An extended memetic algorithm for multiobjective routing and scheduling of airport ground movements with intermediate holding

Tianci Zhang
College of Automobile and Traffic Engineering
Nanjing Forestry University
Nanjing, China
tczhang@njfu.edu.cn

Lilla Beke
School of Engineering and Materials Science
Queen Mary University of London
London, UK
l.beke@qmul.ac.uk

Songwei Liu
School of Engineering and Materials Science
Queen Mary University of London
London, UK
songwei.liu@qmul.ac.uk

Michal Weiszer
School of Engineering and Materials Science
Queen Mary University of London
London, UK
m.weiszer@qmul.ac.uk

Jun Chen*
School of Engineering and Materials Science
Queen Mary University of London
London, UK
jun.chen@qmul.ac.uk

***Abstract*—Routing and scheduling of airport ground movements poses a critical issue for efficient surface operations. For real-world applications, multiple objectives should be considered, leading to a multigraph representation of the search space. Meanwhile, intermediate holding is often needed to a) release availability of scarce resources such as runways and gates for more cost-effective routing and scheduling, b) provide additional solutions in the speed profile database that can be used during routing and scheduling, and c) keep airport ground movements functional during disruptive events that may paralyse part of the taxiway network. This paper presents an extended multiobjective memetic algorithm upon the multigraph model to search for desirable solutions with intermediate holding. The performance of the proposed algorithm is examined with problem instances of different airport layouts. The results demonstrate prominent savings in both time and fuel costs compared with solutions without intermediate holding.**

Keywords—Airport ground movements, intermediate holding, memetic algorithm, multigraph, multiobjective optimisation

I. INTRODUCTION

With the continuous increase of air traffic demand in recent years, more efficient utilisation of scarce airport resources such as runways and gates has attracted lots of research effort [1]–[3]. The movement of aircraft on taxiways is crucial for efficient use of airport resources as it is the link between runway and gate operations [4], [5]. Aircraft should be routed on the taxiways considering multiple conflicting objectives such as minimising time and fuel costs [6]. Minimising the time cost can help to reduce the occupancy of airport resources so as to increase the overall operational efficiency. However, extra fuel consumption could be needed in the meantime since aircraft has to use a larger acceleration rate to reduce taxiing time. This will increase the economic cost and lead to environmental concerns [7], [8]. Hence it is necessary to handle the conflicting objectives with multiobjective optimisation approaches, providing Pareto optimal solutions for the controllers to make final decisions.

*Corresponding author. This work is supported in part by the Engineering and Physical Sciences Research Council under grant EP/N029496/1, EP/N029496/2, EP/N029356/1, EP/N029577/1 and EP/N029577/2 and the research start-up fund from NJFU (163106035, CX2019019).

Aircraft will be routed from the runway to the gate or vice versa during taxiing following an assigned path along the taxiway. Accurate calculation of the costs depends on the detailed description of movements on the path, which is usually described by speed profiles [9]. Meanwhile temporal information such as arrival time at intermediate waypoints can also be determined with speed profiles, which is useful for avoiding conflicts with other aircraft. The problem of finding desirable path and speed profile combinations is referred to as routing and scheduling.

Early work deals with routing and scheduling based on a simple graph model of airport taxiway layout [10]. A node of the graph represents an intersection or intermediate waypoint of the taxiway. An edge of the graph is the taxiway between two connected nodes. In a simple graph, there can only be one edge between two nodes. To avoid conflicts, each edge can only be occupied by one aircraft at a time. The time interval in which an edge is occupied will be reserved. The remaining intervals along the time axis indicate time windows in which an edge is available. The simple graph model with time windows associated to each edge provides a basis for avoiding conflicts between aircraft. Despite the convenience of constructing the simple graph model and its effectiveness in handling conflicts, heuristic solution strategies have to be adopted which separate routing and scheduling, as we have to assume constant speed in routing and consequently can only explore a very limited search space for path and speed profile combinations.

To avoid the abovementioned disadvantage of the simple graph based approach, a multigraph model for routing and scheduling is proposed in more recent work [11]. In a multigraph, there can be more than one edge between two nodes, corresponding to different candidate speed profiles. Desirable candidate speed profiles can be generated offline and saved in a database for more efficient computation at runtime [12]. This also enables time consuming calculation of the Pareto front with respect to multiple conflicting objectives. In this way, routing and scheduling can be represented by a multiobjective shortest path problem (MSPP) on the multigraph. Time windows upon each edge of the taxiway will act as additional constraints to ensure conflict-free solutions.

Enumerative methods such as label setting and label correcting algorithms prove to be among the most effective

2022 IEEE Congress on Evolutionary Computation (CEC) | 978-1-6654-6708-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/CEC55065.2022.9870239

approaches for MSPP [13], [14]. They extend the labeling solution techniques for single objective shortest path problems such as the Dijkstra's algorithm [15] or the A* algorithm [16] to multiple objectives. When certain conditions such as additivity of the cost are met, they can find the whole Pareto front exactly. However, as the complexity of the specific variant of MSPP on the multigraph is NP-hard, there could be problem instances which are not tractable for exact solution approaches in reasonable computational time. Moreover, when search for routing, the speed profile to a following node depends on the speed profile to the current one. The costs are often nonadditive as a result [11]. This also prevents satisfactory performance of enumerative methods. Additionally, we also need to handle constraints described by time windows on each edge of the taxiway. This can make the preserved partial solutions infeasible during the search of enumerative methods, resulting in failure in finding a satisfactory solution.

