2012), Paris (2013), Moscow (2018), Shanghai (2019), and Mexico City (2021) [3].



(a) Plain view



(b) 3D view

Fig. 1. Typical configuration of Point Merge system

Fast-time simulations has been used to compare PM with the conventional vectoring method using the metrics of mean controller task load, number of instructions to pilots, and fuel consumption [6]. Other simulations have demonstrated how PM can be adapted to typical terminal area configurations with benefits in terms of staffing, predictability, and continuous descent [7]. PM for converging runways has been studied via real-time simulations [4] and compared with vectoring, taking the Istanbul International Ataturk Airport as an example. An optimization framework of PM-based autonomous system has been proposed [8] for busy airports such as Beijing Capital International Airport (BCIA). All of these studies used centralized control for the PM system.

Generally, the related works can be divided into two aspects: 1). Optimization study; 2). Modelling and Simulation (M&S) study. Regarding the first aspect, Mixed Integer Linear Programming (MILP) is found to be the most popular choice [9-13]. Apart from that, heuristic algorithms, such as PSO [14, 15] and Genetic Algorithm (GA) [16], were also tried to deal with the optimization problems of PM system. For the other aspect, nearly all the studies of PM M&S have focused on the numerical simulations to evaluate the PM performance, which may simplify the models or ignore the details of aircraft dynamics, communication, human factors, and the interactions between controller and aircraft.

The primary motivation for this new research is that, to the best knowledge we have, there is no previous study which established the decentralized control for PM. To optimise the arrival sequence, a PM-based autonomous arrival manage system was studied in [8, 24] by using simplified numerical models, but they are based on numerical simulations only, and there is no communication considered between the aircraft. In contrast, this paper explores the possibility of building the decentralized control schemes in a PM system. Decentralized control avoids the single point of failure associated with centralized control and is typically more scalable. However, it demands more coordination between agents, especially in safety-critical resource sharing like PM. This paper explores the performance trade-offs introduced.

Few studies have modelled the PM system by using the state/behaviour-based methods, such as Finite State Machine (FSM), and behaviour tree. This is a secondary motivation of the works performed in this paper. FSMs have been used and developed in many different fields, such as computer science [17], transportation [18, 19], infrastructures resilience [20], operational research [21, 22], war gaming [23], etc. One of main reasons for using FSM here is that the transient state of every airplane (as well as the controller) in the PM system can be easily and correctly obtained during the offline or real-time simulations. This fact makes it straightforward and easy when debugging the PM system or finding out the most important state/behaviour for the system-level evaluation metrics. Another important reason of using FSM is that the uncertainty and human factors (even completely new agents) can be easily added with FSM due to its flexibility of modelling multiple states and hierarchical structures.

This paper is organized as follows: the fundamentals of PM and FSM-based agent modelling will be reviewed in the next section. After that, the agent-based PM system with a centralized controller will be discussed in Section 3. In addition, to achieve the decentralized control for PM system, three message sending schemes are proposed in Section 4, which can remove the above centralized controller while not affecting the desired sequence control. And, in Section 5, a dynamic scenario is further given to compare these schemes. Finally, conclusions are drawn in Section 6.

2. Point Merge and Agent-based Modelling Fundamentals

2.1. Point Merge system

As mentioned, a PM system consists of a merge point and pre-defined sequencing legs. These legs are designed as arcs, which should be parallel and vertically separated. Fig. 1 shows a typical configuration of PM with two legs (opposite directions) using the plan and 3D views. In this configuration, the aircraft from two legs are traditionally managed by terminal controllers for turning to the merge point. This turning sequence is achieved through a single 'Direct-To' instruction sent to each aircraft, as soon as the preceding aircraft enters an internal area (i.e., the required spacing from the preceding one is obtained) [2].

In this study, aircraft are assumed to fly at a specified speed, no matter before/on the leg or after turning. Regarding the required safety spacing, the whole PM area could be divided into four parts, as separated by three dotted arcs in Fig. 1(a). For simplicity, this paper only uses a half the arc radius to define the internal area, which means the arc radius is double the minimum aircraft separation.

In practice, the distance between sequence legs and the merge point can be designed with scalability, because PM offers a scalable design allowing procedure designers to reflect local needs [2]. This is a trade-off between environmental/fuel efficiency and PM system capacity. Therefore, different geometry settings were used in the literature. For example, in [4], the length of the sequencing legs (L_{leg}) is set as 36 Nautical Miles (NMs), the distance (radius) between legs and the merge point for the inner leg (R_1) is given 21 NM and for outer leg (R_2)

23 NM. Differently, in [6], L_{leg} is 30 NM for the outer sequencing leg while 27.5 NM for the inner, and R_1 is given as 26 NM. To ensure that the aircraft in two legs would have the same safety spacing for turning, in this paper, R_1 is assumed to be equal to R_2. H_1 is 8 NMs, H_2 is 7 NMs, and the radius of Leg 1 in plain view is 30 NMs. Therefore, R_1 and R_2 are both 31.0483 NMs in 3D, and radius of the internal area is 15.524 NMs.

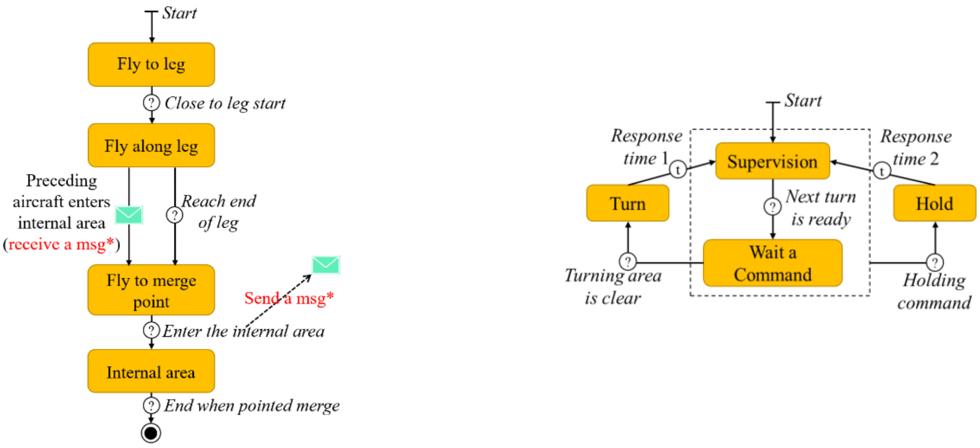
2.2. FSM and agent modelling

An FSM (also called a finite state automaton) is a computational model that can be implemented with hardware or software, based on a finite number of states, potential transitions between the states, and causes (e.g., events, actions, conditions, processes) inducing a transition from one state to another [20, 25]. In computer science, FSMs have been used in behaviour modelling, sequential circuits, software engineering, communication protocols, and the objective-oriented systems for decades [17, 26].

