A Rolling Window with Genetic Algorithm Approach to Sorting Aircraft for Automated Taxi Routing

Alexander E.I. Brownlee
Division of Computing Science and Mathematics
University of Stirling
Stirling, UK
alexander.brownlee@stir.ac.uk

John R. Woodward
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, UK
j.woodward@qmul.ac.uk

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

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

ABSTRACT

With increasing demand for air travel and overloaded airport facilities, inefficient airport taxiing operations are a significant contributor to unnecessary fuel burn and a substantial source of pollution. Although taxiing is only a small part of a flight, aircraft engines are not optimised for taxiing speed and so contribute disproportionately to the overall fuel burn. Delays in taxiing also waste scarce airport resources and frustrate passengers. Consequently, reducing the time spent taxiing is an important investment. An exact algorithm for finding shortest paths based on A* allocates routes to aircraft that maintains aircraft at a safe distance apart, has been shown to yield efficient taxi routes. However, this approach depends on the order in which aircraft are chosen for allocating routes. Finding the right order in which to allocate routes to the aircraft is a combinatorial optimization problem in itself.

We apply a rolling window approach incorporating a genetic algorithm for permutations to this problem, for real-world scenarios at three busy airports. This is compared to an exhaustive approach over small rolling windows, and the conventional first-come-first-served ordering. We show that the GA is able to reduce overall taxi time with respect to the other approaches.

CCS CONCEPTS

• **Computing methodologies** → **Search methodologies**; • **Applied computing** → **Transportation**;

KEYWORDS

transportation, aircraft taxiing, routing, permutations, genetic algorithm

GECCO '18, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5618-3/18/07…$15.00
https://doi.org/10.1145/3205455.3205558

ACM Reference Format:
Alexander E.I. Brownlee, John R. Woodward, Michal Weiszer, and Jun Chen. 2018. A Rolling Window with Genetic Algorithm Approach to Sorting Aircraft for Automated Taxi Routing. In *GECCO '18: Genetic and Evolutionary Computation Conference, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3205455.3205558

1 INTRODUCTION

We need to make better use of existing aviation infrastructure as worldwide air traffic is predicted to increase 1.5 times by 2035 [16]. This means fitting more aircraft into the same runways, terminal gates and taxiways. Although taxiing is only a small part of a flight, the inefficient operation of aircraft engines at taxiing speed can contribute disproportionately to the overall fuel burn. These effects apply particularly at larger airports, where ground manoeuvres are more complex, but also for short-haul operations, where taxiing represents a larger fraction of a flight. It is estimated that fuel burnt during taxiing alone represents up to 6% of fuel consumption for short-haul flights resulting in 5 m tonnes of fuel burnt per year globally, equating to 1.6 billion pounds given today's fuel prices. This situation is only set to get worse as more aircraft are squeezed in to the existing infrastructure.

Allocating routes to taxiing aircraft can be modelled as finding shortest paths across a graph, with the extension of a time dimension. A shortest path algorithm based on the well known A* algorithm is able to sequentially allocate efficient routes to aircraft, avoiding conflicts by routing around other aircraft, or waiting for other aircraft to move. However, this approach is affected by the order in which aircraft are presented for routing. This paper proposes the application of a rolling window approach, based on the receding horizon concept [6, 7, 21, 22, 37], incorporating a permutation-encoded genetic algorithm (GA), to find the best sequence in which to allocate routes to the aircraft. This approach is compared with the baseline method, First-Come-First-Served (FCFS), and an exhaustive search within small rolling windows. The approach is applied to three international airports: Doha, Hong Kong and Beijing Capital. We show that the GA was able to find better routings than FCFS and small window exhaustive search, by virtue of being able to explore much larger windows. The approach is able to reduce the delays by using FCFS by over 70%. This simple

application of a GA represents an effective improvement for aircraft taxi routing that could have substantial impact.

We begin with a summary of related work in Section 2 and introduce background concepts including the taxiing problem and A* algorithm in Section 3. We describe our rolling window methodology and the genetic algorithm in Section 4. Experiments and results for the three airports considered are given in Sections 5 and 6, before we draw our conclusions in Section 7.

2 RELATED WORK

Optimisation of taxi routes (or *Ground Movement*) is a challenging problem that has been tackled in a variety of ways. Comprehensive reviews of this area are [2] and [3]. Early approaches [18, 19, 30] used a list of routes that were either human-designed or generated before the algorithm was run using a shortest path algorithm. Metaheuristics were then used to choose an appropriate route and wait points for each aircraft. Genetic algorithms have also been used to evolve the routes rather than choosing predefined ones [24]. Alternative efforts including [13, 17, 20, 36] formulated Ground Movement as a mixed-integer linear programming problem. A recent approach using pre-computed routes formulated Ground Movement as a job-shop scheduling problem [1].