Metaheuristics such as the genetic algorithm (GA) is a more popular choice as they can easily handle the nonadditivity and deal with additional constraints more flexibly [17], [18]. Among the various metaheuristic approaches, memetic algorithms have demonstrated superiority in solving many hard optimisation problems of engineering applications [19]. They can exploit problem knowledge for individual improvement on a local basis in additional to populational cooperation and competition as in conventional GAs. In the previous work, a memetic algorithm combining GA and local search was proposed to improve the convergence properties of GA by exploiting the search space around a given solution [20], [21], which can successfully solve the routing and scheduling problem with the multigraph model for real-world problem instances. However, when feasible solutions cannot be found due to the time window constraints, aircraft has to be postponed before starting taxiing. In practice, there is a need to allow intermediate holding. For example, arrival aircraft usually cannot hold at the runway exit before taxiing. Departing aircraft should also release the gate when required by other aircraft. Intermediate holding will enlarge the solution space from the optimisation viewpoint, leading to potentially more cost-effective solutions than postponing at the beginning. Adding intermediate holding is effectively equivalent to having not only nondominated speed profiles but also dominated ones in the speed profile database, shedding light on how to construct a more complete offline database. Furthermore, as envisioned in the European Air Traffic Management (ATM) Master Plan [22], a fundamental digital transformation in ATM will increase levels of automation. This fundamental shift calls for a decision support system that is more resilient against unprecedented events at all layers of traffic management so that any disruptive events would only cause controllable disruption to runway and gate operations.

In light of the above discussions, this paper presents an extended memetic algorithm for multiobjective routing and scheduling of airport ground movements with intermediate holding, in order to find more realistic and cost-effective solutions. It follows a similar algorithmic design approach to [21]. The main contributions are listed in the following:

a) Based on the multigraph model for routing and scheduling, an encoding strategy is designed to enable holding at intermediate nodes. It extends the encoding strategy proposed in [21] with an additional list for holding times. Suitable mutation and crossover operations are also devised to handle the holding times. An integrated parameter tuning approach is applied to configure the algorithm.

b) The effectiveness and advantages of the proposed algorithm is demonstrated using problem instances of different airport layouts. An analysis of the holding time and cost savings compared to solutions without intermediate holding is presented. The performance in response to disruptive events is also examined.

The rest of the paper is organised as follows: Section II describes the airport layout model, which provides the basis to apply time window constraints, and the multigraph model for routing and scheduling. Section III presents an overview of the proposed memetic algorithm and the methodological ingredients, including the encoding strategy, mutation and crossover operations for population evolution, local search method and holding time perturbation approach. Section IV presents experimental results and discussions. Section V concludes the paper and presents future research directions.

II. Airport Layout and Multigraph Model

A. The airport layout model

The task of routing is to find feasible paths from the runway exit to the gate or vice versa. A simple graph is constructed directly according to the airport layout, which is the basis for the multigraph model with speed profiles.

In this paper, a directed graph $G_0 = (V_0, E)$ is used to model the airport layout, which contains the geographical information of taxiways. Nodes V_0 correspond to runway exits, gates, taxiway intersections or other intermediate positions on the taxiway. There will be an edge $e_{i,j} \in E$ between nodes i and j $(i, j \in V_0)$ if they are connected by a taxiway section without other intermediate nodes. The length of an edge is no longer than the minimum safe separation between two aircraft [21].

In addition to the geographical information, there are also time constraints on each edge of the graph to avoid conflicts between aircraft. Specifically, a set of time windows F_e will be defined for each edge $e \in E$, indicating all the time intervals during which the corresponding taxiway section is available. The time windows will be maintained dynamically. When a routing and scheduling solution is assigned to an aircraft, the path and corresponding schedules to reach each node along the path will be used to calculate the start time t_r and end time t_d on each taxiway section. Then the time interval $[t_r, t_d]$ will be removed from the time windows F_e for the corresponding edge. Moreover, for nearby edges with a distance smaller than the minimum separation to the current edge, the time windows of the nearby edges will also be updated by removing the time interval $[t_r, t_d]$.

B. The multigraph model

A multigraph model $G = (V, A)$ is constructed based on the layout model G_0 and speed profiles as follows.

A group of consecutive edges in the layout model forms a straight segment, if the angle between two neighbouring edges is small enough (<30 degrees [12]). If the angle of an edge is larger than 30 degrees with respect to the predecessor edge, it will form a turning segment. Speed profiles are generated separately for straight and turning segments.

For each segment, multiple Pareto optimal speed profiles are precomputed depending on the weight category of aircraft with respect to the conflicting objectives [4], [12]. The speed

profile contains acceleration, constant speed, deceleration and rapid deceleration phases where the duration of each phase determines the taxiing time and fuel cost of the speed profile. Consequently, there will be more than one speed profile connecting two endpoints of a segment. Each speed profile forms an arc of G, making it a multigraph. Note that the concept of arc is used for G to avoid confusion with the concept of edge in the layout model G_0. An arc of the multigraph model G is denoted as $(v,w)^k$, where v, w are the endpoint nodes and k is the parallel arc index indicating arc k connecting v and w. The set of arcs connecting v and w is $A_{(v,w)} = \{(v,w)^k \mid 1 \le k \le N\}$, with N being the number of precomputed speed profiles for each segment. For a given path passing v and w, the predecessor edge determines which arcs from $A_{(v,w)}$ are feasible. For arc $(v,w)^k$, there is a cost vector $c_{(v,w)^k} = (c_1, c_2)$ determined by the corresponding speed profile, where c_1 and c_2 correspond to the taxiing time and fuel cost, respectively. An illustration of the layout and multigraph models is presented in Fig. 1 [11].