More specifically, this study uses FSM on the Anylogic platform, which supports agent-based, discrete event, and system dynamics simulation methods [27]. Three methods can work independently and/or in a hybrid approach. The Agent-Based Modelling (ABM) [28] with FSM is adopted, which is based on the State-chart entry point, one or more states, the final state, and the transitions to link these states [29]. In addition, system dynamics (with the Aircraft Agent) is also used here to model the flight of aircraft.

3. Point Merge system with a centralized controller

In this paper, a controller-based (centralized) PM system is first studied, which have three types of agent in the system: Main Agent, Aircraft Agent, and Controller Agent. The Main Agent is the top-level agent where mission scenario is modelled, for example, simulation environment, leg path, set parameters/variables, add agents, etc. The other two agents should be embedded into the Main Agent to join the entire simulation. The model diagrams of Aircraft Agent and Controller Agent are depicted in Fig. 2.



(a) Aircraft Agent (b) Controller Agent

Fig. 2. FSM-based agent models

The Aircraft Agent is the key part in PM system. Generally, this agent is to model all the potential states of aircraft in the PM system. As shown in Fig. 2(a), in this study, there are four states modelled in this agent: a) Fly to leg, b) Fly along leg, c) Fly to merge point, d) Internal area, which explicitly represent four stages of an airplane during the simulation. The message sending and receiving should be crucial. As marked in red, when the aircraft receives a turning message (msg) or "Direct-to" command in the State "Fly along the leg", it can turn to the merge point, which is a normal turn in PM system. If it reaches the end of sequencing leg, it also goes to the "Fly to merge point" State but it may cause a loss-of-separation issue. Therefore, this is not a safe situation, and the PM system is then regarded to be over-capacity. In this study, the frequency of aircraft reaching the end of the leg, and hence losing separation downstream, will be the primary metric for the evaluation in Section 5.

As shown in Fig. 2(b), there are four states and two commands considered in the Controller Agent. Obviously, the "Turn" State is making a turning command for the next aircraft, and the "Hold" is making a holding command when there is a safety issue. Since holding command is more emergency than the turning, a high-level state is made here to include two states: "Supervision" and "Wait a Command". That means, no matter in which state, the Controller Agent will go to the "Hold" State when a safety issue happens. After making a command, there is a response time delay (manually set) before entering the "Supervision" State.

In the Controller Agent, following the First-Come-First-Serve (FCFS) rule, there is a function to monitor all the waiting arrivals and determine which plane is the next to turn. The most important part of this system is when and how the aircraft agents turn to the merge point. In the following contents, decentralized control is introduced into the PM system which can determines the turning sequence but without a controller. The decentralized control is achieved by the communication (sending message) schemes between the aircraft. The centralized control will be a benchmark for the evaluation of different communication schemes, see Section 5.

4. Decentralised control for Point Merge system

This section will explore the possibility of establishing the PM system in a decentralized way. More specifically, all the arriving aircraft need to send messages to each other and determine the turning sequence by themselves. In this way, the previous centralized controller agent can be removed from the PM system partly or fully in different scenarios.

The proposed decentralized approach for PM is inspired by and based on the Self-organizing Time Division Multiple Access (STDMA), which is an alternative channel access protocol for communications [30]. Generally, STDMA is a channel access technique using time frames and slots to achieve a distributed slot reservation process for all the joining nodes (stations). The nodes are using the same frame/slot duration, but the start/end time is different due to the various random entry. In the following sections, the principles of STDMA will be first discussed, and then the proposed three decentralized schemes will be introduced.

4.1. Principles of STDMA

STDMA is a decentralized medium access control method. All the joining members are sharing the same channel for communication. Time is divided into frames, which consist of a few time slots. In each frame, every station/node will randomly select several free slots to transmit their messages.

Principles of STDMA are summarized as: After joining the channel, in the *initialization* phase, an agent listens to the channel for a complete frame, without sending anything. In this phase, the current slot allocation status can be learned by the joining agent. In the following Network Entry (NE) phase, the station randomly accesses a free slot to announce its presence and its first slot to be reserved. After the NE slot, the *first frame* phase immediately begins; during that, the further slot reservations should be announced by the joining agent. The *continuous* phase will operate the same as the first frame, where all the slot reservations should be transmitted before starting a new frame.

In a communications setting, transmitting stations can use more than one slot per frame. However, for decentralized PM, each agent reserves only one slot per frame in the continuous phase. Since STDMA develops a shared understanding of allocation of slots to transmitting aircraft, it also develops a consistent ordering of aircraft. Then, each aircraft can know its predecessor's identity, and knows it is time to turn towards the merge point when its predecessor signals it is safe to do so. This paves a way for the proposed decentralized control of PM system.

In the remainder of this section, three variants of STDMA will be outlined. The first is close to the original implementation of STDMA for communications. The second and third variants add extra effort to improve slot ordering. In a communication setting, slots and frames are very short in duration, and ordering has little impact on aggregate performance. However, for point merge, poor ordering could lead to excess distance flown or even loss of separation, so extra effort on ordering is desirable.

4.2. Proposed three schemes for decentralized control

Three different control schemes are proposed here based on the principles of STDMA. After reservation, each plane only sends a message once per frame. In addition, the initialization phase continues to be the listening period to confirm the current reservation situation but, after that, the network entry is an option for the proposed schemes (no additional phase). Regarding the turning sequence for the arrivals, in all three schemes, the aircraft in the same channel would turn in the order of reserved slots, but the slot reservation approach varies.