An alternative approach [32] generates the routes for aircraft each time, allowing the route and its timings to be tailored to account for the movements of other aircraft. Ravizza et al. described the Quickest Path Problem with Time Windows (QPPTW) algorithm, an adaptation of Dijkstra's shortest path algorithm that adds a time dimension and accounts for the movements of previously-allocated aircraft. The A* algorithm variant for taxiing that we use tackles the problem in much the same way [26]. While the approach will find the quickest path for an aircraft given existing movements, Ravizza et al. noted that QPPTW was dependent on the order in which aircraft were chosen for allocating routes: this also applies to the A* approach we use. This is the problem we tackle in the present paper.

Often the focus is on optimising taxi times, but other objectives have attracted some attention, particularly reducing aircraft emissions and fuel consumption due to taxiing [11, 12, 17, 35]. An advantage of accurate and efficient taxi routing is that aircraft can be made to wait at the gate until the last possible minute before starting engines and pushing back, saving fuel that would otherwise be wasted waiting in a queue at the runway [4].

Metaheuristics have been applied to other aviation optimisation problems including gate assignment [8], runway sequencing [7], and flight path planning [31]. The Rolling Window or Receding Horizon approach has been demonstrated to be useful in handling dynamic problems in aviation [6, 7, 13, 21, 22, 37] and a wide variety of other application domains (e.g. wireless base station planning [38]).

Metaheuristics have also found application in a variety of other routing and transportation problems including GPS navigation [33] and railway scheduling [15].

3 BACKGROUND

3.1 Taxiing and Ground Movement

Allocation of routes for aircraft taxiing or *Ground Movement* is a combined routing and scheduling problem [3]. Time-efficient routes must be allocated to aircraft seeking to pass through the taxiways between the runways and gates or stands. For safety reasons, it is crucial that two aircraft never conflict with each other throughout the taxiing process. All routes have to respect allocated runway times, taxi route restrictions, and safety constraints on the proximity of other aircraft. At airports with low air traffic, with only one or two aircraft moving at any one time, routes could be assigned using shortest path algorithms like Dijkstra's or A*. However, interactions between moving aircraft mean that a more sophisticated approach is required at busier airports. At busy airports, taxiways almost always cross each other. In addition, with obstacles such as runways aircraft must be carefully timed to cross busy junctions, reducing the need to change speed or stop. Both reducing speed or stopping can cause a knock-on effect in terms of delay, and also increases fuel consumption

3.2 A*

The routing algorithm used in this paper is a multi-objective shortest path A* algorithm [26] adapted to the ground movement problem. Although in the present work we only use one objective (taxi time) we intend to extend this to also consider fuel consumption, which has a complex relationship with speed. For each aircraft, routed sequentially, a set of optimal routes is found from the start node to the end node. Then, the fastest route is reserved. Aircraft are routed on a directed graph, representing the taxiway layout, where nodes represent gates, stands, taxiway intersections, intermediate points and runway exits and edges represent taxiways. In order to ensure conflict-free routes, each edge can be occupied by only one aircraft at a time. This also applies to any edges close to one holding an aircraft (within a radius of 60m) that would cause a conflict. Edges are connected together to form straight and turning segments. The angle between adjacent edges is less than 30 degrees in straight segments. For each straight segment, its taxi time corresponds to the fastest speed profile, i.e. maximum acceleration (0.1 g, where g is 9.8 ms^{-2}), followed by maximum allowed constant speed (30 knots) phase and braking (0.1 g). For turning segments, constant speed (10 knots) is assumed. A more detailed description of speed profile generation is in [12]. In order to accelerate the search, while not compromising the optimality of routes, a heuristic function is used. The heuristic function provides estimates of costs from the all nodes to the end node. For taxi time, it is calculated as the length of the shortest path to the end node divided by the maximum allowed speed (30 knots).

The advantage of this approach is that the route of the aircraft is not pre-determined, allowing greater flexibility for solutions. The times of entry/exit at the runway are fixed for departures and arrivals. The sequential approach to allocating taxi routes will then be used to minimise the taxi time for each individual aircraft given the planned movement for the aircraft which have already been routed.

4 METHODOLOGY

The natural order in which to allocate routes to aircraft is first-come-first-served (FCFS). If departing and arriving aircraft require routes allocated at the same time, arriving aircraft are allocated routes first because the departing aircraft can simply wait at the gate.

