Journal of Scheduling manuscript No.
(The final publication is available at link.springer.com)

An Analysis of Constructive Algorithms for the Airport Baggage Sorting Station Assignment Problem

Amadeo Ascó · Jason A. D. Atkin · Edmund K. Burke

Received: date / Accepted: date

Abstract The appropriate assignment of airport resources can greatly affect the quality of service which airlines and airports provide to their customers. Good assignments can help airlines and airports keep to published schedules by minimising changes or delays while waiting for resources to become available.

In this paper, we consider a resource allocation problem, namely the problem of assigning available baggage sorting stations to flights which have already been scheduled and allocated to stands. A description and model for the problem are presented, illustrating the different objectives which have to be considered. A number of constructive algorithms for sorting station assignments are then presented and their effects are compared and contrasted when different numbers of sorting stations are available. In particular, it can be observed that the appropriate algorithm selection is highly dependent upon whether or not reductions in service time are permitted and upon the flight density in relation to the number of sorting stations.

Finally, since these constructive approaches produce different solutions which are better for different trade-offs of the objectives, we utilise these as initial solutions for an evolutionary algorithm as well as for an Integer Linear Programming (ILP) model in CPLEX. We show that in both cases they are helpful for improving the results which are obtainable within reasonable solution times.

Keywords Airport Baggage Sorting Stations · Scheduling · Heuristics · Constructive Algorithms · Greedy Algorithms

1 Introduction

The mishandling of baggage in airports has been one of the more important passenger issues for several years in both Europe and the U.S.A. It was ranked third in complaints after cancellations and delays in the 2009 report of the Air Transport Users Council (2009), and its importance was further emphasized in the April 2010 report of the Office of Aviation Enforcement and Proceedings (U.S. Department of Transportation (2010)) where over a hundred thousand baggage reports were logged, ranking baggage complaints in second place. Expected increases in civil air traffic (ICAO (2010) and Federal Aviation Administration (2010)) will continue to increase the complexity of these problems. Systems to improve this situation are therefore extremely valuable.

A stand is an area on the ground where aircraft are parked. Gates are the exits from the terminal through which passengers pass to reach the aircraft. Stands may be located at gates, in which case the terms stands and gates are often used synonymously, or may be remote stands, located elsewhere on the airport surface, for which passengers will usually catch a bus from the terminal to the aircraft. In order to increase the number of gates at a terminal, piers often protrude, with gates

Amadeo Ascó
School of Computer Science
University of Nottingham
E-mail: a.asco@cs.nott.ac.uk

Dr. Jason A. D. Atkin
School of Computer Science
University of Nottingham
E-mail: jason.atkin@nottingham.ac.uk

Prof. Edmund K. Burke
Department of Computing and Mathematics
University of Stirling
E-mail: e.k.burke@stir.ac.uk

along one or both sides, as illustrated in Figure 2. Baggage sorting stations are often located at the bases of the piers, rather than along the sides, as also illustrated in Figure 2.

Baggage which is checked-in will travel through the baggage system to baggage sorting stations where it is temporarily stored. Baggage handlers at the sorting stations will sort the baggage and load it onto baggage carts or into special containers which go directly into the aircraft. These sorting stations are not usually co-located with the aircraft, so the pairing of sorting stations to gates is important. Furthermore, baggage often arrives or accumulates at the sorting station prior to the aircraft arriving at the gate, so the sorting station is needed earlier than the gate, and usually for a longer period.

Better modelling and solution of resource allocation operations at the stands, including baggage sorting station allocation, is also extremely important in improving the overall departure system. As stated in Atkin et al (2008), it is often impractical to accurately predict pushback times in advance due to the uncertainty involved in the turnaround process. Collaborative decision making (see European Airport CDM) is key for improving this situation and gaining the huge potential benefits. Such decision making requires more accurate predictions for process finish times. Reducing the probability of baggage being in the wrong place or mixed with baggage for other flights, and reducing the distance to get baggage from the sorting station to the aircraft, will all aid this. Reducing this uncertainty can allow runway sequencing to be performed at the stands, allowing delays to be decreased significantly Atkin et al (2012).