III. METHODOLOGY FOR ROUTING AND SCHEDULING

A. Problem description

The goal of the routing and scheduling problem is to find the set of Pareto optimal trajectories Θ for a given aircraft with start node v_{O} and end node v_{D}, with respect to the objectives of taxi time and fuel consumption. A trajectory $\theta \in \Theta$ describes the movement of the aircraft in time and space by specifying a sequence of arcs in the multigraph G and the holding times t_i $(1 \le i \le |\theta| - 1)$ before entering the ith arc:

$$\theta = \left\langle (v_{\mathrm{O}}, v_2)^{k_1}, t_1 \right\rangle, \left\langle (v_2, v_3)^{k_2}, t_2 \right\rangle, \ldots, \left\langle \left(v_{|\theta|-1}, v_{\mathrm{D}}\right)^{k_{|\theta|-1}}, t_{|\theta|-1} \right\rangle.$$

In order to ensure the feasibility of a trajectory, the following constraints should also be satisfied:

a) For each intermediate arc of the trajectory, the arc index should be consistent with the predecessor edge in the trajectory. Specifically, if we denote the predecessor edge of arc $(v_i, v_{i+1})^{k_i}$ in trajectory θ as $e = pred((v_i, v_{i+1}) | \theta)$, then k_i should be within $I(e, (v_i, v_{i+1}))$, which is the set of parallel arc indices connecting node v_i and v_{i+1} befitting predecessor edge e. The set $I(e, (v_i, v_{i+1}))$ is determined by collecting the indices of arcs in $A_{(v_i, v_{i+1})}$ which can be accessed from predecessor edge e. This ensures a realistic speed profile along the whole trajectory.

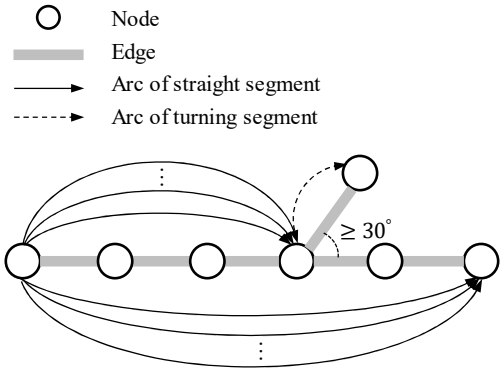


Fig. 1. Illustration of the layout and multigraph models.

b) Time windows should be met by the trajectory. For each edge e contained in the trajectory θ, the times to arrive at and leave e are determined by the speed profiles and holding times in θ. Both the arrival and leaving times should lie within one of the time windows of F_e. This ensures conflict-free occupancy of taxiway sections in the existence of other taxiing aircraft.

c) The trajectory should not pass a node more than one time, which avoids loops in the path.

For a feasible trajectory θ, the cost vector $c(\theta)$ can be calculated by accumulating the arc costs considering the impact of holding:

$$c(\theta) = \sum_{\theta} \hat{c}_{(v_i, v_{i+1})^{k_i}},$$

where $\hat{c}_{(v_i, v_{i+1})^{k_i}}$ is the cost vector of arc $(v_i, v_{i+1})^{k_i}$ summed with the extra time and fuel costs due to additional deceleration and acceleration required for implementing intermediate holding.

B. The proposed memetic algorithm

We extend the algorithm proposed in [21] to also consider solutions with intermediate holding. For completeness, a high-level description of the algorithm is given, with a focus on the changes required to enable intermediate holding.

1) The overall algorithm

The proposed memetic algorithm can be described by the pseudocode in Algorithm 1. In Line 1, heuristic initialisation is adopted for the populations as described in [21]. Then the mutation (Line 3), crossover (Line 4), local search (Line 5) and holding time perturbation (Line 6) operations are iteratively applied to create new offsprings. Each operation is conducted with a specified probability. Mutation is applied before crossover to increase the diversity of candidate solutions. The multiobjective evolutionary strategy is based on fast nondominated sorting and binary tournament selection with crowded-comparison [23] (Line 7).

After a trajectory is selected for the current aircraft from the returned solutions, the time windows of each edge along

Algorithm 1. Pseudocode of the proposed algorithm for routing and scheduling with intermediate holding.

input: Layout graph G_0, multigraph G, aircraft with start node n_{O} and end node n_{D}, and parameters for the memetic algorithm.

output: The set of Pareto optimal trajectories from n_{O} to n_{D}.

1: *pop* ← Initialisation(*popsize*)
2: **while** not converged and within time budget:
3: *newpop* ← Mutation(*pop*)
4: *newpop* ← Crossover(*newpop*)
5: *newpop* ← LocalSearch(*newpop*)
6: *newpop* ← HoldTimePerturb(*newpop*)
7: *pop* ← Selection(*pop, newpop*)
8: **end while**
9: **return** the current candidate solutions in *pop*

the trajectory will be updated by removing the corresponding traversal time interval. In this way, conflict-free solutions can be found for the forthcoming aircraft.

2) Encoding and decoding strategies

In the proposed algorithm, the trajectory should first be encoded into a form suitable for evolutionary computation. The direct variable length representation [24] based trajectory encoding strategy is used in this paper. It specifies a trajectory with a list of alternating nodes and parallel arc indices, and a separate list of holding times corresponding to the arcs. In this way, the trajectory θ described in Section III-A can be encoded as

$$enc(\theta) = \left\langle \left(v_{\mathrm{O}}, k_1, v_2, k_2, \ldots, v_{|\theta|-1}, k_{|\theta|-1}, v_{\mathrm{D}} \right), \left(t_1, t_2, \ldots, t_{|\theta|-1} \right) \right\rangle .$$

The decoding of a solution is straightforward for the abovementioned encoding strategy if the solution is feasible. However, if the solution has loops, the encoded solution cannot be decoded to a legal trajectory. In this case only the part from the start node to the position where the first loop occurs will be preserved and a penalisation will be applied for not reaching the end node.

3) Mutation and crossover

After the population is initialised using the abovementioned strategy, mutation and crossover will be applied iteratively during the search for desirable solutions.

A random walk based mutation is used which first selects a node from a candidate solution randomly, and then changes the path starting from the selected node by a random walk. The predecessor edge at the selected node will be considered in the random walk. This ensures a physically feasible trajectory after mutation. Note that the holding time for the newly found part of mutated solution will be set to zero.