The names of three control schemes are: the arbitrary reservation with network entry, the contiguous sequence with direct reservation, and the contiguous sequence with network entry. They are short as the Arbitrary Reservation, the Direct Reservation, and the Network Entry scheme, respectively. The slot reservation is randomly determined for the Arbitrary Reservation but contiguous for the other two, which result in first-come, first-served (FCFS) ordering. The diagrams of three schemes are illustrated in Fig. 3, using simple reservation cases.

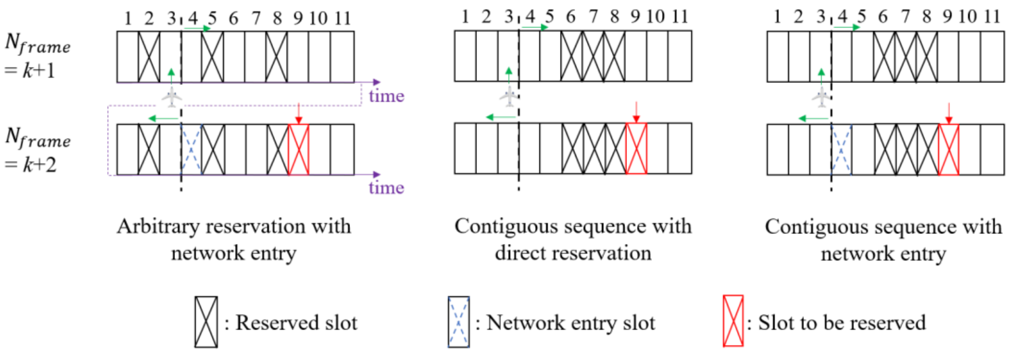


Fig. 3. Proposed three schemes for the decentralized control

4.2.1. Arbitrary Reservation

As shown on the left of Fig. 3, the Arbitrary Reservation scheme randomly chooses a slot to reserve, from all the free ones. The green arrows indicate when a new aircraft first tries to join the sequence. It begins by listening for one whole frame, in this case, 11 slots, and learns that slots 2, 5, 8 are already taken by receiving reservation messages from other aircraft. Then, this aircraft randomly chooses Slot 9 to reserve. A NE slot (Slot 4) is used to announce its presence. Even for the situation when the first slot after listening is not free, the closest free slot will be chosen as the entry slot. This entry slot may not be the to-be-reserved slot because the slots are not reserved contiguously.

In this way, the slot reservation can be flexible. The NE slot is used at the closest slot (after the listening phase). This scheme has more chance than the other two schemes to announce the presence before the to-be-reserved slot due to the arbitrary reservation.

However, the disadvantage of this scheme is from the determination of the sequence. Slots are randomly reserved, and the aircraft will just follow the reserved sequence to turn (the smaller slot number, the higher priority). In this situation, some aircraft who reserved a large slot number can be delayed much for turning, due to the jump in of other aircraft with smaller slot numbers.

4.2.2. Direct Reservation

As shown in the middle of Figure 3, the green arrows indicate the listening frame of a new aircraft who first tries to join the sequence. After the listening phase, it learns that slots 6, 7, 8 are already taken. This scheme is using the contiguous sequence for slot reservation. Therefore, this airplane will reserve Slot 9 and, after listening, it should wait for 5 slots to directly announce its presence and reservation.

An obvious drawback of Direct Reservation is that, when the slot to be reserved is far away from the first slot, after the listening frame, the joining aircraft needs to wait for a long time before other aircraft become aware of it. The waiting time might even be close to the length of one frame.

4.2.3. Network Entry

Compared with Direct Reservation, the only difference of Network Entry (NE) scheme is that it adds a NE slot after the listening frame, announcing in advance the intent to reserve a later slot in the frame. As shown on the right of Fig. 3, it uses the first available free slot, in this case 4, to announce its presence to reserve Slot 9. Note that if the nearest available free slot is 9, this Scheme is identical to Direct Reservation.

Therefore, in this Scheme, the airplane may not need to wait a long time to announce the reservation when the slot to be reserved is far. That problem would be more likely to happen if there were a lot of reserved slots during the listening frame. However, as mentioned, if the first slot after listening is not free, this Network Entry scheme would be the same with the Direct Reservation.

4.3 Message sending and re-slot after collision

A "collision" refers here to the condition when two aircraft both try to reserve the same slot, having been unaware of each other doing overlapping listening phases. In the event of a collision, we assume that aircraft still receive all messages sent: i.e., that there is some underlying messaging system that delivers all messages within the duration of the slot. We further assume that every message includes the "joining time", i.e., when an aircraft first started listening. Hence the aircraft with the lowest joining time is taken to have secured the reserved slot. Other aircraft in the collision now follow a process with their updated knowledge of the allocations and aircraft.

Each aircraft in a collision should send a "ReSlot" message to all others, together with its joining time. When an aircraft in collision receives this kind of message, the joining time needs to be compared with its local joining time. After the comparison, the updated slot number should be broadcast globally. Fig. 5 shows the message sending in a collision example with two aircraft.

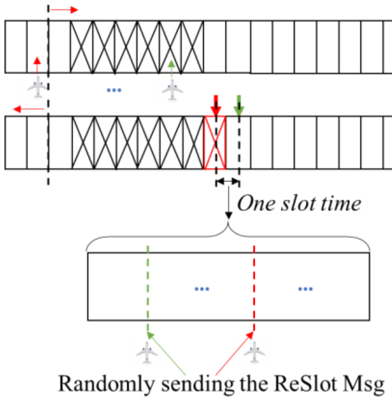


Fig. 5. Diagram for addressing the collision.

Therefore, the general rule of addressing collision is to compare the joining time of aircraft and re-allocate the slots for them. In this way, the collision can be addressed in a short time thus no risk to delay the slot reservation.

4.4. FSM modelling for decentralized control

To implement the message sending between aircraft, an STDMA agent is proposed in this section. It is a separate agent from the above flight agent; However, this STDMA agent should be embedded into the flight agent explicitly for the communication (receiving and sending messages), and it will be turned on when the airplane enters the communication area (near airport).

Fig. 4 shows the diagram of STDMA agent for the decentralized control. Generally, there are six states: Listening, NetworkEntry, Wait, ReSlot, Reserved, and Leave. Regarding the design of transitions, there are two types used: one is condition-based, the other is remaining time-based (marked by red rectangle).

