Preference-based evolutionary algorithm for airport surface operations

Michal Weiszer,[*] Jun Chen,[†] Paul Stewart[‡]
and Xuejun Zhang.[§]

April 10, 2018

Abstract

In addition to time efficiency, minimisation of fuel consumption and related emissions has started to be considered by research on optimisation of airport surface operations as more airports face severe congestion and tightening environmental regulations. Objectives are related to economic cost which can be used as preferences to search for a region of cost efficient and Pareto optimal solutions. A multi-objective evolutionary optimisation framework with preferences is proposed in this paper to solve a complex optimisation problem integrating runway scheduling and airport ground movement problem. The evolutionary search algorithm uses modified crowding distance in the replacement procedure to take into account cost of delay and fuel price. Furthermore, uncertainty inherent in prices is reflected by expressing preferences as an interval. Preference information is used to control the extent of region of interest, which has a beneficial effect on algorithm performance. As a result, the search algorithm can achieve faster convergence and potentially better solutions. A filtering procedure is further proposed to select an evenly distributed subset of Pareto optimal solutions in order to reduce its size and help the decision maker. The computational results with data from major international hub airports show the efficiency of the proposed approach.

Keywords— airport ground operations; runway scheduling; multiobjective optimisation; preference search

1 Introduction

Twice as many passengers are predicted to be carried by air traffic in 2030 compared to 2013 [1]. With this continuous growth and no actions taken, congestion

[*] M. Weiszer is with the School of Engineering and Materials Science, Queen Mary University of London, UK. e-mail: m.weiszer@qmul.ac.uk

[†] J. Chen is with the School of Engineering and Materials Science, Queen Mary University of London, UK. e-mail: jun.chen@qmul.ac.uk

[‡] P. Stewart is with the Institute for Innovation in Sustainable Engineering, Derby, UK. e-mail: p.stewart1@derby.ac.uk

[§] X. Zhang is with the School of Electronic and Information Engineering, Beihang University and National Key Laboratory of CNS/ATM, Beijing, PR China. e-mail: zhxj@buaa.edu.cn

will become a serious problem for many airports together with a significant environmental impact. As a result, a lot of attention has been attracted towards research on airport operations on the surface [2–10] and near airspace [11, 12].

Recently, the Active Routing (AR) approach for airport ground movement has been introduced [3,4,13,14] with the aim of providing near-optimal nondominated speed profiles and routes for taxiing aircraft. AR enables the routing and scheduling of taxiing aircraft, which was previously based on distance, emphasising time efficiency, to be optimised with regard to richer information embedded within speed profiles. These include the taxiing times, the corresponding fuel consumption, and the associated economic implications, i.e. cost of taxi time and fuel [4]. Results in [4, 9] demonstrated a significant trade-off between taxi time and fuel consumption using different speed profiles and routes, which facilitates multi-objective decision making (e.g. selecting the taxi time efficient solutions in the peak period and fuel efficient ones in the off-peak period as in [4]). The real-time application of the AR framework can be achieved using a pre-computed database of nondominated speed profiles for key building blocks (i.e. straight taxiway segments) of the airport layout [13]. The database acts as a middleware to effectively separate the speed profile generation module from the routing and scheduling module. Furthermore, the airport ground movement problem has been shown in [14] to have an impact on another critical surface operation, runway scheduling.

Due to the multi-objective nature of speed profile generation, routing and scheduling in the AR framework implies an existence of Pareto optimal solutions for different conflicting objectives. This gives rise to the following issues: 1) which Pareto optimal solutions should be selected and stored in the database when the size of the Pareto optimal set is extensively large; 2) which routing and scheduling solution should be selected and implemented for the airport ground movement and the runway scheduling problem.

State-of-the-art approaches for the multi-objective optimisation problem considered in this study, e.g. those in [7, 15], are proved to be computationally demanding for larger and complex instances. Therefore, a legitimate approach in this case is multi-objective evolutionary optimisation algorithms, as they are suitable for complex optimisation problems and have the ability to find multiple near-Pareto optimal solutions in a single run compared to the classical optimisation methods [16]. Traditionally, multi-objective evolutionary optimisation algorithms have emphasised on the search for a complete Pareto optimal set. It is often the case that a decision maker (DM) is expected to select a preferred solution from the obtained Pareto optimal set, i.e. a posteriori, according to his/her preferences. However, the complete Pareto optimal set may be difficult to approximate and an unconverged Pareto front does not allow the DM to find an ideal solution to his/her preferences [17]. As a consequence, a solution with higher taxi time and fuel consumption may be chosen by the DM. On the contrary, if DM's preferences are considered before the search, i.e. a priori, the optimisation algorithm can concentrate on guiding the search to a preferred region of interest (RoI), making the search more computationally efficient with faster convergence [18, 19]. The search by the optimisation algorithm can be also steered in an interactive manner, in which the DM progressively articulates the preferences during the search.

Scalarizing functions [20] involving some additional parameters corresponding to DM's preference are often used to transform a multi-objective problem into a single-objective one. However, this approach may be counterproductive [21]. The DM cannot investigate other optimal or near-optimal solutions and their properties corresponding to the preference information if only a single solution is found during the search. Moreover, in practice, the preferences are often only vague as relative weighing of the priorities is usually approximate. Therefore, the preferences are better to be utilised to search for a RoI rather than a single solution in order to take into account such uncertainty.

Research on incorporating preferences into evolutionary algorithms has been active in the last two decades. For a recent review, see [22]. There are several ways of expressing preferences [23]. In addition to use weights, reference points [21, 24], aspiration levels or goal vectors to represent the desired values of objectives, the DM can also specify a utility function, preference [25] or outranking relation [26]. Weights or trade-off information (i.e. how many units in one objective is at most worth a unit improvement in another objective) are often used to express preferences. Examples include an evolutionary algorithm in [27] with maximally acceptable trade-off rate between objectives, a weight distribution function in [28] and a modified Nondominated Sorting Genetic Algorithm-II (NSGA-II) [16] with reference direction (weights) [17, 18]. The dominance relation is modified according to the distance to the reference point in [29] or aspiration level satisfaction in [30]. An achievement scalarizing function taking into account reference point is used to prefer some solutions closer to the RoI in [17, 25, 31].

Although the abovementioned approaches assume inherently approximate preferences and search for the RoI instead of a single solution, sometimes more information about the preference is available. For example, an interval provides more information (upper and lower bound) about the underlying uncertainty in the preference compared to the single value of the reference point, weight, etc. The uncertainty in preferences should be linked to the size of the RoI, with the RoI adjusted accordingly. Usually, a user defined parameter is introduced to control the extent of the RoI [18, 21, 24]. From a practical point of view, setting up this parameter is not intuitive and the DM can control the extent of the RoI only approximately. As a result, improper parameter setting will lead to either a too wide RoI, wasting computational resources, or a too narrow RoI, not including preferred solutions. In more recent development, Tchebycheff weights that minimise the weighted Tchebycheff distance from the ideal point in [19] and objective function values in a co-evolutionary algorithm [32] can be expressed as an interval. Also, a brushing technique [32] enables the DM to conveniently specify a range of preferred objective function values by drawing in the objective space. However, the specification of these ranges in [19, 32] is left completely to the DM.

In addition, a scalarizing function (i.e. weighted aggregation) and its corresponding parameters (i.e. weights) can express approximate preference. Scalarizing functions have been used in decomposition evolutionary algorithms [33] to convert a multi-objective problem into a set of single-objective subproblems. By varying the parameters of the scalarizing function, different solutions are obtained during the search and combined to provide a Pareto set. The DM can select multiple parameters according to his/her approximate preference, or the parameters can be set by the algorithm as in [34] to find solutions close to a reference point. However, the decomposition based approach has the following disadvantages: 1) if a weighted sum is used as a scalarizing function, the decomposition based algorithm cannot reach solutions on a non-convex Pareto front; 2) if a reference point is used, then it leads to a problem of controlling the extent of the RoI as described earlier; and 3) selecting evenly distributed parameters does not always result in even distribution of solutions. Therefore, in this paper, the scalarizing function is used only for controlling the extent of the RoI, rather than for decomposition. To the best of our knowledge, this is the first application of the scalarizing function for this purpose, especially in the context of airport surface operations.

A scalarizing function can be a cost function, i.e. a unit of each objective has an economic value: the price of fuel or cost corresponding to a minute of delay spent at the airport surface or emissions produced. Economic costs are commonly used in airport sector as incentives [4] when several stakeholders (e.g. airlines, airport) are involved in airport surface operations. However, unit cost are often available as approximate values, reflecting the range of inputs used for their calculation, e.g. different costs for different airlines. The assumption made in this paper is that the uncertainty in the unit cost can be modelled as the RoI. The linear cost function of the unit cost and the total objective function values can evaluate each solution in terms of the total costs, determining the solution with minimal cost from the Pareto set. By varying the unit costs within an interval, reflecting the approximate preference, different solutions will have the minimum cost. A set of these solutions will then form the RoI.

Once the RoI is found, the DM can select a single preferred routing and scheduling solution to be implemented. However, due to the large number of optimal solutions in the RoI, especially in the presence of many objectives, such a decision making process can pose a significant cognitive load on the DM. Therefore, a representative subset of solutions in the RoI is often favoured by the DM. This subset should be uniformly-distributed, i.e. with uniform distances between the solutions in the objective space. Too large distance can result in too big difference between two alternative solutions. The same requirement applies to the subset of speed profiles which are stored in the database with a finite size. In this case, reducing the number while retaining an uniform distribution of available speed profiles is essential in reducing the search space of the airport ground movement and runway scheduling problem.

The issue of finding a set of well-distributed optimal solutions in the objective space has been addressed by preferring less crowded regions such as in NSGA-II [16] or by a controllable distance between the solutions during the search (a priori) [19, 24, 35, 36], after (a posteriori), e.g. by filtering methods in [37]. The uniform spread of solutions in NSGA-II [16] is achieved by a crowding distance favoring solutions further apart from each other, however without any control by the user over the desired distance. In [35], ϵ-dominance divides the objective space into hyper-boxes with equal size of ϵ and only one solution is retained within each box. However, the location of the solution within the box is not considered, which may result in an uneven distribution. R-NSGA-II [21] prefers only one solution within the ϵ-neighbourhood in a similar way to ϵ-dominance . The territory concept presented in [19] prevents solutions to have smaller than the pre-defined distance. Although the territory concept is similar to ϵ-dominance, it prevents losing solutions towards the extremes of the Pareto front [19]. Similarly, network suppression threshold in [36] controls the minimum euclidean distance of solutions. However, the solutions can still be unevenly dispersed as 1) the maximum distance between the solutions is not restricted in

[19,36], and 2) only maximum distance is considered in [19]. The reference-point-based many-objective evolutionary algorithm proposed in [24] evenly distributes reference points so their projections on the Pareto front result in well-distributed solutions. However as noted in [38], in practical problems with constraints or discontinuities in the Pareto front, even though reference points are evenly selected, the algorithm may not end up distributing all solutions uniformly on the Pareto front. In contrast to a priori methods, a posteriori filtering methods select a subset of solutions after the search has finished. This is advantageous as the filtering method can consider all solutions found after the search, not only a few constantly changing solutions generated during each iteration of the search algorithm. A review of a posteriori methods is given in [37].

In the light of the discussion above, a gap in the existing research is identified: ability to handle approximate or uncertain preferences while finding uniform distribution of solutions on the Pareto front. Therefore, in this paper, we introduce a multi-objective evolutionary optimisation (EMO) framework that addresses the above issues. In particular, the novelty of this study can be summarised as follows:

1. An EMO framework is introduced which can handle approximate preferences. The scalarizing function and its corresponding parameters (i.e. an interval of economic value for a unit of each objective) specify the RoI. Specially designed new crowding distance for the replacement procedure of the EMO framework controls the extent of RoI efficiently, which has a direct impact on the algorithm's performance. The main idea is that using interval preference information to define the extent of the RoI results in better solutions found by the EMO algorithm than the algorithms which use user defined parameters for this purpose.

2. A new filtering procedure is proposed to find a representative uniformly-distributed subset of solutions, which further improves the ability of the DM to select a cost efficient solution.

3. The proposed EMO algorithm and filtering procedure are applied to an integrated ground movement and runway scheduling problem formulated in our previous work [14] with an additional objective corresponding to emissions.

The algorithm design choices, i.e. crowding distance and filtering procedure have a practical relevance in terms of saved total time, fuel, emissions and economic costs.

