An online speed profile generation approach for efficient airport ground movement

Tianci Zhang[a,b], Meng Ding[a,*], Hongfu Zuo[a], Jun Chen[b], Michal Weiszer[b], Xiaoyan Qian[a], Edmund K. Burke[c]

[a] College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, 29 Jiangjun Avenue, Nanjing 211106, China
[b] School of Engineering and Materials Science, Queen Mary University of London, Mile End Road, London E1 4NS, U.K.
[c] School of Electronic Engineering and Computer Science, Queen Mary University of London, Mile End Road, London E1 4NS, U.K.

* Corresponding author.

E-mail addresses: zhangtiancy@163.com (T. Zhang); nuaa_dm@nuaa.edu.cn (M. Ding); rms@nuaa.edu.cn (H. Zuo); jun.chen@qmul.ac.uk (J. Chen); m.weiszer@qmul.ac.uk (M. Weiszer); qianxiaoyan@nuaa.edu.cn (X. Qian); ekb@qmul.ac.uk (E.K. Burke).

An online speed profile generation approach for efficient airport ground movement

Abstract: The precise guidance and control of taxiing aircraft based on four-dimensional trajectories (4DTs) has been recognised as a promising means to ensure safe and efficient airport ground movement in the context of ever growing air traffic demand. In this paper, a systematic approach for online speed profile generation is proposed. The aim is to generate fuel-efficient speed profiles respecting the timing constraints imposed by routing and scheduling, which ensures conflict-free movement of aircraft in the planning stage. The problem is first formulated as a nonlinear optimisation model, which uses a more flexible edge-based speed profile definition. A decomposed solution approach (following the framework of matheuristic) is then proposed to generate feasible speed profiles in real time. The decomposed solution approach reduces the nonlinear optimisation model into three tractable constituent problems. The control point arrival time allocation problem is solved using linear programming. The control point speed allocation problem is solved using particle swarm optimisation. And the complete speed profile between control points is determined using enumeration. Finally, improved speed profiles are generated through further optimisation upon the feasible speed profiles. The effectiveness and advantages of the proposed approach are validated using datasets of real-world airports.

Keywords: Airport ground movement; Fuel consumption; Matheuristics; Nonlinear optimisation; Speed profile

1. Introduction

As many airports approach their maximum capacity, the ever growing air traffic demand starts to put more pressure on airport ground movement operations (Eurocontrol, 2013). Congestion will frequently occur even with the support of advanced ground movement management systems. This leads to both economic and environmental concerns, such as excessive delay, increased fuel consumption and emissions. A non-negligible factor contributing to congestion is the lack of effective information sharing and collaboration between stakeholders. Aircraft usually taxi along standard routes and avoid conflict in a purely reactive way as a result. This leads to unnecessary uncertainty and inefficiency, making it difficult to maximise the utilisation of the existing airport infrastructure. To address the above issues, collaborative decision support systems (Eurocontrol, 2012; FAA, 2012) are being developed, and research into new operational concepts based on such systems is under investigation (JPDO, 2007).

The concept of trajectory-based taxi operations has been recently proposed to achieve more efficient ground movement (Okuniek et al., 2016). Instead of using standard taxi routes and purely reactive surface guidance, the trajectory-based approach generates conflict-free four-dimensional trajectories (4DTs) for all aircraft on the airport surface, and uses more sophisticated on-board or ground-based guidance technologies to keep pilots aware of the 4DT conformance status and the required maneuvers during taxiing (Bakowski et al., 2015; Biella et al., 2015; Foyle et al., 2011; Haus et al., 2011; Jones et al., 2014). This will largely eliminate the temporal uncertainty in the taxiing phase, making the holistic optimisation of interconnected airport operations (such as departure management and runway scheduling) more tractable (Weiszer et al., 2015a). Furthermore, the application of trajectory-based operations will also enable a smoother transition between the en route and ground movement phases, facilitating the development of the future generation of airspace systems (Eurocontrol, 2015; FAA, 2016).

On the airport, 4DTs are generally described by the route aircraft should follow and the detailed speed profiles along the route. To generate conflict-free 4DTs, a two-stage 4DT design procedure is developed in Cheng and Sweriduk (2009) and Cheng (2004). In the first stage, a desirable taxi route and the corresponding required times of arrival at selected waypoints (e.g., taxiway intersections or runway crossings) are determined during routing and scheduling. In the second stage, speed profiles complying with the designated route and the required times of arrival are generated for guidance. Although speed profiles are generated online in this approach, the fuel-efficiency cannot be ensured due to the stringent constraint of the required times of arrival (Chen et al., 2016a). Another 4DT generation approach is typified by the Active Routing framework (Chen et al., 2016a; Chen et al., 2016b). Active Routing aims to produce greener and more cost-effective 4DTs by combining routing, scheduling and speed profile generation into one integrated multi-objective optimisation framework. In this way, different costs (e.g., taxi time and fuel consumption) of each candidate 4DT can be precisely evaluated in routing and scheduling, and a set of conflict-free 4DTs with nondominated costs can be found out for each aircraft. This enables controllers to select suitable 4DTs for a group of aircraft according to the current scenario. However, due to the restriction of the computational cost, currently the Active Routing framework generates speed profiles offline for different taxiway segments and then recompose them to form a complete speed profile for a certain route in routing and scheduling.

In this paper, we propose an improved online speed profile generation approach, which can be embedded within the two-stage 4DT design procedure as well as the Active Routing framework. With a more flexible speed profile generation model and a specifically developed solution approach, feasible and fuel-efficient speed profiles can be generated online according to the current situation. This will facilitate the generation of conflict-free 4DTs under various scenarios, especially when recovery planning is needed due to unprecedented events.

The rest of the paper is organised as follows. Section 2 presents a review of relevant approaches and summarises the contributions of this paper. Section 3 formulates the speed profile generation problem as a nonlinear optimisation model. Section 4 introduces the proposed solution approach. The performance of the proposed method is validated in Section 5 using problem instances based on real-world airport layouts. Conclusions and future directions are presented in Section 6.

2. Literature review and contributions

2.1. Literature review

The problem of 4DT-based ground movement planning has attracted significant attention in recent years (Atkin et al., 2010; Marín, 2006). A full 4DT for ground movement specifies the expected positions of an aircraft at all time during taxiing. Earlier work has aimed to determine the partially defined 4DT (often known as time-based taxi trajectory) for an aircraft, which consists of a taxi route and the corresponding required times of arrival at control points.

Here, control points refer to the artificial waypoints set along taxiways for safe separation between aircraft during routing and scheduling. The taxiway between two adjacent control points is referred to as an edge. The control points dividing turning and straight segments of a specific taxi route are referred to as critical points. In addition to the critical points, other intermediate control points may exist on each segment of the taxi route. It should be noted that for guidance purpose, not all required times of arrival at control points will be displayed to pilots, in order to avoid impact on their work load and situation awareness due to frequent checking of the time. In Smeltink and Soomer (2004) and Roling and Visser (2008), aircraft are scheduled using the fixed standard taxi routes. In this case, only scheduling is applied to generate the required times of arrival at control points. In Balakrishnan and Jung (2007), Deau et al. (2009), Gotteland and Durand (2003) and Montoya et al. (2010), multiple standard taxi routes are available for each aircraft. Routing and scheduling are then carried out in this reduced search space. Although using standard taxi routes makes the problem more tractable, better solutions may be obtained using the complete search space (i.e., all taxi routes). In light of this, in Clare and Richards (2011), García et al. (2005) and Marín (2006), routing and scheduling are conducted upon all taxi routes. However, the underlying difficulty with these approaches often makes the computational cost too high for real-world problems. This is especially true when airport traffic is subject to unexpected events and a fast re-planning is mandatory (Clare and Richards, 2011; Marín, 2013). In view of this, methods based on sequential planning have been proposed (Lesire, 2010; Ravizza et al., 2014; Zhang et al., 2016). By dealing with one aircraft at a time, the sequential approach is more computationally efficient and flexible, requiring less interference to the already moving aircraft.

Given a time-based taxi trajectory, the ground movement process of aircraft will be inherently regulated by the speed limit constraints and required times of arrival at control points. However, inefficient movement may still occur as there are still many degrees of freedom for the movement between two control points (Chen et al., 2016a; Cheng and Sweriduk, 2009). A possible solution is to further plan detailed speed profiles between control points (Bakowski et al., 2013; Cheng and Sweriduk, 2009; Cheng et al., 2008). This leads to a full 4DT with which the expected position of the aircraft at any time during taxiing can be determined. In Bakowski et al. (2013), Cheng and Sweriduk (2009) and Cheng et al. (2008), the taxi route for each aircraft is determined using dynamic programming. The required times of arrival at control points are then determined to avoid conflicts between aircraft. Finally, detailed speed profiles between control points complying with the required times of arrival are generated after setting the speeds at control points. However, it is unclear how the detailed speed profiles can be generated with respect to certain objectives such as fuel consumption. Moreover, as the required arrival times at control points are determined using average taxiing speeds, it is possible that unnecessary accelerations or decelerations will exist in the resulting speed profiles.

In light of these issues, an integrated approach (i.e., the Active Routing framework) for full 4DT planning is proposed in Chen et al. (2016b) and Chen et al. (2016a). Active Routing proactively considers the taxiing speed profile along with routing and scheduling. To improve the solution efficiency, a database of optimised speed profiles is developed beforehand (Weiszer et al., 2015b). The database consists of desirable speed profiles for every possible segment of the taxiway. During 4DT planning, a complete speed profile for a certain route is produced by connecting the precomputed speed profiles for the constituent segments. To facilitate such connection, the speed at the beginning or end of a segment takes some fixed values in the precomputed speed profiles. This approach is not flexible enough and inevitably increases the chance in violating the time window constraints imposed on the constituent edges of each segment. Moreover, when airport traffic is congested or under severe disruption, it will be more likely to fail in selecting a feasible 4DT from the precomputed speed profiles in the database. Generating new speed profiles using either the existing metaheuristic (Chen and Stewart, 2011; Chen et al., 2016b) or heuristic approaches (Weiszer et al., 2014) will not meet the computational time limit of re-planning. The requirement for a fast online speed profile generation approach is pressing.