The agent starts with the Listening state, which is followed by the first reservation announcement state, NetworkEntry. Before entering this entry state, as marked in "Transition 1" of Fig. 4, this agent needs to select the NE slot and the reserved slot from all the free slots. Then, when it is the time for the announcement, the airplane needs to send the entry message with the slot number they want to reserve (during the "Transition 2"). Obviously, three schemes have different rules for the slot number choosing and when to send this entry message but, the general agent design and the following collision operation are very similar.

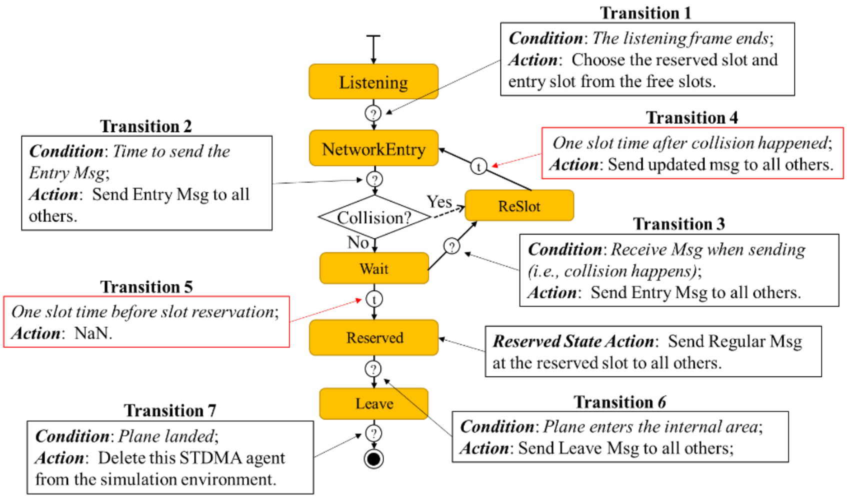


Fig. 4. FSM-based agent for decentralized control (STDMA agent)

After the collision and possible ReSlot state (see the next subsection), the agent goes to the Reserved state assuming that the above messages have been sent successfully to all the other aircraft in communication area. In that way, the plane just needs to send the message at this reserved slot regularly in every framework, as marked in the "*Reserved State Action*" of Fig. 4.

The final state would be the Leave. When the plane enters the internal PM area, the "Transition 6" will be triggered, and the agent goes to the Leave state. It needs to send the Leave message to others in "Transition 6", especially for the next turning. Noting that there is no Turning state in STDMA agent because it focuses on the slot reservation and message sending. Only after the slot is successfully reserved, the plane knows the identity of its predecessor, it turns when its predecessor sends "NextTurn" message. However, when turning, the state in STDMA agent is unchanged because it is still in the regular Reserved state (until leaving the communication channel).

Therefore, all three schemes can basically use this agent design but the detailed functions in each state vary. For example, after Listening, the slot selection (entry and reserve) would be the same for Direct Reservation and Network Entry, but the Arbitrary Reservation is different; And the slot to send the entry message would be the latest available slot for Arbitrary Reservation and Network Entry, but Direct Reservation is directly using the slot to be reserved, no NE slot added. The comparison of three schemes is summarised in Table I.

Table I. Comparison of three decentralized schemes

Scheme	Includes NE slot?	Contiguous reservation?	Jump-in situation	"ReSlot" message sending after "Collision" triggered
Arbitrary Reservation	Yes	No	Possible	Not necessary when aircraft in collision reserved different slot
Direct Reservation	No	Yes	No	All necessary if "Collision" triggered
Network Entry	Yes	Yes	No	All necessary if "Collision" triggered

5. Simulation of the decentralized control

5.1. Scenario and parameters

A dynamic scenario with two sequencing legs is given for the evaluation of the three schemes. As shown in Fig. 6, the total simulation time for each design point is always 6 hours. In every case, 30 aircraft are approaching Leg 1 along a fixed, straight arrival route, spread at time interval T_{Gap}. For Leg 2, another 30 aircraft are approaching along a separate route, also spread at interval T_{Gap}, but delayed with respect to those on Leg 1 by time $T_{leg2,S}$. At low value of the offset delay $T_{leg2,S}$, the two streams overlap, and we expect aircraft to be delayed on the sequencing legs to maintain downstream separation. The goal of the PM system is to absorb peaks in aircraft flow, and by reducing $T_{leg2,S}$, the system is subjected to a sharper peak as the two arriving flows overlap for longer.

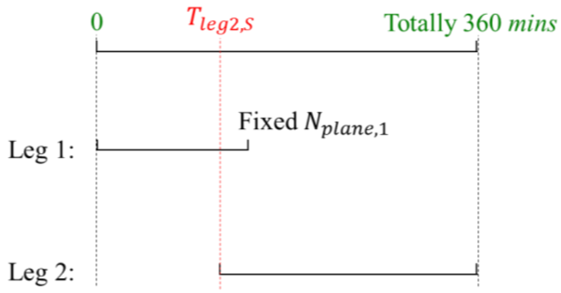


Fig. 6. A dynamic scenario for the case study

The key parameters consider in this case include the Leg 2 starting time $T_{leg2,S}$, plane spacing time T_{Gap}, onboard distance from the leg start point $S_{onboard}$. Other important design parameters are the number of slots in one STDMA frame N_{slot}, STDMA slot duration T_{slot} and the distance $S_{onboard}$ from the leg start point at which aircraft join the STDMA system.

The geometry parameters of PM system have been discussed in Section 2.1. Regarding the onboard distance $S_{onboard}$, it should be well selected. The reason is that if a small value is chosen, the leg usage could be affected too much by the Listening phase; if a big value is selected, the slots can be always successfully reserved before reaching the leg start thus, the leg usage would be quite similar for all three schemes. After trial-and-error, $S_{onboard}$ is chosen as 30 *km* (16.1987 NMs).

For simplicity, T_{slot} is fixed as 12*s*, corresponding to a medium message-sending service quality as defined by [31]. However, the slot number in one framework N_{slot} will be studied as a design variable (see Section 5.3).