The rest of the paper is organised as follows. Details and related work about the integrated optimisation problem, consisting of the ground movement problem and runway scheduling, are provided in Section 2. Section 3 describes the EMO framework incorporating the preferences and uniform distribution of solutions. Section 4 presents experiments with the proposed algorithm on data instances from Manchester, Beijing Capital International and Doha International Airports. Lastly, Section 5 draws conclusions and future work.

2 Airport ground movement and runway scheduling problem

This section provides a description of the models for the integrated airport ground movement and runway scheduling problem. A table with complete notation is in Appendix.

2.1 Related work

The ground movement problem has been mostly investigated with the aim of minimisation of the total taxi time or time associated objective [2]. Integer programming [5, 10] or graph-based approaches such as [6, 39] have been employed to tackle this problem. Apart from the taxi time, a limited number of studies considered fuel burn as an objective. Research on the stand holding problem [40–42] minimises the fuel burn by holding the aircraft at the stand with inactive engines for as long as possible. However, the stand holding problem does not consider fuel variations during taxiing due to different acceleration. Results in [9, 43] demonstrated that the minimum taxi time results in a higher fuel burn caused by heavy and multiple accelerations/decelerations required to achieve fast movement. Also, as shown in [44], emissions are a conflicting objective with taxi time and fuel consumption. In light of this, the recently proposed AR framework [3, 4] provided a holistic decision making framework for multi-objective routing and scheduling of taxiing aircraft. This renders the AR approach the ability to search for efficient taxiing in terms of not only time, but also fuel consumption and emissions. A similar approach is introduced in [45], considering taxi time, deviation from departure slots and emissions for ground movement. Unlike the multi-objective AR framework, [45] aggregates all objectives together in an scalarized objective function using weights. Therefore, only one solution is obtained after a single run of the algorithm.

The runway scheduling problem has often an objective related to delay, makespan of the schedule, the number of changes in comparison with the First-come-first-served (FCFS) sequence, or various combinations. Approaches employed to solve this problem include hybrid tabu search [46], dynamic programming [47], branch and bound [48] and genetic algorithms [49]. For a detailed review on this topic, see [50]. Routing, sequencing and scheduling aircraft in terminal area around the runway has also attracted attention of researchers [11, 12, 51–55]. Similarly to the runway scheduling problem, approaches for the terminal area often have a time related objective, such as minimisation of delay from scheduled landing/take off times, or minimum distance in case of routing [56].

As pointed out in [2], the ground movement problem is interconnected with runway scheduling. The ground movement needs to make sure that the departing aircraft can reach the runway on time. Similarly, the runway schedule determines the time and sequence for the arriving aircraft to taxi. Researchers have started to consider the ground movement and runway scheduling in an integrated manner. A two-stage model was adopted in [8] where a runway sequence is fixed first by a branch and bound algorithm, and then the ground movement problem is solved by a genetic algorithm considering a delay related objective. An integer programming proposed in [15] decomposes the integrated problem in a similar way. Runway scheduling constraints are imposed on ground movement in the mixed integer linear programming model in [7]. These approaches can result in a suboptimal solution as the problems are treated in separate stages. More recently, a heuristic for the integrated problem has been introduced in [57] and [58] considering only the time related objective. In addition to ground movements and runway scheduling, the model in [59] integrates also scheduling aircraft in the terminal area. In contrast to the previous single-objective approaches, a multi-objective genetic algorithm proposed in [14] for the integrated ground movement and runway scheduling problem takes into account aircraft speed profiles, improving both time and fuel efficiency. In light of this, in this study, a multi-objective and integrated modelling approach similar to [14], but additionally incorporating emissions as another objective and preferences, is adopted. Minimising time and fuel consumption due to surface operations bears significant economic implications to airlines and airports [4]. Also, minimisation of emissions is often considered by airports as shown in [60] in order to mitigate their environmental impact.

2.2 Ground movement problem

The ground movement problem aims to obtain conflict-free routes and schedules with minimum taxi time, fuel and emissions for all aircraft taxiing between gates/stands and runway or in an opposite direction. The total taxi time t^{taxi}, total fuel consumption f^{taxi} and total emissions ε^{taxi} for all aircraft $i = 1, \dots, h$ are calculated as outlined in Algorithm 1. As described in our previous works [13, 14], the ground movement problem is divided into two parts:

1. Preprocessing: the nondominated speed profiles for key building blocks of the airport layout are found and stored in a database in Lines 1–4.
2. Routing and scheduling: the routes and schedules for all taxiing aircraft are found in Lines 5–17, using the speed profiles for building blocks retrieved from the database.

In Line 1, building blocks are identified from a graph representation of airport taxiway layout [13]. The building blocks include all straight segments of taxiways, separated by turning segments. Using building blocks and turning segments, any route between gates/stands and runway (or vice versa) can be recreated. Then, nondominated speed profiles for each building block are found in Line 2. Taxiing on each building block is divided into four phases: acceleration, constant speed, deceleration and rapid deceleration, representing a simplified typical taxiing behaviour as shown in Fig. 1. As described in [3], by varying acceleration rate and the length of each phase, different speed profiles can be explored. In order to have an unrestricted search space for the integrated optimisation problem, all speed profiles, without considering preferences, are explored in Line 2. Without loss of generality, a recently proposed Population Adaptive Based Immune Algorithm (PAIA) [36] is adopted. PAIA achieved good results in terms of performance indicators for the speed profile optimisation problem compared to other algorithms [3]. However, it should be noted that any search algorithm can be employed for this task. The set of nondominated speed profiles is filtered through the filtering procedure described in the next section for uniformly-distributed nondominated speed profiles which

Algorithm 1: Outline of the ground movement problem.

```
   /* Preprocessing                                              */
1  Identify building blocks of airport layout;
2  Find speed profiles for building blocks by PAIA;
3  Filter speed profiles;
4  Store speed profiles in database;
   /* Routing and scheduling                                     */
5  for $i = 1, \ldots, h$ do
6      if $i \in D$ then
7          Find route $q_i$ starting at $t_i^{base} + x_i$;
8      else
9          Find route $q_i$ starting at $t_i^r$;
10     end
11     Given $q_i$, determine $S_i$;
12     forall $s \in S_i$ do
13         Retrieve speed profile $y_i$ from database;
14         Calculate $t_{i,s}, f_{i,s}, \varepsilon_{i,s}^{pp}$;
15     end
16     Reserve route $q_i$ with taxi time $\sum_{s=1}^{|S_i|} t_{i,s}$
17 end
18 Calculate $t^{taxi}, f^{taxi}, \varepsilon^{taxi}$;
```

are then stored in the database in Line 4. Note, that Lines 1–4 are run before the preference-based EMO framework described later in this paper.

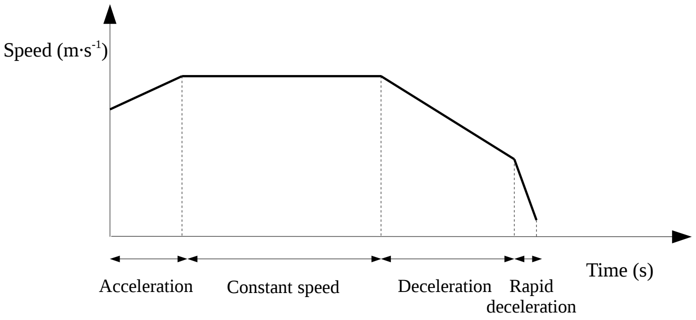


Figure 1: An example of a speed profile with four phases.

Lines 5–18 detail the routing and scheduling part for each aircraft $i = 1, \ldots, h$. The aircraft are ordered with respect to their pushback/landing times. For departures (Line 7), aircraft start taxiing at time $t_i^{base} + x_i$, where x_i is a decision variable and corresponds to time in seconds for which the departing aircraft is held at the gate after the baseline departure time t_i^{base}. Arriving aircraft (Line 9), start taxiing at landing time t_i^r. For each departing/arriving aircraft i, route q_i is found by the heuristic k-QPPTW algorithm described in [9]. In order to keep computational times reasonable, in this paper, for each arriving aircraft i, only the fastest route q_i in terms of taxi time is generated, based on assumed constant speed 15.43 m/s (30 kn) for straight segments and

5.14 m/s (10 kn) for turns. Due to the assumption of generating only the fastest route based on the constant speed, this route is slightly worse than the optimal one. By generating additional routes up to 10 (the 2nd fastest route, 3rd fastest route, ...), an improvement of 2–5% and 2–7% in taxi time and fuel consumption, respectively, was observed in [4] compared to the fastest route. In this paper, the generated routes take into account previously routed aircraft and do not change with subsequently processed aircraft. Based on route q_i, a set of building blocks S_i, corresponding to q_i is determined in Line 11. For all $s \in S_i$, speed profile y_i is then retrieved from the database. y_i is a decision variable that determines which speed profile is retrieved from N^{sp} speed profiles. Then, taxi time $t_{i,s}$, fuel consumption $f_{i,s}$ and emissions $\varepsilon_{i,s}^{pp}$ of pollutant pp for building block s are determined as follows. The taxi time $t_{i,s}$ of aircraft i required to travel through building block s following speed profile y_i consists of partial times $t_p^{phase}(y_i)$ spent in each taxiing phase p:

$$t_{i,s} = \sum_{p=1}^{4} t_p^{phase}(y_i). \tag{1}$$

As described in [3], fuel consumption needed to follow a speed profile depends on thrust levels η which are determined for each taxiing phase p. During deceleration and rapid deceleration, $\eta = 5\%$ of full rated power, and during turning $\eta = 7\%$ [61]. For acceleration and constant speed phase, η is calculated in (2) where *weight* is the weight of the aircraft, *acc* is the acceleration rate, $\mu \cdot weight \cdot g^{acc}$ is the rolling resistance force and F_{oo} is the maximum power output of the jet engine. μ is rolling resistance coefficient, set to 0.015 [3] in this paper. $g^{acc} = 9.81 \text{ m} \cdot \text{s}^{-2}$ is the gravitational acceleration. Note, that the air resistance is not assumed here due to low speeds involved.

$$\eta = \frac{weight \cdot acc + \mu \cdot weight \cdot g^{acc}}{F_{oo}} \tag{2}$$

The thrust level η corresponds to a fuel flow $\phi_p(y_i)$ which is calculated by linearly interpolating or extrapolating fuel flow values for $\eta = 7\%$ and $\eta = 30\%$ reported in ICAO database, following the approach in [61]. The fuel consumption $f_{i,s}$ of aircraft i for building block s following speed profile y_i is defined in (3).

$$f_{i,s} = \sum_{p=1}^{4} \phi_p(y_i) \cdot t_p^{phase}(y_i) \tag{3}$$

Given a calculated fuel flow $\phi_p(y_i)$, the corresponding emission indices $EI^{pp}(\phi_p(y_i))$ (g of pollutant for each kg of burned fuel) for pollutant pp is a function of $\phi_p(y_i)$. $EI^{pp}(\phi_p(y_i))$ can be obtained using curves fitted to the values reported in ICAO Emissions Databank, similarly as in [44]. Finally, the total emission $\varepsilon_{i,s}^{pp}$ of pp of aircraft i during taxiing on s is calculated in (4). It should be noted that the relationship between fuel flow $\phi_p(y_i)$ and emission indices $EI^{pp}(\phi_p(y_i))$ is non-linear and inversely proportionate for hydrocarbons (HC) and carbon monoxide (CO) pollutants, i.e. a higher fuel flow yields less pollutants per kg of fuel than the lower fuel flow does.

$$\varepsilon_{i,s}^{pp} = \sum_{p=1}^{4} \phi_p(y_i) \cdot t_p^{phase}(y_i) \cdot EI^{pp}(\phi_p(y_i)) \tag{4}$$

In the previous research [44], it was shown that HC and CO pollutants for light and medium category aircraft are strongly correlated with the taxi time. Therefore, for light and medium category aircraft it is assumed that minimising taxi time will minimise HC and CO emissions at the same time. The nitrous pollutants (NOx) are linearly dependent on fuel flow for light, medium and heavy category aircraft, and for that reason are not considered in the search. For the heavy category aircraft, the HC and CO pollutants are considered as an individual objective. However, due to the strong correlation between them, it is sufficient to include only one of them, e.g. HC, in the optimisation. As a result, only HC is considered as an additional objective for heavy aircraft in this paper. For light and medium category aircraft, only taxi time and fuel consumption is considered during the search. After efficient solutions in terms of these two objectives are found, the value of HC is calculated for aircraft in these categories according to (4).