A model is presented for the baggage sorting station assignment problem at airports, along with a consideration of the various objectives. The problem can be observed to be a multi-objective resource constrained assignment problem, where the aim is to assign the limited baggage handling resources amongst the various flights which have to be serviced. Research into a similar problem was performed in Abdelghany et al (2006) which uses a constructive algorithm, but various questions were left unanswered. This paper aims to answer these questions and to perform a rigorous analysis of the effects and benefits of various constructive algorithms for the problem, with a view to utilising these when providing initial solutions for further search methods. The intention is not to determine the 'perfect' algorithm for constructing a sorting station assignment, but instead to understand the effects and trade-offs of different choices.

An example of flight assignments is shown in Figure 1, which can be considered to be a type of Gantt chart, where the vertical axis represents the stands and the horizontal axis shows the time of day. Each rectangle on the diagram represents a specific flight and shows the times at which the flight will use the stand. Each stand is numbered: the first digit is the terminal number, the second digit is the pier number, and the last two digits are the individual stand identification. For example the top row shows five flights assigned to stand 1101, which refers to terminal 1, pier 1 and stand 1.

The root of the problem for baggage sorting station assignment is that baggage sorting stations are required for a longer period than gates, so there can be no one-to-one correspondence between baggage sorting stations and stands, and ideal locations cannot be guaranteed. Indeed, there should also ideally be a buffer time between sorting station usages, to reduce the risk of small perturbations affecting the assignments and of the mixing of baggage between flights, but the contention for baggage sorting stations means that this sometimes has to be reduced or eliminated. One of the purposes of this paper is to better understand the way in which the potential reductions in buffer times affect the various algorithms.

This is not an easy problem to solve, as will be apparent from Section 4. There are a number of objectives to consider in the baggage sorting station assignment problem (for example, maximising the number of assignments, maximising available buffer times and assigning flights to the closest sorting stations) and these are in obvious conflict with each other. Any solution method needs to take this into account. Some of these objectives are easier than others to handle independently, and the robustness objective can be particularly hard to optimise, but the interaction of the objectives results in a much more complex overall problem. In particular, different constructive algorithms will be observed in this paper to perform better for different objectives. Hybridisation of the algorithms themselves or the appropriate utilisation or recombination of solutions from different algorithms may potentially lead to assignments which better reflect the overall objectives.

This paper is structured as follows: Firstly, the problem description and model are presented, followed by a description of the algorithms considered. The results of applying these algorithms to the problem are then provided and various observations are made and explanations given. Results are then provided to illustrate the benefits of using the constructed solutions as initial values for other solution methods. Finally some conclusions are presented in Section 5.