Ravizza et al. [32] noted approximately a 1% reduction in total taxi time by using a bespoke swap heuristic. This was embedded within their QPPTW algorithm. Wherever an aircraft deviated from its shortest possible path (either by having to stop, or taking a longer route), the conflicting aircraft causing the delay is identified, and the two aircraft are swapped and re-routed. If this results in a shorter total taxi time for the two aircraft, the new routes are kept, if not, the initial routes are retained. The approach we take is a separate search over permutations of aircraft, which can be independent of the specific shortest path algorithm used.

4.1 Rolling Windows

We can consider a window containing the next w waiting aircraft, and assign routes to aircraft for different potential orderings. A simple approach will examine all $w!$ permutations of aircraft in a window of size w. For a given window size w, all permutations of Aircraft 1 to w will be considered, and the routes allocated to the aircraft in the order specified by each permutation. The quickest routing is chosen then those aircrafts' routes are fixed. The window then rolls to Aircraft $w+1$ to $2w$, within which all permutations are considered, and routes allocated also accounting for the now chosen routes for Aircraft from the first window. This continues, with the window rolling forward w aircraft at a time, allocating routes to aircraft then fixing them so that subsequent aircraft avoid earlier ones. This is more formally set out in Algorithm 1 and illustrated in Figure 1.

However, this exhaustive approach is limited by the long run time of the routing algorithm. For example, around 5 seconds to allocate routes to 10-20 aircraft for Doha Airport, and up to 200 seconds for Beijing. These observations suggest a window size of more than 5 or 6 aircraft is not practical. However, at busy airports it can be the case that 10–15 aircraft are taxiing at any one time, any of which may be in conflict, so a larger window size might yield some improvement in the routes. This motivates use of a heuristic search method.

4.2 Genetic Algorithm

We now consider a simple implementation of a genetic algorithm to searching the permutations of aircraft within the larger rolling windows, in place of the exhaustive search described in the previous section. The GA will be run once for each window to determine the permutation that results in the shortest overall taxi time once aircraft routes have been allocated. The window will then move to the next set of aircraft and the GA will run again.

The GA encodes the aircraft in a window as a list of integers, each being an index into the aircraft list. The encoding is such that each aircraft is represented once and only once. The population is initialised with a set of permutations generated uniformly at random, with one seed solution in which the aircraft are arranged in FCFS order. Four elite solutions are carried over from one generation to the next. 2-tournament selection is used to choose parents for recombination. Recombination is the order-crossover [14], in which a sub-sequence of indices from the first parent is chosen at random and copied to the offspring. The offspring is then completed by adding the remaining indices in-order from the second parent. Mutation used two operators, chosen at random with an equal probability. The Swap Mutation operator [29], which takes two indices in the permutation and swaps them. Displacement mutation [27] takes a random sub-sequence within the permutation and shifts it to another position chosen at random.

Algorithm 1 Procedure for allocating routes to set of Aircraft $A = a_0, \ldots, a_{n-1}$, for a window. Note, there is a correction for the final window where its size is smaller than usual to avoid running off the end of the aircraft list: this is omitted for simplicity. Function reserve(r,G) reserves the route r on taxiway graph G.

```
Inputs: G,A,w        ▷ taxiway graph, set of aircraft, windowSize
s = 0                                                ▷ window start
while windowStart < n do
    W = a_s,...,a_{s+w}                  ▷ Aircraft in the current window
    minP = null, minTime = +∞
    for all Permutations p over W do ▷ can be run in parallel
        if s > 0 then                              ▷ if not on first window
            reserve((r_{a_{s-w}}, ..., r_{a_{s-1}}),G)
                ▷ Reserve routes for previous window's
                ▷ aircraft on taxiways, so the present
                ▷ windows aircraft avoid them
        end if
        Let time=0
        for a in W do
            r_a = allocateRoute(a)          ▷ Apply A* to find route
            t = length(r_a)                        ▷ (length in seconds)
            time += t
            reserve(r_a, G)                  ▷ Ensures remaining
                ▷ aircraft in the present window avoid a
        end for
        if time < minTime then
            minTime = time
            minP = p
        end if
        Clear reservations on G
            ▷ Allows next permutation to be checked afresh
    end for
    Apply routes found for minP to W
    s += windowSize
end while
```

5 EXPERIMENTS

5.1 Airport Scenarios

In this paper, we use a set of instances of real arrival and departure flights from 3 airports: Doha International Airport (DOH), Hong Kong International Airport (HKG) and Beijing Capital International (PEK). The complexity of the taxiway layout ranges from simple (DOH), medium (HKG) to complex (PEK). The instances consist of