After mutation, crossover is applied to the current population. The crossing site is selected as a node present in both parents. Two children are then generated by swapping the node and parallel arc sequences of the parents after the selected crossing site. The holding times of the arcs in the swapped parts are set to zero, because the timing of their traversal has generally changed. It should be noted that illegal paths with loops may occur after crossover. A repair mechanism will be applied to eliminate loops. In case the predecessor constraint is violated after repairing, the part up to the violation node will be taken as the child, and a penalisation will occur in fitness evaluation.

4) Local search

A specifically designed local search approach is applied in the proposed memetic algorithm. In each iteration, we will use local search to improve candidate solutions with some probability. A modified Dijkstra's algorithm is utilised in the local search to find the shortest path between two randomly chosen nodes of a candidate solution, based on a scalarisation of the cost vectors with random weights. The modification includes ignoring arcs in the search process that would violate the time windows. The newly found path will replace a part of the original candidate solution. In case no feasible path is found in local search, the partial solution up to the node where local search is performed will be returned as the new candidate solution, and a large penalisation will be added in the fitness function for not reaching the end node.

5) Holding time perturbation

At initialisation, holding times are set to zero. To search for suitable holding times, an additional operator of holding time perturbation is incorporated in the proposed algorithm. During the search, the holding time perturbation operator is applied after mutation and crossover in each iteration, where three operations are executed on the holding times: (1) a holding time interval of 30, 60 or 120 seconds is inserted at a random position, (2) a randomly chosen holding time interval is shifted forward to the previous segment, and (3) a randomly chosen holding time interval is decreased by 5 seconds. For the investigated problem, it is good to keep the lengths of holding short to keep the search space tractable and reduce the cost incurred by holding.

6) Fitness function and constraint handling

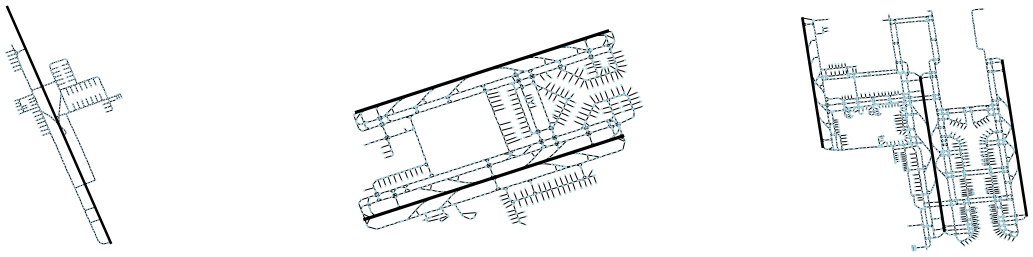
The fitness function is defined as the sum of the costs $c(\theta)$ and penalties applied for constraint violations. We apply static penalties for solutions not reaching the end node or violating the time window constraints as in [21]. In each case, the penalty is defined as a penalty base p_0 multiplied by predefined penalty weights. The penalty base p_0 depends on the maximum cost of arcs in the multi-graph *G* or the number of violated time windows. For the case of not reaching the end node, the fuel cost will be penalised with a larger penalty weight. While for time window violations, the time cost will be penalised more. This helps to make the solutions not biased to a specific objective.

IV. Experimental Results

A. Datasets and parameter settings

Three datasets of real arrival and departure flights are used for the experimental studies of the proposed algorithm, which correspond to the Doha International Airport (DOH), Hong Kong International Airport (HKG) and Beijing Capital International (PEK), respectively. The layout graphs of the airports are shown in Fig. 2.

The datasets contain flights on 16 Mar. 2014 (17:00-23:00) for DOH, 17 Jan. 2017 (0:00-24:00) for HKG and 9 Jul. 2014 (9:00-14:00) for PEK. The taxiway layout and flight instances are obtained from freely-available data on OpenStreetMap.org and FlightRadar24.com respectively, and further processed by the GM tools presented in [25]. The landing/pushback time, gate/runway exit and weight category of aircraft are specified for each flight. The information of the datasets are summarised



(a) (b) (c)

Fig. 2. The layout graphs. (a) DOH, (b) HKG, (c) PEK.

in Table I. Moreover, artificial datasets based on HKG will also be used to simulate higher level of traffic densities than in nominal operations. Specifically, the artificial dataset with more aircraft will be denoted by HKG plus the increased percent. For example, the artificial dataset with 25% more aircraft in the same time duration will be referred to as HKG25.

The speed profile database is generated using the method presented in [12]. Aircraft weight categories induce differences in speed profile generation, as they affect movement parameters [26]. For each segment, 20 Pareto optimal speed profiles are generated offline and saved in the database. At runtime, a specific number of evenly distributed speed profiles for each segment will be selected from the database to balance the computational cost and solution quality [11], [21].

The aircraft are routed sequentially according to the readiness time for taxiing. The economic cost is used to select a preferred trajectory from the candidate solutions so that time window is updated before the next aircraft is routed. The economic cost weights for time and fuel are $w_1 = 0.469$ and $w_2 = 0.710$ respectively according to [26], which can be set to other values without the loss of generality.

The probabilities for mutation, crossover, local search, holding time perturbation and the population size of the proposed algorithm are presented in Table II, which are tuned with the irace package [27]. Other algorithm specific parameters are set according to [21]. All the experiments are run on a personal computer with 3.70GHz Intel Core i9-10900K CPU and 64GB RAM.

B. Results

1) Application study

We first investigate the performance of the proposed algorithm with an application study which resembles the operations in a nominal application scenario. In this scenario, aircraft in each dataset are routed sequentially from the first one to completion according to the readiness time. The time windows are updated once a trajectory has been selected for the current aircraft.

TABLE I. THE INVESTIGATED DATASETS.

	DOH	HKG	PEK
Nodes	434	1309	3194
Edges	436	1491	3928
Gates	55	160	286
Runway exits	14	38	53
Arrivals	103	290	185
Departures	77	216	164
Total flights	180	506	349

TABLE II. PARAMETERS OF THE PROPOSED ALGORITHM.