2.2. Contributions

Considering the issues of feasibility, fuel-efficiency and flexibility, an improved online speed profile generation approach is proposed in this paper. The main contributions can be summarised as follows:

1) As the time window constraints are imposed upon edges between control points, speed profiles are also defined upon edges in the proposed approach, instead of the entire segment between critical points as in existing approaches (Cheng and Sweriduk, 2009; Weiszer et al., 2015b). This not only improves the flexibility to generate feasible speed profiles, but also enlarges the search space to find more desirable solutions.

2) The arrival times at both the critical and control points can be selected from given ranges, instead of using fixed time points (Cheng and Sweriduk, 2009). And the speeds at both the critical and control points can be specifically allocated according to the arrival time requirements, instead of assuming fixed values beforehand at the critical points (see e.g., Weiszer et al. (2015b)). These also make the proposed approach more flexible and facilitate the search for more desirable speed profiles.

3) A nonlinear optimisation model is formulated for the online speed profile generation problem. A matheuristic-based solution approach is developed accordingly, which can generate feasible speed profile in real time. Improved solutions can then be found using nonlinear optimisation techniques within the given computational time limit.

3. Problem description and formulations

As mentioned in Section 2.1, control points (including critical points) divide the taxiway into edges. Conflict-free time-based taxi trajectories can be generated by assuring that each edge is occupied by no more than one aircraft at a time. When the time-based taxi trajectory for an aircraft is determined, the number of control points along the taxi trajectory is also known. For clarity, in this paper, the edge between two consecutive control points p_i and p_{i+1} is referred to as edge i. Unlike previous studies where a specific required time of arrival is assigned to each control point p_i, this paper uses time-based taxi trajectories with an arrival time interval $ATW_i = [ts_i, te_i]$ imposed at each p_i, as shown in Fig. 1. Conflict-free movements can be ensured as long as aircraft enter edge i within the designated ATW_i. For more details of generating ATW_i, interested readers are referred to Zhang et al. (2017).

The concern of this paper is to generate a fuel-efficient speed profile in real time, which complies with the given time-based taxi trajectory. To make the problem tractable, a piece-wise linear speed profile model (see Fig. 2(a)) similar to Weiszer et al. (2014) is

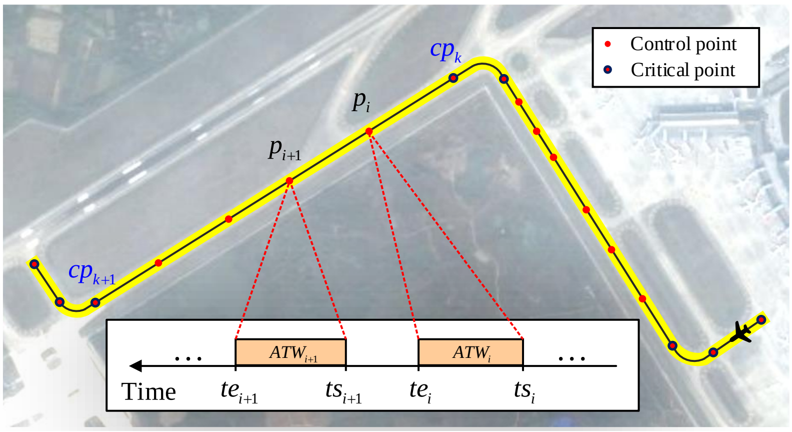


Fig. 1. An illustration of a time-based taxi trajectory on the Nanjing Lukou Airport. The background satellite image is taken from Google Maps.

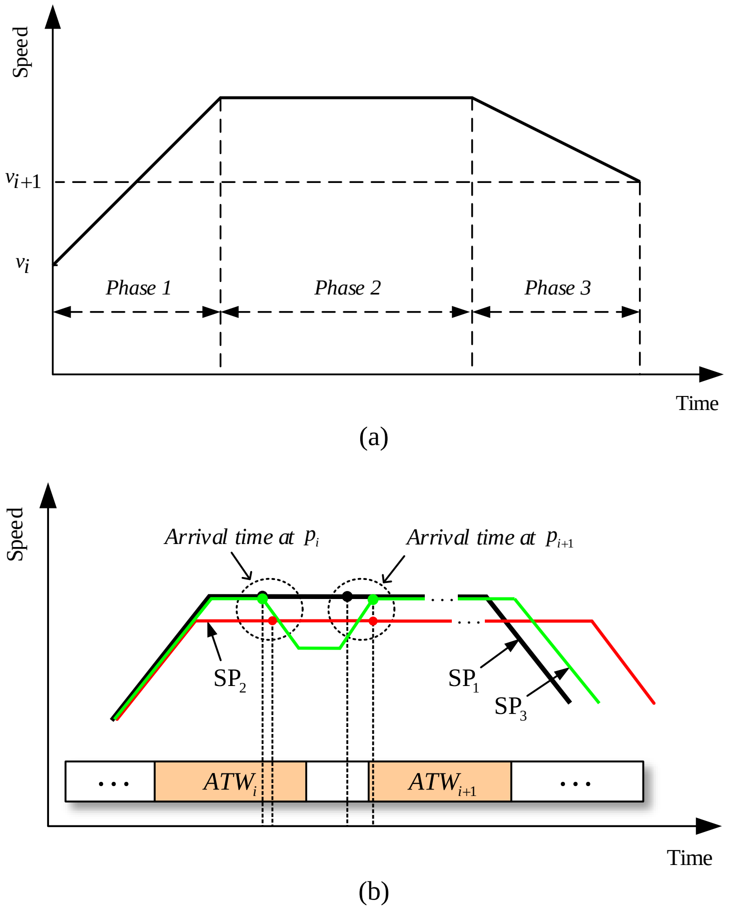


Fig. 2. Illustration of the speed profile model. (a) A typical speed profile for edge i with three phases. Phase 1: acceleration, Phase 2: constant speed, Phase 3: deceleration. $t_{i,k}$ is the duration of phase k. (b) Different speed profiles for a segment containing more than one edges. SP$_1$: The fastest speed profile generated based on the segment between two adjacent critical points, which cannot meet ATW_{i+1}. SP$_2$: A feasible speed profile generated based on the segment between two adjacent critical points and reduced maximum taxiing speed, which leads to longer taxi time. SP$_3$: A feasible speed profile generated with the presented model which is based on the edge between two adjacent control points. It can meet ATW_{i+1} by adjusting the speed profile for edge i, without having to slow down in other edges of the segment.

adopted but applied to each edge. The speed profile model consists of three phases. The first and last phases correspond to a constant acceleration/deceleration rate. The second phase corresponds to a constant speed. It is worth noting that not all the three phases are necessarily required for each edge, as will be determined by the solution algorithm. Comparing with the previous studies (Chen et al., 2016a; Chen et al., 2016b; Weiszer et al., 2014; Weiszer et al., 2015b) in which the speed profile model is applied to the entire segment between two adjacent critical points (e.g., cp_k and cp_{k+1} in Fig. (1)), the proposed method provides a means to adjust the speed profile for each edge between intermediate control points (see Fig. 2(b)), making it possible to generate fuel-efficient speed profiles along the whole route while still satisfying all the arrival time interval requirements.

3.1. Fuel consumption modelling

The main objective is to minimise the overall fuel consumption of the aircraft travelling along the given time-based taxi trajectory. For edge *i*, the fuel consumption g_i is determined by (1).

$$g_i = \sum_{k=1}^{3} f_{i,k} \cdot t_{i,k} \tag{1}$$

where $f_{i,k}$ is the fuel flow rate in phase *k*, and $t_{i,k}$ is the duration of phase *k*. $f_{i,k}$ is determined by the thrust level $\varepsilon_{i,k}$ of the engine, which is defined as follows.

$$\varepsilon_{i,k} = \left(m \cdot a_{i,k} + \mu \cdot m \cdot g\right) / F_0 \tag{2}$$

where *m* is the mass of aircraft, *g* is the acceleration of gravity, μ is the rolling resistance coefficient, F_0 is the maximum power output of the engine, and $a_{i,k}$ is the acceleration rate in phase *k*. The dependence between $f_{i,k}$ and $\varepsilon_{i,k}$ is provided by the International Civil Aviation Organization Emissions Databank (Nikoleris et al., 2011). Interpolation/extrapolation is used for $f_{i,k}$ if the corresponding $\varepsilon_{i,k}$ is outside of the defined values in the databank. For braking, the thrust level is assumed to be around 5% (Chen et al., 2016b; Weiszer et al., 2014).