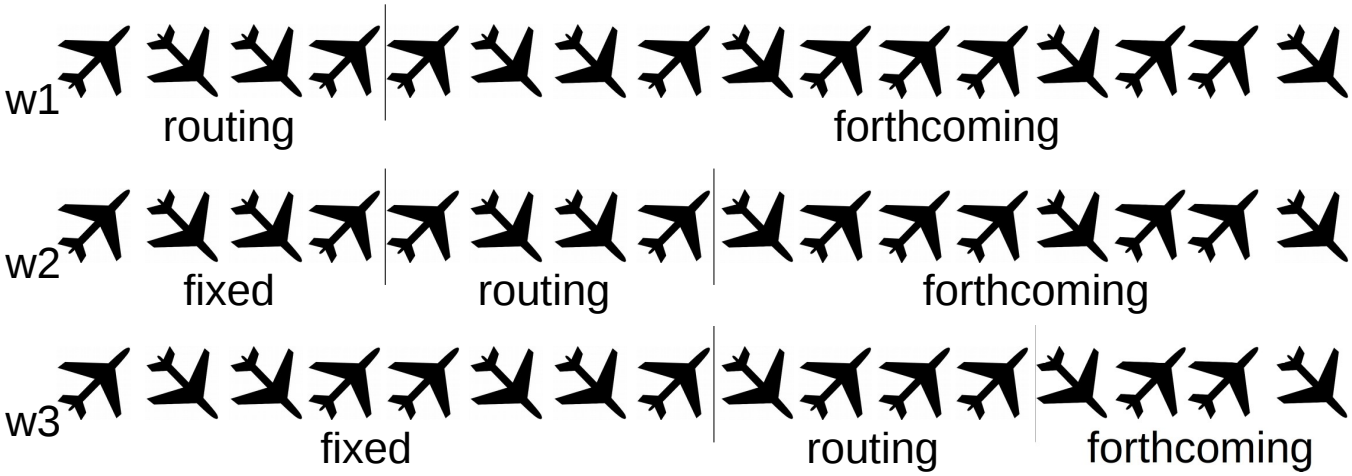

Figure 1: Illustration of a rolling window of size four (aircraft are a mixture of arrivals and departures)

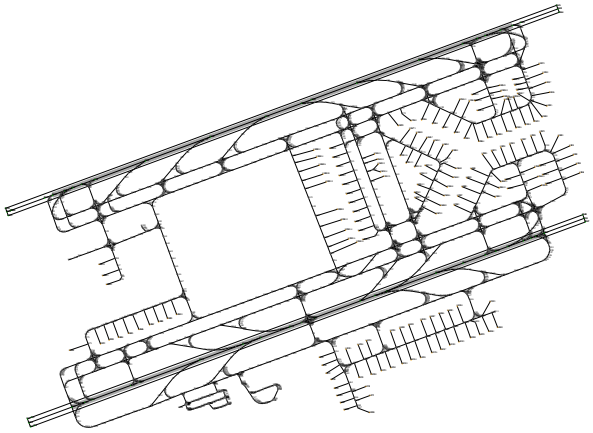

Figure 2: Layout of taxiways at Hong Kong International Airport

following number of aircraft: 180 (DOH) from 16.3.2014, 506 (HKG) from 17.1.2017 and 349 (PEK) from 9.7.2014. The data provided specifies landing/pushback times, gates/runway exits and weight category for each flight. The taxiway graph of Hong Kong International Airport is illustrated in Figure 2. The graph specifying the airport layout and movement data were prepared using the GM Tools[1] [10].

In all of the experiments, the route allocations were run in up to 24 parallel threads. This depends on the number of permutations being tested: obviously smaller GA population sizes, and window sizes 2 and 3 with 2 and 6 permutations respectively, did not use all these threads).

[1]https://github.com/gm-tools/gm-tools/wiki

5.2 Parameter tuning

There is a balance to be made in where the effort of evaluating permutations takes place. In the dynamic and rapidly changing environment of the airport, there is only a certain allowable length of time that we can take to find a solution for each group of aircraft. This is an important physical constraint if we aim for our approach to be used in real time under operating constrains and conditions. This time limit corresponds to an upper limit on the number of permutations that we can sample for a set of aircraft. The following two factors trade-off against each other:

- a larger window size means a larger search space, so potentially needing a longer-running GA, but fewer windows will be required to cover all aircraft
- in common with many other applications, the GA can run for a fixed length of time, meaning either a larger population for few generations, or a smaller population for many generations
- the run length for the routing algorithm increases with the number of aircraft, but due to set up times there is not an easy rule of thumb to determine the optimal number of aircraft to route with respect to running time