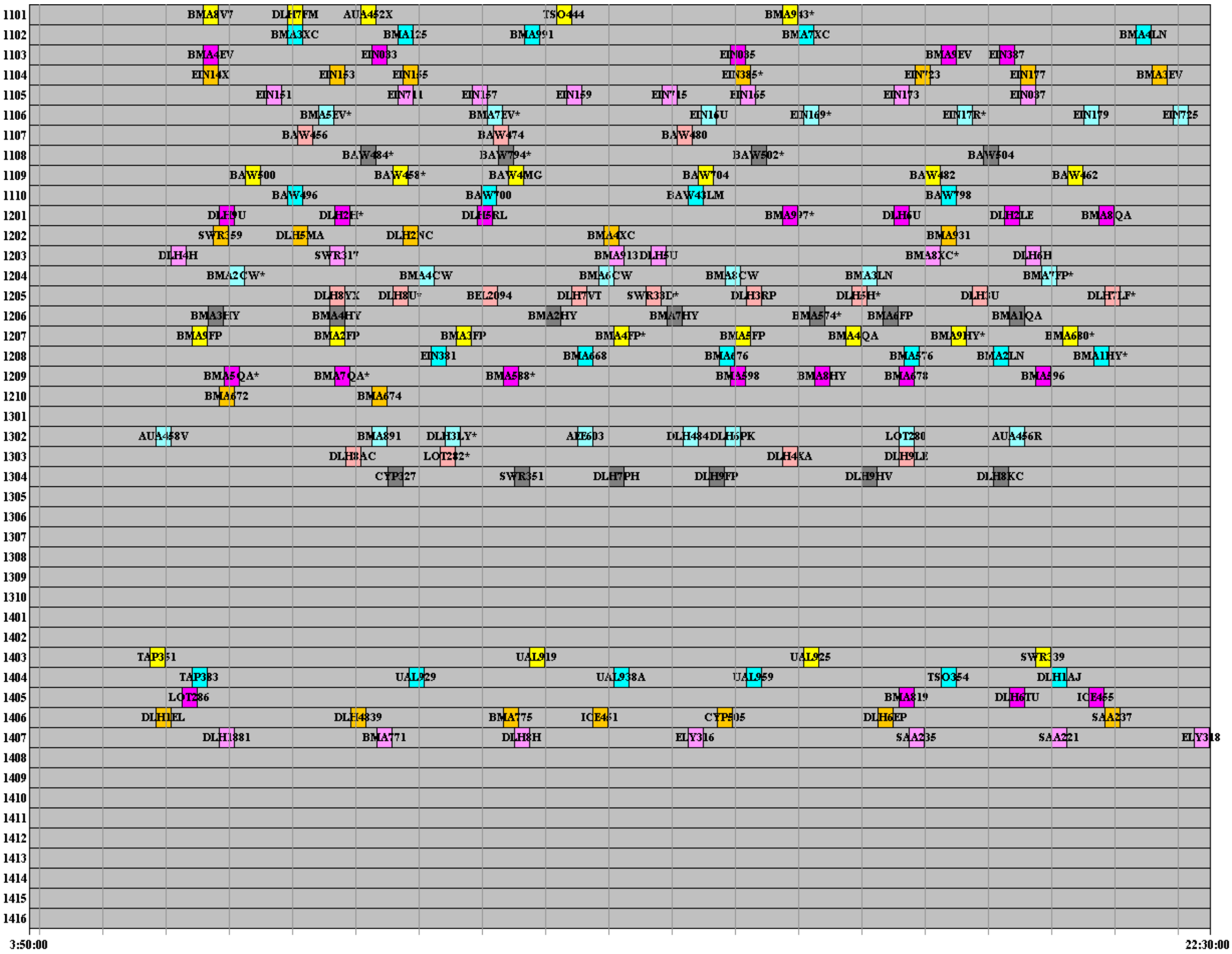

Fig. 1 Allocation of 46 stands to 163 flights from 1st March 2010.

2 Problem Description and Model

The problem under consideration in this paper may be summarised as the assignment of available baggage sorting stations to flights which have already been scheduled. In the baggage sorting station assignment problem, the flights will already have been assigned to stands, which are often grouped along piers around the terminals, and there will usually be some bias in this allocation, according to airline preferences.

2.1 Airport Layout

Airport geometry plays an important role in the assignment of resources and the safety of the airport operations. An overview of the airport configurations and technologies for the transportation of passengers and baggage was presented by Pitt et al (2002), who concentrated on airport configurations and the availability of different types of resources. Rijsenbrij and Ottjes (2007) provided an overview of different elements of the baggage handling system and gave a description of the way in which baggage is currently handled, identifying potential areas of improvement.

Figures 2 and 3 provide stylised diagrams of two example layouts. The stands are grouped on piers, which have their baggage sorting stations at their bases, placed perpendicularly to the pier. For any stand it will be better to assign the luggage to the sorting stations on the same side of the same pier. Alternatively, more distant sorting stations could be used, but these are less preferable. A ‘cost’ or ‘distance’ (d_{ij} in this model) can be associated with a stand-sorting station pairing and one aim is to reduce this cost by assigning as many flights as possible to their preferred sorting stations. Provided that the distances are specified, the model which is presented in this paper can be used to represent many different topologies. For example, Figure