3.2. Mathematical models and constraints

Let τ_i and v_i be the arrival time and speed at control point p_i, respectively, and *n* be the number of control points on the given taxi trajectory, the speed profile generation problem is formulated as (3)-(12).

$$\min \sum_{i=1}^{n-1} g_i \tag{3}$$

$$\text{s.t.} \ \sum_{k=1}^{3} t_{i,k} = \tau_{i+1} - \tau_i , \quad \forall\ i \in \{1,\dots,n-1\} \tag{4}$$

$$a_{i,k} = 0 \Rightarrow t_{i,k} = 0 , \quad \forall\ i \in \{1,\dots,n-1\},\ k \in \{1,3\} \tag{5}$$

$$v_{\min}^{i} \le v_i + a_{i,1} \cdot t_{i,1} \le v_{\max}^{i} , \quad \forall\ i \in \{1,\dots,n-1\} \tag{6}$$

$$v_i + a_{i,1} \cdot t_{i,1} = v_{i+1} - a_{i,3} \cdot t_{i,3} , \quad \forall\ i \in \{1,\dots,n-1\} \tag{7}$$

$$v_i \cdot t_{i,1} + 0.5 \cdot a_{i,1} \cdot t_{i,1}^2 + \left(v_i + a_{i,1} \cdot t_{i,1}\right) \cdot t_{i,2} + v_{i+1} \cdot t_{i,3} - 0.5 \cdot a_{i,3} \cdot t_{i,3}^2 = D_i ,\ \forall\ i \in \{1,\dots,n-1\} \tag{8}$$

$$ts_i \le \tau_i \le te_i , \quad \forall\ i \in \{1,\dots,n\} \tag{9}$$

$$v_{\min}^{i} \le v_i \le v_{\max}^{i} , \quad \forall\ i \in \{1,\dots,n\} \tag{10}$$

$$a_{i,k} \in K_i , \quad \forall\ i \in \{1,\dots,n-1\},\ k \in \{1,3\} \tag{11}$$

$$t_{i,k} \geq 0\,, \quad \forall\; i \in \{1,\ldots,n-1\},\;\; k \in \{1,3\} \tag{12}$$

Here, the acceleration/deceleration rate $a_{i,k}$ and duration $t_{i,k}$ of each phase are the primary decision variables, which determine the total fuel consumption in (3). $a_{i,k}$ adopts the convention that a positive value corresponds to accelerating while a negative value corresponds to decelerating. τ_i and v_i are intermediate variables which take values from the given ranges in (9) and (10), respectively. In (10), $v^i_{\min}$ and $v^i_{\max}$ are the minimum and maximum speed allowed in edge *i*, respectively. Unlike Chen et al. (2016b) where a continuous $a_{i,k}$ is used, we use a finite set K_i of acceleration/deceleration rates in (11). This results in throttle controls set up in discrete steps, amenable to human manual control in practice (Cheng and Sweriduk, 2009). (4) is the constraint on the control point arrival time. (5) prevents unrealistic time allocation for the acceleration or deceleration phase. (6) limits the taxiing speed to the allowed range. (7) validates the constant speed phase. (8) checks if aircraft will travel exactly the length of edge *i* (denoted as D_i) following the generated speed profile, where $v_i \cdot t_{i,1} + 0.5 \cdot a_{i,1} \cdot t^2_{i,1}$, $\left(v_i + a_{i,1} \cdot t_{i,1}\right) \cdot t_{i,2}$, and $v_{i+1} \cdot t_{i,3} - 0.5 \cdot a_{i,3} \cdot t^2_{i,3}$ are the movement distances for the three phases of the speed profile on edge *i* (see Fig. 2(a)), respectively.

The dynamic nature of airport ground movement operations may require speed profiles to be re-generated in real time according to the current ground movement scenarios. This is especially the case under severe disruption. However, it would be computationally expensive to exactly solve this optimisation problem. Fortunately, many real-world applications only require a satisfactory solution. In this case, decomposition using problem specific heuristics can offer such a solution with acceptable computational cost. Therefore, we first decompose the above model into two constituent problems, i.e., optimising τ_i and searching for suitable v_i. As $a_{i,k}$ is defined on a limited finite set of discrete values, an enumeration based approach will then be employed to provide the most desirable $a_{i,k}$ and $t_{i,k}$, hence the fuel-efficient speed profile, within this finite set. The proposed approach is detailed in the next section.

4. Solution approach

A matheuristic-based solution approach is developed in this section to generate feasible speed profiles efficiently. This ensures at least one feasible speed profile can be found for each aircraft. When there is still remaining computational time after finding a feasible speed profile, general nonlinear optimisation techniques can be utilised to search for more desirable speed profiles.

The proposed approach for feasible speed profile generation consists of three steps, as illustrated in Fig. 3. Firstly, τ_i is allocated using mathematical programming. It is worth noting that $\left[ts_i, te_i\right]$ provides a feasible range to adjust τ_i so that $\tau_{i+1} - \tau_i$ approximates the traversal time of edge *i* in the precomputed speed profile. Secondly, the feasible control point speed v_i is allocated using a metaheuristic approach as this particular constituent problem is nonlinear and nonconvex. The aim is to minimise the estimated deviation from the control point speeds in the precomputed speed profile. Finally, detailed speed profiles (defined by $a_{i,k}$ and $t_{i,k}$) are generated. As τ_i and v_i have been allocated in the first two steps, the detailed speed profile can be determined for each edge independently. The above approach utilises the interoperation of metaheuristics and mathematical programming, and follows the framework of matheuristics (Fischetti and Fischetti, 2016).

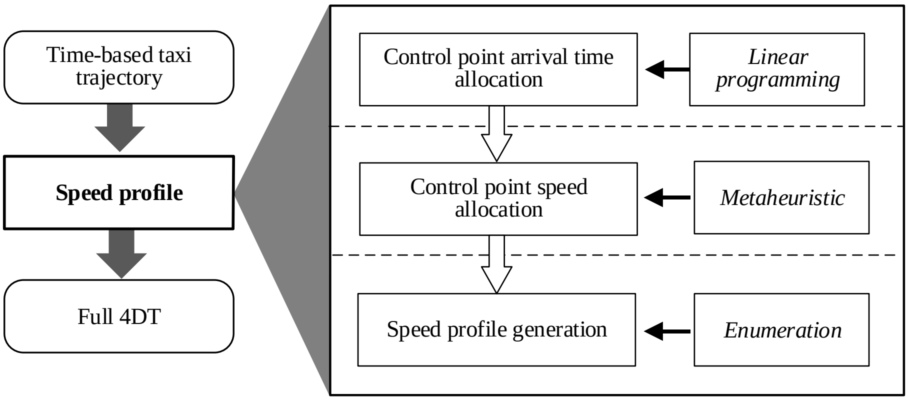

Time-based taxi trajectory
Speed profile
Full 4DT
Control point arrival time allocation
Linear programming
Control point speed allocation
Metaheuristic
Speed profile generation
Enumeration

Fig. 3. Diagram of the proposed approach for feasible speed profile generation.

4.1. Control point arrival time allocation

According to the spatial layout of the taxi trajectory, we can identify straight and turning segments as described in Chen et al. (2016b). The desirable traversal time T_{ref}^{i} of edge *i* with respect to certain speed limits is known according to the precomputed speed profiles (Weiszer et al., 2015b). The control point arrival time is allocated by minimising the difference between $\tau_{i+1} - \tau_i$ and T_{ref}^{i} as well as the total taxi time along the assigned route τ_n :

$$\min \ o_1 = w_1 \cdot \sum_{i=1}^{n-1} y_i + w_2 \cdot \tau_n \tag{13}$$

$$\text{s.t.} \ T_{ref}^{i} \le \tau_{i+1} - \tau_i + y_i , \quad \forall \ i \in \{1, \ldots, n-1\} \tag{14}$$

$$\tau_{i+1} - \tau_i - y_i \le T_{ref}^{i} , \quad \forall \ i \in \{1, \ldots, n-1\} \tag{15}$$

$$T_{\min}^{i} \le \tau_{i+1} - \tau_i \le T_{\max}^{i} , \quad \forall \ i \in \{1, \ldots, n-1\} \tag{16}$$

$$ts_i \le \tau_i \le te_i , \quad \forall \ i \in \{1, \ldots, n\} \tag{17}$$

Here, w_1 and w_2 are user specified weights with $w_1 \gg w_2$. The auxiliary variable y_i is introduced to linearise the implicit objective $\left|\tau_{i+1} - \tau_i - T_{ref}^{i}\right|$, as indicated by (14) and (15). (16) requires that the time spent on edge *i* is between the minimum and maximum traversal times $T_{\min}^{i}$ and $T_{\max}^{i}$. This implicitly puts a constraint on the taxiing speed of the related edge, ensuring the allocated control point arrival time is suitable for finding out feasible control point speeds in the next step. (17) ensures τ_i is within the given arrival time interval.

(13)-(17) form a linear programming problem which can be solved efficiently using commercial solvers (e.g., CPLEX).

4.2. Control point speed allocation

Similar to arrival times at control points, the desirable control point speeds can also be determined according to the precomputed speed profiles. We allocate the control point speeds by minimising the level of deviation from the desirable control point speeds, as described in (18)-(21).

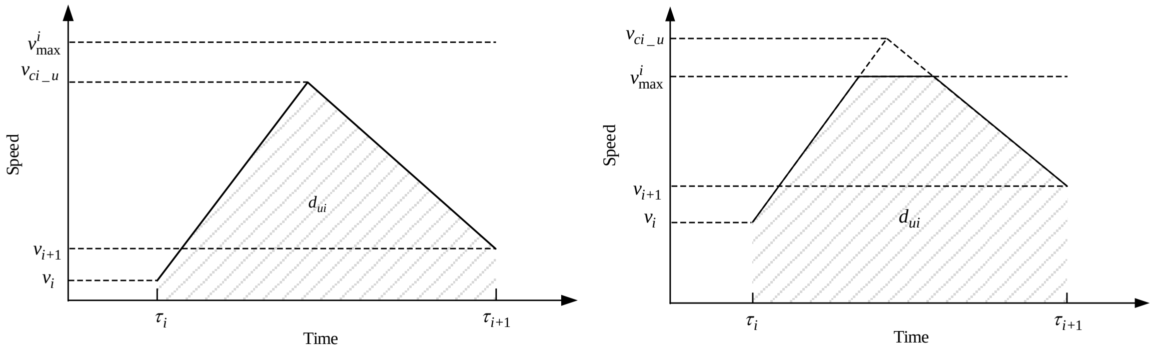
$$\min \ o_2 = \sum_{i=1}^{n} u_i \tag{18}$$

$$\text{s.t.} \ -a_{\max}^{i} \le \left(v_i - v_{i+1}\right) / T_i \le a_{\max}^{i} , \quad \forall \ i \in \{1, \ldots, n-1\} \tag{19}$$

$$d_{li} \le D_i \le d_{ui} , \quad \forall \ i \in \{1, \ldots, n-1\} \tag{20}$$