There are four metrics considered for the evaluation of proposed three schemes (in the simulation of 6 hours):

1) Leg finish frequency (%), the number of aircraft turning on reaching the end of the leg, divided by the total aircraft number. A non-zero value indicates going over capacity.
2) Average leg usage (*km*): For every plane, the leg usage counts from the leg start to the turning point, maximum the whole leg. Lower leg usage is better, implying less excess flight distance.
3) Average message sending number, which equals all message number divided by the total aircraft number.
4) Average slot reservation time: For every plane, it counts from the end of "Listening" state and the beginning of "Reserved" state in the STDMA agent (see Fig. 4).

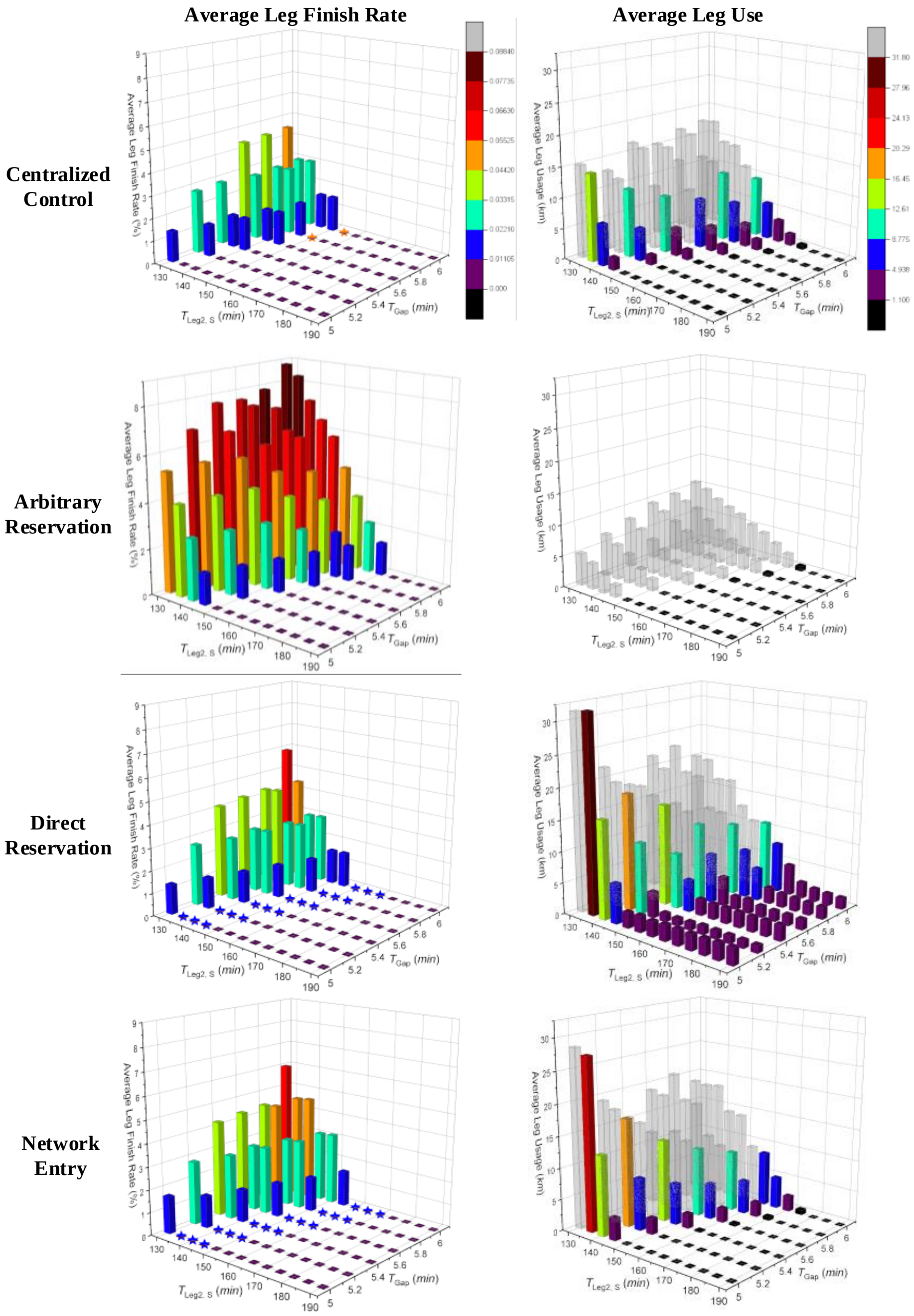


Fig. 7. Peak traffic results of sequencing leg capacity

5.2. Simulation results of peak traffic capacity

Fig. 7 shows the leg finish frequency (left column) and leg usage (right column) results based on two variables, $T_{leg2,S}$ and T_{Gap} for the centralized scheme and three decentralized schemes. The third variable N_{slot} is fixed at 15 for all of these simulations. For any control scheme, if the leg finish frequency is non-zero, the leg use results would be incomparable due to the loss of separation. Therefore, leg usage bars are marked in grey for cases with non-zero leg finish frequency. The other leg use results are coloured according to the colour bar.

Looking first at the leg finish frequencies, in all schemes, as the offset delay $T_{leg2,S}$ decreases or the aircraft spacing T_{Gap} increases beyond a certain boundary, aircraft begin to reach the end of the legs due to greater overlap between the two arriving flows. Hence non-zero leg finishing indicates when a scheme has exceeded its capacity to absorb a peak in demand. While this transient capacity level is difficult to quantify directly, schemes can be compared by observing the region of ($T_{leg2,S}, T_{Gap}$) space where their leg finish frequency is zero. Fig. 8 summarizes this information in terms of the zero-contours for all the plots in the left column of Fig. 7. There are 78 design points sampled in this design space. Centralized control covers the largest area (orange), and Arbitrary Reservation covers the smallest (green). For the covering area of Direct Reservation and Network Entry, there are 18 more non-leg-finish points (blue stars) than Arbitrary Reservation, while 2 less points (orange stars) than Centralized.

Centralized control here acts as the benchmark scheme which does not consider any delay for the control and sequencing. Therefore, predictably, the leg capacity results are better than all the decentralized schemes. Arbitrary Reservation shows much poorer capacity than the other two decentralized schemes. For the leg finish results of Direct Reservation & Network Entry, as marked by the blue stars, there are 18 design points with zero leg finish but non-zero for Arbitrary Reservation. This illustrates how the extra effort in the STDMA scheme to achieve FCFS ordering has a significant impact on capacity.