The run time for our A* implementation for a permutation of 20 aircraft on Hong Kong Airport is approximately 40s (DOH is around 5s, PEK is around 200s). Ideally we want the runs to complete in real time, so at 120s separation between aircraft, we have 2400s per window, so only about 60 evaluations allowed per window: 3 per aircraft. We can (and do) improve this situation by running evaluations of the permutations in parallel: with 24 threads we are limited 1440 evaluations for windows of 20 aircraft, or 72 evaluations per aircraft. Obviously we could add additional cores within reason to extend this further (at least 5x the number so that we can achieve the same run times for PEK as for HKG). The above might seem like a long-winded way of reaching the average figure of 72s per aircraft: it needs to be thought of from the perspective of a reasonable window length (in the range of 10-60) because due to set-up time, a run of the routing algorithm for a single aircraft

is still several seconds, and while not constant is fairly consistent across this range of window sizes. These run times do not impact on the potential taxi time savings of the approach: the reason we limit the times is to ensure a high enough throughput that aircraft can have their routes allocated before they are due to start taxiing.

In order to find a good balance between the population size, GA evaluation limit and window size, we ran a preliminary parameter tuning experiment on a subset of 120 aircraft at Hong Kong Airport. At 72 evaluations per aircraft this means we have 8640 evaluations for all 120 aircraft. Population size and window size were varied, then the GA evaluation limit computed by dividing the total number of allowed evaluations (8640) by the number of windows required to cover the 120 aircraft. The tuning was carried out using the Sequential Model-based Algorithm Configuration (SMAC) tool [23], with a limit of 1000 GA runs. The tuning also included the mutation and crossover rates for the GA: these are simply the proportion of offspring that crossover and mutation were applied to.

The best parameters found by SMAC were as follows:

- GA population size: 17
- Window size: 56
- GA evaluation limit (computed from the above): 4032
- Crossover rate: 0.226
- Mutation rate: 0.496

At this point we have made the assumption that the configuration for HKG will carry over to the other airports. Strictly speaking the parameter tuning should be carried out for each, but in this study we are more interested in proving the concept rather than ensuring maximal performance.

6 RESULTS

Table 1 summarises the results. The first block of the table gives total taxi times in seconds for all routed aircraft at each airport when using the different approaches. The lower bound was determined by running the routing algorithm for each aircraft in isolation, avoiding any waiting times or detours. This is of course may not be achievable, but gives us some indication of performance. FCFS is the result when using the standard first-come-first-served order for allocating routes. "Exhaustive WSx" are the figures for the exhaustive rolling window approach, with window sizes of 2–6. "GA RW" are the average figures from 20 repeat runs of the GA rolling window approach with the tuned parameters (including a window size of 56). "GA all aircraft" was a single run of the GA conducted as a sanity check, working over the permutation of all aircraft, running for 3000 evaluations with a population size of 20 (this would be no use in practice as the run time was several hours).

The "gap" values in the lower half of Table 1 are the gap or delay in seconds between the lower bound and the result found by each of the methods. The "reduction of gap" values are the relative improvement of the GA WS56 approach over FCFS, Exhaustive WS2 and Exhaustive WS3. The exhaustive search over rolling windows of 6 for HKG and 5 and 6 for PEK is omitted; in these cases the run times were prohibitive for practical use.

It is clear that all of the approaches offer an improvement over FCFS, reducing the gap between the routed times and the theoretical lower bound for the situation where no aircraft are in conflict with each other. In general, increasing the window size means that

Table 1: Results: total taxi time at each airport when using each method ('Exhaustive' abbreviated to 'Exh.'). For the GA WS56 rolling window results, this is the mean over 20 runs, with standard deviation given in subscript

	Total taxi time (s)		
Method	DOH	HKG	PEK
Lower bound	32 430	128 763	88 119
FCFS	32 685	129 366	88 707
Exh. WS2	32 685	129 229	88 703
Exh. WS3	32 685	129 274	88 542
Exh. WS4	32 515	129 066	88 640
Exh. WS5	32 515	128 954	
Exh. WS6	32 684		
GA RW	**32 435** $_{35}$	**128 852** $_{100}$	**88 246** $_{79}$
GA all aircraft	32 515	129 070	88 192
FCFS gap	255	603	588
Exh. WS2 gap	255	466	584
Exh. WS3 gap	255	511	423
Exh. WS4 gap	85	303	521
Exh. WS5 gap	85	191	
Exh. WS6 gap	254		
GA RW gap	**64**	**89**	**127**
GA all aircraft gap	85	2307	1073
% reduction of gap over FCFS	74.90	85.24	78.40
% reduction of gap over Ex. WS2	74.90	80.90	78.25
% reduction of gap over Ex. WS3	74.90	82.58	69.98

permutations leading to faster routes can be found, although in some cases (e.g. DOH window sizes 5 and 6), increasing the size actually causes the delays to increase. This is because the change in window size causes conflicting [2] aircraft to fall on the boundary between windows where they previously did not: when this happens they cannot be swapped to improve the conflict situation.

