View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Repository@Nottingham

Journal of Applied Operational Research *(2015) Vol. 7, No. 2, 68–79* ISSN 1735-8523 (Print), ISSN 1927-0089 (Online)

www.orlabanalytics.ca

The effects of pushback delays on airport ground movement ☆

Christofas Stergianos [1,*], Jason Atkin [1], Patrick Schittekat [2], Tomas E. Nordlander [2], Chris Gerada [1] and Herve Morvan [1]

[1] Institute for Aerospace Technology, The University of Nottingham, Nottingham, UK

[2] SINTEF ICT, Department of Applied Mathematics, Oslo, Norway

Received 14 October 2015
Accepted 07 November 2015

Keywords:
Airport operations
Ground movement optimisation
Pushback delay
Routing
Scheduling

Abstract—With the constant increase in air traffic, airports are facing capacity problems. Optimisation methods for specific airport processes are starting to be increasingly utilised by many large airports. However, many processes do happen in parallel, and maximising the potential benefits will require a more complex optimisation model, which can consider multiple processes simultaneously and take into account the detailed complexities of the processes where necessary, rather than using more abstract models. This paper focuses on one of these complexities, which is usually ignored in ground movement planning; showing the importance of the pushback process in the routing process. It investigates whether taking the pushback process into consideration can result in the prediction of delays that would otherwise pass unnoticed. Having an accurate model for the pushback process is important for this and identifying all of the delays that may occur can lead to more accurate and realistic models that can then be used in the decision making process for ground movement operations. After testing two different routing methods with a more detailed pushback process, we found that many of the delays are not predicted if the pushback process is not explicitly modelled. Having a more precise model, with accurate movements of aircraft is very important for any integrated model and will allow ground movement models to be of use in more reliable integrated decision making systems at airports. Minimising these delays can help airports increase their capacity and become more environmentally friendly.

Published 01 December 2015

Copyright © ORLab Analytics Inc. All rights reserved.

Introduction

Over the years, airports have become increasingly busy and many are already facing capacity problems. There is a considerable amount of research into optimising the processes at the airports. Successful optimisation of these processes can save considerable fuel and emissions, and reduce delays. The ground movement of aircraft is one of the most important operations and includes a number of sub-problems that can be optimised (Atkin *et al*. 2010). For example, departing aircraft will first push back from the gates (the pushback process), then taxi around the airports (the taxi process) and queue for the runway (the runway sequencing process).

Ground operations can be divided into several sub-problems, such as the runway sequencing problem (Bennell *et al*. 2011, Apice *et al*. 2014), which can involve an explicit model for the ground movement element (Atkin *et al*. 2007); the gate allocation problem (Bouras *et al*. 2014, Dorndorf *et al*. 2007); and the ground movement routing and scheduling problem itself (Rolling and Visser 2008, Atkin *et al*. 2010, Ravizza *et al*. 2014a). These problems interact with each other and the solution of one can affect another. There has been some research towards the integration of processes (Kjenstad *et al*. 2013). Taking the interactions between problems into account within models can increase the accuracy of the models, in terms of modelling the real world behaviour, as well as increasing the applicability of the results. This paper considers the integration of the pushback process into the ground movement problem. Although the ground movement problem has received significant research attention, there has been very little consideration of the pushback process. Tu *et al*. (2008) attempted to identify the delays that happen during the routing process with the use of statistical analysis. They took into consideration a number of trends and patterns like weather impact, delay built up from previous flights, seasonal and daily patterns, in order to predict the difference between the scheduled time and actual time that an aircraft was going to start the pushback process. Neuman and Atkin (2013) attempted to find the conflicts that may occur because of the pushback process or the conflicts that happen close to the gates in order to better allocate aircraft to gates. Atkin *et al*. (2013)

---

[*] *Correspondence*: Christofas Stergianos, Institute for Aerospace Technology, The University of Nottingham, Nottingham, NG7 2RD, UK
E-mail: christofas.stergianos@nottingham.ac.uk

☆ This work is distributed under the terms of a self-archiving open access article which permits unrestricted noncommercial use, distribution and reproduction in any medium, provided the original authors and source are credited. Copyright © 2015 the authors.

used a model to predict the total delays for aircraft (at the stands or the runway) in order to absorb more of this time at the stand, before the pushback process of the aircraft commences. Consideration of the time which was needed to perform the pushback process and start the engines was an important element of the system. Cheng (1998) developed a model that predicts and resolves conflicts on the taxiways close to the gates, in order to minimise the delay. Burgain *et al.* (2012) used a stochastic model of surface operations to control the pushback clearances based on the number of aircraft that are taxiing. However these models do not explicitly examine the effects of the pushback process upon the ground movement, instead focusing on the minimisation of the total travel time and/or queuing time at the runway.

Ravizza *et al*. (2013) used a statistical approach to predict total taxi times, then Ravizza *et al*. (2014b) compared a number of approaches for this, but understanding the real ground movement problem requires a better understanding of where delays actually occur as well as the total unimpeded delays. This paper proposes that, in order to achieve a more realistic model which will be able to assess the effects of the pushback delays, the pushback process needs to be explicitly modelled within the routing process. This involves taking into consideration the elements which are known to affect this delay and ensuring that the delay occurs at the same position in the model as it does in the real world; by the gates, where pushback occurs, rather than being spread over the entire taxi duration. This proposal is evaluated in this paper using two different methods for aircraft routing, showing that the pushback delays have a measurable effect in both cases.

Consideration of the size of the aircraft and the morphology of the taxiways are two important aspects of the model, which influence the precision of the results. Accurate times as well as accurate sequencing of aircraft movements are key for building increasingly precise integrated models and will allow these models to be used not only for predictions but also for reliable integrated decision making systems at airports.