After $t_{i,s}, f_{i,s}, \varepsilon_{i,s}^{pp}$ are determined, the route q_i with taxi time $\sum_{s=1}^{|S_i|} t_{i,s}$ is reserved for aircraft i. Lines 5–17 are repeated until all aircraft are processed. Finally, the total taxi time t^{taxi}, the total fuel consumption f^{taxi} and emission ε^{taxi} for all aircraft $i = 1, 2, \ldots, h$, are calculated in (5)–(7). Due to reasons mentioned above ε^{taxi} only considers $\varepsilon_{i,s}^{HC}$.

$$t^{taxi} = \sum_{i=1}^{h} \sum_{s=1}^{|S_i|} t_{i,s}, \tag{5}$$

$$f^{taxi} = \sum_{i=1}^{h} \sum_{s=1}^{|S_i|} f_{i,s}, \tag{6}$$

$$\varepsilon^{taxi} = \sum_{i=1}^{h} \sum_{s=1}^{|S_i|} \varepsilon_{i,s}^{HC}. \tag{7}$$

2.3 Runway scheduling problem

The runway scheduling problem aims to find landing and take-off times for all aircraft arriving or departing from a given runway. The objectives of runway scheduling are the minimum delay, fuel consumption due to waiting and associated emissions subject to safe separation between aircraft. In this study, landing times are fixed and only take-off times are subject to search. This is due to the fact that holding arriving aircraft still in air has more impact on the air traffic control system compared to departing aircraft on the airport surface.

The minimum amount of time which must elapse between subsequent aircraft using the runway is due to air turbulence (called wake vortices) generated by departing/landing aircraft and in-flight separation constraints imposed by different speeds of airborne aircraft. In this study, only separation to prevent wake vortices is considered for simplicity.

The set of all h aircraft is denoted as H and consists of A arriving and D departing aircraft. The function $W(w_j, w_i)$ returns the wake vortex separation for a pair of subsequent (leading/trailing) aircraft j, i with weight categories w_j and w_i. The values of required separations between a leading and trailing aircraft departing/arriving on the runway are given in Table 1. As can be seen,

Table 1: Separations in seconds between departures (D) and arrivals (A) for different weight categories w: heavy (Hv), medium (M) and light (L) [15].

		Trailing					
		A-Hv	A-M	A-L	D-Hv	D-M	D-L
Leading	A-Hv	96	157	207	60	60	60
	A-M	60	69	123	60	60	60
	A-L	60	69	82	60	60	60
	D-Hv	60	60	60	96	120	120
	D-M	60	60	60	60	60	60
	D-L	60	60	60	60	60	60

longer separations are generally needed when the heavier aircraft is followed by a lighter one. For aircraft departing in the order of j, i, e, the triangle inequality $W(w_j, w_i) + W(w_i, w_e) \geq W(w_j, w_e)$ holds.

Let t_i^r be the actual landing or take-off time for arriving and departing aircraft i, respectively. t_i^r is given for arriving aircraft $i \in A$. For departures, i.e. $i \in D$, t_i^d is the arrival time at the runway holding point which is calculated as $t_i^d = t_i^{base} + \sum_{s=1}^{|S_i|} t_{i,s}$. If the difference between t_i^d and t_j^r of previous aircraft j complies with the minimum separation $W(w_i, w_j)$, aircraft i can take-off without delay, i.e. $t_i^d = t_i^r$. Otherwise, departing aircraft i at the runway holding point needs to postpone its take-off as defined in (8), subject to $t_i^r \geq t_i^d$ and $t_i^r \geq t_j^r + W(w_i, w_j)$.

$$t_i^r = \begin{cases} t_i^d & \text{if } t_i^d - t_j^r \\ & \geq W(w_i, w_j), \\ t_i^d + W(w_i, w_j) & \\ -(t_i^d - t_j^r) & \text{otherwise.} \end{cases} \tag{8}$$

Therefore, the waiting time t_i^w of the departing aircraft $i \in D$ is equal to $t_i^w = t_i^d - t_i^r$. In this paper, we assume that no departure slots are prescribed, and aircraft can take off immediately as long as it is safe to do so.

Minimisation of the total runway delay t^{rwy}, as defined in (9), is the first objective of the runway scheduling. The second objective to be minimised is the total runway fuel f^{rwy} used during t_i^w by jet engines with idle fuel flow ϕ_{w_i}. The third objective is to minimise emissions ε^{rwy} associated with the fuel f^{rwy}.

$$t^{rwy} = \sum_{i=1}^{|D|} t_i^w, \tag{9}$$

$$f^{rwy} = \sum_{i=1}^{|D|} t_i^w \cdot \phi_{w_i}. \tag{10}$$

$$\varepsilon^{rwy} = \sum_{i=1}^{|D|} t_i^w \cdot \phi_{w_i} \cdot EI^{pp}. \tag{11}$$

The idle fuel flow ϕ_{w_i} and emissions EI^{pp} correspond to fuel flow and emissions, respectively, and are based on the International Civil Aviation Organisation

(ICAO) engine database for $\eta = 5\%$ of the representative aircraft. Similarly, as for the ground movement problem, only HC emissions are considered in (11). For heavy category aircraft, HC emission serves as an individual objective, whereas for light and medium category aircraft emissions are calculated after the search.

2.4 Integrated optimisation problem

Following the modelling approaches in Sections 2.2 and 2.3, in this section, the ground movement and runway scheduling problem are integrated into a multi-objective optimisation problem. The objective functions considered are (12)-(14). g_1 corresponds to the total time, g_2 is the fuel consumption and g_3 is the total emission of HC. Each objective function consists of two components. The first component is for ground movement, and the second one is for the runway scheduling part. Each component is based on equations detailed in Sections 2.2 and 2.3.

$$\min g_1 = t^{taxi} + t^{rwy}, \tag{12}$$

$$\min g_2 = f^{taxi} + f^{rwy}, \tag{13}$$

$$\min g_3 = \varepsilon^{taxi} + \varepsilon^{rwy}. \tag{14}$$

For the integrated optimisation problem, the decision variables are the following: the pushback time x_i for departing aircraft $i \in D$ and the speed profile $y_i \in \{1, \ldots, N^{sp}\}$ for all aircraft $i \in H$. x_i determines when aircraft i starts ground movement and therefore arrival time at the runway holding point. In this paper, x_i is limited to an integer value within $\{0, \ldots, 300\}$. y_i determines the duration of ground movement for arriving/departing aircraft i and arrival time at the runway holding point in case of departures. The complete solution to the integrated optimisation problem is represented as a vector of integer values: $\{ x_1, x_2, \ldots, x_{|D|}, y_1, y_2, \ldots, y_h\}$. Given the decision variables, the objective function values g_1, g_2, g_3 are determined by:

1. t^{taxi}, f^{taxi}, ε^{taxi},
2. t^{rwy}, f^{rwy}, ε^{rwy}.

In order to search for the values of decision variables, the EMO framework is adopted and is described later.

2.5 Economic values of objectives

In practice, each unit of objectives g_1, g_2, g_3 defined in Section 2.4 correspond to an economic value, i.e. a unit cost. Economic costs are recognised in [4] as incentives and an efficient way to consider different stakeholders' interests during airport surface operations: the airlines and the airport. As described in [4], the following costs are related to airport surface operations (this includes taxiing and waiting at the runway):

- time dependent cost (€·s^{-1}) related to g_1 which consists of:
  - aircraft maintenance cost: maintenance is needed at a regular time intervals,

  - aircraft opportunity cost: revenues missed because time during taxiing or waiting at the runway is not used for profitable service,
  - other operational costs related to aircraft (e.g. crew salaries)
  - airport opportunity cost: revenues missed because infrastructure at the airport is not used for profitable service,

- fuel costs (€·kg^{-1}) related to g_2,
- emission costs (€·g^{-1}) related to g_3 used in this paper.

Emission costs are a monetary value for each g of the pollutant emitted. Some airports have already applied an emissions charge scheme [60] due to which airlines need to pay for their emissions. However, the schemes usually lump HC and NOx emissions together and the detailed information of charges per unit for individual pollutants is missing. Therefore, in this study, no monetary value for g_3 is assumed and the decision is left to the DM.

However, some unit costs are often difficult to determine and only approximations are available. For example, a unit cost of g_1 depends on maintenance cost or crew salaries which can be different for different airlines. Therefore, searching for a RoI with solutions within a range of objective function values, corresponding to certain ranges of unit costs for g_1, g_2 bears more pragmatic meaning. The scalarizing function used in this work is defined as a function $C^{total}(\mathbf{z}_k, \mathbf{c})$ which evaluates a solution z_j found during the search in terms of the total monetary cost for nr^{obj} objectives, using a unit cost vector $\mathbf{c} = [c_1, \ldots, c_{nr^{obj}}], c_m \geq 0$ for $m = 1, \ldots, nr^{obj}$ for each objective:

$$C^{total}(\mathbf{z}_k, \mathbf{c}) = \sum_{m=1}^{nr^{obj}} c_m \cdot g_m(\mathbf{z}_k). \tag{15}$$

If no information about the economic value for objective m is available, $c_m = 0$. It should be noted that (15) can be rewritten such that different c_m is used for aircraft in different weight categories and then multiplied by the sum of the objective function values of aircraft belonging to that weight category. For example, different fuel price can be used for heavy category aircraft if more equal consideration of heavy (which consume more fuel) and medium aircraft is desired.

The vector $\bar{\mathbf{c}} = [\bar{c}_1, \ldots, \bar{c}_u]$ defines the most probable unit costs $\bar{c}_m \geq 0$ for $m = 1, \ldots, nr^{obj}$. Furthermore, to include the uncertainty in the unit costs, for each $\bar{c}_m$ an upper bound $\bar{c}_m^{upper}$ and lower bound $\bar{c}_m^{lower}$ is defined such that $\bar{c}_m^{lower} < \bar{c}_m < \bar{c}_m^{upper}$. It should be noted that $\bar{c}_m^{lower}, \bar{c}_m, \bar{c}_m^{upper}$ can be elicited from the DM or from the data held by the airport, e.g. the range of fuel price during a certain period.

In the next section, we show how the preferred region defined by the DM or based on scalarizing function and unit costs of objectives is constructed during the search by the EMO framework, to guide the preference-based EMO algorithm towards the RoI. This is carried out in conjunction with a filtering procedure to obtain uniformly-distributed solutions.

3 EMO framework with preferences

In this section, we describe the EMO framework based on the preferred region and filtering procedure to find solutions in the RoI for the integrated optimisation problem as defined in Section 2.4. The structure of the EMO framework is outlined in Algorithm 2. Lines 1–13 describe the Preference-based EMO algorithm (P-EMOA) and Line 14 refers to the filtering procedure. P-EMOA is a derivative of a generic EMO algorithm, such as NSGA-II [16]. In Line 1, the initial population is filled with solutions with random values of decision variables. Each solution in the population is assigned objective function values g_1, g_2, g_3 in Lines 2–4, as described in Section 2.4. In Line 6, solutions with better objective function values are selected for reproduction. Reproduction is performed by applying a 2-point crossover, with a given probability, to two parent solutions. In Line 9, mutation randomly changes the value of decision variables according to the mutation rate. Then, the solution is assigned the values of g_1, g_2, g_3. Solutions, surviving to the next generation, are selected in Line 12. The replacement procedure in P-EMOA is adapted to incorporate preferences. A generic EMO favours non-dominated solutions which are in less ”crowded“ regions of the whole objective space, such as in the standard NSGA-II. In P-EMOA, the replacement prefers solutions which are in the RoI as well as not crowded within it. For this purpose, the replacement procedure with a modified crowding distance taking into account these two requirements is adopted and will be described in the next section. The loop in Lines 5–13 is repeated until the maximum number of generations is reached. The final population is then filtered by a filtering procedure to obtain a representative subset of solutions.

Algorithm 2: Structure of the EMO framework.

```
   /* P-EMOA                                                    */
1  create initial population;
2  for each solution in population do
3      calculate $g_1, g_2, g_3$;
4  end
5  while the maximum number of generations is not reached do
6      select good solutions for reproduction;
7      apply 2-point crossover;
8      for each solution in population do
9          perform mutation;
10         calculate $g_1, g_2, g_3$;
11     end
12     replace population;
13 end
   /* Filtering procedure                                       */
14 apply filtering procedure to the final population
```

3.1 Replacement procedure for P-EMOA

During each generation, a non-dominated sorting [16] is performed during the replacement to identify the non-dominated fronts. Within each front, surviving solutions are selected based on the preferred region of the Pareto front of solutions in the objective space. The preferred region is defined as follows.