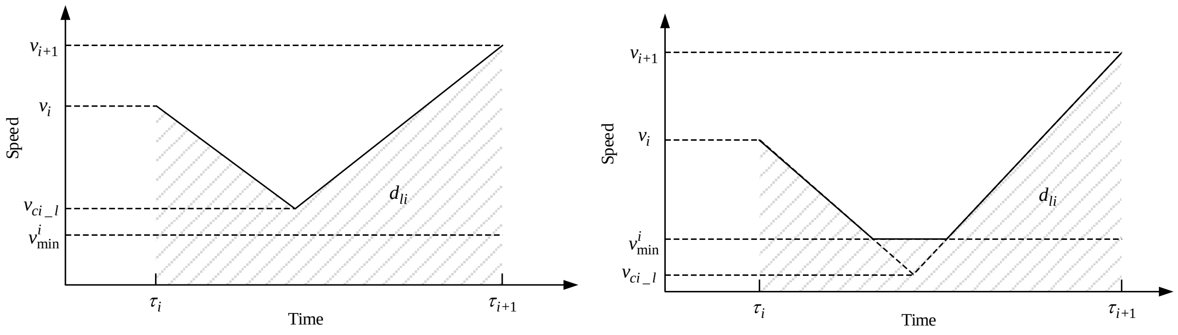
$$v_{\min}^i \le v_i \le v_{\max}^i, \quad \forall\ i \in \{1,\ldots,n\} \tag{21}$$

In (18), $u_i = \left|v_i - v_{ref}^i\right|$ represents the level of deviation from the desirable control point speed v_{ref}^i at control point p_i. (19) ensures that the allocated speed v_i is feasible to reach the ending speed v_{i+1} for each edge *i*, where $T_i = \tau_{i+1} - \tau_i$ is the traversal time in edge *i*. (20) ensures that T_i provides enough time for an aircraft to traverse distance D_i, where d_{ui} and d_{li} are the longest and the shortest distance an aircraft can travel over T_i, respectively. d_{ui} is achieved by first accelerating and then decelerating with the maximum rate $a_{\max}^i$ as illustrated in Fig. 4. d_{li} is achieved by first decelerating and then accelerating with $a_{\max}^i$ as shown in Fig. 5. The values of d_{ui} and d_{li} are determined according to (22) and (23), respectively.



(a) (b)

Fig. 4. Upper bound of the travelling distance. (a) $v_{ci\_u} \le v_{\max}^i$, (b) $v_{ci\_u} > v_{\max}^i$.



(a) (b)

Fig. 5. Lower bound of the travelling distance. (a) $v_{ci\_l} \ge v_{\min}^i$, (b) $v_{ci\_l} < v_{\min}^i$.

$$d_{ui} = \begin{cases} \left(2 \cdot v_{ci\_u}^2 - v_i^2 - v_{i+1}^2\right) / \left(2 \cdot a_{\max}^i\right) & ,\text{if } v_{ci\_u} \le v_{\max}^i \\ \left(2 \cdot v_{ci\_u}^2 - v_i^2 - v_{i+1}^2\right) / \left(2 \cdot a_{\max}^i\right) - & \\ \left(v_{ci\_u} - v_{\max}^i\right)^2 / a_{\max}^i & ,\text{otherwise} \end{cases} \tag{22}$$

where $v_{ci\_u} = 0.5 \cdot \left(v_i + v_{i+1} + a_{\max}^i \cdot T_i\right)$.

$$d_{li} = \begin{cases} \left(v_i^2 + v_{i+1}^2 - 2\cdot v_{ci\_l}^2\right)/\left(2\cdot a_{\max}^i\right) & \text{,if } v_{ci\_l} \geq v_{\min}^i \\ \left(v_i^2 + v_{i+1}^2 - 2\cdot v_{ci\_l}^2\right)/\left(2\cdot a_{\max}^i\right) + & \\ v_{\min}^i \cdot \left(T_i - \left(v_i + v_{i+1} - 2\cdot v_{\min}^i\right)/a_{\max}^i\right) & \text{,otherwise} \end{cases} \tag{23}$$

where $v_{ci\_l} = 0.5\cdot\left(v_i + v_{i+1} - a_{\max}^i \cdot T_i\right)$.

As this problem is nonlinear and nonconvex, it is often hard to find global optimal solutions within given computational time limit. In light of this, metaheuristics are suitable approaches to find a good solution efficiently. Here particle swarm optimisation is adopted as a typical choice due to its proven capability of achieving fast convergence. The pseudo-code is presented in Algorithm 1. In particle swarm optimisation, the position of a particle in the search space represents a potential solution of the problem (Kennedy and Eberhart, 1995; Shi

Algorithm 1: Control point speed allocation

```
Input: Edge lengths ${D_i}$, travelling times ${T_i}$ and desirable control point speeds ${v_{ref}^i}$
Output: Control point speeds ${v_i}$
$MaxIter = 200$, $PopSize = 200$, $c_1 = c_2 = 2.05$;
for $j = 1 : PopSize$ do
    Randomly initialise position $x_j$ and velocity $s_j$;
    Initialise local best position: $x_j^* = x_j$;
Initialise global best position: $x^* = \arg\min\left(fit_j\right)$;
$k = 1$;
while $k \leq MaxIter$ do
    for $j = 1 : PopSize$ do
        $x_{\max} = \arg\max\left(fit_j\right)$;
        $x_{\min} = \arg\min\left(fit_j\right)$;
        $p_r = e^{-\left(fit_{\max} - fit_{\min}\right)/fit_{\min}}$;
        if $Rand < p_r$ then
            if $k < MaxIter/2$ then
                global randomisation: $x^* = Rand\cdot\left(x_{\max} - x_{\min}\right) + x_{\min}$;
            else
                local randomisation: $x_j^* = Rand\cdot\left(x_{\max} - x_{\min}\right) + x_{\min}$;
        $w = 0.9 - 0.5\cdot k/MaxIter$;
        $s_j = w\cdot s_j + c_1\cdot Rand\cdot\left(x_j^* - x_j\right) + c_2\cdot Rand\cdot\left(x^* - x_j\right)$;
        $x_j = x_j + s_j$;
        if $fit_j < fit_j^*$ then
            $x_j^* = x_j$;
    if $\min\left(fit_j^*\right) < fit^*$ then
        $x^* = \arg\min\left(fit_j^*\right)$;
    $k = k + 1$;
return $x^*$;
```

and Eberhart, 1998). To avoid premature convergence, a position randomisation scheme as presented in Zhou et al. (2011) is adopted (Lines 12-16). For our problem, a particle *j* is defined in an *n*-dimensional search space (i.e., the dimensionality is equal to the number of control points). Its position x_j specifies the allocated control point speeds. The fitness value of *j* is defined as

$$fit_j = \left(1+e_j\right)\cdot o_{2,j}$$

where $o_{2,j}$ is the objective value for each x_j. e_j is a penalty for constraint violation in order to push solutions back into the feasible regions. If x_j is infeasible, the deviation from the left or right bound of each violated constraint, i.e., (19) and (20), will be added to e_j.

4.3. Speed profile generation

After obtaining τ_i and v_i in the first two steps, the speed profile generation problem can be described by (3)-(8), (11) and (12) (see Section 3.2). This problem is solved independently over each edge *i* as τ_i and v_i are now fixed. Only two independent variables $a_{i,1}$ and $a_{i,3}$ are considered in this problem as $a_{i,2}=0$. An enumeration based approach is used to search for the best values of $a_{i,1}$ and $a_{i,3}$ with respect to the fuel consumption over a finite set K_i of acceleration/deceleration rates described in Section 3.2. For the combination of $a_{i,1}$ and $a_{i,3}$, there are four possible cases.

Case 1: $a_{i,1}, a_{i,3} \neq 0$.

All the feasible solutions in this case can be obtained by enumerating the combination of nonzero $a_{i,1}$ and $a_{i,3}$ within set K_i and checking the following conditions:

According to (8), we have

$$h_2 \cdot v_{i,2}^2 + h_1 \cdot v_{i,2} + h_0 = 0$$

where $v_{i,2}$ is the value of the constant speed in the second phase, and

$$h_2 = a_{i,1} - a_{i,3}$$

$$h_1 = 2\cdot\left(a_{i,1}\cdot a_{i,3}\cdot T_i + a_{i,3}\cdot v_i - a_{i,1}\cdot v_{i+1}\right)$$

$$h_0 = a_{i,1}\cdot v_{i+1}^2 - a_{i,3}\cdot v_i^2 - 2\cdot a_{i,1}\cdot a_{i,3}\cdot D_i$$

To find out potential solutions, we first check whether there is a root $v_{i,2}$, satisfying

$$v_{\min}^i \le v_{i,2} \le v_{\max}^i$$

If such $v_{i,2}$ exists, we need to further make sure that $t_{i,1}$, $t_{i,2}$ and $t_{i,3}$ satisfy the following inequalities:

$$0 \le t_{i,1}, t_{i,2}, t_{i,3} \le T_i$$

where

$$t_{i,1} = \left(v_{i,2} - v_i\right) / a_{i,1}$$

$$t_{i,3} = \left(v_{i+1} - v_{i,2}\right) / a_{i,3}$$

$$t_{i,2} = T_i - t_{i,1} - t_{i,3}$$

Case 2: $a_{i,1} = 0$, $a_{i,3} \neq 0$.

In this case, the first phase of the speed profile is not existing. Therefore, $t_{i,1}=0$, and $a_{i,3}$, $t_{i,3}$ and $t_{i,2}$ can be determined as follows:

$$a_{i,3} = 0.5\cdot\left(v_{i+1} - v_i\right)^2 / \left(D_i - v_i \cdot T_i\right)$$

$$t_{i,3} = \left(v_{i+1} - v_i\right) / a_{i,3}$$

$$t_{i,2} = T_i - t_{i,3}$$

Case 3: $a_{i,1} \neq 0$, $a_{i,3} = 0$.

Similar to Case 2, in this case, the third phase $a_{i,3}$ is not existing. Therefore, $t_{i,3} = 0$, and $a_{i,1}$, $t_{i,1}$ and $t_{i,2}$ can be determined as follows:

$$a_{i,1} = 0.5 \cdot \left(v_{i+1} - v_i\right)^2 / \left(v_i \cdot T_i - D_i\right)$$

$$t_{i,1} = \left(v_{i+1} - v_i\right) / a_{i,1}$$

$$t_{i,2} = T_i - t_{i,1}$$

For Case 2 or 3, the solution is feasible only when $a_{i,3}$ or $a_{i,1}$ is within set K_i.

Case 4: $a_{i,1} = a_{i,3} = 0$.