This paper presents our ongoing development of a more integrated and detailed model for the ground movement of aircraft. Section 2 describes the problem of the pushback process. Section 3 presents our solution approach. Section 4 states the results and Section 5 concludes the paper and proposes future work.

Problem description

The pushback process (which is the part of the ground movement process where the aircraft pushes back from the gate and starts its engines) is a crucial point where delays can (and do) happen. While an aircraft is being pushed back and its engines are started, it can block other aircraft that are moving around the airport. The pushback and engine start-up process is often a time-consuming process. While this is happening, other aircraft may not able to pushback if they are using stands that are close by. In cases where the taxi area around the gates is not wide enough to be simultaneously used by two aircraft, a taxiway may be blocked by the aircraft for the duration of the process. In some cases, for safety reasons, airlines do not allow another aircraft to enter or to pushback onto the taxi area around the gates when another aircraft which is starting its engines, due to the size of and limited manoeuvrability within these areas. In summary, pushback operations for one aircraft can delay other aircraft. The reverse can also happen, where an aircraft may not be permitted to start the pushback process until another aircraft has passed. This is the case whenever the area that they would push back to will not be free for the entire duration of the process. Figure 1 shows how delays can happen, illustrating how an aircraft pushing back would prevent another aircraft passing, or an aircraft passing could prevent a pushback.

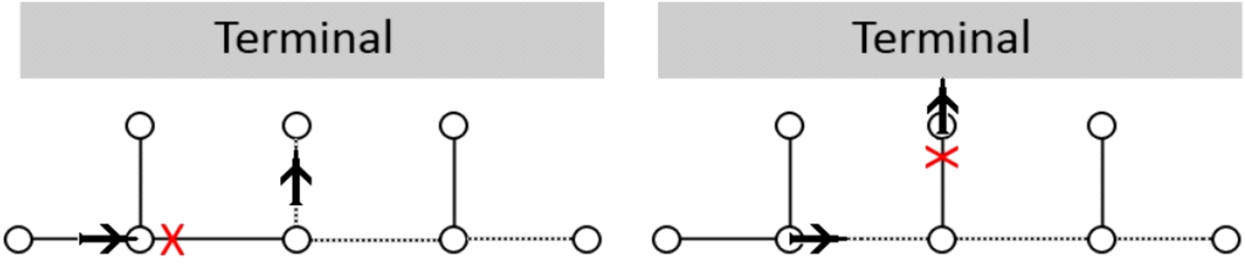

Figure 1. Causes of pushback delays, delaying other aircraft or the aircraft pushing back

Pushback delays can cause significant delays at airports and add uncertainty to the predicted position of an aircraft. The absence of consideration of these factors in a model can lead to further unpredicted delays later on down the path of an aircraft, since the delay in the routing for one aircraft may cause knock-on interactions with later aircraft. A take-off sequencing system would usually require knowledge of how early an aircraft can reach the runway, so any unpredicted delays may affect the feasibility of potential take-off sequences, compromising the feasibility of these sequences. An accurate model for scheduling and routing aircraft ground movement is important for providing any automated decision support to improve runway operations. Reducing waiting time at the runway by even a small percentage can save significant amounts of fuel, which directly influences the cost, as well as the carbon dioxide emissions. Reducing the delays and having improved ground movement can also increase the capacity of the airport.

The aim of this paper is to investigate and evaluate the effects upon the paths and schedules of explicitly taking into consideration the aircraft that are being pushed back. To do this, the pushback operation modelling will put a larger initial delay on the aircraft when they are near to the stands, rather than adding the additional pushback time as extra taxi time, distributed evenly across the taxi path. It is important to quantify any accuracy benefits, such as improved predictions of delays, and this will be performed by considering two different routing methods and evaluating the differences between the delays when the pushback process delays are and are not explicitly considered. A weighted graph with edges and vertices is used to model (parts of) an airport, with aircraft travelling along the edges between the vertices, as illustrated in Figure 1.

Solution approach

Stergianos *et al*. (2015) investigated the ways in which the pushback process affects the accuracy of the routing process by using two variants of a ground movement algorithm and a real world data instance (Method 1, below). This current paper performs a much improved analysis of the effects by considering two different routing and scheduling methods (based upon Kjenstad *et al*. 2013 and Stergianos *et al*. 2015). A modified version has been created of both of these, for use in this evaluation, so that two variants of each can be compared; one that explicitly includes the pushback delays, and one which includes the pushback time as extra taxi time. In addition to executing both algorithms on the original ground movement instance from the paper of Stergianos *et al*. (2015), additional, more complex, data instances are used in the evaluation, which were specifically produced to make the problem harder, simulating and evaluating the effects at busier airports.
The constants/input values and decision variables/outputs which are used in the explanations of the methods are summarised in Table 1.

Table 1. Table of definitions for Method 1 (Algorithms 1 and 2) and Method 2 (Models 1 & 2).