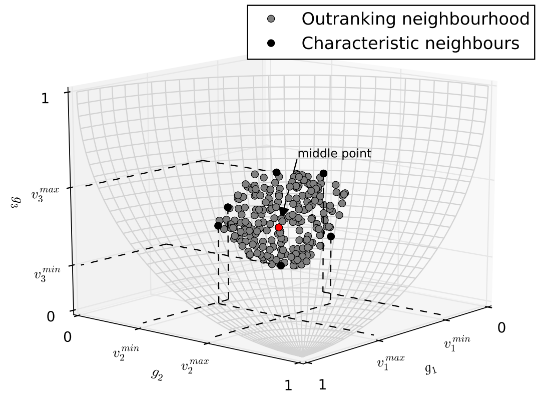


Figure 2: Preferred region on Pareto front of the convex DTLZ2 problem [62], for $\bar{\mathbf{c}} = [1, 1, 1]$, $\bar{c}_1^{lower} = \bar{c}_2^{lower} = \bar{c}_3^{lower} = 0.8$, $\bar{c}_1^{upper} = \bar{c}_2^{upper} = \bar{c}_3^{upper} = 1.2$.

Firstly, the middle point $\mathbf{z}^C$ is identified as the most preferred solution within the front. Next, boundary solutions $\mathbf{z}_a^B$, called the characteristic neighbours are selected such that for each objective g_m an interval $[v_m^{min}, v_m^{max}]$ is defined for $m = 1, \ldots, nr^{obj}$ as shown in Fig. 2. The middle point together with the boundary solutions specify the RoI. It should be noted, that the middle point and the boundary solutions can be selected by the DM interactively via the brushing technique [32]. As described in Section 2.5, a scalarizing function C^{total} can be utilised for finding the middle point and the boundary solutions. In such case, $\bar{c}_m^{lower}, \bar{c}_m, \bar{c}_m^{upper}$ need to be provided by the DM. The middle point is determined as a solution $\mathbf{z}_k$ which has a minimum scalarizing function value with respect to the unit cost vector $\bar{\mathbf{c}}$:

$$\mathbf{z}^C = \arg\min_{\mathbf{z}_k}(C^{total}(\mathbf{z}_k, \bar{\mathbf{c}})). \tag{16}$$

In order to determine boundary solutions, we define a boundary unit cost vector $\bar{\mathbf{c}}^B$. The vector $\bar{\mathbf{c}}^B$ is a vector with boundary unit cost values, i.e. c_m equals to $\bar{c}_m^{lower}$ or $\bar{c}_m^{upper}$. Therefore, for u objectives, this leads to $nr^B = 2^{nr^{obj}}$ boundary unit cost vectors $\bar{\mathbf{c}}^B$, which are the complete combinations of upper/lower bounds, e.g. $\bar{\mathbf{c}}^B = [\bar{c}_1^{lower}, \bar{c}_2^{upper}, \bar{c}_3^{upper}]$. Each a-th $\bar{\mathbf{c}}^B$ corresponds to a characteristic neighbour, which is a solution $\mathbf{z}_k$ with a minimum scalarizing function value using $\bar{\mathbf{c}}_a^B$:

$$\mathbf{z}_a^B = \arg\min_{\mathbf{z}_k}(C^{total}(\mathbf{z}_k, \bar{\mathbf{c}}_a^B)). \tag{17}$$

All $\mathbf{z}^B$ define a RoI, within which solutions with minimum $C^{total}(\mathbf{z}_k, \bar{\mathbf{c}}_{\mathbf{m}})$ for any unit cost vector c_m between $\bar{c}_m^{lower}$ and $\bar{c}_m^{upper}$ are located, as illustrated in Fig. 2. Furthermore, the RoI obtained from the scalarizing function can be further modified by the DM during each generation.

In order to focus the search on the RoI, the crowding distance cd_k is modified as follows. For each $\mathbf{z}_k$ within the same non-dominated front, cd_k is calculated as:

$$cd_k = \begin{cases} \infty & \text{if } \mathbf{z}_k \text{ is the middle point } \mathbf{z}^C, \\ M + d_k^{cd} & \text{else if } \mathbf{z_j} \text{ outranks } \mathbf{z}^C, \\ 1/C^{total}(\mathbf{z}_k, \bar{\mathbf{c}}) & \text{ otherwise.} \end{cases} \tag{18}$$

If the solution $\mathbf{z}_k$ is the middle point, $cd_k = \infty$ and ensures that it is always selected for the next generation. Otherwise, cd_k is determined based on the outranking relation [26] of $\mathbf{z}_k$ to $\mathbf{z}^C$. The middle point $\mathbf{z}^C$ is outranked by $\mathbf{z}_k$ (denoted as $\mathbf{z}_k S \mathbf{z}^C$) if $v_m^{min} \leq g_m(\mathbf{z}_k) \leq v_m^{max}$ for all objectives $m = 1, \ldots, nr^{obj}$. All $\mathbf{z}_k$ outranking $\mathbf{z}^C$ form the outranking neighbourhood. Solution $\mathbf{z}_k S \mathbf{z}^C$ if all g_m for $m = 1, \ldots, nr^{obj}$ are within $[v_m^{min}, v_m^{max}]$. The interval $[v_m^{min}, v_m^{max}]$, called the veto interval, for objective m is determined by nr^B characteristic neighbours:

$$v_m^{min} = min\{g_m(\mathbf{z}_a^B), a = 1, \ldots, nr^B\}, \tag{19}$$

$$v_m^{max} = max\{g_m(\mathbf{z}_a^B), a = 1, \ldots, nr^B\}. \tag{20}$$

Determining whether $\mathbf{z}_k$ belongs to the outranking neighbourhood does not depend on its C^{total} value as defined in (15). This is important, as it is well known (e.g. [63]) that such weighted sum aggregation can discover solutions located only on a convex part of the Pareto front.

Solutions $\mathbf{z}_k$ belonging to the outranking neighbourhood have the crowding distance equal to $M + d_k^{cd}$, where $M = 10^6$ in this paper. d_k^{cd} estimates the distance to the neighbouring solutions in (21), sorted according to [16]:

$$d_k^{cd} = \sum_{m=1}^{u} \frac{g_m(z_{j+1}) - g_m(z_{j-1})}{g_m^{\max} - g_m^{\min}}. \tag{21}$$

Where $g_m^{\min}$,$g_m^{\max}$ correspond to the minimum and maximum value of objective g_m encountered during the search so far. With M set to big positive integer, cd_k ensures that the replacement performs its intended function, i.e. always preferring solutions from the characteristic neighbourhood, whereas the diversity within RoI is maintained by favouring solutions which are more distant from each other by adding d_k^{cd}.

For solutions outside the characteristic neighbourhood, cd_k is inversely related to $C^{total}(\mathbf{z}_k, \bar{\mathbf{c}})$, with larger cd_k assigned to solutions with smaller $C^{total}(\mathbf{z}_k, \bar{\mathbf{c}})$. As a result, solutions $\mathbf{z}_k$ with smaller $C^{total}(\mathbf{z}_k, \bar{\mathbf{c}})$ and therefore smaller distance to the RoI are favoured. If a scalarizing function is not used, euclidean distance from the nearest characteristic neighbourhood solution can be used instead. This is particularly important if there are not enough solutions in the outranking neighbourhood to fill the population such as in early stages of evolution or due to the nature of the optimisation problem.

Finally, solutions are selected according to descending value of cd_k until the population is filled. After the specified number of generations has elapsed, the final population containing solutions from the RoI is archived into set R. However, as mentioned previously, the possibly large number of solutions and their uneven distribution pose a difficulty for the DM to make a decision, particularly in the case of higher number of objectives. Therefore, the same filtering procedure used to fill the database is applied again to obtain a representative subset of uniformly-distributed solutions from R.

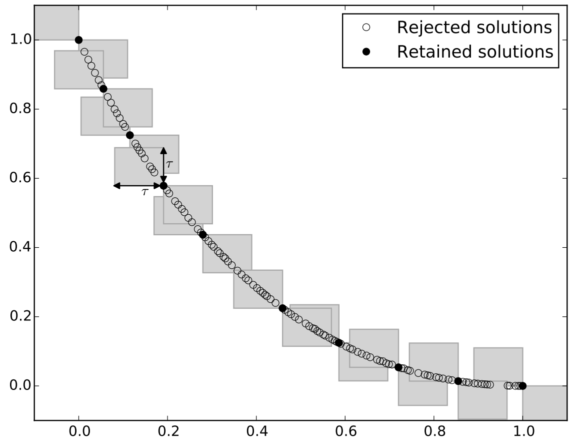


Figure 3: Illustration of the territory concept.

3.2 Filtering procedure of Pareto optimal solutions

As discussed in Section 2.2, a representative subset of solutions helps the DM in the selection process and reduces the number of solutions to be stored in a database. The procedure is performed in two steps:

1. $R \rightarrow R^*$ using territory concept [19],
2. $R^* \rightarrow R^{**}$ using the ξ-heuristic proposed in this paper.

Firstly, R is filtered using the territory concept to obtain an initial set R^*. However, as discussed below, the distance between solutions after filtering in the first step may still be non-uniform. Therefore, in the second step, R^* is further refined into R^{**} using the evenness measure.

In the first step, only solutions $\mathbf{z}_k$ which do not have any other solution $\mathbf{z}_l$ in their territory are kept. The territory of $\mathbf{z}_k$ is defined as the region within a distance τ of $\mathbf{z}_k$ in each objective among the regions that neither dominate nor are dominated by $\mathbf{z}_k$ as shown in Fig. 3.

The first step of the filtering procedure is detailed as follows. For each solution $\mathbf{z}_k$ in R:

1. If R^* is empty, $\mathbf{z}_k$ is accepted into R^*. Otherwise, proceed to the next step.
2. Objective values g_m of $\mathbf{z}_k$ are normalised as defined in (22), where g_m^{max}, g_m^{min} refer to the maximum/minimum objective value g of the m-th objective in R.
$$\overline{g_m} = \frac{g_m - g_m^{min}}{g_m^{max} - g_m^{min}} \tag{22}$$
3. The rectilinear distance d_{kl}^{rect} of $\mathbf{z}_k$ is calculated to each solution $\mathbf{z}_l$ in R^*:
$$d_{kl}^{rect} = \sum_{m=1}^{nr^{obj}} |\overline{g_m}(\mathbf{z}_k) - \overline{g_m}(\mathbf{z}_l)| \tag{23}$$

4. A solution $\mathbf{z}_l^*$ with the smallest rectilinear distance $d_{kl^*}^{rect}$ to $\mathbf{z}_k$ is found as $l^* = \arg\min_l(d_{kl}^{rect})$.