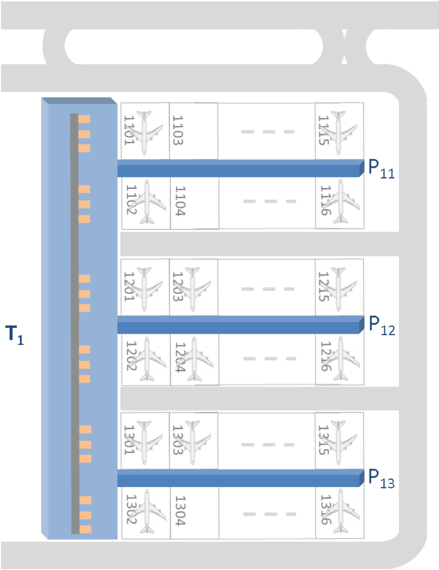

Fig. 2 Simple one terminal topology with 3 piers and 48 stands.

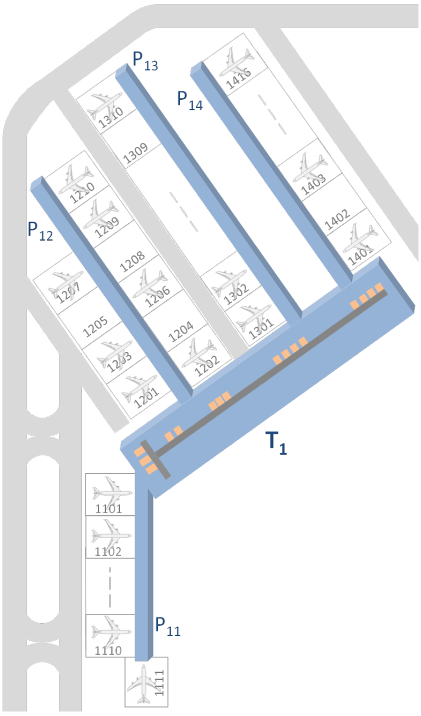

Fig. 3 Simple London Heathrow T1 topology with 4 piers and 46 stands.

2 shows the generic topology which was used in Ascó et al (2011) whereas a closer representation of London Heathrow Terminal 1 is shown in Figure 3. Rather than being specific to the example layouts provided here, the model which is utilised in this paper is appropriate for any airport where there are groupings of aircraft/gates which enforce a sorting station group preference (such as when aircraft are on piers) and where there is a distance or cost metric for the assignment of a sorting station to a flight. For example, at some airports the sorting stations may be at the ends of the piers, as in Figures 2 and 3. In others, the sorting stations may be between the gates, in which case the preference for the distance/cost of assigning flight-sorting station pairs may be much stronger, whereas the group/pier preference may not be so strong. The model given here is equally valid in either case.

2.2 Input Data and Constants

Table 1 summarises the various constants which are used in the model described in this paper.

Table 1 List of the constants and input values for the model.

Name	Description
N	The total number of baggage sorting stations under consideration.
M	The total number of flights to which sorting stations should be assigned.
T_j	The base service time for flight j.
B_j	The desired buffer time for flight j.
R_j	The maximum reduction of service time allowed for flight j (we assume $R_j = B_j$ for this paper, so that the buffer time can be reduced but the base service time cannot).
e_j	The end service time for flight j.
t_j	The target starting service time for flight j, $t_j = e_j - T_j - B_j$, assuming the full buffer time is available.
C_j	A flight specific constant representing the amount of baggage to be processed for flight j. This determines the difficulty involved in assigning the flight to a sorting station which is further away. For example, this may represent the number of delivery trips required to move the baggage from the sorting station to the aircraft. In the absence of baggage load figures, we set $C_j = 1$ for all aircraft for the results in this paper.
d_{ij}	The distance between baggage sorting station i and flight j.

2.3 Service Period

A service period is associated with each departing flight, during which the baggage for the flight is accumulated at the assigned baggage sorting station and finally loaded onto baggage carts for transfer to the flight. This service period may (optionally) be extended by applying a buffer time, since it is preferable to have a gap between the servicing of consecutive flights by the same sorting station.