The GA with rolling window approach (GA RW) was able to find the permutation leading to the shortest overall taxi times for all three airports. In all, the reduction in the delay (gap) over the minimal times compared to FCFS was between 74 and 85%, with similar reductions for the exhaustive rolling window approaches with small window sizes.

The sanity-check single GA run on the whole permutation for each airport was not able to improve on the GA with rolling window: we expect that this was due to the large search space involved but this requires further investigation.

The reductions in delay with respect to the total taxi times may seem insignificant in terms of percentages, but it is important to see how this translates into financial cost for this real world problem. Given a lower bound of 128 763s for the movements of the

[2] *conflicting* in the sense that they are competing for a route rather than risking safety

aircraft at Hong Kong Airport, reducing the delays by 514s (0.40%) may seem trivial. However, this is for 506 aircraft, part of one day of movements. For the 421 000 aircraft movements in 2017, this amounts to a saving of 427 656s. Following the assumption of [9] that a single aisle jet aircraft uses 25 pounds of fuel per minute while taxiing, and an assumed $5 US per gallon of fuel (one gallon being 6.7lbs), the savings in fuel at Hong Kong Airport alone would be approximately $133k per year. However, it should be noted that other sources question the actual fuel rate for taxiing, which is possibly slightly overestimated by ICAO [25, 28].

7 CONCLUSION AND FUTURE WORK

Growing air traffic means heavier demands on existing airport infrastructure including taxiways. Although taxiing is only a small part of an overall flight, ground movement is is responsible for a disproportionate amount of fuel burn. This is because jet engine are very efficient at cruising speed, but relatively inefficient while moving on the ground at low speed. Consequently, even small improvements in efficiency in taxiing will reduce economic and environmental costs, notwithstanding reduced frustration for delayed passengers.

An existing method based on A* has proven successful for routing aircraft without conflicts, but its performance is dependent on the sequence in which aircraft are presented to it for routing. We proposed the application of a rolling window approach incorporating a permutation-encoded genetic algorithm to find the best sequence in which to allocate routes to the aircraft. This was compared with FCFS, and an exhaustive search within small rolling windows. When applied to Doha, Hong Kong and Beijing Capital International Airports, the GA based approach was able to find better routings than FCFS or exhaustive search. Delays added by FCFS were reduced by over 70%. This simple application of a relatively naive genetic algorithm represents an effective improvement for aircraft taxi routing that could have substantial impact.

In terms of future directions to explore, rather than moving the complete window forward each time, a more continuous rolling window could be considered. For window size W, the order would be determined, $n < W$ aircraft would be dispatched, the window moved on by n, and the next order for the next W aircraft found (potentially reordering the last $W - n$ of the first window's aircraft). This would avoid the anomalous situation we observed where conflicting aircraft span the boundary of a window and can not be reordered. Additionally, given that one of the key obstacles encountered in this work was the long running time of the routing algorithm, we are investigating more advanced search algorithms such as model based algorithms (e.g. [5]) and surrogates (one possibility for permutations being [34]). Our model of aircraft movement also incorporates an estimation of fuel consumption for the allocated routes and we are particularly interested in exploring what the possible multi-objective trade-off between fuel consumption and taxi time looks like. Understanding this will be an important part of tackling the problem of airport congestion as air traffic continues to grow.

ACKNOWLEDGEMENT

Work funded by UK EPSRC [grants EP/N029577/2 and EP/N029496/2].

DATA ACCESS STATEMENT

The taxiway layout and aircraft movement data cannot be shared due to licensing restrictions.

REFERENCES

[1] L. Adacher, M. Flamini, and E. Romano. 2018. Airport Ground Movement Problem: Minimization of Delay and Pollution Emission. *IEEE Trans. on Intelligent Transportation Systems* PP, 99 (2018), 1–10. https://doi.org/10.1109/TITS.2017.2788798

[2] Jason A.D. Atkin. 2013. Airport Airside Optimisation Problems. In *Automated Scheduling and Planning*, A. Sima Uyar, Ender Ozcan, and Neil Urquhart (Eds.). Studies in Computational Intelligence, Vol. 505. Springer Berlin Heidelberg, 1–37. https://doi.org/10.1007/978-3-642-39304-4_1

[3] J. A. D. Atkin, E. K. Burke, and S. Ravizza. 2010. The Airport Ground Movement Problem: Past and Current Research and Future Directions. In *4th International Conference on Research in Air Transportation*. 131–138.