Constants	*Explanation*
F	The set of all flights.
n	The total number of flights, $\|F\|$.
$f \in F := \{1, \ldots, n\}$	A flight.
$s_{f,i}$	The time that flight f starts traveling towards the i^{th} vertex of its path, $i \in \{1 \ldots k_f\}$
p_f	The pushback duration for flight f.
t_f	The time at which flight f should commence its push back – i.e. the starting time for aircraft f in the datasets.
$w_{f,i}$	The weight (necessary taxi time) of the i^{th} edge of flight f's path, which connects the $(i\text{-}1)^{th}$ vertex of the path with the i^{th} vertex.
m_f	The minimum time that it can take for an aircraft f to reach the runway from the gate.
Variables	*Explanation*
C	The set of all identified conflicts where two flights wish to use the same vertex at the same time.
$c := \{f_1, f_2, i_1, i_2\} \in C$	A conflict between two flights f_1 and f_2 at a vertex, where the conflict vertex is the i_1^{th} vertex on the path for flight f_1, and the i_2^{th} vertex on the path for flight f_2.
$f_1(c), f_2(c), i_1(c), i_2(c)$	Functions which will return the relevant element (the element with the same name as the function name) of the conflict c.
k_f	The total number of vertices on the allocated path for flight f.
T_f	The total routing time calculated by the algorithm for flight f not including the pushback time.
T'_f	The total routing time calculated by the algorithm for flight f, including the pushback time.
d_f	Total calculated delay for aircraft f. (A detailed explanation can be found on in the discussion of Equation 7 below)

Method 1

The first method (Method 1) makes use of two algorithms (labelled Algorithm 1 and Algorithm 2) that implement the Quickest Path Problem with Time Windows (QPPTW) algorithm, a routing and scheduling algorithm which was developed by Gawrilow *et al*. (2008) and later modified by Ravizza *et al*. (2014a) in order to be more suitable for airports. We refer the reader to Ravizza *et al*. (2014a) for full algorithm details and discuss only the extensions in this paper. In summary, the algorithm is an extension of Dijkstra's algorithm, which considers multiple aircraft in turn, rather than a single shortest path. When the path for the current aircraft is considered, all of the paths which were found for previous aircraft are taken into account, using time windows to block the graph edges for a specified time during which they are in use.

Algorithm 1: this is a typical implementation of the QPPTW algorithm, as described in Ravizza *et al*. (2014a) and routes a number of aircraft without taking the blocking which can occur during the pushback process into consideration. In order to provide a fair comparison, rather than modelling the pushback delay by the stand, the algorithm instead delays the aircraft from setting off until the pushback duration has expired by delaying the start time of the operation. i.e. the start time for any aircraft f in Algorithm 1 is given by $T_f + p_f$. This ensures that the pushback and engine start-up operations will occur out of the way (at the stands) and will not delay any other aircraft while they occur. Again, to ensure a fair comparison, the calculated total taxi time is given by Equation 1, adding the pushback delay to the final routing time for each aircraft.

$$Total\ taxi\ time = \sum_{f=1}^{n} (T_f + p_f) \tag{1}$$

Algorithm 2: this is an extension of Algorithm 1, and includes the pushback duration at the start of the movement, moving the aircraft into the first vertex (where it would be located while it starts its engines) and then delaying it from commencing its journey until its pushback and engine start-up operation would have completed. For this duration it will be blocking the part of the taxi area into which it will push back, potentially delaying other aircraft. Algorithm 2 will start the routing process for aircraft f (which now includes the pushback process) at time t_f and the final total routing time will be determined by Equation 2, since the pushback delay has already been included in the routing time.

Algorithm 2 requires an adaptation of the QPPTW algorithm. In order to have a more precise routing process, the pushback procedure was added to the QPPTW algorithm. Simply adding the pushback delay into the total taxiing time (as was done for Algorithm 1) cannot guarantee to identify delays which are specifically associated with the pushback process. The weight of an edge is the travel time to traverse that edge. In the extended algorithm (Algorithm 2), all of the edges that the aircraft checks for the first move have been modified to have their weight increased by the pushback duration. This ensures that the pushback time is considered, but is all allocated by the gates, as would actually happen.

$$Total\ taxi\ time = \sum_{f=1}^{n} T'_f \tag{2}$$

With Algorithm 2, the total routing time for an aircraft T_f will not only include the pushback duration for this aircraft p_f but will also include all of the delays that are caused during the pushback process as well. These delays can be caused by delaying an aircraft's own pushback process (not being able to pushback immediately due to traffic) or by delaying its taxi operations, due to being blocked by other aircraft which are pushing back.

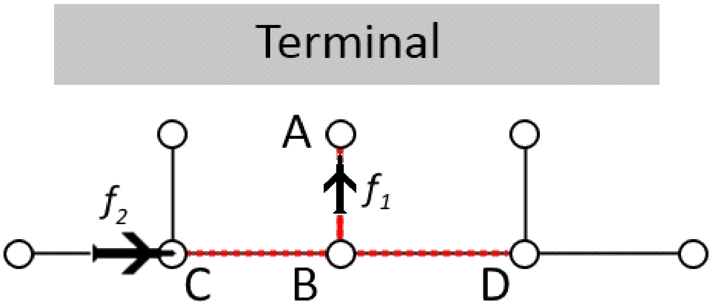

Figure 2. Blocked edges during pushback

Figure 2 shows an aircraft f_1 that is pushing back from vertex A to vertex B. The new weight of the edge AB w'_{AB} which is used by the modified algorithm is can be calculated by $w'_{AB} = w_{AB} + p_f$, where w_{AB} is the normal weight of the edge. The pushback duration p_f is determined by the size of the aircraft *f*.

The QPPTW algorithm finds the shortest path, taking into consideration the added delay. All of the edges that are connected to the first edge are blocked, preventing other aircraft from coming too close to the aircraft which is pushing back. In the example in Figure 2, this means that all of the edges AB, BC, BD are blocked for the entire duration on the pushback process (w'_{AB}). Blocking the edges ensures that the aircraft will reach its destination in the shortest amount of time allowing for the fact that edges can be used by a maximum of one aircraft at a time.

Figure 2 also illustrates the situation where there is another aircraft f_2 that has to wait for aircraft f_1 to finish the pushback process. Aircraft such as f_2 that get blocked have to either wait or choose a longer path if there is one. The QPPTW algorithm which is used will ensure that the path is reallocated appropriately.