Parameter	Value
Population size	120
Mutation prob.	0.14
Crossover prob.	0.96
Local search prob.	0.02
Hold time pert. prob.	0.27

The results on the three datasets with the time budget of 10 seconds are presented in Table III. This time budget is used according to the decision time requirement described in [28]. Table III presents the average number of Pareto optimal solutions found by the algorithm for each aircraft, the average holding time at the origin before taxiing, the average holding time for the whole trajectory, and the average time, fuel and economic costs for the selected trajectories. The baseline results of the algorithm without enabling intermediate holding are also presented for comparison. For clarity, the baseline is implemented by setting the holding time perturbation probability to zero to eliminate intermediate holding. If no feasible solution is found, it will postpone the start time of taxiing and try to solve the problem again. This process will be repeated each time no feasible solution can be found in the given time budget.

The results in Table III demonstrate the effectiveness of the proposed algorithm on airports with different layout complexities. Compared with the baseline, the average time, fuel and economic costs on the DOH and HKG datasets can be reduced with the proposed algorithm by enabling intermediate holding and flexibly determining the holding time at the origin. Moreover, as less holding time is required at the origin, more efficient use of the runways and the gates is enabled. However, the average number of feasible solutions is less than the baseline due to the fact that the proposed algorithm usually requires more computational time to converge with the holding time perturbation operations. For the same reason, the benefit of the proposed algorithm is not prominent within the specified time budget of 10 seconds for more complex airport layouts such as PEK.

To demonstrate the impact of computational time, Table IV presents the results with increased time budgets on the PEK dataset. It can be observed that when the time budget is enlarged, better solutions can be found by the proposed

TABLE III. RESULTS ON THE THREE DATASETS.

	DOH		**HKG**		**PEK**	
	Prop.	**Base.**	**Prop.**	**Base.**	**Prop.**	**Base.**
#Solutions	20.1	**20.3**	5.7	**6.0**	5.0	**5.3**
Hold time at origin (s)	**1.5**	4.3	**1.9**	2.4	1.7	1.7
Overall hold time (s)	**1.5**	4.3	**2.3**	2.4	2.3	**1.7**
Time cost (s)	**184.4**	190.8	**264.2**	265.0	269.9	**267.5**
Fuel cost (kg)	**85.7**	87.5	**132.9**	132.9	87.2	**86.7**
Eco. cost	**147.3**	151.6	**218.3**	218.7	188.5	**187.0**

TABLE IV. RESULTS ON PEK WITH DIFFERENT TIME BUDGETS.

	10s		**25s**		**50s**	
	Prop.	**Base.**	**Prop.**	**Base.**	**Prop.**	**Base.**
Hold time at origin (s)	1.7	1.7	**1.0**	1.7	**1.2**	2.1
Overall hold time (s)	2.3	**1.7**	**1.1**	1.7	**1.2**	2.1
Time cost (s)	269.9	**267.5**	**263.3**	264.1	**260.5**	262.2
Fuel cost (kg)	87.2	**86.7**	**85.0**	85.1	**84.0**	84.5
Eco. cost	188.5	**187.0**	**183.8**	184.3	**181.8**	183.0

algorithm. The average time, fuel and economic costs can be decreased, and the benefit of the proposed algorithm becomes clearer (note, the economic costs in the table are not in any monetary units). Therefore, for applications on larger airports such as PEK, more time budget will have to be spared to ensure improved performance of the proposed algorithm.

In addition to the airport layout complexity, the level of traffic densities also impacts the performance of the proposed algorithm. Generally, higher level of traffic densities leads to more competitive use of the taxiways, hence will be more challenging to find desirable solutions. To demonstrate the advantage of the proposed algorithm for higher level traffic densities, tests on the artificial datasets of HKG25, HKG50 and HKG75 are performed. The results of average holding times and costs are presented in Table V. The baseline approach cannot hold during taxiing and sometimes have to use lower speeds on the entire segment to meet the time window of an edge within the segment. In contrast, the proposed algorithm can properly add holding time during taxiing to meet the time window without dropping the speeds on other edges. Moreover, it can automatically assign the holding time at the origin to appropriate values. These result in more cost-effective solutions in all the investigated cases.

To show the performance of the proposed algorithm, a detailed illustration of the economic cost relative to the baseline on the HKG75 dataset is presented in Fig. 3. For clarity, the corresponding time and fuel costs are also presented. The costs are accumulated for all the aircraft routed so far as indicated in the horizontal axis. It can be observed that for the initial few aircraft, as the constraining effect of the time windows is negligible, there is no need for intermediate holding to avoid conflicts. The proposed algorithm could find inferior solutions due to the extra computational cost of holding time perturbation within the given time budget. However, when the number of routed aircraft is large enough as is common in real-world applications, tighter time window constraints exist, and the savings in the economic cost become more prominent compared with the baseline. The Wilcoxon rank sum test upon the two economic costs has a p-value of 0.04, indicating the proposed algorithm has significant improvement.

2) Algorithmic study

The results in the previous section demonstrate the performance of the solution algorithms for real-world applications from a statistical perspective. However, the time windows could be different for a specific aircraft when different algorithms are used, since different solutions for previously routed aircraft might be generated. In this section, we further compare the proposed algorithm and the baseline algorithm without intermediate holding under the same time window constraints, in order to provide a fair comparison from

TABLE V. RESULTS ON ARTIFICIAL DATASETS WITH HIGHER LEVEL OF TRAFFIC DENSITIES.