In this case, only the second phase exists. Therefore, the following two conditions should be checked for feasibility:

$$T_i \cdot v_i = D_i$$

$$v_i = v_{i+1}$$

If the above equations hold, the rest variables are determined:

$$t_{i,1} = t_{i,3} = 0, \;\; t_{i,2} = T_i$$

The solution approach is summarised in Algorithm 2. We first enumerate all feasible solutions in the four cases (Lines 3-8). In Line 4, the function *FeasibleSolution* returns all the feasible solutions in the specified case using the corresponding method presented above. $s_j = \left(a_{i,1}^j, a_{i,3}^j, t_{i,1}^j, t_{i,2}^j, t_{i,3}^j\right)$ is the *j*-th feasible solution found in this case and $\left\{s_j\right\}$ is the set of all feasible solutions. Once all feasible solutions are found, we choose the best speed profile with the smallest fuel consumption (Lines 9-11).

Algorithm 2: Speed profile generation for edge i

```
     Input: Travelling time $T_i$, distance $D_i$ and control point speeds $v_i$ and $v_{i+1}$
     Output: Speed profile for edge $i$
1:   Initialise the feasible solution list $\mathcal{F} = \varnothing$ and the related objective value list $\mathcal{V} = \varnothing$;
2:   $k = 0$;
3:   for $case = 1:4$ do
4:       $\left\{s_j\right\} = FeasibleSolution\left(T_i, D_i, v_i, v_{i+1}, case\right)$;
5:       foreach $s \in \left\{s_j\right\}$ do
6:           $k = k + 1$;
7:           $\mathcal{F}\left(k\right) = s$;
8:           $\mathcal{V}\left(k\right) = g_i\left(s\right)$;
9:   if $k > 0$ then
10:      Find the index $k^*$ with the minimum objective value $\mathcal{V}\left(k^*\right)$;
11:      return $\mathcal{F}\left(k^*\right)$;
12:  else
13:      return $null$;
```

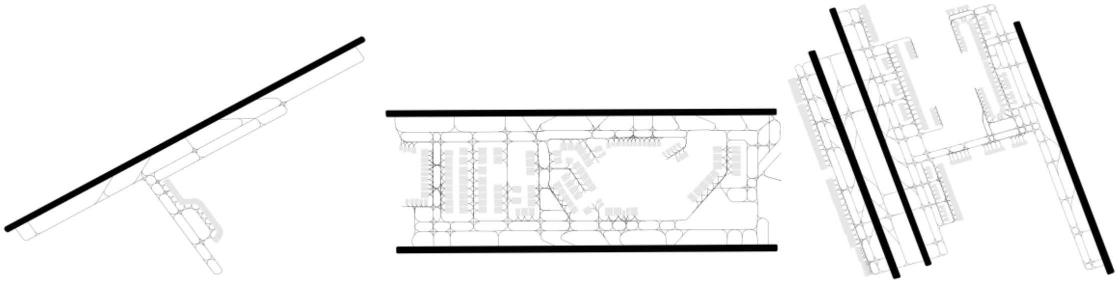
5. Experimental evaluation

5.1. Datasets and settings

To test the performance of the proposed speed profile generation approach, three baseline datasets of aircraft with time-based taxi trajectories (i.e., p_i and ATW_i) are generated using the shortest path algorithm in Zhang et al. (2017). A brief description of the datasets is presented in Table 1. The three datasets correspond to Nanjing Lukou Airport (NKG), London Heathrow Airport (LHR) and Shanghai Pudong Airport (PVG), respectively. The layouts of the taxiways are illustrated in Fig. 6. It is worth noting that the readiness time and gates/runways of the aircraft in each dataset are randomly assigned. The time-based taxi trajectories are generated sequentially according to the readiness time of aircraft. The datasets cover time-based taxi trajectories (possibly along nonstandard taxi routes) for both unimpeded and congested ground movement cases. This makes it possible to test the performance of the proposed speed profile generation approach with respect to different situations that may occur in trajectory-based taxi operations. For the three datasets described in Table 1, the average readiness time difference between consecutive aircraft is 2, 4 and 2 minutes, respectively. The NKG dataset corresponds to a high traffic density due to the limited scale of the taxiway and runway. The LHR dataset corresponds to a low traffic density, where more aircraft can move unimpededly. The PVG dataset corresponds to a medium traffic density. Additionally, different traffic densities based on the PVG dataset are derived by increasing or decreasing the readiness time difference between consecutive aircraft and generating new time-based taxi trajectories. Two derived datasets (i.e., PVG-L and PVG-H) are considered to investigate the impact of traffic densities on the proposed speed profile generation approach. For PVG-L (PVG-H), the readiness time difference between consecutive aircraft is decreased (increased) by 50% based on the PVG dataset.

Table 1. Description of the datasets.

Dataset	Number of edges in the taxiway	Number of aircraft in the dataset		
		Departure	Arrival	Total
NKG	150	523	477	1000
LHR	777	502	498	1000
PVG	1300	501	499	1000



(a) NKG (b) LHR (c) PVG

Fig. 6. Illustration of the taxiway layouts for the datasets. (a) Nanjing Lukou Airport (NKG), (b) London Heathrow Airport (LHR), (c) Shanghai Pudong Airport (PVG).

In speed profile generation, aircraft related parameters will take the representative values

Table 2. Default aircraft specifications.

Parameter	Value
Takeoff weight m	78000 kg
Rated output F_0	222.4 kN
Fuel flow at 7% F_0	0.101 kg/s
Fuel flow at 30% F_0	0.291 kg/s

presented in Table 2, without loss of generality (Chen et al., 2016b). As the proposed approach is able to generate speed profiles online, the aircraft related parameters can be determined according to the actual aircraft model in real-world applications. The maximum taxiing speed is set to 10 knots (5.14 m/s) for turning segments and 30 knots (15.43 m/s) for straight segments, respectively. The maximum acceleration or deceleration rate is set to 1 m/s^2, similar to the earlier work (Weiszer et al., 2015b). The parameters for the particle swarm optimisation algorithm (see Algorithm 1) are set according to Shi and Eberhart (1998) and Zhou et al. (2011). The desirable speed profiles for unimpeded movement are precomputed by solving (3)-(12) offline and without considering the arrival time requirements in (9).

Based on the above datasets, the proposed speed profile generation approach is tested and compared with other solution approaches. Unless otherwise mentioned, the computational time limit for each aircraft is set to 10 seconds in the experiments according to the requirement to process the initial route (ICAO, 2004). The best solution found so far will be returned when the computational time limit is reached. All experiments are run on a personal computer with 2.5GHz Intel i7-4710 CPU and 8GB RAM. The algorithms are implemented in Matlab. CPLEX 12.6.3 is the linear solver used in the proposed approach. Knitro 9.0 and LocalSolver 7.0 are the nonlinear solvers used in the comparative study.

5.2. Results

5.2.1. Classification of the aircraft in each dataset

Different ground movement cases may exist for the aircraft in each dataset. The first case is that aircraft can move unimpededly with the precomputed speed profile along the given route. For this case, it is straightforward to use the precomputed speed profile as the solution. The second case is that the precomputed speed profile can become feasible by adding holding (or buffer) time at the starting position, as described in Weiszer et al. (2015b). For this case, the proposed approach can determine the holding time in the first step (see Section 4.1). Finally, there is the third case for which it is not possible to find a feasible solution by merely adding holding time to the precomputed speed profile. For this case, feasible speed profiles need to be generated online according to the speed and timing constraints described in Section 3.2, which often require holding time as well.

We first classify the aircraft in each dataset into the three cases mentioned above. To this end, an iterative method similar to Weiszer et al. (2015b) is utilised, which iteratively increases the values of the holding time (starting from zero) with a fixed step size and checks if the precomputed speed profile is feasible with respect to the arrival time interval constraints in each iteration. If the precomputed speed profile is feasible with no holding time, the aircraft belongs to the first case. If the precomputed speed profile becomes feasible after adding a nonzero holding time, the aircraft is classified into the second case. Otherwise, the speed profile of the aircraft needs to be generated online according to the speed and timing constraints, which corresponds to the third case. The numbers of aircraft in different cases are summarised in Table 3. It can be noticed that about one third of the aircraft can move unimpededly in each dataset. For about half of the aircraft in each dataset, feasible solutions can be found by adding holding time to the precomputed speed profiles. The rest aircraft need online speed profile generation, the number of which tends to increase in higher traffic density situations.

Table 3. Summary of the numbers of aircraft in different cases.

	Dataset				
	NKG	LHR	PVG	PVG-L	PVG-H
The first case: unimpeded movement	257	364	277	375	225
The second case: adding holding time	575	599	588	581	493
The third case: generating speed profile online	168	37	135	44	282

5.2.2. Results for aircraft with added holding time

The proposed approach can automatically determine the optimal holding time for all the aircraft belonging to the second case (see Section 4.1). The computational time for each aircraft is 0.01-0.32 seconds (with an average of 0.03 seconds), which is fast enough for application in a dynamic environment (ICAO, 2004). Table 4 shows the resulting fuel consumption and holding time for each dataset. The results indicate a trend of increased holding time in higher traffic density situations. For contrast, the holding time of the iterative method mentioned in Section 5.2.1 is also presented in Table 4. Due to the restriction of the step size, the holding time of the iterative method is slightly larger than that of the proposed approach.

Table 4. Average fuel consumption and holding time for the second case.

	Dataset				
	NKG	LHR	PVG	PVG-L	PVG-H
Fuel cons. of the proposed approach (kg)	40.7	46.7	48.8	49.5	56.4
Holding time of the proposed approach (s)	70.2	62.6	60.5	30.0	248.8
Holding time of the iterative method (s)	71.1	63.6	61.6	31.1	249.8

5.2.3. Results for aircraft with online generated speed profiles

1) Result of the proposed approach

For the aircraft belonging to the third case, the proposed approach will generate feasible speed profiles according to the arrival time interval requirements and then search for improved solutions through further optimisation. The average fuel consumption and holding time for both the initial feasible speed profiles and the optimised ones are summarised in Table 5. It can be noticed that the average fuel consumption is effectively reduced after optimisation, while the resulting holding time is increased in some datasets. The increase of holding time may contribute to the fuel consumption for arrivals, but it will not increase the fuel consumption for departures, as aircraft can wait at the gate with engines off (Ravizza et al., 2014). Fig. 7 shows two typical examples of the speed profiles in this case. It can be noticed that small overshoots in the initial feasible solutions are eliminated through further optimisation, while quicker speed profiles with possibly larger holding time can also be found.