5. The maximum scaled absolute objective difference δ between $\mathbf{z}_k$ and $\mathbf{z}_l^*$ is found according to (24).

$$\delta = \max_{m=1,2,\ldots,nr^{obj}} |\overline{g_m}(\mathbf{z}_k) - \overline{g_m}(\mathbf{z}_l^*)| \tag{24}$$

6. If $\delta \geq \tau$, $\mathbf{z}_k$ is accepted and moved into R^*. Otherwise, $\mathbf{z}_k$ is rejected.

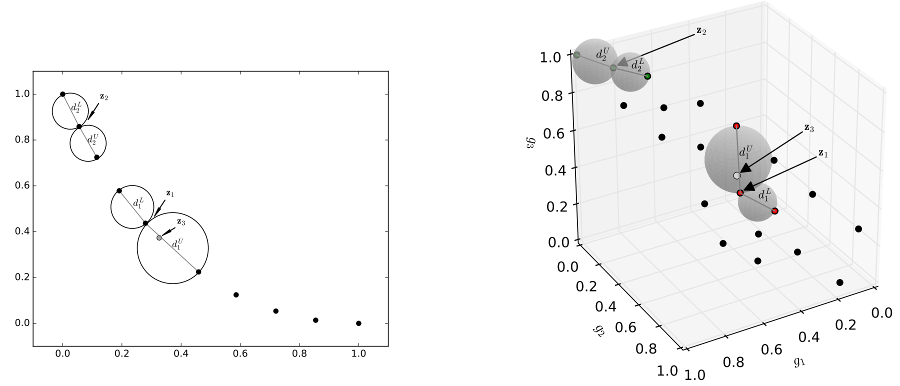
The filtering procedure in the first step depends on the value of τ, effectively controlling the size of the archive population. Two strategies for setting up the value of τ are proposed: 1) If a fixed number of solutions N^R is preferred, the value of τ is set to $\tau = 1/(\sqrt[nr^{obj}-1]{N^R} + o)$, with $o = 0$ initially. If the number of solutions obtained using the initial τ is less than N^R, o is iteratively increased until the number of filtered solutions equals N^R. 2) If a minimum difference inc_m in objective m between any two solutions is required, $\tau = \overline{inc_m}$, where $\overline{inc_m}$ is normalised similarly as in (22). This ensures that δ is at least $\overline{inc_m}$ in (24). Instead of setting up different inc_n for each objective n with a different scale than objective m, an initial inc_m is set up for any m objective. In order to set inc_n for objective $n \neq m$, the objective normalisation for n can be adjusted with adj_n as outlined in (25). adj_n is derived from (26) and adjusts the scale of n such that τ (based on inc_m) corresponds to inc_n. The difference inc_n can be also based on the economic value, in which case inc_n multiplied by c_n corresponds to the desired difference in monetary units.

$$\overline{g_n} = \frac{g_n - g_n^{min}}{g_n^{max} - g_n^{min} + adj_n} \tag{25}$$

$$\tau \cdot (g_n^{max} - g_n^{min} + adj_n) = inc_n \tag{26}$$

The filtering in the first step provides a good initial set R^*. However, solutions in R^* may be closer than τ in some dimensions as only maximum distance is considered in (24). Also, as δ is not required to be strictly equal to τ, and indeed it can be any value larger or equal to τ, a large distance between solutions is possible. Therefore, R^* needs to be refined by a ξ-heuristic based on the evenness measure [64] in the second step. The evenness measure is defined as follows. For each solution $\mathbf{z}_k$ in R^*, two spheres (circles in case of 2 objectives or hyper-spheres for more than 3 objectives) are constructed, as illustrated in Fig. 4. The first sphere is the smallest sphere that can be constructed between $\mathbf{z}_k$ and any other solution in R^*. The diameter of this sphere is denoted as d_k^L. The second sphere is the largest sphere that can be constructed between $\mathbf{z}_k$ and any other solution in R^* such that no other solution in R^* is within the sphere. The diameter of this sphere is denoted as d_k^U. The diameters d_k^L and d_k^U are calculated as euclidean distances between corresponding points. The evenness measure for R^* is then defined in (27), where σ_d and $\hat{d}$ refer to the standard deviation and mean, respectively, of the set of all diameters for all solutions in R^*: $\{d_1^L, d_2^L, \ldots, d_k^L\} \cup \{d_1^U, d_2^U, \ldots, d_k^U\}$.

$$\xi = \frac{\sigma_d}{\hat{d}} \tag{27}$$



(a)

(b)

Figure 4: Illustration of the evenness measure for (a) 2-objective case, (b) 3-objective case.

A set of solutions is exactly evenly distributed when $\xi = 0$, i.e. all diameters are equal. This is illustrated in Fig. 4, where $d_1^U > d_1^L$ for $\mathbf{z_1}$, in contrast to $d_2^U \approx d_2^L$ for $\mathbf{z_2}$.

The set R^* is refined by the ξ-heuristic by replacing solutions $\mathbf{z}_k$ from R^* with better solutions $\mathbf{z}_l$ from R. The ξ-heuristic replaces solutions based on the value of absolute difference $|d_i^L - \hat{d}|$ and $|d_i^U - \hat{d}|$. A small value of $|d_i^L - \hat{d}|$, $|d_i^U - \hat{d}|$, respectively indicates that d_i^L, d_i^U are close to an ideal value, approximated by $\hat{d}$. The ξ-heuristic is performed as follows. For each solution $\mathbf{z}_k$ in R:

1. The nearest solution $\mathbf{z}_l$ from R^* is determined in terms of euclidean distance.

2. Solution $\mathbf{z}_l$ is temporarily removed from R^* and $\mathbf{z}_k$ temporarily inserted to R^*.

3. Then, diameters d_k^L and d_k^U are calculated as euclidean distance between $\mathbf{z}_k$ and the corresponding points on the sphere.

4. If $|d_k^L - \hat{d}| < |d_i^L - \hat{d}| \wedge |d_k^U - \hat{d}| < |d_i^U - \hat{d}|$, solution $\mathbf{z}_k$ is accepted into R^* and $\mathbf{z}_l$ is finally removed. For example, solution $\mathbf{z}_3$ replaces $\mathbf{z}_1$ in Fig. 4. Otherwise, $\mathbf{z}_k$ is rejected and $\mathbf{z}_l$ is returned to R^*.

After all solutions from R are tested, $R^{**} = R^*$.

4 Computational results and discussion

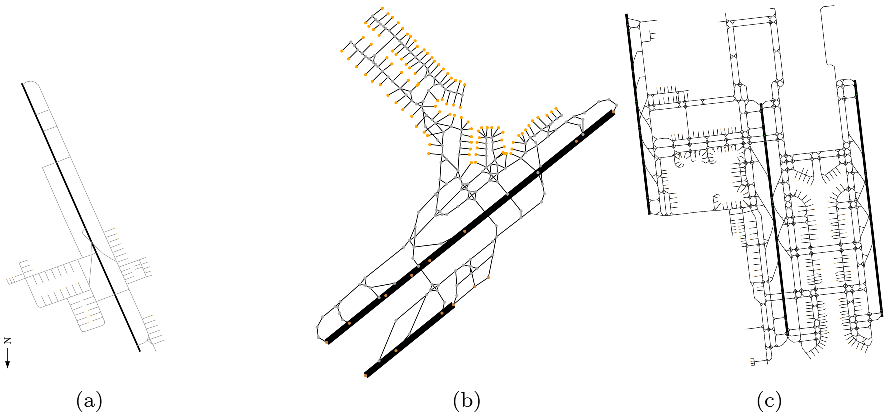
4.1 Experimental setup

The proposed EMO framework was tested on a set of instances of real arrival and departure flights from 3 airports: Manchester (MAN), Beijing Capital International (PEK) and Doha International Airport (DOH). The complexity of the taxiway layout ranges from simple (DOH), medium (MAN) to complex (PEK) as can be seen in Fig. 5. The data provided specified landing/pushback

Table 2: Data instances.

Instance	Aircraft	Arrivals	Departures	Date
*man*1	25	13	12	12:00, 3.9.2011
*man*2	21	12	9	21:00, 3.9.2011
*man*3	24	8	16	18:00, 3.9.2011
*man*4	36	15	21	07:00, 3.9.2011
*man*5	13	4	9	9:00, 3.9.2011
*man*6	21	11	10	6:00, 3.9.2011
*doh*1	21	17	4	19:00, 16.3.2014
*doh*2	21	19	2	21:00, 16.3.2014
*pek*1	17	3	14	13:00, 9.7.2014
*pek*2	20	6	14	13:00, 9.7.2014

times and gates/runway exits for each flight. *man* instances are extracted from Nottingham ASAP instances [1]. Complete data instances are available at http://dx.doi.org/10.5281/zenodo.1197292. The details of instances are given in Table 2. The size of instances ranges from 13 to 36 aircraft with different mix of arrivals and departures, representing a balanced set of different traffic levels.



(a)

(b)

(c)

Figure 5: A directed graph representation of the airport surface for (a) Doha International Airport, (b) Manchester, (c) Beijing Capital International Airport.

As a simplification, all aircraft have been categorised into 3 weight categories (light, medium, heavy) and representative aircraft is designated for each category: Learjet 35A, Airbus A320 and Airbus A333 for light, medium and heavy category, respectively. The specifications of the representative aircraft are used for calculation of wake vortex separations, fuel burned and emissions. For each building block, the number of speed profiles which are saved into the database

[1] http://www.asap.cs.nott.ac.uk/external/atr/benchmarks/data/groundMovement/MAN_OSM_Benchmark_20111029_GM.tx

was set to $N^{sp} = 10$ for light and medium category aircraft. For heavy category aircraft, $N^{sp} = 20$ as 3 objectives are considered during the speed profile generation, in contrast to 2 objectives for light and medium category aircraft, as explained in Section 2.2.

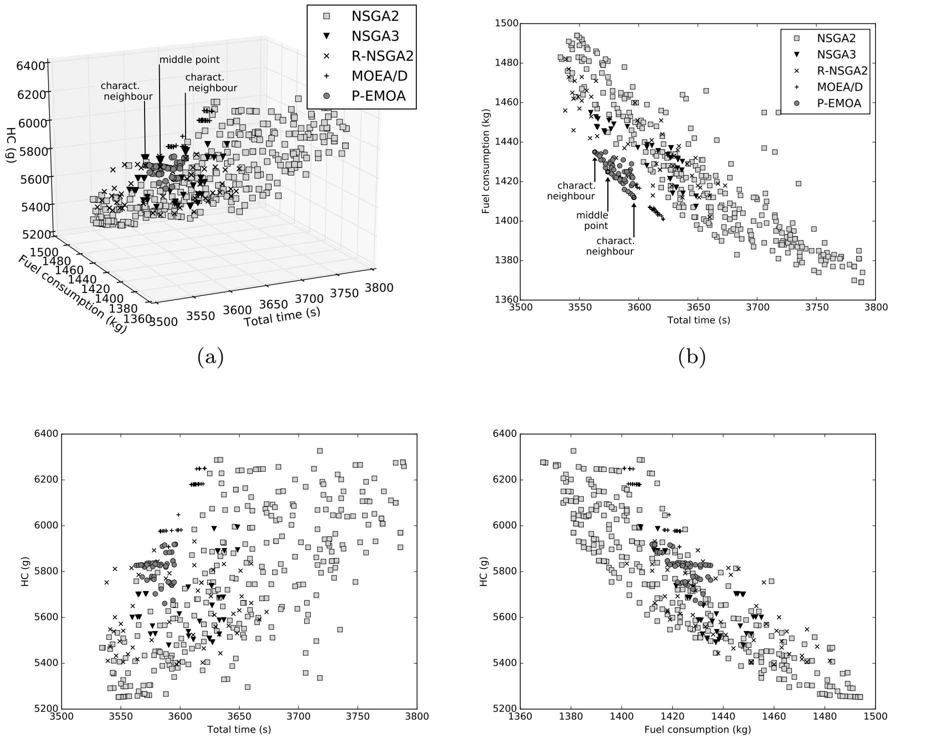
The EMO framework is implemented using the Inspyred package for Python [65]. Based on initial experiments with instance $man1$, the termination criteria for EMO framework was set to 50 generations and the number of individuals in population was 50. The crossover operator was set to a two-point crossover with probability 1, the mutation operator randomly changed a single gene with a probability 0.1. A scalarizing function described in Eq. 15 and a unit cost vector $\bar{\mathbf{c}} = [0.469, 0.71, 0]$ was used for the RoI specification. $\bar{\mathbf{c}}$ is a vector of cost of delay, set similarly as in [14] and fuel price (as of 14.1.2014), all in Euro. The unit cost bounds in this paper, without loss of generality, were set to $\bar{\mathbf{c}} \pm 20\%$: $\bar{\mathbf{c}}^{upper} = [1.2 \times 0.469, 1.2 \times 0.71, 0]$ and $\bar{\mathbf{c}}^{lower} = [0.8 \times 0.469, 0.8 \times 0.71, 0]$. After the search, the resulting solutions were filtered to obtain $N^R = 10$ solutions.

4.2 Computational results

The performance of the proposed two-phase EMO framework (P-EMOA with the filtering procedure) was compared to other evolutionary algorithms: NSGA-II [16] without preference information, R-NSGA-II [21], MOEA/D [33] and NSGA-III [24] with preferences. MOEA/D decomposes the optimisation problem into subproblem based on weights. As preference in this paper is formulated using a scalarizing function and unit cost bounds, it directly applies to MOEA/D. More specifically, we use (15) as a decomposition method with weights randomly generated within $\bar{\mathbf{c}}^{upper}$ and $\bar{\mathbf{c}}^{lower}$ as defined in Section 4.1. R-NSGA-II and NSGA-III require reference points as an input, which were generated as follows. In each generation of R-NSGA-II and NSGA-III, the middle point and characteristic neighbours are calculated as defined in Section 3.1 and used as reference points. The crossover and mutation settings for NSGA-II, R-NSGA-II, MOEA/D and NSGA-III were identical as for P-EMOA. For R-NSGA-II, $\epsilon = 0.1$ is used to control the extent of obtained solutions. Other parameters of algorithms were left unchanged from the ones suggested in the original studies. It should be noted that although $c_3 = 0$, all algorithms consider and minimise all 3 objectives and c_3 applies only to preference information.