	HKG25		HKG50		HKG75	
	Prop.	**Base.**	**Prop.**	**Base.**	**Prop.**	**Base.**
Hold time at origin (s)	**2.7**	4.0	**5.6**	9.3	**12.8**	14.9
Overall hold time (s)	**3.5**	4.0	**7.9**	9.3	**14.0**	14.9
Time cost (s)	**282.3**	282.6	**296.6**	303.5	**315.6**	316.1
Fuel cost (kg)	**140.9**	141.7	**152.7**	154.7	**156.0**	158.8
Eco. cost	**232.5**	233.2	**247.5**	252.1	**258.8**	261.0

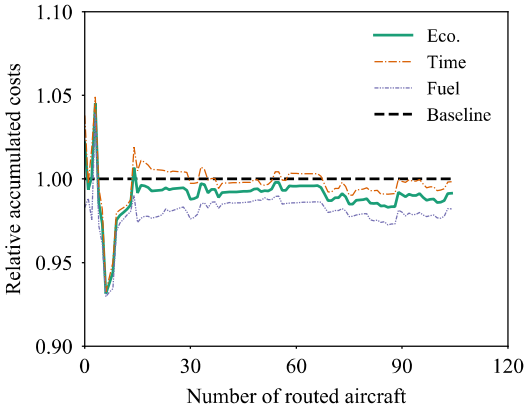


Fig. 3. The relative accumulated costs on HKG75.

the algorithmic perspective. The test is implemented on the HKG dataset by routing each aircraft using the two algorithms respectively. For each aircraft, the same time windows are used for either algorithm to search for desirable solutions. After both algorithms have finished searching, a trajectory is selected from the solutions returned by the last applied algorithm. Then the time windows are updated according to the selected trajectory before dealing with the next aircraft.

Fig. 4 shows the resulting costs using the two algorithms with the time budget of 10 seconds. For clarity, the relative economic costs with respect to the baseline results are presented for each individual aircraft. We find that for 3/4 of aircraft, the proposed algorithm can found solutions superior or identical to the baseline.

Fig. 5 shows a typical example for which an improved solution is found by the proposed algorithm. For clarity, the readiness time of the aircraft is aligned to zero in the horizontal axis. The vertical axis represents the labels of edges along the selected trajectory. It can be observed that the baseline approach cannot find a feasible solution without postponing the start time by one minute. In contrast, the proposed algorithm is able to find a desirable path with suitable initial and intermediate holding time.

It should be noted that there are cases for which the proposed algorithm find inferior solutions in the abovementioned test. An analysis of the results demonstrates that inferior solutions are mainly due to nonconvergence of the algorithm within the limited time budget. As a result, trajectories with larger economic costs could be returned. When we increase the time budget to ensure convergence, better solutions can be found using the proposed algorithm. Table VI presents 1/10 randomly selected examples of problem instances for which inferior solutions are returned by

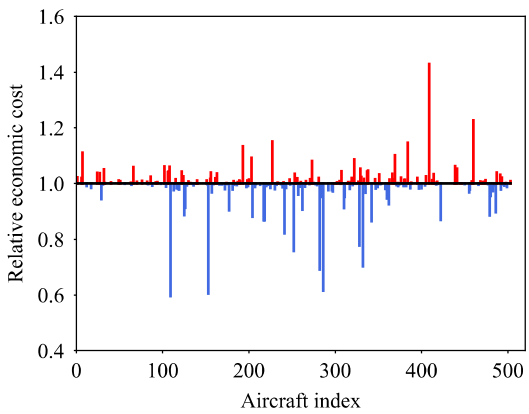


Fig. 4. Comparison of the economic costs for individual aircraft.

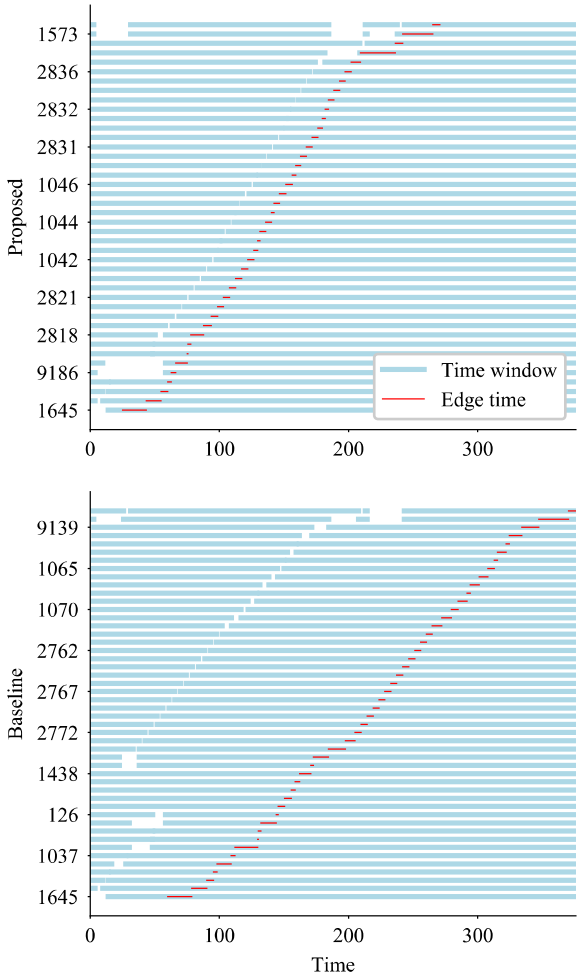


Fig. 5. Illustration of solutions found by different algorithms. The blank regions between time windows are edge times reserved for previously routed aircraft.

TABLE VI. COMPARISON OF THE RESULTS (ECONOMIC COSTS) BEFORE AND AFTER CONVERGENCE.