Moving to the right column of Fig. 7, leg usage is largely irrelevant for the Arbitrary Reservation scheme. Although low for its limited region of safe operation, this suggests that the first onset of any overlap between the flows results in an immediate loss of separation, due to the random sequencing and hence waste of leg capacity. Comparing Direct Reservation and Network Entry, though they show identical leg finish results, the leg usage of Direct Reservation is generally larger than the Network Entry. Centralized control shows the lowest leg usage of all, to be expected because of the absence of time needed to determine sequencing. However, leg usage for centralized is not much lower than for Network Entry.

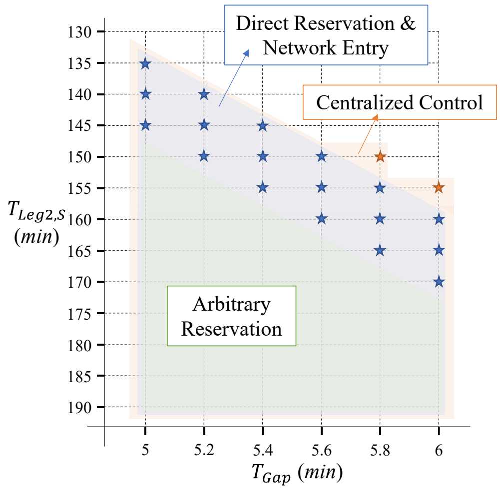


Fig. 8. Zero leg-finish ranges of four control schemes

Fig. 9 shows the number of messages sent by each of the three decentralized scheme. Similar to above leg usage results, there is a clear decreasing trend as $T_{leg2,S}$ climbs for all three schemes because the traffic overlapping of two legs become smaller. The Direct Reservation can send less messages than the other two when T_{Gap} is larger than 5.4; The reason is that there are not many aircraft communicating at the same time thus the slot can be easily found to reserve for a new plane (just one slot is needed). In contrast, when T_{Gap} equals 5 and $T_{leg2,S}$ is small (crowded scenarios), the Arbitrary Reservation send the least messages mainly due to its flexibility, less collisions happened than the other two schemes.

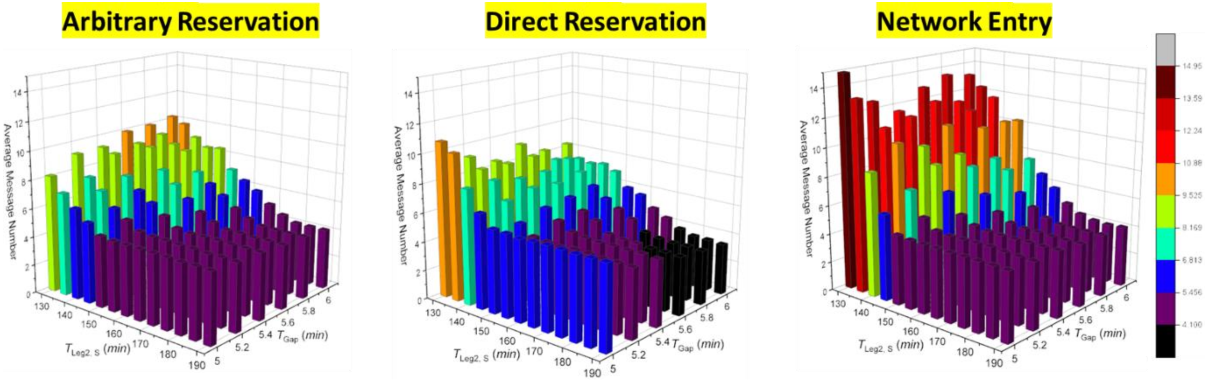


Fig. 9. Average message sending number

The Network Entry scheme generally requires sending more messages than the other two, especially when $T_{leg2,S} \leq 140$. The reasons include two aspects: 1). Compared with Direct Reservation, it requires an additional network entry slot for reservation; 2). Compared with Arbitrary Reservation, Network Entry would have higher probability to address collisions (by sending more messages), because Arbitrary Reservation randomly select the slot to reserve. In Arbitrary Reservation, if two or more adjacent aircraft select different slots, there is no need to send the ReSlot messages even though they chose the same network entry slot.

Fig. 10 shows the mean time of slot reservation in three decentralized schemes. It records the time consumption between the end of "Listening" state and the beginning of "Reserved" state (see Fig. 4). Since both N_{slot} and T_{slot} are fixed in this design space, in every scheme, the slot reservation time performs a small fluctuation in this space. Obviously, Direct Reservation suffers a long reservation time (between 1.2 and 1.8 *mins*), which is much larger than the other two. This explains the higher leg usage for Direct Reservation seen in Fig. 7, since aircraft will sometimes be flying along the leg before confirmation of their slot. Since Arbitrary Reservation and Network Entry use the closest available slot as the network entry for reservation, the general reservation time is close to one slot duration, i.e., T_{slot}=0.2 *mins*.

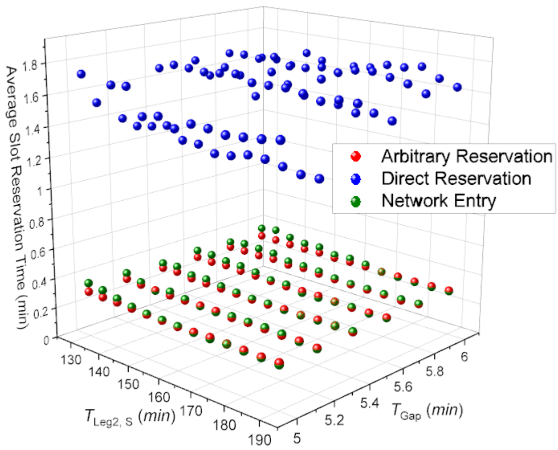


Fig. 10. Average slot reservation time

5.3. Sensitivity to number of slots

The leg finish and leg use results of N_{slot} sensitivity analysis are further given, as shown in Fig. 11, with the aircraft entry spacing T_{Gap} fixed at 5*min* throughout. The offset delay $T_{leg2,S}$ is varied here between 100 and 200, a wider and coarser sampling than the previous experiments in Fig. 7. Only the centralized schemes are considered here, as slots are not considered in the centralized control. As in Fig. 7, all the leg use results, whose leg-finish rate is non-zero, are marked in grey for the PM separation concern.