Fig. 6 shows an example of the Pareto front for $pek1$ instance. As can be seen, P-EMOA focused its search on the RoI. Setting no preference for the emissions $c_3 = 0$ produced a 3 dimensional front defined by the middle and characteristic neighbour points based only on unit costs incurred by c_1, c_2. Within this front, emissions are not restricted. Due to the inverse and non-linear relationship between fuel flow and emissions per kg of fuel described in Section 2.2, solutions with higher fuel consumption due to a higher acceleration rate and fuel flow have less emissions in total. From a visual comparison, P-EMOA resulted in a better convergence compared to other algorithms. For NSGA-II, this is expected, as it concentrates on the whole Pareto front. MOEA/D produced inferior solutions, although not very far from the ones generated by P-EMOA. Both R-NSGA-II and NSGA-III resulted in worse convergence compared to P-EMOA. Also, for both MOEA/D and NSGA-III, the resulting fronts tend to have solutions not evenly distributed.

In order to fairly compare the performance of P-EMOA with NSGA-II, R-



(a)

(b)

(c)

(d)

Figure 6: Pareto fronts from an experiment for *pek*1 instance shown (a) in a 3-objective view , (b) g_1, g_2 projection, (c) g_1, g_3 projection, (d) g_2, g_3 projection.

NSGA-II, MOEA/D and NSGA-III, a quantitative measure is further calculated. As the aim of the search in this study is to find solutions corresponding to the DM's preferences, a measure which can take into account the utility of obtained solutions according to preferences is needed. Traditional measures such as the hypervolume indicator [66], can be unsuitable for preference-based algorithms [67], as they do not take into account the preference information. For example, one set of solutions can have better hypervolume value compared to another set, even if it is further away from the RoI. Few performance measures have been introduced in the literature recently [67, 68], designed specifically for preferences expressed as reference points. These measures assume that the RoI is defined as solutions within a radius (set as a user parameter) from the reference point. However, this is different to the definition of RoI used in this work, i.e. solutions within a range of objective function values, determined by the characteristic neighbours, corresponding to $\bar{c}_m^{lower}$ and $\bar{c}_m^{upper}$ for objectives $m = 1, \ldots, nr^{obj}$. As a results, selecting solutions within a radius from the reference point may include solutions which are not in the RoI. Furthermore, as explained below, an improperly selected RoI can affect search performance. Therefore, the R3 indicator [69] is employed as it directly incorporates weights into the evaluation procedure, which enables to define an interval of weights, similar to $\bar{c}_m^{lower}$ and $\bar{c}_m^{upper}$. The R3 indicator evaluates Pareto set approximations based on a utility value of individual solutions. Let the weighted linear utility function $u(\lambda, P_A)$ of the Pareto front P_A be the minimum value of the scalarizing function (15) obtained across all solutions in P_A:

$$u(\lambda, P_A) = (\min_{\forall \mathbf{z_j} \in P_A} \{|C^{total}(\mathbf{z}^*, \lambda) - C^{total}(\mathbf{z}_k, \lambda)|\}). \tag{28}$$

where $\lambda \in \Lambda$ is a weight vector, $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ and $\mathbf{z}^*$ is the ideal point, i.e. the minimum values of objectives found during the experiments. Note, that in this case the lower value of the utility function corresponds to lower economic costs, thus better solution. In our case, the set Λ consists of 10,000 randomly generated vectors with the bounds $\lambda_1 \in (0.8 \times 0.469, 1.2 \times 0.469), \lambda_2 \in (0.8 \times 0.71, 1.2 \times 0.71), \lambda_3 = 0$, according to preferences set in Section 4.1.

Then, the I_{R3} indicator for two approximation sets P_A, P_B can be calculated as:

$$I_{R3}(P_A, P_B) = \frac{\sum_{\lambda \in \Lambda} [u(\lambda, P_B) - u(\lambda, P_A)]/u(\lambda, P_B)}{|\Lambda|} \tag{29}$$

Positive values of I_{R3} mean that P_A is preferable to P_B and larger values show bigger difference between the two sets.

Table 3: Average I_{R3} indicator for P-EMOA and one of the investigated algorithms for 30 runs before applying the filtering procedure and after.

	NSGA-II		R-NSGA-II		MOEA/D		NSGA-III	
	Before	After	Before	After	Before	After	Before	After
*man*1	0.1367	0.1692	0.0901	0.1061	-0.0024	-0.0024	0.1899	0.2028
*man*2	0.0929	0.1102	0.0439	0.0518	0.0398	0.0310	0.2039	0.2013
*man*3	0.0834	0.1076	0.1021	0.1112	0.0085	0.0030	0.1985	0.1983
*man*4	0.2110	0.2443	0.1419	0.1600	0.0846	0.0844	0.2501	0.2540
*man*5	0.0775	0.1214	0.0966	0.1163	0.1218	0.1185	0.2406	0.2429
*man*6	0.1630	0.2081	0.1169	0.1404	0.1347	0.1323	0.2563	0.2685
*doh*1	0.1161	0.1484	0.1372	0.1633	0.0067	0.0032	0.2035	0.2079
*doh*2	0.1103	0.1478	0.1235	0.1515	0.0079	0.0086	0.1979	0.2065
*pek*1	0.1470	0.1845	0.1079	0.1396	0.0934	0.0898	0.2118	0.2189
*pek*2	0.1351	0.1703	0.0933	0.1071	-0.0112	-0.0128	0.1960	0.2049

Average I_{R3} indicators for P-EMOA and one of the NSGA-II, R-NSGA-II, MOEA/D, NSGA-III algorithms over 30 runs are given in Table 3. The results presented are before applying the filtering procedure and after. The I_{R3} indicator is calculated before the filtering procedure, where all solutions generated by the algorithms are considered, and after the filtering, where only $N^R = 10$ solutions from each algorithm are compared. For comparisons with NSGA-II, R-NSGA-II and NSGA-III, all values are positive, meaning that P-EMOA resulted in a better convergence, i.e. better solutions in terms of utility than NSGA-II, R-NSGA-II and NSGA-III for both filtered and unfiltered solutions. For MOEA/D, most values are relatively small and positive, indicating a slightly worse performance compared to P-EMOA.

Table 4: Average utility for $man1$ for 10 runs of algorithms.

R-NSGA-II		NSGA-III	
ϵ	u	r	u
0.5	84.3981	1	83.5896
0.2	83.3158	0.75	85.7992
0.1	79.4351	0.5	82.9054
0.05	73.3123	0.35	82.6533
0.01	74.0890	0.25	82.1816
0.005	76.3028	0.15	85.7848
0.001	75.3737	0	89.4186

Despite R-NSGA-II and NSGA-III use preference information to focus their search, I_{R3} indicator showed worse convergence compared to P-EMOA. From algorithm point of view, it should be noted that the difference between R-NSGA-II, NSGA-III and P-EMOA lies in their replacement procedures, which selects solutions surviving to the next generation. R-NSGA-II and NSGA-III emphasise solutions associated with reference points. Solutions are associated to the closest reference point in terms of distance in objective space. In R-NSGA-II, all solutions near the reference point are associated with it and only one solution within a ϵ-neighbourhood is preferred [21]. In NSGA-III, all solutions near the reference point are associated with it and the preferred solutions are selected randomly [24]. Table 4 summarises results of experiments analysing the relationship between the size of space associated with a reference point and convergence conducted on $man1$ instance. The convergence is expressed by average utility u as defined in (28) over 10 runs of the investigated algorithms. For R-NSGA-II, ϵ value was varied, whereas for NSGA-III, the solutions associated with a reference point were ordered in terms of their distances to the reference point, and only a fraction r of closest solutions were considered. As can be seen, the size of space associated with a reference point determined by ϵ and r affects convergence to the RoI. For comparison, P-EMOA achieved $u = 72.4613$. In P-EMOA, outranking neighbourhood comprises a different portion of objective space compared to the space associated with a reference point in R-NSGA-II and NSGA-III. For larger values of ϵ and r, the space associated with a reference point is too large and with too diverse solutions, leading to worse convergence. On the other hand, for small values of ϵ and r the space associated with a reference point is too small and convergence is compromised due to lost of solution diversity.

To illustrate the practical relevance of the proposed approach, $pek1$ instance and Pareto fronts (before applying the filtering procedure) from Fig.6 are further analysed as a representative example. The solution with minimum g_1 from all experiments has objective values [3534, 1483, 5468]. The solution with minimum taxi time is often preferred in the previous research on the ground movement problem [2]. Suppose, that the DM is interested in solutions with minimum C^{total} for unit costs $\bar{\mathbf{c}} = [0.469, 0.71, 0]$. The objectives for such solutions are in Table 5. The best convergence of P-EMOA resulted in minimum values of g_1, g_2, C^{total} compared to other algorithms. In comparison with the solution

Table 5: Objectives g_1, g_2, g_3 with minimum C^{total} for unit costs $\bar{\mathbf{c}} = [0.469, 0.71, 0]$ for $pek1$ instance.

Algorithm	g_1 (s)	g_2 (kg)	g_3 (g)	C^{total} (EUR)
P-EMOA	3574	1425	5837	2688
NSGA-II	3581	1439	5796	2701
R-NSGA-II	3545	1446	5812	2689
MOEA/D	3583	1423	5976	2691
NSGA-III	3564	1448	5701	2700

with minimum taxi time, the best solution in cost for P-EMOA could save 58 kg of fuel at the expense of g_1, g_3. The difference for g_1, g_2 and C^{total} of P-EMOA and NSGA-II (with the highest C^{total}) is 7 s, 14 kg and 13 EUR, respectively. Beijing Capital International Airport had 567,759 departures/landings in 2013 [70]. Projecting the differences in values for P-EMOA and NSGA-II for the whole year would result in 233,783 s, 467,566 kg and 434,169 EUR, respectively. The annual difference between the solution with minimum taxi time, and the best solution in cost for P-EMOA results in 1,937,060 kg of fuel. As can be seen, even relatively small difference in this case can results in substantial amounts given the annual traffic numbers. For objective g_3, P-EMOA resulted in worse value than NSGA-II, as $\bar{c}_3 = 0$. However, if the DM is interested in lower values of g_3, he/she can choose a solution [3587,1431,5661] with the minimum value of g_3 for P-EMOA with $C^{total} = 2698$, which is still lower than C^{total} of the best solution in cost for NSGA-II.

I_{R3} is not the only measure that can be applied to compare the performance of algorithms. The consideration of distribution of solutions is also important from the DM's perspective. The distribution of solutions in the Pareto front in terms of the evenness measure as defined in (27) is compared in Table 6. The results for P-EMOA are given after applying the filtering procedure (the 1st column) and before (the 2nd column). Firstly, we compare P-EMOA without the filtering procedure with other algorithms. P-EMOA without the filtering procedure resulted in the similar evenness measure as NSGA-II, due to the similar niching mechanism by the crowding distance in (21). On the other hand, R-NSGA-II produced more evenly distributed solutions. Allowing only one solution within its ϵ-neighbourhood in R-NSGA-II actively limits the distance between solutions in contrast to the crowding distance. MOEA/D and NSGA-III which rely on the distribution of reference points and weights for the even distribution of solutions achieved worse results than P-EMOA without the filtering procedure. The solutions obtained by P-EMOA without the filtering procedure give a good starting point for the filtering procedure, after which P-EMOA achieved better evenness measure than other algorithms. The a posteriori filtering procedure complements the a priori crowding distance niching in P-EMOA. Note, that the filtering procedure can be used with any algorithm such as NSGA-II, R-NSGA-II, MOEA/D and NSGA-III. The filtering procedure applied to R-NSGA-II can obtain the same of slightly better evenness measure than P-EMOA. However, only P-EMOA could obtain an even distribution and good convergence of solutions at the same time as documented in Table 3 and

Table 6: Average evenness measure for 30 runs of algorithms.