In both Algorithms 1 and 2 of Method 1, aircraft are considered in the order in which they can start, which is information calculated from the available data set. This means that the aircraft which is planned to start-up first will usually be able to take the shortest path, the second aircraft will take the quickest path considering any blocking by the first aircraft, and so on, as discussed in Ravizza *et al*. (2014a).

Method 2

Method 2 assumes that all aircraft will be assigned to their shortest paths, regardless of the movement of other aircraft. This method makes use of the vertices that each aircraft traverses, instead of blocking the edges that were used (as the QPPTW algorithm does). The time that flight *f* commences its journey to vertex *i* is denoted $s_{f,i}$. Only one aircraft can use any vertex *v* at any time, so a different aircraft can only use the vertex when the current aircraft uses the next vertex.

The second method consists of 4 stages:

1. Find the shortest path for all of the aircraft, to determine the allocated paths.
2. Find all of the conflicts between aircraft (aircraft that will require the same vertex at the same time).
3. Solve the LP model for all of the known conflicts (the LP model is explained below).
4. Check if there are new conflicts and if so then go to step 3.

In order to find the shortest path for step 1 a simple version of the QPPTW algorithm (similar to Dijkstra's algorithm) is used for this process. This simplified QPPTW algorithm both finds the shortest paths (step 1), and also finds the initial conflicts at the same time (step 2). A linear programming formulation is then used to solve the routing problem and determine the order in which aircraft will pass vertices where there is contention (step 3). Where further conflicts are found, additional constraints are added to the model and it is re-solved until no further conflicts exist (step 4).

For the 2nd step it is important to find all of the conflicts that happen when the aircraft use the shortest path that was found in the previous step. In order to identify a conflict the movements of every aircraft are stored in each vertex every time that it is used. So if any vertex is used by more than one aircraft at the same time, a conflict is added to the list of conflicts. Any conflict is between only 2 aircraft, although each aircraft can have multiple conflicts with other aircraft, and 2 aircraft can conflict with each other multiple times along their path. For the 3rd step, a linear optimisation model is solved. The constraints and objective function for this model are shown below:

Constraints

$$s_{f,1} \geq t_f \qquad \forall\, f \in F \tag{3}$$

$$s_{f,2} \geq s_{f,1} + w_{f,1} + p_f \qquad \forall\, f \in F \tag{4}$$

$$s_{f,i+1} \geq s_{f,i} + w_{f,i} \qquad \forall\, i \coloneqq \{2 \ldots k_f - 1\}, \quad \forall\, f \in F \tag{5}$$

Constraint 3 ensures that all aircraft start after a set start time for the aircraft. This time is allocated to each aircraft and is forbidding the program to make them start earlier. Constraints 4 and 5 ensure that an aircraft cannot enter the next vertex on its path any earlier than the time at which it enters the current vertex, plus the time to traverse (i.e. the weight of) the edge between the two vertices. Constraint 4 ensures that the aircraft spends extra time on its first vertex to simulate the pushback operation (which will also delay the time at which any other aircraft can enter that vertex). Note that p_f will be 0 for the versions that do not include the pushback process.

If conflicts are found, additional constraints are added to resolve the conflicts, ensuring that one of the aircraft cannot use the vertex until the other has reached the following vertex. One of the disjunctive constraints (6a) or (6b) will be added for each conflict.

$$s_{f_2(c),i_2(c)} \geq s_{f_1(c),i_1(c)+1} \qquad \forall\, c \in C \tag{6a}$$

$$s_{f_1(c),i_1(c)} \geq s_{f_2(c),i_2(c)+1} \qquad \forall\, c \in C \tag{6b}$$

For efficiency reasons and to reduce the number of iterations, these are actually applied to the next *l* vertices, where *l* is the number of vertices which the two aircraft share after the vertex where they first conflict. i.e. if they enter the shared path in a specific order, they must traverse all shared vertices in that order:

$$s_{f_2(c),i_2(c)+j} \geq s_{f_1(c),i_1(c)+j+1} \qquad \forall\, j \in \{0 \ldots l-1\},\ \ \forall\, c \in C \tag{6'a}$$

$$s_{f_1(c),i_1(c)+j} \geq s_{f_2(c),i_2(c)+j+1} \qquad \forall\, j \in \{0 \ldots l-1\},\ \ \forall\, c \in C \tag{6'b}$$

Objective function

The objective function measures the times at which the aircraft reach the final vertices in their journeys, which is equivalent to the objective function for the first (QPPTW-based) method, allowing a comparison to be made between the two methods.

$$minimize \sum_{f=1}^{n} s_{f,k_f}$$

Two variants of the model are evaluated, and each can be evaluated with and without pushback delays. In the first variant (Model 1), the prioritisation constraints (Inequalities 6a or 6b) are applied in order to prioritise the aircraft that would reach the vertex first, even though the aircraft may use the vertices for different durations (e.g. aircraft which are pushing back will use the vertex for longer than aircraft which are going past). In the second variant (Model 2), this prioritisation is reversed, so priority will be given to the aircraft which starts moving towards the vertex second. Since the pushback operation is time consuming, this will usually be the one which is already taxiing rather than the one which is pushing back. This models what happens at real airports more often, since it is often better to avoiding asking an aircraft which is already moving to stop. This latter approach also turns out to be more similar to the usual case for the QPPTW approach (which prioritises the aircraft which started its move first), since when an aircraft which is pushing back comes into contention with one which is already moving, the one which was already moving will almost always have commenced its pushback earlier (it has had to complete its pushback and taxi to the vertex where the problem occurs before it comes into contention with the aircraft which is pushing back).