These lead to smoother and more fuel-efficient speed profiles.

Table 5. Average fuel consumption and holding time of the proposed approach for the third case.

	Dataset				
	NKG	LHR	PVG	PVG-L	PVG-H
Fuel cons. before optimisation (kg)	64.6	69.7	66.2	60.7	109.8
Fuel cons. after optimisation (kg)	61.5	67.0	63.2	58.1	105.9
Holding time before optimisation (s)	166.7	10.8	38.5	12.0	247.6
Holding time after optimisation (s)	175.9	9.2	40.3	10.4	255.8

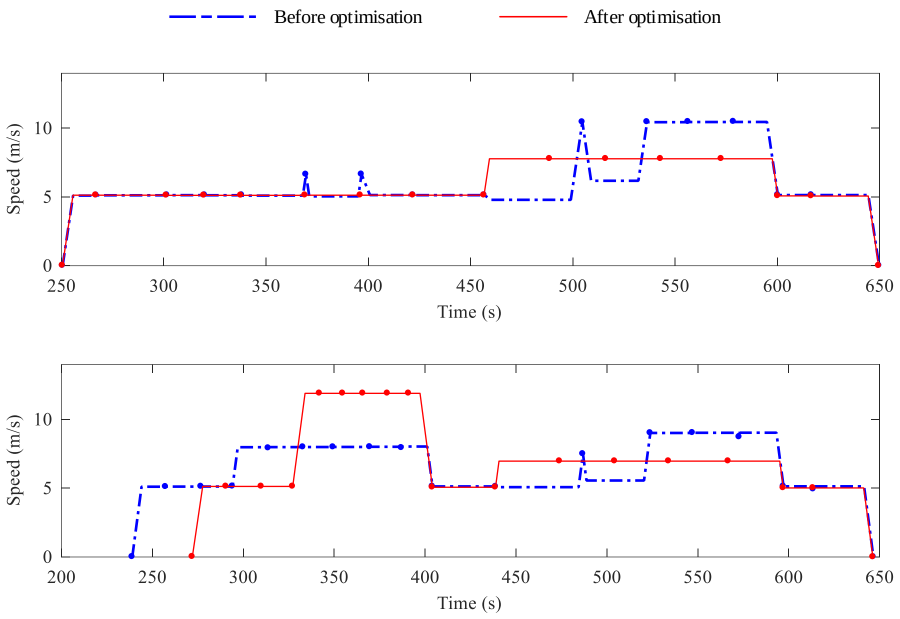


Fig. 7. Examples of speed profiles generated by the proposed approach. The dots on speed profiles correspond to control points.

To show the infeasibility issue of the precomputed speed profile for the third case and how it is resolved by the proposed approach, a typical example is presented in Fig. 8. To illustrate the problem, the starting time of the precomputed speed profile is set to the same value as the starting time of the feasible solution generated by the proposed approach. The left and right outer vertical dotted lines in Fig. 8 indicate the arrival time at control point p_8 and p_{15} for the speed profile generated by the proposed approach, respectively. The inner two vertical dotted

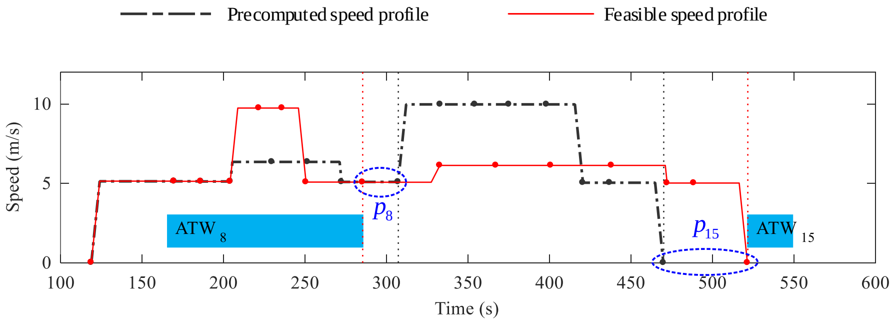


Fig. 8. An illustration of the (infeasible) precomputed speed profile and the feasible speed profile generated by the proposed approach.

lines indicate the arrival time at control point p_8 and p_{15} for the precomputed speed profile, respectively. The filled rectangles with annotations are the given arrival time intervals at control point p_8 and p_{15}, respectively. It can be noticed that the precomputed speed profile arrives too late at control point p_8 to meet ATW_8, while arriving too early at control point p_{15} to meet ATW_{15}. It is therefore not possible to find a feasible solution by only adjusting the holding time of the precomputed speed profile. The proposed approach can generate a feasible solution by flexibly adjusting the speed and arrival time at each control point while deciding a suitable holding time at the same time. The resulting arrival time at p_8 and p_{15} can meet the required arrival time intervals. However, by comparing the arrival times of the two speed profiles at the ending control point (i.e., p_{15}) we can see that the taxi time increases about 50 seconds for the feasible speed profile. This inevitably leads to increased fuel consumption compared with the precomputed speed profile, which is the cost we have to pay to resolve the infeasibility of the precomputed speed profile.

2) Comparison with the nonlinear solvers

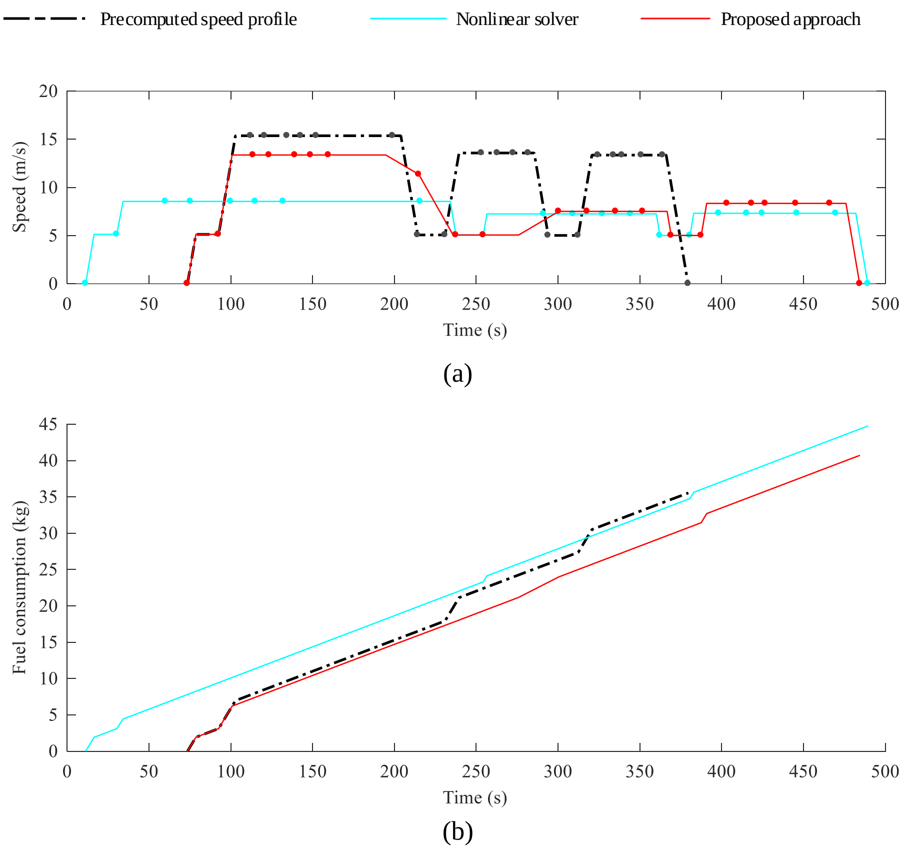
To show the advantages in generating feasible and efficient speed profiles for the third case, the proposed approach is compared with two representative nonlinear solvers (i.e., Knitro and LocalSolver), which can be utilised to solve the nonlinear optimisation problem described in (3)-(12) directly. The precomputed speed profiles are used as initial solutions for the nonlinear solvers. It is worth noting that good initial solutions are important for the nonlinear solvers to search for desirable solutions. Otherwise, the search can be trapped in inferior or infeasible regions. However, as the precomputed speed profiles may be infeasible or notably different from the optimal speed profile when the arrival time requirements are considered, the nonlinear solvers may fail to find feasible or desirable solutions in some cases.

Table 6 compares the results of the proposed approach and the first nonlinear solver (i.e., Knitro). Knitro fails to find feasible solutions for a number of aircraft in each dataset, as shown in the first row of Table 6. For fair comparison, only results of the feasible instances are presented in the following rows. The results indicate that the proposed approach and the nonlinear solver can find solutions with similar fuel consumption and taxi time. For some aircraft, however, the proposed approach can find more efficient solutions with larger holding time. A typical example is shown in Fig. 9(a). In this example, the precomputed speed profile is shown for reference, but it cannot provide a feasible solution by only adding holding time. It is worth noting that the origin of Fig. 9(a) corresponds to the readiness time of the aircraft. The solutions found by the proposed approach and the nonlinear solver both have holding

Table 6. Comparison with the nonlinear solver (Knitro).

		Dataset				
		NKG	LHR	PVG	PVG-L	PVG-H
Nonlinear solver	No. of infeasible instances	26	7	33	8	166
	Avg. fuel consumption (kg)	59.1	65.9	59.6	58.3	73.4
	Avg. taxi time (s)	647.7	727.0	649.2	634.1	804.9
	Avg. holding time (s)	174.1	12.5	32.1	12.6	128.5
Proposed approach	Avg. fuel consumption (kg)	59.1	65.9	59.6	58.3	73.3
	Avg. taxi time (s)	647.5	727.0	649.2	633.8	803.7
	Avg. holding time (s)	174.3	11.3	32.2	12.9	129.4

times. However, the movement starts earlier for the solution found by the nonlinear solver, as no holding time is added in the initial solution. This leads to longer taxi time and more fuel consumption as shown in Fig. 9(a) and Fig. 9(b), respectively.