	P-EMOA	P-EMOA	NSGA-II	R-NSGA-II	MOEA/D	NSGA-III
		w/o filtering				
*man*1	0.1983	0.5704	0.6130	0.5108	1.1168	0.9151
*man*2	0.2267	0.8060	0.9545	0.6436	0.9199	1.0900
*man*3	0.2290	0.7517	0.8532	0.5330	0.9214	1.0756
*man*4	0.2235	0.5857	0.6358	0.4814	0.8790	0.8032
*man*5	0.2437	0.7925	0.8043	0.5072	0.8863	1.0888
*man*6	0.2378	0.6141	0.6066	0.4927	1.1480	0.7500
*doh*1	0.2320	0.5087	0.5964	0.4149	1.3990	0.7091
*doh*2	0.2262	0.5440	0.5886	0.4740	0.8765	0.6908
*pek*1	0.2273	0.6089	0.6309	0.4721	1.1227	0.7067
*pek*2	0.2235	0.5937	0.6647	0.4777	1.4012	0.8399
Avg	0.2268	0.6376	0.6948	0.5007	1.0671	0.8669

Table 6. Given the relatively good performance of R-NSGA-II in the evenness measure, one could wonder if it could improve its convergence using the same mechanism as described in Section 3.1 to restrict its RoI in each generation. However, R-NSGA-II uses ϵ-neighbourhood, which in combination with restricted RoI would reduce the number of surviving solutions. More solutions found within the veto interval do not result in more solutions in RoI, as only one solution is allowed within its ϵ-neighbourhood. Therefore, a negative impact on its convergence is expected.

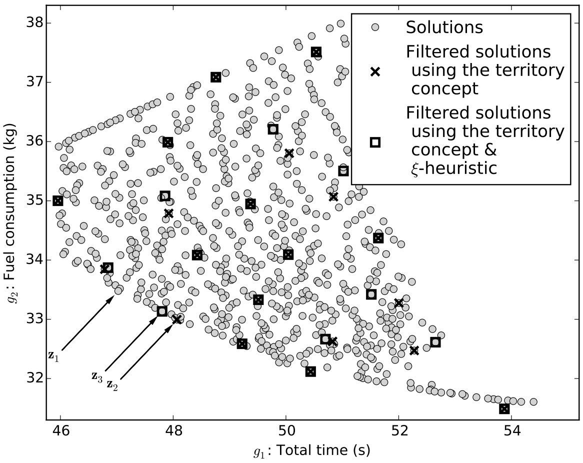


Figure 7: Pareto front for speed profile optimisation problem with filtered solutions.

The filtering procedure is further analysed in Table 7. The filtering procedure selects the solutions in two steps using: 1) the territory concept [19], 2)

ξ-heuristic as described in Section 3.2. Table 7 gives results of the each step applied to nondominated speed profiles for MAN and PEK airports as described in Section 2.2 and solutions of the integrated ground movement and runway scheduling problem in Section 2.4. Note, that for DOH airport a subset of speed profiles from PEK was used, thus no search and filtering was required. As it can be seen, the ξ-heuristic improved the evenness measure by 18% on average after the first step. An example of such filtering is shown in Fig. 7. The evenness measure ξ as defined in (27), improved from 0.2357 to 0.1315 in Fig. 7. Improved distribution has a practical implication for the DM. If the filtered solutions are not evenly distributed, some regions of the Pareto front will be covered less. In such a case, if the solution with minimum $C^{total}(\mathbf{z}_k, \bar{\mathbf{c}})$ lies in such regions, the nearest selected solution will be located far from the this solution, resulting in a solution with higher C^{total}, i.e. the total monetary cost being chosen by the DM. For example, suppose the unit costs were equal $c_1 = 3$, $c_2 = 8$, $c_3 = 0.2$, the solution with minimum $C^{total}(\mathbf{z}_k, \bar{\mathbf{c}})$ would be $\mathbf{z}_1 =$[47.03, 33.48, 191.51] with economic cost $C^{total}(\mathbf{z}_1, \bar{\mathbf{c}}) = 447.20$ in Fig. 7. Considering only filtered solutions from the first step in Fig. 7, the solution with minimum $C^{total}(\mathbf{z}_k, \bar{\mathbf{c}})$ is $\mathbf{z}_2 =$[48.07, 32.99, 200.30] with $C^{total}(\mathbf{z}_2, \bar{\mathbf{c}}) = 448.22$. For filtered solutions from the second step, the solution with minimum $C^{total}(\mathbf{z}_k, \bar{\mathbf{c}})$ is $\mathbf{z}_3 =$[47.80, 33.13, 196.29] with $C^{total}(\mathbf{z}_3, \bar{\mathbf{c}}) = 447.73$. As can be seen, more even distribution of filtered points resulted in a lower C^{total} (economic costs) for the efficient solution compared to the points obtained from the first step. Minimum C^{total} are important when selecting a subset of nondominated speed profiles for the database and economically efficient routes and schedules based on the speed profiles. For each aircraft $i = 1, 2, \ldots, h$ its set S_i usually consists of several building blocks for which speed profiles are retrieved from the database. For example, for 17 aircraft of $pek1$ instance 80 building blocks were needed. If we assume, that for each building block a similar difference $C^{total}(\mathbf{z}_2, \bar{\mathbf{c}}) - C^{total}(\mathbf{z}_3, \bar{\mathbf{c}}) = 0.49$ is between evenly distributed solutions and less evenly distributed ones, then the difference is 39.2 EUR in total. Again, for a yearly traffic of 567,759 departures/landings in 2013 [70] for Beijing Capital International airport this would result in 1,309,185 EUR difference. Therefore, any economic cost difference between the selected and efficient solution quickly adds up, emphasising the importance of even distribution.

5 Conclusion

In this paper, a multi-objective evolutionary optimisation framework with preferences and filtering procedure is proposed for the integrated optimisation problem. This problem combines airport ground movement and runway scheduling problem. The proposed approach was tested on data from major international hub airports. Two challenging tasks were addressed by the proposed evolutionary framework: 1) ability to handle approximate or uncertain preferences expressed as an interval of the economic value for a unit of each objective; 2) finding uniform distribution of solutions on the Pareto front.

To tackle the first challenge, a newly designed crowding distance within the replacement procedure of the evolutionary algorithm takes into account unit

Table 7: Average evenness measure for two steps of the filtering procedure.

Instance	Step1	Step2
Speed profiles *man*	0.4576	0.3836
Speed profiles *pek*	0.5584	0.4914
*man*1	0.2581	0.1983
*man*2	0.2767	0.2267
*man*3	0.2831	0.2290
*man*4	0.2748	0.2235
*man*5	0.2933	0.2437
*man*6	0.2873	0.2378
*doh*1	0.2823	0.2320
*doh*2	0.2785	0.2262
*pek*1	0.2800	0.2273
*pek*2	0.2771	0.2235

cost range for taxi delay and fuel consumption, to control the extent of the RoI. Efficient control of the extent of the RoI during the evolution successfully improved the convergence of the algorithm. Such improvement has a practical relevance, as better solutions in terms of total time, fuel consumption and emissions can be discovered. The computational results illustrated the scale of the potential savings compared to other baseline algorithms.

The second challenge was tackled by the proposed filtering procedure which was applied to select a uniformly distributed subset of speed profiles to be stored in the database and solutions from the RoI to be presented to the DM. Experiments highlighted the importance of having evenly distributed solutions in terms of the saved costs.

The computational experiments suggest that airports could benefit from adoption of the proposed approach for decision support.

For the future research, the proposed a posteriori filtering procedure could be embedded within EMO to select evenly distributed solutions during evolution. Also, the idea of including uncertainty into preference information in a systematic manner deserves more attention. As an example, the uncertainty could be expressed in terms of fuzzy values. Finally, finding not only optimal but also robust solutions (robust against deviation in taxi time, departure and arrival time) is of high importance when dealing with a real-world problem like airport operations. To this aim, elements of robust optimisation could be included within the search algorithm.

Data Access statement

The data and code used in this paper can be accessed at http://dx.doi.org/10.5281/zenodo.1197292.

Acknowledgement

This work is supported in part by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/H004424/1, EP/N029496/1 and EP/N029496/2.

Appendix. Notation

Table 8: Notation.

	Description		
g_1	total time objective	N_{sp}	number of speed profiles
g_2	fuel consumption objective	C^{total}	scalarizing function
g_3	total emission objective	$\mathbf{z}_k$	solution vector
H	Set of all aircraft	$\mathbf{c}$	vector of unit costs c_m
h	The number of all aircraft	m, n	objective index
A	Set of arriving aircraft	nr^{obj}	number of objectives
D	Set of departing aircraft	$\bar{\mathbf{c}}$	vector of the most probable unit costs $\bar{c}_m$
i, j, e	Aircraft index	$\bar{c}_m^{upper}, \bar{c}_m^{lower}$	upper/lower bound of unit cost $\bar{c}_m$
y_i	speed profile of aircraft i	$\mathbf{z}^C$	middle point
q_i	route of aircraft i	$\mathbf{z}_a^B$	characteristic neighbour
x_i	pushback time of aircraft i	a	characteristic neighbour index
acc	acceleration rate	v_m^{min}, v_m^{max}	minimum and maximum bounds of objective m
t_i^{base}	baseline departure time	$\bar{\mathbf{c}}^B$	boundary unit cost vector
s	building block index	nr^B	number of boundary unit cost vectors
$t_p(y_i)$	taxi time spent in phase p of y_i	cd_k	crowding distance of solution k
p	taxiing phase	M	big number constant
η	thrust level	d_k^{cd}	neighbouring solution distance of solution k
Thr	thrust	R	set of solutions from RoI
F_{oo}	maximum power output of jet engine	R^*, R^{**}	set of filtered solutions from RoI
$weight$	aircraft weight	τ	territory distance
μ	rolling resistance coefficient	g_m^{max}, g_m^{min}	the maximum/minimum of the m-th objective
$\phi_p(y_i)$	fuel flow in phase p of y_i	$\overline{g_m}$	normalised value of the m-th objective
$EI^{pp}(\phi_p(y_i))$	amount of pollutant for each kg of burned fuel for pp given $\phi_p(y_i)$	d_{kl}^{rect}	rectilinear distance between solutions k and l
pp	pollutant	δ	The maximum scaled absolute objective difference
S_i	Set of building blocks for the route of aircraft i	N^R	preferred number of solutions in R
t^{taxi}	total taxi time	o	territory adjustment
f^{taxi}	total taxi fuel consumption	inc_n	preferred minimum difference in objective n
ε^{taxi}	total taxi emission	$\overline{inc_m}$	normalised inc_n
$t_{i,s}$	taxi time for building block s of aircraft i	adj_n	adjustment for objective n
$f_{i,s}$	fuel consumption for building block s of aircraft i	d_k^L, d_k^U	diameters of the smallest/largest sphere for solution k
$\varepsilon_{i,s}^{pp}$	emissions of pp for building block s of aircraft i	ξ	evenness measure
W	wake vortex separation function	σ_d	standard deviation of diameters
w_i	weight category of aircraft i	$\hat{d}$	mean of diameters
t_i^r	landing/take-off time for aircraft i	ϵ	extent of obtained solutions for R-NSGA-II
t_i^d	arrival time at the runway holding point for aircraft i	u	utility function
t_i^w	waiting time for aircraft i	λ	weight vector
$\phi_{w_i}^{idle}$	idle fuel flow for w_i	Λ	set of weight vectors
t^{rwy}	total runway delay	P_A, P_B	Pareto fronts
f^{rwy}	total runway fuel consumption	$\mathbf{z}^*$	ideal point
ε^{rwy}	total runway emission	r	fraction of solutions for NSGA-III
k, l	solution index	g^{acc}	gravitational acceleration

References

[1] ICAO. (2014) Annual Report of the ICAO Council: 2013 The World of Air Transport. [Online]. Available: http://www.icao.int/annual-report-2013/Pages/the-world-of-air-transport-in-2013.aspx

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

[3] J. Chen, M. Weiszer, P. Stewart, and M. Shabani, "Toward a More Realistic, Cost-Effective, and Greener Ground Movement Through Active Routing - Part I: Optimal Speed Profile Generation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1196–1209, May 2016.

[4] J. Chen, M. Weiszer, G. Locatelli, S. Ravizza, J. Atkin, P. Stewart, and E. K. Burke, "Towards a More Realistic, Cost Effective and Greener Ground Movement through Active Routing: A Multi-Objective Shortest Path Approach," *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[5] A. Marín and E. Codina, "Network design: taxi planning," *Annals of Operations Research*, vol. 157, no. 1, pp. 135–151, 2008.

[6] C. Lesire, "An Iterative A* Algorithm for Planning of Airport Ground Movements," 19th European Conference on Artificial Intelligence (ECAI)/6th Conference on Prestigious Applications of Intelligent Systems (PAIS), Lisbon, Portugal, August 16-20, 2010.

[7] G. Clare and A. Richards, "Optimization of Taxiway Routing and Runway Scheduling," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1000–1013, Dec 2011.