As for the QPPTW algorithm, the model was developed in order to run both with and without the explicit pushback process delays. As for the QPPTW algorithm, the aircraft start times are delayed by the pushback duration for the version that does not take the pushback process into consideration.

In order to be able to compare all of the algorithms and models, an effective way to calculate the delays was needed. In order to make sure that all delays are found, even the ones that are caused by taking longer paths when path allocation was involved, the minimum routing times were calculated for all gates. Dijkstra's algorithm is sufficient for this, so a simpler version QPPTW algorithm (without the time windows) was executed for each of the gates (twice, once for arrivals and once for departures), on an empty airport without the enhancements which block edges and readjust the time windows. Once the minimum times had been found (the quickest path, without any delay) it was easy to establish the exact additional delay that each aircraft had, regardless of whether this delay was due to waiting for other aircraft to move, or increased taxi time due to taking a longer path, re-routing around any blocks (e.g. pushback blocks), on the optimal path. Given the preceding calculations and definitions, the delay for each aircraft can be calculated using Equation 7, where m_f is the time which would be taken on the shortest path for aircraft *f*. The total delay is then the sum of all delays for individual aircraft.

$$d_f = T_f - m_f \tag{7}$$

Results

Both routing methods were executed using different instances for Stockholm's Arlanda airport, the largest airport in Sweden (http://www.asap.cs.nott.ac.uk/external/atr/benchmarks/index.shtml, accessed 18 November 2015). Instance 1 is based on historical data from the Swedish Air Navigation Service Provider (ANSP) and was the basis for the rest of the instances. It includes 54 aircraft set to depart within 2 hours. Figure 3 is a simple map of the paths and gates that an aircraft can use on Arlanda airport. The gates are highlighted with large dots and the main runway for departures (01L) is highlighted with arrows. Runway 01L is usually used for departures and 01R (right side of the picture) is used for arrivals (or the inverse). Runway 26 (top side of the picture) is newer and used in off-peak conditions so mainly the departure runway is shown in the figure.

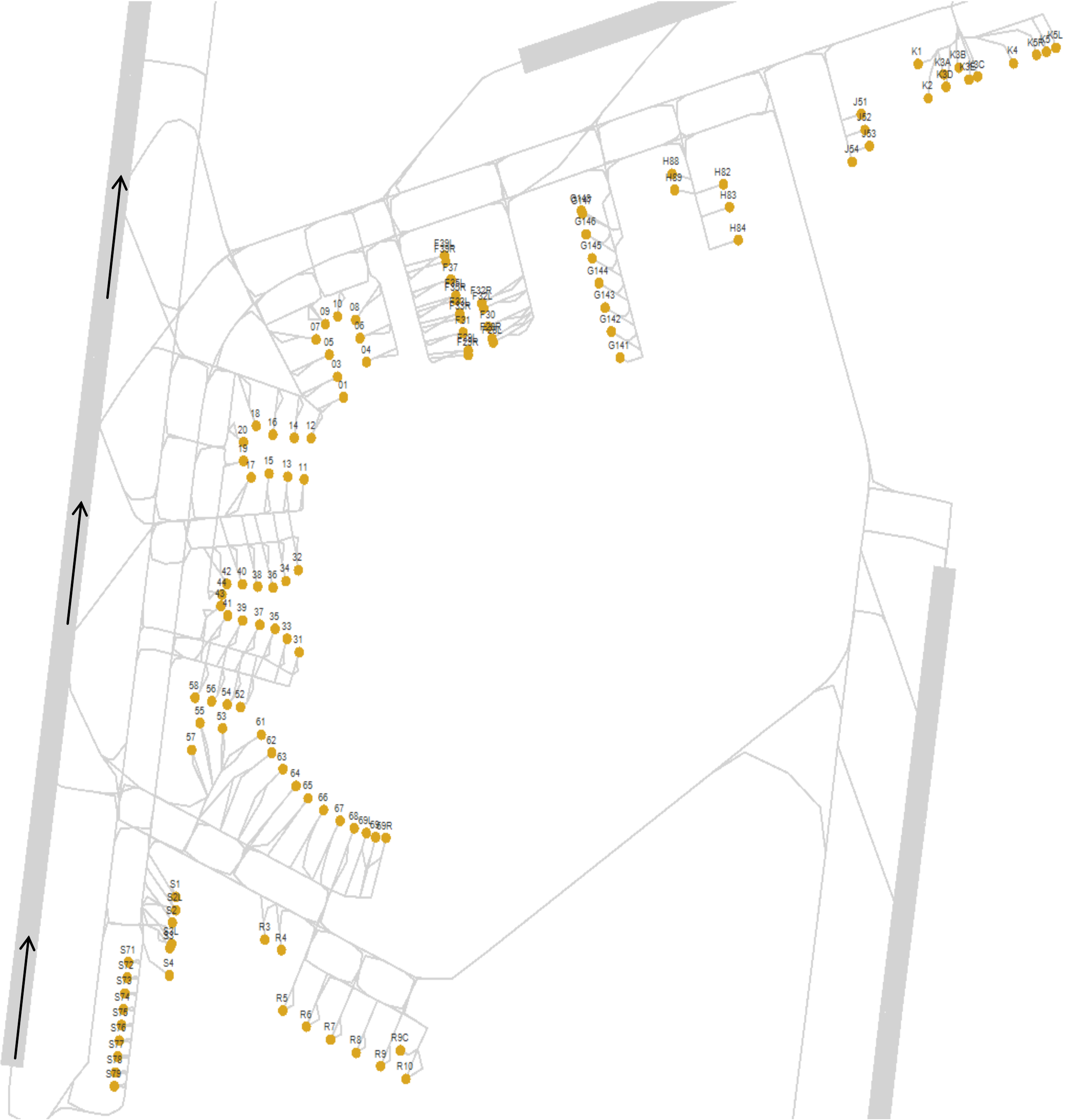

Figure 3. Layout of Arlanda airport

The remaining instances were developed to simulate the effects of heavier airport loads, by altering the data from the Swedish ANSP (by adding more aircraft and assigning them to different gates), creating problems of increasing difficulty. These resulting instances therefore have different characteristics for sparsity of movements and complexity of solution. Although the algorithm can handle different sizes of aircraft, in these instances all aircraft were considered to be medium weight class.