(a)

(b)

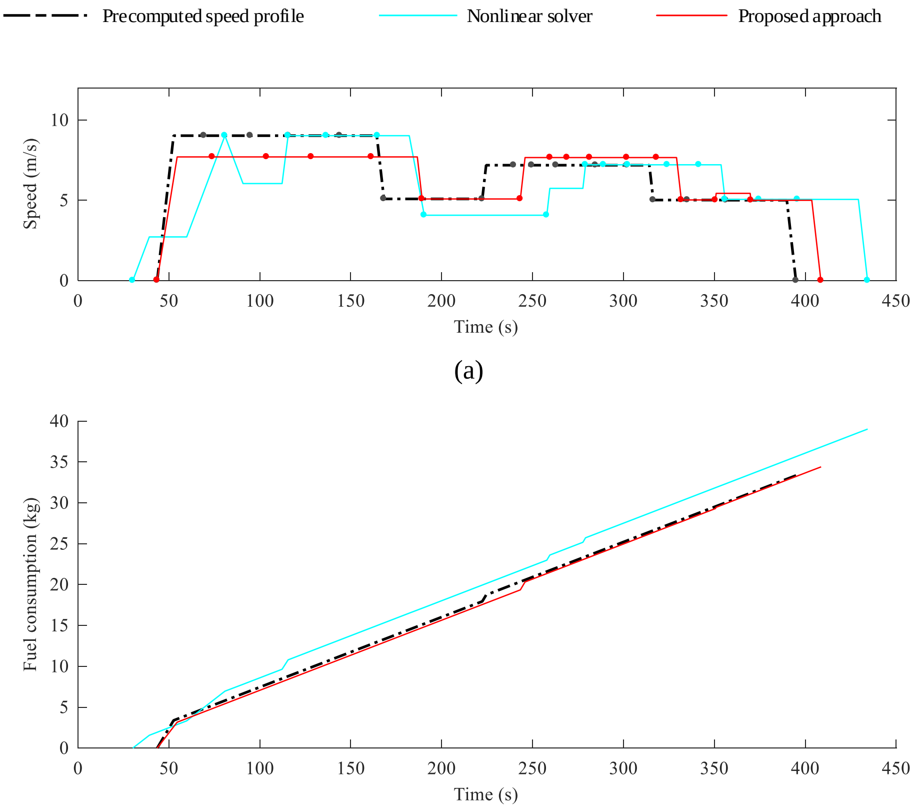
Fig. 9. A comparison of the solutions found by the proposed approach and the nonlinear solver (Knitro). (a) The speed profile, (b) The fuel consumption.

Table 7 compares the results of the proposed approach and the second nonlinear solver (i.e., LocalSolver). It is worth noting that LocalSolver is based on a metaheuristic framework. It may take a long time to find a feasible solution when the initial solution is infeasible, often much longer than the given computational time limit. For this reason, the number of infeasible instances in Table 7 is even more than those in Table 6. Moreover, the fuel consumption and taxi time of the feasible solutions found within the computational time limit also tend to be larger than those of the proposed approach due to slow convergence to the (local) optimal

Table 7. Comparison with the nonlinear solver (LocalSolver).

		Dataset				
		NKG	LHR	PVG	PVG-L	PVG-H
Nonlinear solver	No. of infeasible instances	145	26	103	35	257
	Avg. fuel consumption (kg)	59.8	58.8	57.5	56.6	65.0
	Avg. taxi time (s)	639.1	637.8	614.4	611.0	701.9
	Avg. holding time (s)	200.6	12.6	52.3	9.5	100.5
Proposed approach	Avg. fuel consumption (kg)	56.1	57.5	55.7	54.2	63.9
	Avg. taxi time (s)	610.8	626.9	598.5	589.6	687.4
	Avg. holding time (s)	220.8	11.8	58.5	12.4	107.9

solution. Fig. 10 shows a typical example of the result. In Fig. 10(a), the speed profile generated by the nonlinear solver has a smaller starting time and a larger finishing time, which means a longer overall taxi time. In addition, it also has more accelerations in the middle of the speed profile. These result in more fuel consumption as shown in Fig. 10(b).



(a)

(b)

Fig. 10. A comparison of the solutions found by the proposed approach and the nonlinear solver (LocalSolver). (a) The speed profile, (b) The fuel consumption.

3) Results under different computational time limits

Apart from providing feasible and fuel-efficient solutions in the normal cases mentioned above, another motivation of the proposed approach is to provide an immediate solution to unprecedented events. In such a case, a fast response is of paramount importance. To validate the effectiveness of the proposed approach with respect to this purpose, several more stringent computational time limits are imposed to test the proposed approach (see Fig. 11). The proposed approach can find a feasible solution with 0.3-0.9 seconds (0.4 seconds in average) for the aircraft belonging to the third case using the parameter settings described in Section 5.1, which can meet the reprocessing time requirement for tactical changes (ICAO, 2004). After finding a feasible solution, the proposed approach will further optimise the speed profile within the computational time limit. It is worth noting that the fuel consumption of the final speed profile is usually larger than that of the precomputed speed profile, due to the need of longer taxi time and/or extra acceleration/deceleration operations to meet the arrival time interval requirements in such congested situations. Fig. 11 shows the total fuel consumption increase relative to the precomputed speed profiles under different computational time limits. It can be noticed that when more computational time is available, feasible speed profiles with less fuel consumption can often be found. Moreover, the fuel consumption increase also depends on the traffic density. Larger increase of fuel consumption occurs for the dataset with higher traffic densities (e.g., NKG and PVG-H) due to the impact of increased congestion, while the fuel consumption increase for the lower density dataset (e.g., LHR and PVG-L) is much smaller.

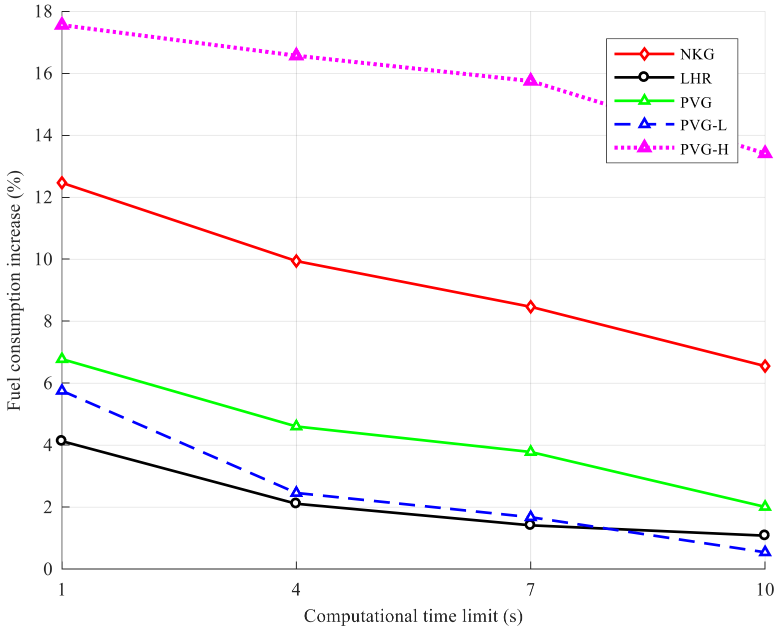


Fig. 11. The fuel consumption increase relative to the precomputed speed profile under different computational time limits.

6. Conclusions

This paper has presented a systematic approach to online speed profile generation based on time-based taxi trajectories, which proactively considers the arrival time requirements at each control point. The proposed approach models speed profiles over individual edges, and allows automatic determination of the arrival time and speed at both control and critical points. This is more flexible than existing approaches which define speed profiles on segments and assume fixed speeds at the critical points. This paper formulates the online speed profile generation problem as a nonlinear optimisation model, and proposes a matheuristic-based solution approach to find feasible solutions in real time. Improved solutions can then be obtained through further optimisation upon the initial feasible solutions. Experimental results confirm the effectiveness of the proposed speed profile generation approach with datasets of different airport layouts and traffic densities, and demonstrate its advantages in flexibility, feasibility and fuel-efficiency compared with other approaches. The results also indicate the importance of a fast computation capability for fuel-efficiency in congested situations. More efficient solutions can likely be found with increased computing power within the given computational time limit.

It is envisioned that the enhanced flexibility in deciding control point arrival times and speeds and the real-time solution capability of the proposed approach will be very useful for online adaption or re-planning of the 4DTs due to disruptions. In these cases, we are able to adjust the speed profile in some edges (instead of the entire segment) to avoid conflicts. In future, we will investigate the performance of the proposed approach with respect to different kinds of disruptions. The result will facilitate the development of more robust and resilient 4DT planning systems. Also, with increasing concerns of the aviation industry’s impact on environment, more environmentally related objectives such as emissions should be considered during speed profile generation. Preliminary results have demonstrated the advantage of estimating emissions based on detailed speed profiles (Chen et al., 2015; Sweriduk et al., 2011). Moreover, in practice there are other factors that could affect the fuel consumption such as the warming-up of the engines. More precise assessment of the benefits of trajectory-based taxi operations should therefore be conducted through e.g. pilot-in-the-loop evaluation when the supporting technologies/tools are ready. Meanwhile, the impact of the trajectory-based taxi operations on other practical approaches for reducing fuel consumption and emissions (e.g., single-engine taxiing) should also be investigated.

Finally, it should be noted that due to the introduction of arrival time intervals, the proposed speed profile generation approach may impact the coordination between the air navigation service provider and aircraft operators in the two-stage 4DT generation procedure, as the timing constraints need to be set by the air navigation service provider before speed profile generation (Cheng and Sweriduk, 2009; Cheng, 2004). This issue can be avoided if speed profiles are generated in a separate 4DT system and then displayed to the air navigation service provider and the flight deck (Okuniek et al., 2016). However, this needs electronic transmission of the 4DT information via data link communications (Bakowski et al., 2012; Wargo and Hurley, 2012). Moreover, as the utilisation of nonstandard taxi routes in trajectory-based taxi operations may affect the cognitive performance of controllers, improved decision support systems with user friendly human-machine interfaces are also needed to achieve the envisioned benefits of trajectory-based taxi operations (Carstengerdes et al., 2013).