[8] R. Deau, J. Gotteland, and N. Durand, "Airport surface management and runways scheduling," in *Proceedings of the 8th USA/Europe Air Traffic Management Research and Development Seminar, Napa, CA, USA*, 2009.

[9] S. Ravizza, J. Chen, J. A. Atkin, E. K. Burke, and P. Stewart, "The trade-off between taxi time and fuel consumption in airport ground movement," *Public Transport*, vol. 5, no. 1-2, pp. 25–40, 2013.

[10] P. C. Roling and H. G. Visser, "Optimal Airport Surface Traffic Planning Using Mixed-Integer Linear Programming," *International Journal of Aerospace Engineering*, vol. vol. 2008, p. 11, 2008.

[11] L. Bianco, P. Dell'Olmo, and S. Giordani, "Scheduling models for air traffic control in terminal areas," *Journal of Scheduling*, vol. 9, no. 3, pp. 223–253, 2006. [Online]. Available: http://dx.doi.org/10.1007/s10951-006-6779-7

[12] M. Samà, A. D'Ariano, and D. Pacciarelli, "Rolling horizon approach for aircraft scheduling in the terminal control area of busy airports," *Transportation Research Part E: Logistics and Transportation Review*, vol. 60, no. 0, pp. 140–155, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S136655451300118X

[13] 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, Part A, pp. 127–145, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X1500251X

[14] M. Weiszer, J. Chen, and G. Locatelli, "An integrated optimisation approach to airport ground operations to foster sustainability in the aviation sector," *Applied Energy*, vol. 157, pp. 567–582, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306261915004948

[15] M. Frankovich and D. Bertsimas, "Air Traffic Flow Management at Airports: A Unified Optimization Approach," in *Tenth USA/EUROPE Air Traffic Management Research & Development Seminar*, June 2013.

[16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[17] K. Deb and A. Kumar, "Light beam search based multi-objective optimization using evolutionary algorithms," in *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on.* IEEE, 2007, pp. 2125–2132.

[18] J. Branke and K. Deb, "Integrating user preferences into evolutionary multi-objective optimization," in *Knowledge incorporation in evolutionary computation.* Springer, 2005, pp. 461–477.

[19] İ. Karahan and M. Köksalan, "A territory defining multiobjective evolutionary algorithms and preference incorporation," *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 4, pp. 636–664, 2010.

[20] K. Miettinen and M. M. Mäkelä, *OR spectrum*, vol. 24, no. 2, pp. 193–213, 2002.

[21] K. Deb, J. Sundar, N. Udaya Bhaskara Rao, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 273–286, 2006.

[22] R. Purshouse, K. Deb, M. Mansor, S. Mostaghim, and R. Wang, "A review of hybrid evolutionary multiple criteria decision making methods," in *Evolutionary Computation (CEC), 2014 IEEE Congress on*, July 2014, pp. 1147–1154.

[23] C. C. Coello, "Handling preferences in evolutionary multiobjective optimization: A survey," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1. IEEE, 2000, pp. 30–37.

[24] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 4, pp. 577–601, 2014.

[25] A. López-Jaimes and C. A. C. Coello, "Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems," *Information Sciences*, vol. 277, no. 0, pp. 1–20, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025514004708

[26] A. Jaszkiewicz and R. Słowiński, "The 'Light Beam Search'approach–an overview of methodology applications," *European Journal of Operational Research*, vol. 113, no. 2, pp. 300–314, 1999.

[27] J. Branke, T. Kaußler, and H. Schmeck, "Guidance in evolutionary multi-objective optimization," *Advances in Engineering Software*, vol. 32, no. 6, pp. 499–507, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0965997800001101

[28] T. Friedrich, T. Kroeger, and F. Neumann, "Weighted preferences in evolutionary multi-objective optimization," in *AI 2011: Advances in Artificial Intelligence.* Springer, 2011, pp. 291–300.

[29] L. Ben Said, S. Bechikh, and K. Ghédira, "The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making," *Evolutionary Computation, IEEE Transactions on*, vol. 14, no. 5, pp. 801–818, 2010.

[30] J. Molina, L. V. Santana, A. G. Hernández-Díaz, C. A. Coello Coello, and R. Caballero, "g-dominance: Reference point based dominance for multiobjective metaheuristics," *European Journal of Operational Research*, vol. 197, no. 2, pp. 685–692, 2009.

[31] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evolutionary Computation*, vol. 17, no. 3, pp. 411–436, 2009.

[32] R. Wang, R. C. Purshouse, I. Giagkiozis, and P. J. Fleming, "The iPICEA-g: a new hybrid evolutionary multi-criteria decision making approach using the brushing technique," *European Journal of Operational Research*, vol. 243, no. 2, pp. 442–453, 2015.

[33] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[34] A. Mohammadi, M. N. Omidvar, and X. Li, "Reference point based multi-objective optimization through decomposition," in *Evolutionary Computation (CEC), 2012 IEEE Congress on.* IEEE, 2012, pp. 1–8.

[35] K. Deb, M. Mohan, and S. Mishra, "Evaluating the &epsi;-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions," *Evolutionary computation*, vol. 13, no. 4, pp. 501–525, 2005.

[36] J. Chen and M. Mahfouf, "A Population Adaptive Based Immune Algorithm for Solving Multi-objective Optimization Problems," in *Artificial Immune Systems*, ser. Lecture Notes in Computer Science,

H. Bersini and J. Carneiro, Eds. Springer Berlin Heidelberg, 2006, vol. 4163, pp. 280–293. [Online]. Available: http://dx.doi.org/10.1007/11823940_22

[37] S. L. Faulkenberg and M. M. Wiecek, "On the quality of discrete representations in multiple objective programming," *Optimization and Engineering*, vol. 11, no. 3, pp. 423–440, 2010.

[38] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: handling constraints and extending to an adaptive approach," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 4, pp. 602–622, 2014.

[39] S. Ravizza, J. A. Atkin, and E. K. Burke, "A more realistic approach for airport ground movement optimisation with stand holding," *Journal of Scheduling*, pp. 1–14, 2013.

[40] J. A. Atkin, E. K. Burke, and J. S. Greenwood, "TSAT allocation at London Heathrow: the relationship between slot compliance, throughput and equity," *Public Transport*, vol. 2, no. 3, pp. 173–198, 2010.

[41] ——, "A comparison of two methods for reducing take-off delay at London Heathrow airport," *Journal of Scheduling*, vol. 14, no. 5, pp. 409–421, 2011.

[42] P. Burgain, E. Feron, and J. Clarke, "Collaborative virtual queue: Benefit analysis of a collaborative decision making concept applied to congested airport departure operations," *Air Traffic Control Quarterly*, vol. 17, no. 2, pp. 195–222, 2009.

[43] J. Chen and P. Stewart, "Planning aircraft taxiing trajectories via a multiojective immune optimisation," in *Natural Computation (ICNC), 2011 Seventh International Conference on*, vol. 4, 2011, pp. 2235–2240.

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

[45] C. Evertse and H. Visser, "Real-time airport surface movement planning: Minimizing aircraft emissions," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 224–241, 2017.

[46] J. A. D. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson, "Hybrid Metaheuristics to Aid Runway Scheduling at London Heathrow Airport," *Transportation Science*, vol. 41, no. 1, pp. 90–106, 2007.

[47] H. Balakrishnan and B. Chandran, "Scheduling aircraft landings under constrained position shifting," in *AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO*, 2006.

[48] G. Sölveling and J.-P. Clarke, "Scheduling of airport runway operations using stochastic branch and bound methods," *Transportation Research Part C: Emerging Technologies*, vol. 45, pp. 119–137, 2014, advances in Computing and Communications and their Impact on Transportation

Science and Technologies. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X14000631

[49] X.-B. Hu and E. Di Paolo, "Binary-Representation-Based Genetic Algorithm for Aircraft Arrival Sequencing and Scheduling," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 2, pp. 301–310, June 2008.

[50] J. Bennell, M. Mesgarpour, and C. Potts, "Airport runway scheduling," *Annals of Operations Research*, vol. 204, no. 1, pp. 249–270, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10479-012-1268-1

[51] M. C. R. Murça and C. Müller, "Control-based optimization approach for aircraft scheduling in a terminal area with alternative arrival routes," *Transportation Research Part E: Logistics and Transportation Review*, vol. 73, pp. 96–113, 2015.

[52] M. Samà, A. D'Ariano, P. D'Ariano, and D. Pacciarelli, "Optimal aircraft scheduling and routing at a terminal control area during disturbances," *Transportation Research Part C: Emerging Technologies*, vol. 47, pp. 61–85, 2014, special Issue: Towards Efficient and Reliable Transportation Systems.

[53] ——, "Scheduling models for optimal aircraft traffic control at busy airports: Tardiness, priorities, equity and violations considerations," *Omega*, vol. 67, pp. 81–98, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305048316301165

[54] M. Samà, A. D'Ariano, F. Corman, and D. Pacciarelli, "Metaheuristics for efficient aircraft scheduling and re-routing at busy terminal control areas," *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 485–511, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968090X16301504

[55] A. Lieder and R. Stolletz, "Scheduling aircraft take-offs and landings on interdependent and heterogeneous runways," *Transportation Research Part E: Logistics and Transportation Review*, vol. 88, pp. 167–188, 2016.

[56] A. V. Sadovsky, D. Davis, and D. R. Isaacson, "Separation-compliant, optimal routing and control of scheduled arrivals in a terminal airspace," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 157–176, 2013.

[57] U. Benlic, A. E. Brownlee, and E. K. Burke, "Heuristic search for the coupled runway sequencing and taxiway routing problem," *Transportation Research Part C: Emerging Technologies*, vol. 71, pp. 333–355, 2016.

[58] J. Guépet, O. Briant, J.-P. Gayon, and R. Acuna-Agost, "Integration of aircraft ground movements and runway operations," *Transportation Research Part E: Logistics and Transportation Review*, vol. 104, pp. 131–149, 2017.

[59] M. Samà, A. D'Ariano, F. Corman, and D. Pacciarelli, "Coordination of scheduling decisions in the management of airport airspace and taxiway operations," *Transportation Research Procedia*, vol. 23, pp. 246–262, 2017,

papers Selected for the 22nd International Symposium on Transportation and Traffic Theory Chicago, Illinois, USA, 24-26 July, 2017.

[60] J. D. Scheelhaase, "Local emission charges – A new economic instrument at German airports," *Journal of Air Transport Management*, vol. 16, no. 2, pp. 94–99, 2010, selected Papers from the Air Transport Research Society Conference Athens, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0969699709000878

[61] T. Nikoleris, G. Gupta, and M. Kistler, "Detailed estimation of fuel consumption and emissions during aircraft taxi operations at Dallas/Fort Worth International Airport," *Transportation Research Part D: Transport and Environment*, vol. 16, no. 4, pp. 302–308, 2011.

[62] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-Objective Optimization Test Problems," in *Congress on Evolutionary Computation (CEC 2002)*. IEEE Press, 2002, pp. 825–830.

[63] R. T. Marler and J. S. Arora, "The weighted sum method for multi-objective optimization: new insights," *Structural and Multidisciplinary Optimization*, vol. 41, no. 6, pp. 853–862, 2010. [Online]. Available: http://dx.doi.org/10.1007/s00158-009-0460-7

[64] A. Messac and C. A. Mattson, "Normal constraint method with guarantee of even representation of complete Pareto frontier," *AIAA journal*, vol. 42, no. 10, pp. 2101–2111, 2004.

[65] A. Garrett, "Inspyred," 2012, (Version 1.0) [software]. Inspired Intelligence Initiative. Retrieved from http://inspyred.github.com.

[66] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms — A comparative case study," in *Parallel Problem Solving from Nature — PPSN V*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 292–301.

[67] K. Li and K. Deb, "Performance Assessment for Preference-Based Evolutionary Multi-Objective Optimization Using Reference Points," Tech. Rep., Jan. 2016, tech. Rep. COIN Report No. 2016001.

[68] A. Mohammadi, M. N. Omidvar, and X. Li, "A new performance metric for user-preference based multi-objective evolutionary algorithms," in *2013 IEEE Congress on Evolutionary Computation*. IEEE, 2013, pp. 2825–2832.

[69] M. P. Hansen and A. Jaszkiewicz, *Evaluating the quality of approximations to the non-dominated set*. IMM, Department of Mathematical Modelling, Technical Universityof Denmark, 1998.

[70] A. council international, "Aircraft Movements 2013 FINAL," online, 2014. [Online]. Available: http://www.aci.aero/Data-Centre/Annual-Traffic-Data/Movements/2013-final