Again, Arbitrary Reservation does worst in terms of capacity, with any offset $T_{leg2,S}$=140 or lower causing aircraft to reach the end of the leg and lose separation. Direct Reservation can handle the $T_{leg2,S} = 140$ case only when $N_{slot} > 25$, while Network Entry maintains separation for all cases with $T_{leg2,S} = 140$. This difference was not seen in Fig. 7, where Network Entry and Direct Reservation has identical capacity. The reason for the difference can be seen in the right column of Fig. 11, where Direct Reservation exhibits a clear increase of the leg usage as N_{slot} climbs. The reason is that, with a fixed T_{slot}, increasing N_{slot} means a longer frame duration, which can prolong the waiting time between listening and slot reservation in Direct Reservation. Network Entry and Arbitrary Reservation do not suffer from this effect, by design.

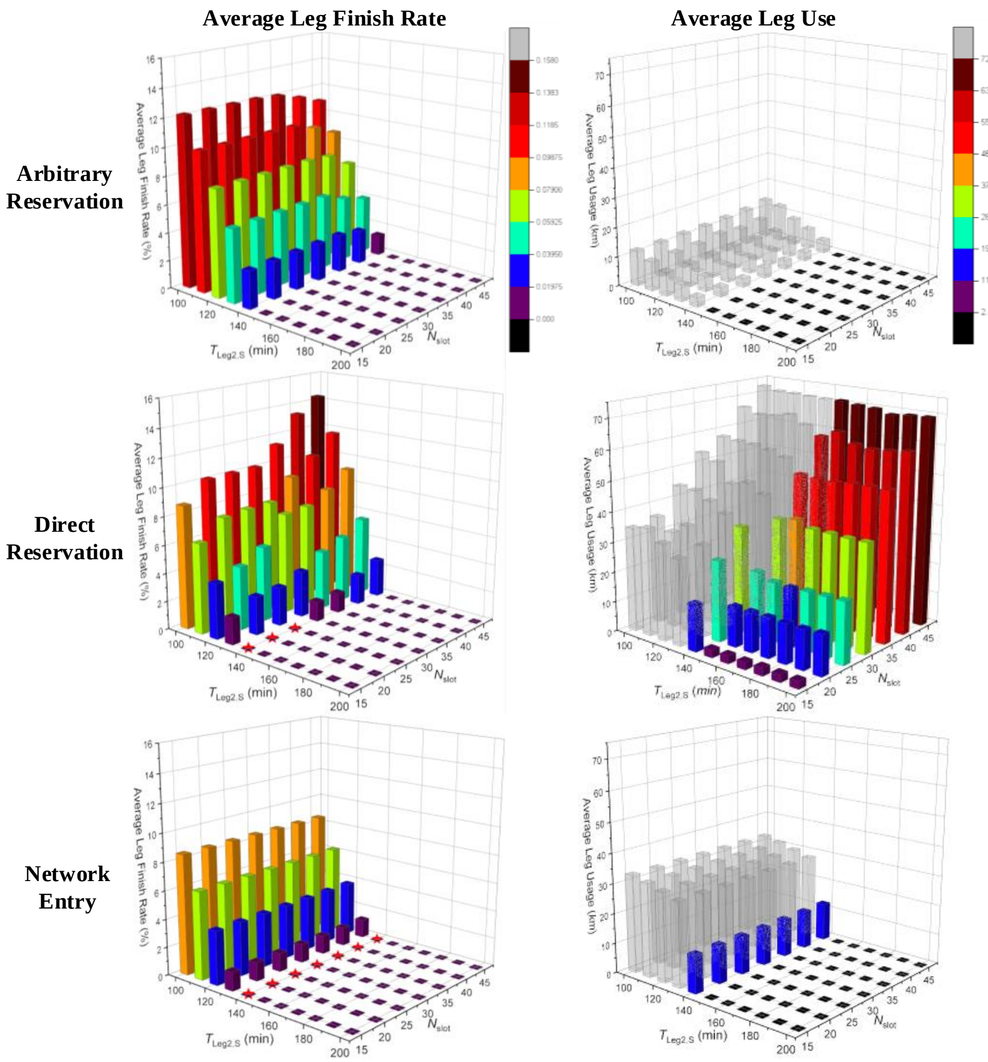


Fig. 11. Average leg finish rate and leg use results in N_{slot} sensitivity analysis.

6. Conclusion

In this paper, agent-based modelling employing finite state machines is used to simulate a point merge system for managing the sequence of arriving aircraft in a terminal area. Three decentralized control schemes have been proposed based on the principles of self-organizing channel access systems from communications. Centralized control has also been studied as a benchmark, and predictably provides the best performance in terms of handling peaks in demand with minimal excess flight distance. Arbitrary sequencing performs worst as the random ordering of aircraft wastes capacity and quickly leads to loss of separation when arriving flows overlap. Direct reservation performs better, but incurs excess flying and high sensitivity to the number of slots in a frame, owing to its need to wait for multiple frames to settle slot allocations. Adding a network entry message to direct reservation dramatically reduces those issues, leading to a decentralized system with capacity and flight lengths only slightly poorer than centralized control.

References

[1] ICAO. "Air traffic recovery is fast-approaching pre-pandemic levels " ICAO. https://www.icao.int/Newsroom/Pages/Air-traffic-recovery-is-fastapproaching-prepandemic-levels.aspx (accessed 16 October, 2022).