Table 2. Specifications of instances.

	No. of Aircraft	*Time span*
Instance 1	54	2h
Instance 2	70	3h 40m
Instance 3	98	4h 50m
Instance 4	118	4h 50m
Instance 5	140	4h 50m

Instances 3 to 5 were made to investigate what happens when there is increased traffic (or even exceptionally high traffic) number on an airport with a poor gate allocation.

A weighted graph for Arlanda airport was used for these experiments. Since the QPPTW algorithm that was used for the core of the routing process works by blocking edges, the maximum distance between two vertices was restricted by inserting vertices into long arcs (therefore splitting the edges and allowing more aircraft to use them), at a spacing of approximately 80 meters, simulating the effects of being able to have multiple aircraft queue one behind another along the taxiway.

The resulting total delays are shown in Tables 3 and 4. The framework was programed in Java and was executed on a personal computer (Intel Core i3, 2.5GHz, 4GB RAM). For Method 2, the framework was programed in Java and all of the LP models were solved using CPLEX (with the use of CPLEX Java libraries for Eclipse). The execution time for both QPPTW algorithms is less than 1.5 second which is fast enough for real time routing. The average time for Algorithm 2 - 5th instance (which is the most computationally demanding instance) was 1452ms. For the linear optimisation models the execution time was usually less than 2 seconds. For instance 5, the problem was solved in 1840ms on average for Model 1and in 2066ms on average for Model 2.

Table 3 shows the total delay and total taxi times for instance numbers 1 and 2. The first two rows show the times that are produced after running the QPPTW algorithms 1 and 2 respectively. Row three and four show the times for the LP Model 1 (with and without taking into consideration the pushback process). Lines five and six show the times for the LP Model 2 (with and without taking into consideration the pushback process). In each case it is apparent that significant additional delays result from the consideration of the pushback delays (Algorithm 2 and 'push' variants of the models). In fact these delays are huge in comparison with the delays without the pushback modelling, which are small, showing that there is relatively little interaction between the aircraft when pushback delays are not considered. This shows the importance for accurate models of including these delays.

Table 3. Total delays and total taxi time for each algorithm/model for instances 1 and 2.

		Instance 1 (2h)		*Instance 2 (3h 40m)*	
	Total:	*Delay [s]*	*Taxi time [s]*	*Delay [s]*	*Taxi time [s]*
Method 1	QPPTW Algorithm 1	89	26606	1	35579
	QPPTW Algorithm 2	1313	28010	1778	37356
Method 2	LP Model 1 no push	45	26562	1	35579
	LP Model 1 push	1022	27719	1234	36812
	LP Model 2 no push	53	26570	19	35597
	LP Model 2 push	1332	28029	1802	37380

Table 4. Total delays and total taxi time for each model/algorithm for running instances 3, 4 and 5.

		Instance 3 (4h 50m)		*Instance 4 (4h 50m)*		*Instance 5 (4h 50m)*	
	Total:	*Delay [s]*	*Taxi time [s]*	*Delay [s]*	*Taxi time [s]*	*Delay [s]*	*Taxi time [s]*
Method 1	QPPTW Alg. 1	50	51669	109	62579	153	73711
	QPPTW Alg. 2	807	52426	1504	63974	2466	76024
Method 2	LP Model 1 no push	20	51639	50	62520	50	73608
	LP Model 1 push	593	52212	1075	63545	2618	76176
	LP Model 2 no push	20	51639	64	62534	64	73622
	LP Model 2 push	780	52399	1373	63843	2297	75855

Table 4 shows the total delay and total taxi times for instance number 3, 4 and 5. For these instances the data is similar, but with gradually increasing traffic. It is apparent from the results that, as the traffic increases, the interactions between aircraft, and hence delays, increase even without the pushback delay modelling, although these delays are still relatively small. These interactions are increasing in a super-linear manner in relation to the increase in the number of aircraft, as would be expected. With the explicit pushback delay modelling included, the consequent delays are much higher, as was observed for instances 1 and 2. It is also obvious that the rate of increase of the delays is rapid as the number of aircraft is increased. This will, therefore, be an even larger problem at busier airports than at quieter airports, with an increasing importance for explicitly considering the pushback delays.

Table 5 shows the details of the flights in instance 1 that are affected by the delays. Flights which are unaffected have been omitted. It is apparent that the delays are actually affecting a small number of flights to a fairly large extent, rather than being evenly spread across many aircraft. This sort of characteristic will make it even more important to understand these delays, since they can affect the predicted taxi times considerably for these aircraft. This will make it increasingly inappropriate to use predictions which do not consider pushback operations within any integrated system. For example, a 3 or more minute discrepancy in predicted arrival time at the runway is likely to make a predicted runway sequence unachievable. Similar results were observed for the remaining instances, 2 to 5. The results for all the instances are summarised in Table 6.

Both of the potential causes for delays were observed to occur in the experiments; aircraft pushing back and blocking the taxi area for other aircraft (i.e. the aircraft pushing back is doing the blocking) and aircraft being prevented from pushing back due to another aircraft passing at the time.