Data access statement

Final URLs to the datasets used in this paper will be provided here.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (U1633105, 61203170), Funding of Jiangsu Innovation Program for Graduate Education (KYLX_0291), Fundamental Research Funds for the Central Universities of China (3082016NS2016061), and the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/N029496/1 and EP/N029496/2. The authors would also like to thank the anonymous reviewers for their detailed and constructive feedback.

References

Atkin, J.A.D., Burke, E.K., Ravizza, S., 2010. The airport ground movement problem: Past and current research and future directions, *Proc. of the 4th International Conference on Research in Air Transportation (ICRAT), Budapest, Hungary*, pp. 131-138.

Bakowski, D.L., Foyle, D.C., Hooey, B.L., Meyer, G.R., Wolter, C.A., 2012. DataComm in flight deck surface trajectory-based operations. *Advances in Human Aspects of Aviation*, 259.

Bakowski, D.L., Hooey, B.L., Foyle, D.C., Wolter, C.A., 2015. NextGen surface trajectory-based operations (STBO): evaluating conformance to a four-dimensional trajectory (4DT). *Procedia*

Manufacturing 3, 2458-2465.

Bakowski, D.L., Hooey, B.L., Foyle, D.C., Wolter, C.A., Cheng, L.W.S., 2013. NextGen surface trajectory-based operations: contingency-hold clearances, *Digital Avionics Systems Conference (DASC), 2013 IEEE/AIAA 32nd*, pp. 1B6-1-1B6-11.

Balakrishnan, H., Jung, Y., 2007. A framework for coordinated surface operations planning at Dallas-Fort Worth International Airport, *AIAA Guidance, Navigation and Control Conference and Exhibit*. American Institute of Aeronautics and Astronautics.

Biella, M., Hahn, K., Zilz, J., Lafferton, H., Vieten, B.D., Danello, F., 2015. Follow-the-greens: towards increased safety and efficiency by the use of airfield ground lighting (AGL). results from a SESAR real time simulation.

Carstengerdes, N., Schaper, M., Schier, S., Metz, I., Hasselberg, A., Gerdes, I., 2013. Controller support for time-based surface management - first results from a feasibility workshop, *Proc. 3rd SESAR Innovation Days*, Stockholm, Schweden.

Chen, J., Stewart, P., 2011. Planning aircraft taxiing trajectories via a multi-objective immune optimisation, *Seventh International Conference on Natural Computation (ICNC)*, Shanghai, pp. 2235 - 2240.

Chen, J., Weiszer, M., Locatelli, G., Ravizza, S., Atkin, J., Stewart, P., Burke, E., 2016a. Toward a more realistic, cost effective and greener ground movement through active routing: a multi-objective shortest path approach. *IEEE Trans Intell Transp Syst* 17(12), 3524 - 3540.

Chen, J., Weiszer, M., Stewart, P., 2015. Optimal speed profile generation for airport ground movement with consideration of emissions, *IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, Las Palmas, pp. 1797-1802.

Chen, J., Weiszer, M., Stewart, P., Shabani, M., 2016b. Toward a more realistic, cost effective and greener ground movement through active routing: part 1-optimal speed profile generation. *IEEE Trans Intell Transp Syst* 17(5), 1196 - 1209.

Cheng, V., Sweriduk, G.D., 2009. Trajectory design for aircraft taxi automation to benefit trajectory-based operations, *Proceedings of the 7th Asian Control Conference*, Hong Kong, pp. 99 - 104.

Cheng, V., Sweriduk, G.D., Yeh, J., Andre, A.D., Foyle, D.C., 2008. Flight-deck automation for trajectory based surface operations, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 18-21.

Cheng, V.H., 2004. Surface operation automation research for airport tower and flight deck automation, *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE, pp. 607-612.

Clare, G.L., Richards, A.G., 2011. Optimization of taxiway routing and runway scheduling. *IEEE Trans Intell Transp Syst* 12(4), 1000-1013.

Deau, R., Gotteland, J.-B., Durand, N., 2009. Airport surface management and runways scheduling, *Proc. 8th USA/Europe Air Traffic Management R&D Seminar*, Napa, USA.

Eurocontrol, 2012. Airport CDM implementation manual version 4.

Eurocontrol, 2013. Challenges of growth 2013: the effect of air traffic network congestion in 2035.

Eurocontrol, 2015. European ATM master plan - edition 2015.

FAA, 2012. US airport surface collaborative decision making (CDM) concept of operations (ConOps) in the near-term: application of surface CDM at United States airports. FAA, June.

FAA, 2016. Next generation air transportation system (NextGen).

Fischetti, M., Fischetti, M., 2016. Matheuristics, in: Martí, R., Panos, P., Resende, M.G.C. (Eds.), *Handbook of Heuristics*. Springer International Publishing, Cham, pp. 1-33.

Foyle, D.C., Hooey, B.L., Bakowski, D.L., Williams, J.L., Kunkle, C.L., 2011. Flight deck surface trajectory-based operations (STBO): Simulation results and ConOps implications, *Proceedings of the Ninth USA/Europe Air Traffic Management Research and Development Seminar (ATM2011)*.

García, J., Berlanga, A., Molina, J.M., Casar, J.R., 2005. Optimization of airport ground operations integrating genetic and dynamic flow management algorithms. *Ai Comm* 18(2), 143-164.

Gotteland, J.-B., Durand, N., 2003. Genetic algorithms applied to airport ground traffic optimization, *Proc. Congr. Evolutionary Computation*. IEEE, Canberra, Australia.

Haus, S., Sendobry, A., Urvoy, C., Klingauf, U., 2011. Control theoretic concept for intuitive guidance of pilots during taxiing, *Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th*. IEEE, pp. 6B3-1-6B3-14.

ICAO, 2004. Advanced surface movement guidance and control systems (A-SMGCS) manual. International Civil Aviation Organization.

Jones, D.R., Prinzel, L.J., Bailey, R.E., Arthur, J.J., Barnes, J.R., 2014. Safely conducting airport surface trajectory-based operations, *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*, pp. 7C2-1-7C2-16.

JPDO, 2007. Concept of operations for the next generation air transportation system.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization, *Proc. of the IEEE International Conference on Neural Networks*, pp. 1942-1948 vol.1944-1948 vol.1944.

Lesire, C., 2010. Iterative planning of airport ground movements, *Proc. of the 4th international conference on research in air transportation (ICRAT 2010)*, Budapest, Hungary, pp. 147-154.

Marín, Á.G., 2006. Airport management: taxi planning. *Ann Oper Res* 143(1), 191-202.

Marín, Á.G., 2013. Airport taxi planning: Lagrangian decomposition. *J Adv Transport* 47(4), 461-474.

Montoya, J., Wood, Z., Rathinam, S., Malik, W., 2010. A mixed integer linear program for solving a multiple route taxi scheduling problem, *Proceedings of the AIAA Guidance, Navigation, and Control Conference*. American Institute of Aeronautics and Astronautics, Toronto, Canada.

Nikoleris, T., Gupta, G., Kistler, M., 2011. Detailed estimation of fuel consumption and emissions during aircraft taxi operations at Dallas/Fort Worth International Airport. *Transportation Research Part D: Transport and Environment* 16(4), 302-308.

Okuniek, J.N., Gerdes, I., Jakobi, J., Ludwig, T., Hooey, B.L., Foyle, D., Jung, Y.C., Zhu, Z., 2016. A concept of operations for trajectory-based taxi operations, *16th AIAA Aviation Technology, Integration, and Operations Conference*, p. 3753.

Ravizza, S., Atkin, J.A., Burke, E.K., 2014. A more realistic approach for airport ground movement optimisation with stand holding. *Journal of Scheduling* 17(5), 507-520.

Roling, P.C., Visser, H.G., 2008. Optimal airport surface traffic planning using mixed-integer linear programming. *Int. J. Aero. Eng.* 2008, 1-11.

Shi, Y., Eberhart, R., 1998. A modified particle swarm optimizer, *IEEE international conference of evolutionary computation*, pp. 69-73.

Smeltink, J.W., Soomer, M.J., 2004. An optimisation model for airport taxi scheduling, *Proceedings of INFORMS Annual Meeting*, Denver, USA.

Sweriduk, G., Cheng, V., Foyle, D., 2011. Models for aircraft surface operations environmental analysis, *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, including the AIAA Balloon Systems Conference and 19th AIAA Lighter-Than*, p. 6880.

Wargo, C.A., Hurley, P., 2012. Data link requirements analysis and benefits of a surface trajectory-based operation, *Aerospace Conference, 2012 IEEE*. IEEE, pp. 1-10.

Weiszer, M., Chen, J., Locatelli, G., 2015a. An integrated optimisation approach to airport ground operations to foster sustainability in the aviation sector. *Appl Energ* 157, 567-582.

Weiszer, M., Chen, J., Ravizza, S., Atkin, J., Stewart, P., 2014. A heuristic approach to greener airport ground movement, *2014 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3280-3286.

Weiszer, M., Chen, J., Stewart, P., 2015b. A real-time active routing approach via a database for airport surface movement. *Transportation Research Part C: Emerging Technologies* 58, 127-145.

Zhang, T., Ding, M., Wang, B., Chen, Q., 2016. Conflict-free time-based trajectory planning for aircraft taxi automation with refined taxiway modeling. *J Adv Transport* 50(3), 326-347.

Zhang, T., Ding, M., Zuo, H., 2017. An improved routing algorithm for efficient airport ground movement.

Zhou, D., Gao, X., Liu, G., Mei, C., Jiang, D., Liu, Y., 2011. Randomization in particle swarm optimization for global search ability. *Expert Syst Appl* 38(12), 15356-15364.