[2] EUROCONTROL, "Point Merge implementation: A quick guide: Simplifying and enhancing arrival operations with closed loop sequencing," 2021.
[3] EUROCONTROL. "Point Merge: Improving and harmonising arrival operations." https://www.eurocontrol.int/concept/point-merge (accessed 1 June, 2022).
[4] Ö. S. Meric and O. Usanmaz, "A new standard instrument arrival: the point merge system," *Aircraft Engineering and Aerospace Technology,* 2013.
[5] L. Boursier, B. Favennec, E. Hoffman, A. Trzmiel, F. Vergne, and K. Zeghal, "Merging arrival flows without heading instructions," in *7th USA/Europe air traffic management R&D seminar*, 2007: Citeseer, pp. 1-8.
[6] 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.
[7] B. Favennec, E. Hoffman, A. Trzmiel, F. Vergne, and K. Zeghal, "The point merge arrival flow integration technique: towards more complex environments and advanced continuous descent," in *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS)*, 2009, p. 6921.
[8] M. Liang, D. Delahaye, and P. Maréchal, "A framework of point merge-based autonomous system for optimizing aircraft scheduling in busy TMA," *5th SESAR Innovation Days, Bologna, Italy,* 2015.
[9] Y. Hong, N. Cho, Y. Kim, and B. Choi, "Multiobjective optimization for aircraft arrival sequencing and scheduling," *Journal of Air Transportation,* vol. 25, no. 4, pp. 115-122, 2017.
[10] Y. Hong, B. Choi, K. Lee, and Y. Kim, "Dynamic robust sequencing and scheduling under uncertainty for the point merge system in terminal airspace," *IEEE Transactions on Intelligent Transportation Systems,* vol. 19, no. 9, pp. 2933-2943, 2017.
[11] S. Lee, Y. Hong, and Y. Kim, "Optimal scheduling algorithm in point merge system including holding pattern based on mixed-integer linear programming," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering,* vol. 234, no. 10, pp. 1638-1647, 2020.
[12] R. K. Cecen, "Multi-objective TMA management optimization using the point merge system," *Aircraft Engineering and Aerospace Technology,* 2020.
[13] R. K. Çeçen, C. Cetek, and O. Kaya, "Aircraft sequencing and scheduling in TMAs under wind direction uncertainties," *The Aeronautical Journal,* vol. 124, no. 1282, pp. 1896-1912, 2020.
[14] Y. Hong, B. Choi, and Y. Kim, "Two-stage stochastic programming based on particle swarm optimization for aircraft sequencing and scheduling," *IEEE Transactions on Intelligent Transportation Systems,* vol. 20, no. 4, pp. 1365-1377, 2018.
[15] Y. Hong, B. Choi, G. Oh, K. Lee, and Y. Kim, "Nonlinear conflict resolution and flow management using particle swarm optimization," *IEEE Transactions on Intelligent Transportation Systems,* vol. 18, no. 12, pp. 3378-3387, 2017.
[16] Y. Tian, C. Xu, M. Sun, C. Li, and R. Sun, "Study on Arrival Aircraft Sequencing Based on Optimization of Point Merge Procedure," *Discrete Dynamics in Nature and Society,* vol. 2021, 2021.
[17] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines-a survey," *Proceedings of the IEEE,* vol. 84, no. 8, pp. 1090-1123, 1996.
[18] S. Hwang, K. Lee, H. Jeon, and D. Kum, "Autonomous Vehicle Cut-In Algorithm for Lane-Merging Scenarios via Policy-Based Reinforcement Learning Nested Within Finite-State Machine," *IEEE Transactions on Intelligent Transportation Systems,* 2022.
[19] Q. Wang, J. Zheng, H. Xu, B. Xu, and R. Chen, "Roadside magnetic sensor system for vehicle detection in urban environments," *IEEE Transactions on Intelligent Transportation Systems,* vol. 19, no. 5, pp. 1365-1374, 2017.
[20] E. Engler, M. Baldauf, F. Sill Torres, and S. Brusch, "Finite state machine modelling to facilitate the resilience of infrastructures: Reflections," *Infrastructures,* vol. 5, no. 3, p. 24, 2020.
[21] L. Hu, C. Zikun, T. Yongliang, W. Bin, Y. Hao, and W. Guanghui, "Evaluation method for helicopter maritime search and rescue response plan with uncertainty," *Chinese Journal of Aeronautics,* vol. 34, no. 4, pp. 493-507, 2021.
[22] R. T. Cahyono, S. P. Kenaka, and B. Jayawardhana, "Simultaneous Allocation and Scheduling of Quay Cranes, Yard Cranes, and Trucks in Dynamical Integrated Container Terminal Operations," *IEEE Transactions on Intelligent Transportation Systems,* 2021.
[23] G. Yuan, L. Hu, and T. Yongliang, "Inverse design of mission success space for combat aircraft contribution evaluation," *Chinese Journal of Aeronautics,* vol. 33, no. 8, pp. 2189-2203, 2020.
[24] M. Liang, D. Delahaye, and P. Maréchal, "Potential operational benefits of multi-layer point merge system on dense TMA operation hybrid arrival trajectory optimization applied to Beijing capital international airport," in *Proceedings of the 7th International Conference on Research in Air Transportation*, 2016.

[25] BRILLIANT. "Finite State Machines." BRILLIANT. https://brilliant.org/wiki/finite-state-machines/ (accessed 1 June, 2022).
[26] K. El-Fakih, R. M. Hierons, and U. C. Türker, "$\mathcal K $ K-Branching UIO Sequences for Partially Specified Observable Non-Deterministic FSMs," *IEEE Transactions on Software Engineering*, vol. 47, no. 5, pp. 1029-1040, 2019.
[27] A. Company. "Anylogic." https://www.anylogic.com/ (accessed.
[28] C. W. Weimer, J. O. Miller, and R. R. Hill, "Agent-based modeling: An introduction and primer," in *2016 Winter Simulation Conference (WSC)*, 2016: IEEE, pp. 65-79.
[29] A. company. "Chapter 7. Designing state-based behavior: statecharts." https://www.anylogic.com/upload/books/new-big-book/7-statecharts.pdf (accessed.
[30] T. Gaugel, J. Mittag, H. Hartenstein, S. Papanastasiou, and E. G. Ström, "In-depth analysis and evaluation of self-organizing TDMA," in *2013 IEEE vehicular networking conference*, 2013: IEEE, pp. 79-86.
[31] E. T. ETSI, "Electromagnetic compatibility and radio spectrum matters (erm); short range devices (srd); radio equipment to be used in the 25 mhz to 1 000 mhz frequency range with power levels ranging up to 500 mw," *European harmonized standard EN*, vol. 300, no. 220, p. v2, 2012.