In most cases where aircraft delay each other with the QPPTW algorithms and the LP Model 2, the delay was experienced by the aircraft that was set to pushback later, as expected. This aircraft will often not be able to start the pushback process at all since the edges in front of the stand would need to be clear for the whole duration of the pushback process. However with the LP Model 1 these kinds of delays were avoided as the aircraft that was set to push back had the priority most of the time (its operation takes longer so it was more likely to start earlier when the two were in contention) and the second type of delay was observed more often.

Comparing the approaches, it can be observed from the results that, even though the pushback process can increase the amount of delay, the LP Model 1 seems to performing better than the LP Model 2 and the QPPTW algorithm for the first four instances. LP Model 1 routes all of the aircraft in advance and still takes the pushback process into consideration, however the main reason that there are fewer delays is that it will allow an aircraft to push back immediately even if it has to delay a taxiing aircraft to do so. This may not be practical at real airports, however. In LP Model 1 if aircraft have the same departure time they can also push aircraft back in parallel, resulting in aircraft not interacting with each other. However, in instance number 5 where there are aircraft departing every 2-3 minutes and it is harder for an aircraft not to interact with another (as it is when pushing back at the same time in the same taxi area for Model 1) QPPTW algorithm and LP Model 2 perform better than LP Model 1.

The QPPTW algorithm has the advantage of being able to re-route aircraft when necessary, whereas the LP models always apply the shortest paths. In this case, however, this advantage seems to be no help. This implies that the shortest path approach works well for Arlanda. Investigating the extent to which this is, or is not the case for other airport layouts, where there are more options for paths with similar lengths, will be an interesting area for future research.

The majority of the additional delay was experienced within the area around the gates and was caused directly by the pushback process. However, in some cases the delays close to the gates also caused later delays, with aircraft being delayed enough to interact with other aircraft later on. For example, in instance 1 with the QPPTW algorithm, aircraft 9 delays aircraft

10 and then aircraft 12 and 10 interact. This can also affect the order in which aircraft arrive at the runway in some cases. For example, with Algorithm 1 aircraft arrive at the runway in the order 10 – 11 – 9 – 12, whereas with Algorithm 2 it is 11 – 9 – 10 – 12. When considering the integration of systems, this can affect the feasibility of potential take-off sequences with appropriate re-sequencing no longer being possible (see Atkin *et al*. 2007), and hence is also important to understand.

Table 5. Flights which are affected by ground movement delays (Instance 1).

			QPPTW Delays (Method 1)			*LP 1 Delays (Method 2)*			*LP 2 Delays (Method 2)*		
Flight no	*Start time*	*Stand*	*Delays Alg1*	*Delays Alg2*	*Diff.*	*No push*	*Push*	*Diff.*	*No push*	*Push*	*Diff.*
1	05:04:12	F37	0	0	-	0	93	93	0	0	-
2	05:04:46	F39R	0	172	172	0	0	-	0	173	173
4	05:13:53	S76	0	0	-	0	87	87	0	0	-
5	05:15:00	S78	0	166	166	0	0	-	0	167	167
7	05:24:10	53	0	0	-	0	92	92	0	0	-
8	05:24:55	57	0	173	173	0	0	-	0	173	173
9	05:29:20	G145	0	0	-	0	0	-	13	273	260
10	05:30:00	G142	0	186	186	0	186	186	0	0	-
12	05:34:45	F37	12	36	24	13	13	-	0	0	-
17	05:45:00	11	23	23	-	0	0	-	0	0	-
21	05:50:20	34	3	3	-	3	3	-	3	3	-
23	05:55:00	F33R	0	0	-	0	0	-	10	10	-
25	05:58:57	41	10	10	-	10	10	-	0	0	-
27	06:04:40	17	0	0	-	6	6	-	6	6	-
28	06:04:53	40	20	20	-	0	0	-	0	0	-
30	06:05:52	36	0	0	-	0	254	254	0	0	-
31	06:09:52	42	0	9	9	0	0	-	0	10	10
36	06:15:20	38	7	7	-	7	7	-	7	7	-
37	06:18:03	32	0	0	-	0	181	181	0	0	-
39	06:22:02	34	0	82	82	0	0	-	0	83	83
40	06:26:14	35	0	0	-	0	84	84	0	0	-
41	06:27:35	39	0	174	174	0	0	-	0	175	175
43	06:30:00	31	0	0	-	6	6	-	0	0	-
44	06:30:00	33	14	252	238	0	0	-	14	252	238
Total			89	1313	1224	45	1022	977	53	1332	1279

Table 6. Results of including the pushback process.

	QPPTW (Method 1)		*LP Model 1(Method 2)*		*LP Model 2(Method 2)*	
	Additional no. of delays	*Additional delay [s]*	*Additional no. of delays*	*Additional delay [s]*	*Additional no. of delays*	*Additional delay [s]*
Instance 1	9	1224	7	977	8	1279
Instance 2	12	1777	11	1233	12	1783
Instance 3	5	757	6	573	5	760
Instance 4	13	1395	8	1025	8	1309
Instance 5	22	2313	16	2568	16	2233

Conclusion

This paper has investigated the importance of the pushback process in the routing and scheduling problem of the ground movement of aircraft. Two different routing methods were considered, with various configurations to examine the effect of the pushback process. All of the methods (QPPTW, LP Model 1 and 2) had versions which did and did not take the pushback process into consideration.

In the first method the ground movement problem is solved using the QPPTW algorithm, which finds the quickest path that an aircraft can take in order to go from point A to point B, taking into consideration the movement of previously routed aircraft. In the second method, routes could be pre-determined for aircraft and the task was only to determine the order in which movement happened. Two models were investigated, which had different prioritisations.