2.4 Buffer Time

A buffer time is applied between two consecutive flights on the same baggage sorting station in order to absorb small disturbances in the real system behaviour. Buffer times are a common means of increasing robustness to avoid delays, as studied by Nikulin (2006) and Mulvey et al (1995). Buffer times were used in the scheduling of baggage sorting stations by Abdelghany et al (2006), and Wu and Caves (2004) used them in the optimization of the aircraft turnaround process. The Airport Gate Assignment Problem (AGAP) has some similar characteristics to the baggage sorting station assignment problem and buffer times were also considered for the AGAP by Hassounah and Steuart (1993), Yan and Chang (1997), Bolat (2000), Yan et al (2002) and Wu and Caves (2004). Yan and Huo (2001) performed a sensitivity analysis on the buffer time for the AGAP, noting that the length of the buffer time significantly influences the gate assignment process, so a reasonable minimum value should be used. Yan et al (2002) looked at the suitability of Flexible Buffer Times (FBT), and showed that given low delays, even short FBTs usually improve real-time objectives. Wei and Liu (2009) showed the feasibility and effectiveness of using a fuzzy model in conjunction with fixed buffer times for the GAP. Wu and Caves (2000) and Wu and Caves (2004) demonstrated the significance of a proper use of schedule buffer time in maintaining schedule punctuality and performance by balancing trade-offs between schedule punctuality and aircraft utilization.

2.5 Decision Variables

Table 2 List of the decision variables which are used in this model.

Name	Description
y_{ij}	Specifies the assignment of flights to sorting stations. $y_{ij} = 1$ if baggage sorting station $i \in \{1, \ldots, N\}$ is assigned to flight $j \in \{1, \ldots, M\}$, and 0 otherwise.
r_j	Specifies the necessary reduction in service time for flight $j \in \{1, \ldots, M\}$, given the assigned starting service time, s_j.
s_j	The assigned starting service time for flight $j \in \{1, \ldots, M\}$, since a sorting station can only service one flight at a time. In this model $s_j = t_j + r_j$.

Table 2 lists the decision variables which are used in this model. The solution algorithms will attempt to find values of y_{ij} and r_j such that the constraints in Section 2.6 are met and the relevant objectives (e.g. maximising the number of assignments and minimising reductions in service times) in Section 2.7 are improved. The actual start of service s_j can be calculated from r_j and t_j ($s_j = t_j + r_j$) and the service time of a flight is the duration from s_j to e_j and the target service time is the duration from t_j to e_j.

2.6 Constraints

The various constraints which apply to baggage sorting station assignments can be summarised as follows.

2.6.1 Assignment Limits

Each flight may be assigned to at most one baggage sorting station, as expressed by Inequality 1. In normal operations, each flight should be assigned to exactly one sorting station, in which case Inequality 1 would instead be an equality. However, in extreme situations, where there are insufficient sorting stations (as considered in this paper) there may be no feasible assignment of flights to sorting stations such that all flights can be assigned. We note that, in some airports, some flights may be assigned to multiple sorting stations, in which case this inequality should reflect that fact and the objectives should be modified accordingly.

$$\sum_{i=1}^{N} y_{ij} \leq 1 \qquad \forall j \in \{1, \ldots, M\} \tag{1}$$

2.6.2 Reduction in Service

Baggage sorting stations can only be used simultaneously by one flight, so it may be necessary to reduce the buffer time between flights in order to assign flights to the same sorting station, in order to increase the number of flights which can be allocated.

For any pair of different flights whose service times overlap, if the overlap in service times is greater than the maximum reduction allowed (R_l for flight l), then both flights cannot be assigned to the same baggage sorting station. Thus, Inequality 2 applies to any such pair of flights, j and l ($j \neq l$), where $t_l < e_j \leq e_l$ and $(e_j - t_l) > R_l$.

$$y_{ij} + y_{il} \leq 1 \tag{2}$$

They may otherwise be assigned to the same baggage sorting station as long as the service time of flight l is reduced sufficiently to remove the overlap. Inequality 3 applies to any such pair of flights, j and l ($j \neq l$), where $t_l < e_j \leq e_l$ and $(e_j - t_l) \leq R_l$. We note that one objective is to minimise these service time reductions, as discussed later.

$$r_l \geq (y_{ij} + y_{il} - 1) \cdot (e_j - t_l) \tag{3}$$

The service time reduction has both upper and lower limits, as expressed by Inequality 4.

$$0 \leq r_l \leq R_l \qquad \forall l \in \{1, \ldots, M\} \tag{4}$$

2.7 Objectives

A number of objectives need to be considered for this problem, and there is a trade-off to be made between them.