[4] Jason A. D. Atkin, Geert De Maere, Edmund K. Burke, and John S. Greenwood. 2012. Addressing the Pushback Time Allocation Problem at Heathrow airport. *Transport. Science* 47, 4 (2012), 584 – 602. https://doi.org/10.1287/trsc.1120.0446

[5] M. Ayodele, J. McCall, O. Regnier-Coudert, and L. Bowie. 2017. A Random Key based Estimation of Distribution Algorithm for the Permutation Flowshop Scheduling Problem. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. 2364–2371. https://doi.org/10.1109/CEC.2017.7969591

[6] J. Bellingham, Y. Kuwata, and J. How. 2003. Stable Receding Horizon Trajectory Control for Complex Environments. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA 2003-5635*.

[7] Una Benlic, Alexander E. I. Brownlee, and Edmund K. Burke. 2016. Heuristic search for the coupled runway sequencing and taxiway routing problem. *Transportation Research Part C: Emerging Technologies* 71 (October 2016). https://doi.org/10.1016/j.trc.2016.08.004

[8] U. Benlic, E.K. Burke, and J.R. Woodward. 2014. Breakout Local Search for Gate Allocation Problem. *Submitted manuscript* (2014).

[9] C Brinton, C Provan, S Lent, T Prevost, and S Passmore. 2011. Collaborative departure queue management: An Example of collaborative decision making in the United States. In *9th USA/Europe Air Traffic Management Research and Development Seminar (ATM2011), Berlin, Germany.*

[10] A. E. I. Brownlee, J. A. D. Atkin, J. A. W. Woodward, U. Benlic, and E. K. Burke. 2014. Airport Ground Movement: Real World Data Sets & Approaches to Handling Uncertainty. In *Proc. PATAT*. York, UK.

[11] J. Chen, M. Weiszer, G. Locatelli, S. Ravizza, J. A. Atkin, P. Stewart, and E. K. Burke. 2016. Toward a More Realistic, Cost-Effective, and Greener Ground Movement Through Active Routing: A Multiobjective Shortest Path Approach. *IEEE Trans. on Intell. Transp. Systems* 17, 12 (Dec 2016), 3524–3540. https://doi.org/10.1109/TITS.2016.2587619

[12] J. Chen, M. Weiszer, P. Stewart, and M. Shabani. 2016. Toward a More Realistic, Cost-Effective, and Greener Ground Movement Through Active Routing–Part I: Optimal Speed Profile Generation. *IEEE Trans. on Intel. Transp. Systems* 17, 5 (2016), 1196–1209. https://doi.org/10.1109/TITS.2015.2477350

[13] G. L. Clare and A. G. Richards. 2011. Optimization of Taxiway Routing & Runway Scheduling. *IEEE T Intell Tran Syst* 12, 4 (2011), 1000–1013. https://doi.org/10.1109/TITS.2011.2131650

[14] Lawrence Davis. 1985. Applying Adaptive Algorithms to Epistatic Domains. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence - Volume 1 (IJCAI'85)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 162–164.

[15] J. Eaton, S. Yang, and M. Gongora. 2017. Ant Colony Optimization for Simulated Dynamic Multi-Objective Railway Junction Rescheduling. *IEEE Transactions on Intelligent Transportation Systems* 18, 11 (Nov 2017), 2980–2992. https://doi.org/10.1109/TITS.2017.2665042

[16] Eurocontrol. 2013. Challenges of Growth 2013-Task 4: European Air Traffic in 2035. (2013).

[17] C. Evertse and H.G. Visser. 2017. Real-time airport surface movement planning: Minimizing aircraft emissions. *Transportation Research Part C: Emerging Technologies* 79 (jun 2017), 224–241. https://doi.org/10.1016/j.trc.2017.03.018

[18] J-B. Gotteland and N. Durand. 2003. Genetic Algorithms Applied to Airport Ground Traffic Optimization. In *Proc. IEEE CEC*. Canberra, Australia, 544–551. https://doi.org/10.1109/CEC.2003.1299623

[19] J-B. Gotteland, N. Durand, J-M. Alliot, and E. Page. 2001. Aircraft Ground Traffic Optimization. In *Proc. Int'l Air Traf. Mgmnt. R&D Seminar.*

[20] J. Guépet, O. Briant, J.P. Gayon, and R. Acuna-Agost. 2016. The aircraft ground routing problem: Analysis of industry punctuality indicators in a sustainable perspective. *European Journal of Operational Research* 248, 3 (feb 2016), 827–839. https://doi.org/10.1016/j.ejor.2015.08.041

[21] X.B. Hu and W.H. Chen. 2005. Genetic algorithm based on receding horizon control for arrival sequencing and scheduling. *Engineering Applications of Artificial Intelligence* 18, 5 (2005), 633 – 642.