In all cases, the pushback process had a considerable effect upon the resulting delays. Failure to consider the pushback process meant that the taxi times for some aircraft could be greatly underestimated, substantially reducing the potential benefits from using a ground movement system. It was observed that, although most of the delays occurred around the stands, where the pushback process happens, in some cases the delays had further effects later on, causing other aircraft to interact. In some cases the interactions even changed the order in which aircraft reached the runways, which could affect the potential benefits from the interaction with a take-off sequencing system (such a system would benefit from increased predictions of times and sequences for aircraft arriving at the runway, see Atkin *et al*. 2008). The pushback operations had an effect in all of the investigated experiments. Interestingly, the ability to re-route aircraft does not seem to help for Arlanda, where it seems to be a sensible option to use the shortest path.

Future research will investigate whether this changes for other airport layout. In addition, the gate allocation obviously has an effect upon the ground movement (see Neuman and Atkin, 2013), since it determines where aircraft start their taxi operations from, and this is another interesting area for future research which we will continue to investigate. In order to maximise the benefits of airport automating in future, it will be increasingly important to consider the integrated problems, and having improved models for what actually happens in the ground movement operations will be important for this.

Acknowledgments—The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no 608322. We wish to thank Åke Wall and Jan-Olof Roos at the Swedish Air Navigation Service Provider (Luftfartsverket), who provided the data, the airport details used in this research, and for their operational expertise. Also, SESAR 10.9.2 "Multiple airport arrival/departure management" for funding the development of the approach to ground movement which is discussed here as Method 2.

References

Apice, C., Nicola, C., Manzo, R., Mocci, V. (2014). Optimal scheduling for aircraft departures. Journal of Ambient Intelligence and Humanized Computing. Vol 5, Issue 6, pp. 799-807

Atkin, J.A.D., Burke, E.K., Greenwood, J.S., Reeson, D. (2007). Hybrid Metaheuristics to Aid Runway Scheduling at London Heathrow Airport. Transportation Science, Vol. 41, Issue 1, pp. 90-106

Atkin, J.A.D., Burke, E.K., Greenwood, J.S., Reeson D. (2008). On-line Decision Support for Take-off Runway Scheduling with Uncertain Taxi Times at London Heathrow Airport. Journal of Scheduling, Vol. 11, Num. 5, pp. 323-346

Atkin, J.A.D., Burke, E.K., Ravizza, S. (2010). 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

Atkin, J.A.D., De Maere, G., Burke, E.K., Greenwood, J.S. (2013). Addressing the Pushback Time Allocation Problem at Heathrow Airport. Transportation Science, Vol. 47, Issue 4, pp. 584-602

Bennell, J.A, Mesgarpour, M., Potts, C.N. (2011). Airport runway scheduling. 4OR. Vol 9, Issue 2 pp.115–138

Bouras, A., Ghaleb, M. A., Suryahatmaja, U.S., Salem, A.M. (2014). The Airport Gate Assignment Problem: A Survey. The Scientific World Journal, vol. 2014, Article ID 923859

Burgain, P., Pinon, O.J., Feron, E., Clarke, J.P., Mavris, D.N. (2012) Optimizing Pushback Decisions to Valuate Airport Surface Surveillance Information. Intelligent Transportation Systems, IEEE Transactions, vol.13, no.1, pp.180-192

Cheng, Y. (1998). Solving Push-Out Conflicts in Apron Taxiways of Airports by a Network-Based Simulation. Computers & Industrial Engineering. Vol. 34, No. 2, pp. 351–369Dorndorf, U., Drexl, A., Nikulin, Y., Pesch, E. (2007) Flight gate scheduling: State-of-the art and recent developments. Omega, 35:326 – 334

Gawrilow, E., Kohler, E., Mohring, R., Stenzel, B. (2008). Dynamic routing of automated guided vehicles in real-time. Mathematics - Key Technology for the Future, pp. 165–178

Kjenstad, D., Mannino, C., Schittekat, P., Smedsrud, M. (2013). Integrated surface and departure management at airports by optimization. 5th International Conference on Modelling, Simulation and Applied Optimization (ICMSAO), pp 1–5

Neuman, U.M., Atkin, J.A.D. (2013). Airport Gate Assignment Considering Ground Movement. 4th International Conference on Computational Logistics, ICCL. Volume 8197, pp. 184–198

Ravizza, S., Atkin, J.A.D., Maathuis, M.H., Burke, E.K. (2013). A combined statistical approach and ground movement model for improving taxi time estimations at airports. Journal of the Operational Research Society, 64: 1347-1360

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

Ravizza, S., Chen, J., Atkin, J.A.D., Stewart, P., Burke, E.K., (2014b) Aircraft taxi time prediction: comparisons and insights. Applied Soft Computing, Vol. 14, Part C, pp. 397-406

Rolling, P.C., Visser, H.G. (2008) Optimal Airport Surface Traffic Planner Using Mixed Integer Linear Programming. International Journal of Aerospace Engineering. Volume 2008, Article ID 732828

Stergianos, C., Atkin, J.A.D., Schittekat, P., Nordlander, T.E., Gerada, C., Morvan, H. (2015). Pushback delays on the routing and scheduling problem of aircraft. 7th International Conference on Applied Operational Research, ICAOR. Lecture Notes in Management Science 7: 34-40

Tu, Y., Ball, M.O., Jank, W.S. (2008). Estimating Flight Departure Delay Distributions – A Statistical Approach with Long-Term Trend and Short-Term Pattern. Journal of the American Statistical Association, Volume 103, Issue 481, pp. 112–125

University of Nottingham, Computer Science (2015) Air Transportation Research - Benchmarks, Retrieved from http://www.asap.cs.nott.ac.uk/external/atr/benchmarks/index.shtml [Accessed 18/11/2015].