Aircraft index	Original		After convergence	
	Prop.	Base.	Prop.	Base.
40	214.0	213.2	**213.2**	**213.2**
86	148.1	144.6	**144.6**	**144.6**
94	111.7	111.2	**111.2**	**111.2**
115	331.1	326.1	296.4	**296.3**
184	316.3	315.8	**308.7**	**308.7**
272	298.5	298.2	**295.8**	**295.8**
307	348.1	333.6	**325.3**	328.1
385	441.1	384.8	**384.3**	**384.3**
439	217.6	217.4	**217.4**	**217.4**
488	324.2	310.4	**266.1**	**266.1**
493	168.3	163.1	**163.1**	**163.1**
505	320.6	318.1	**316.4**	**316.4**

the proposed algorithm in the original test. It also presents results for the selected instances with an enlarged time budget to ensure convergence. After convergence, only in rare cases would the proposed algorithm find inferior solutions, which is caused by premature convergence to local optima due to the stochastic nature of the algorithm. This can be alleviated by adopting the multi-start strategy [29]. Another practical way to resolve this issue is to run multiple times of the proposed algorithm and select the best solution. This can be implemented through parallel computing with little extra computational time.

3) Study on disruptive events handling

With the demonstrated capabilites of solving the multiobjective routing and scheduling problem with time window constraints, the proposed algorithm can provide additional solutions to the speed profile database. Given the flexibility of the proposed algorithm, it also provides a means to deal with disruptive events in ground movements. A case study is considered in this section to demonstrate the capability of the proposed algorithm for disruptive events handling. We enforce a blockage of a taxiway on HKG for one hour (13:00-14:00) for taxiway maintenance. The location of the blocked taxiway on the airport layout graph is shown in Fig. 6. Note that this taxiway would be accessed during the blockage time period by more than 9% of the aircraft according to the results in Section IV-B1.

The proposed algorithm can find solutions straightforward in this case by adding time window constraints to the layout graph model. Specifically, the time interval corresponding to the blockage time period is first removed from the time windows of all the affected edges on the taxiway. Then the proposed algorithm can search for feasible solutions meeting the updated time windows. The resulting paths of the proposed algorithm for the aircraft starting taxiing during the blockage time period are shown in Fig. 6. We can observe that all the paths now bypass the blocked taxiway.

V. CONCLUSIONS

This paper presents an extended memetic algorithm for multiobjective routing and scheduling of airport ground movements. It can find conflict-free taxiing trajectories along the taxiway for inbound and outbound aircraft. The proposed algorithm is designed to accommodate intermediate holding operations and flexibly determine the holding time at the origin position of taxiing. These can reduce the holding time at the origin in many cases, which is beneficial for efficient use of scarce airport resources such as runways and gates.

To investigate the performance for real-world problem instances, the proposed algorithm is tested on datasets of three different airports and compared with the baseline approach without intermediate holding. The results demonstrate the capability of the proposed algorithm to find feasible nondominated trajectories within the expected time budget of 10 seconds. It is also capable of finding desirable solutions under different levels of traffic densities. Compared with the baseline, the proposed algorithm reduces the holding time at the taxiing origin position, transferring some of the required

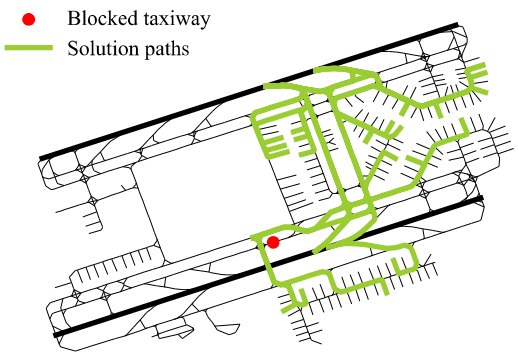


Fig. 6. Illustration of the blocked taxiway and resulting solution paths for the aircraft starting taxiing during the blockage time period.

holding time to intermediate positions on the taxiway, and can therefore release the occupancy of runway exits or gates more quickly for forthcoming aircraft. The comparison with the baseline also demonstrates notable cost savings in different cases. The advantages of the proposed algorithm can be more prominent with more time budgets especially for large airports. This indicates the potential of the proposed algorithm for real-world applications with increased computing power.

Due to the inherent capability of the proposed algorithm in handling time window constraints, it can be used as a complement to the speed profile database which is the cornerstone for real-time routing and scheduling of airport ground movements. In future work, solutions generated by the proposed algorithm will be selected and added to the database for more efficient computation. Moreover, as demonstrated by the case study in this paper, the proposed algorithm can also be directly applied to handle disruptive events in ground movements. It can be immediately adapted to newly added constraints on the taxiway availability and generate feasible solutions accordingly, providing a viable approach for reactive disruption handling on the airport.

For the proposed algorithm, additional improvements can be made to enhance practical performance, including enabling multi-start or parallel computation to promote solution quality. Moreover, holding time variables can be set for the candidates by explicitly searching for holding time intervals that allow the trajectory to satisfy time windows. This can be done through a dedicated local search operator that finds which segments do not comply with time windows and inserts an appropriate amount of holding time before them. This operator will be more costly in terms of computational resources than the holding time perturbation used in the current study, but can provide higher precision.

REFERENCES

[1] G. L. Clare and A. G. Richards, 'Optimization of taxiway routing and runway scheduling', *IEEE Trans Intell Transp Syst*, vol. 12, no. 4, pp. 1000–1013, Dec. 2011, doi: 10.1109/tits.2011.2131650.

[2] U. Neuman and J. D. Atkin, 'Airport gate assignment considering ground movement', in *Computational Logistics*, vol. 8197, D. Pacino, S. Voß, and R. Jensen, Eds. Springer Berlin Heidelberg, 2013, pp. 184–198.

[3] C. Yu, D. Zhang, and H. H. Lau, 'A heuristic approach for solving an integrated gate reassignment and taxi scheduling problem', *Journal of Air Transport Management*, vol. 62, pp. 189–196, 2017.

[4] J. Chen, M. Weiszer, P. Stewart, and M. Shabani, 'Toward a more realistic, cost effective and greener ground movement through active routing: part 1-optimal speed profile generation', *IEEE Trans Intell Transp Syst*, vol. 17, no. 5, pp. 1196–1209, 2016.