[22] X.B. Hu and E. Di Paolo. 2008. Binary-Representation-Based Genetic Algorithm for Aircraft Arrival Sequencing and Scheduling. *Intelligent Transportation Systems, IEEE Transactions on* 9, 2 (June 2008), 301–310.

[23] F. Hutter, H.ËIJH. Hoos, and K. Leyton-Brown. 2011. Sequential Model-Based Optimization for General Algorithm Configuration. In *LION-5 (LNCS)*. 507–523.

[24] Y. Jiang, Z. Liao, and H. Zhang. 2013. A Collaborative Optimization Model for Ground Taxi Based on Aircraft Priority. *Math Probl Eng* 2013 (2013), 1–9. https://doi.org/10.1155/2013/854364

[25] Brian Kim and Jawad Rachami. 2008. Aircraft emissions modeling under low power conditions. In *101st Air and Waste Management Association Annual Conference and Exhibition, Portland, OR, Paper*. Citeseer.

[26] Lawrence Mandow and JosÃĺ Luis PÃĺrez De La Cruz. 2010. Multiobjective A* search with consistent heuristics. *Journal of the ACM (JACM)* 57, 5 (2010), 27.

[27] Zbigniew Michalewicz. 1996. *Genetic Algorithms + Data Structures = Evolution Programs (3rd Ed.)*. Springer-Verlag, London, UK, UK.

[28] Kevin M Morris. 2005. Results from a number of surveys of power settings used during taxi operations. *British Airways. EJT/KMM/1126/14.8* (2005).

[29] I. M. Oliver, D. J. Smith, and J. R. C. Holland. 1987. A Study of Permutation Crossover Operators on the Traveling Salesman Problem. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Application*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 224–230.

[30] B. Pesic, N. Durand, and J-M. Alliot. 2001. Aircraft Ground Traffic Optimisation using a Genetic Algorithm. In *Proc. GECCO*.

[31] Vincent R. Ragusa, H. David Mathias, Vera A. Kazakova, and Annie S. Wu. 2017. Enhanced Genetic Path Planning for Autonomous Flight. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '17)*. ACM, New York, NY, USA, 1208–1215. https://doi.org/10.1145/3071178.3071293

[32] S. Ravizza, J. A. D. Atkin, and E. K. Burke. 2013. A more realistic approach for airport ground movement optimisation with stand holding. *Journal of Scheduling* 17, 5 (Oct 2013), 507–520. https://doi.org/10.1007/s10951-013-0323-3

[33] Daniel H. Stolfi and Enrique Alba. 2017. Computing New Optimized Routes for GPS Navigators Using Evolutionary Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '17)*. ACM, New York, NY, USA, 1240–1247. https://doi.org/10.1145/3071178.3071193

[34] J. Swan. 2017. Harmonic analysis and resynthesis of Sliding-Tile Puzzle heuristics. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. 516–524. https://doi.org/10.1109/CEC.2017.7969355

[35] Michal Weiszer, Jun Chen, and Paul Stewart. 2015. A real-time Active Routing approach via a database for airport surface movement. *Transp. Res. Pt. C: Emerg. Tech.* 58 (2015), 127 – 145. https://doi.org/10.1016/j.trc.2015.07.011

[36] K. Yin, C. Tian, B. X. Wang, and L. Quadrifoglio. 2012. Analysis of Taxiway Aircraft Traffic at George Bush Intercontinental Airport, Houston, Texas. *Transp. Res. Record* 2266, 1 (2012), 85–94.

[37] Z.H. Zhan, J. Zhang, Y. Li, O. Liu, S.K. Kwok, W.H. Ip, and O. Kaynak. 2010. An Efficient Ant Colony System Based on Receding Horizon Control for the Aircraft Arrival Sequencing and Scheduling Problem. *IEEE Transactions on Intelligent Transportation Systems* 11, 2 (2010), 399–412.

[38] Hong-Yuan Zhang, Yu-Geng Xi, and Han-Yu Gu. 2007. A rolling window optimization method for large-scale WCDMA base stations planning problem. *European journal of operational research* 183, 1 (2007), 370–383.