[5] J. Chen *et al.*, 'Toward a more realistic, cost effective and greener ground movement through active routing: a multi-objective shortest path approach', *IEEE Trans Intell Transp Syst*, vol. 17, no. 12, pp. 3524–3540, 2016.

[6] S. Ravizza, J. Chen, J. D. Atkin, E. Burke, and P. Stewart, 'The trade-off between taxi time and fuel consumption in airport ground movement', *Public Transp*, vol. 5, no. 1–2, pp. 25–40, Sep. 2013, doi: 10.1007/s12469-013-0060-1.

[7] J. Chen, M. Weiszer, and P. Stewart, 'Optimal speed profile generation for airport ground movement with consideration of emissions', presented at the IEEE 18th International Conference on Intelligent Transportation Systems (ITSC), 2015, pp. 1797–1802.

[8] C. Evertse and H. G. Visser, 'Real-time airport surface movement planning: minimizing aircraft emissions', *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 224–241, 2017, doi: http://dx.doi.org/10.1016/j.trc.2017.03.018.

[9] T. Zhang *et al.*, 'An online speed profile generation approach for efficient airport ground movement', *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 256–272, 2018, doi: https://doi.org/10.1016/j.trc.2018.05.030.

[10] S. Ravizza, J. A. Atkin, and E. K. Burke, 'A more realistic approach for airport ground movement optimisation with stand holding', *Journal of Scheduling*, vol. 17, no. 5, pp. 507–520, Apr. 2014.

[11] M. Weiszer, E. K. Burke, and J. Chen, 'Multi-objective routing and scheduling for airport ground movement', *Transportation Research Part C: Emerging Technologies*, vol. 119, p. 102734, Oct. 2020, doi: 10.1016/j.trc.2020.102734.

[12] M. Weiszer, J. Chen, and P. Stewart, 'A real-time active routing approach via a database for airport surface movement', *Transportation Research Part C: Emerging Technologies*, vol. 58, pp. 127–145, 2015.

[13] E. Q. V. Martins, 'On a multicriteria shortest path problem', *European Journal of Operational Research*, vol. 16, no. 2, pp. 236–245, 1984, doi: 10.1016/0377-2217(84)90077-8.

[14] A. J. V. Skriver and K. A. Andersen, 'A label correcting approach for solving bicriterion shortest-path problems', *Computers & Operations Research*, vol. 27, no. 6, pp. 507–524, 2000, doi: 10.1016/S0305-0548(99)00037-4.

[15] E. W. Dijkstra, 'A note on two problems in connexion with graphs', *Numer Math*, vol. 1, no. 1, pp. 269–271, Dec. 1959, doi: 10.1007/bf01386390.

[16] P. E. Hart, N. J. Nilsson, and B. Raphael, 'A formal basis for the heuristic determination of minimum cost paths', *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, Jul. 1968, doi: 10.1109/tssc.1968.300136.

[17] C. Chitra and P. Subbaraj, 'A nondominated sorting genetic algorithm solution for shortest path routing problem in computer networks', *Expert Systems with Applications*, vol. 39, no. 1, pp. 1518–1525, 2012, doi: 10.1016/j.eswa.2011.08.044.

[18] C. A. Coello Coello, 'Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art', *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 11, pp. 1245–1287, 2002, doi: 10.1016/S0045-7825(01)00323-1.

[19] F. Neri and C. Cotta, 'Memetic algorithms and memetic computing optimization: A literature review', *Swarm and Evolutionary Computation*, vol. 2, pp. 1–14, Feb. 2012, doi: 10.1016/j.swevo.2011.11.003.

[20] L. Beke, M. Weiszer, and J. Chen, 'A comparison of genetic representations and initialisation methods for the multi-objective shortest path problem on multigraphs', *SN COMPUT. SCI.*, vol. 2, no. 3, p. 176, May 2021, doi: 10.1007/s42979-021-00512-z.

[21] L. Beke *et al.*, 'Memetic algorithms for multiobjective routing and scheduling of airport ground movement', unpublished.

[22] 'European ATM Master Plan'. Dec. 17, 2019 [Online]. Available: https://www.sesarju.eu/masterplan2020

[23] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, 'A fast and elitist multiobjective genetic algorithm: NSGA-II', *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi: 10.1109/4235.996017.

[24] C. W. Ahn and R. S. Ramakrishna, 'A genetic algorithm for shortest path routing problem and the sizing of populations', *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 566–579, Dec. 2002, doi: 10.1109/TEVC.2002.804323.

[25] A. E. I. Brownlee, M. Weiszer, J. Chen, S. Ravizza, J. R. Woodward, and E. K. Burke, 'A fuzzy approach to addressing uncertainty in Airport Ground Movement optimisation', *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 150–175, Jul. 2018, doi: https://doi.org/10.1016/j.trc.2018.04.020.

[26] M. Weiszer, J. Chen, P. Stewart, and X. Zhang, 'Preference-based evolutionary algorithm for airport surface operations', *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 296–316, Jun. 2018, doi: 10.1016/j.trc.2018.04.008.

[27] A. J. Nebro, M. López-Ibáñez, C. Barba-González, and J. García-Nieto, 'Automatic configuration of NSGA-II with jMetal and irace', presented at the Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2019, pp. 1374–1381.

[28] ICAO, 'Advanced surface movement guidance and control systems (A-SMGCS) manual'. International Civil Aviation Organization, 2004 [Online]. Available: https://www.icao.int/Meetings/anconf12/Document%20Archive/9830_cons_en[1].pdf

[29] R. Martí, M. G. C. Resende, and C. C. Ribeiro, 'Multi-start methods for combinatorial optimization', *European Journal of Operational Research*, vol. 226, no. 1, pp. 1–8, Apr. 2013, doi: 10.1016/j.ejor.2012.10.012.