2.7.1 Maximise the number of Assignment of Baggage Sorting Stations

The first and most important objective is to maximise the number of flights assigned to baggage sorting stations, as expressed by Formula 5. In practice at airports, this objective would probably be a hard constraint at most times, since all flights would normally have to be serviced, but we wish to observe the performance of the algorithms when there are too few sorting stations as well as when these are sufficient or plentiful.

$$max \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \tag{5}$$

2.7.2 Minimise Distance

The distance between the assigned baggage sorting stations and the flights should be as short as possible. This objective aims to minimise the inconvenience, work and time involved in getting baggage to the aircraft, as previously discussed, and could relate to preferences rather than strictly to distances, as discussed later. It can be expressed by Formula 6 where $\sum_{i=1}^{N} (y_{ij} \cdot d_{ij})$ corresponds to the distance between flight j and its assigned baggage sorting station.

$$min \sum_{j=1}^{M} \left(C_j \cdot \sum_{i=1}^{N} (y_{ij} \cdot d_{ij}) \right) \tag{6}$$

2.7.3 Robustness

The ability to absorb the effects of uncertainty and variability in a schedule is normally referred to as robustness. More robust assignments are, obviously, preferred over less robust assignments. The size of the gaps between consecutive assignments to the same sorting station can be regarded as a measure of the robustness. Robustness could be increased in a number of ways and two of these are presented below.

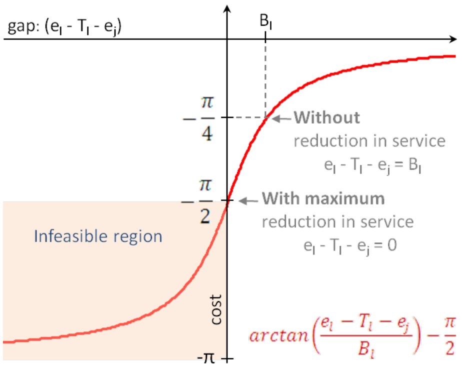

Fig. 4 Plot illustrating how u_{ij} varies depending upon the gap $(e_l - T_l - e_j)$.

Minimise Reduction in Service

Given the detrimental effects that the reduction in service time has for the robustness of the assignment as against real-life delays, it is advisable to minimise the total reduction in service time, thus maximising total buffer times. This objective can be expressed by Formula 7.

$$min \sum_{j=1}^{M} r_j \tag{7}$$

Gaps Between Assignments

Given that it may be necessary to reduce the service times of some flights, it is preferable to have more flights with small reductions in service rather than fewer but larger reductions in service time. A non-linear cost for service time reduction can be used for this purpose.

$$u_{ij} = \begin{cases} \arctan\left(\frac{(e_l - T_l - e_j)}{B_l}\right) - U & \begin{array}{l}\text{if } j < l, y_{ij} = 1,\\ y_{il} = 1 \text{ and}\\ \sum_{k=j+1}^{l} y_{ik} = 1\end{array} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

$$\max \left(\sum_{i=1}^{N} \sum_{j=1}^{M-1} u_{ij} \right) \tag{9}$$

Although many objective functions could model this preference, we will limit our discussion here to the arctangent function, since we have adopted this ourselves for use in our follow-on perturbative algorithms, which we apply to further improve the assignments that are generated using these constructive algorithms (see Section 5). This was previously used by Diepen (2008) and is shown in Equation 8 and Formula 9. This considers the ratio between the actual gap and the desired gap between two consecutive flights which have been assigned to the same sorting station. Since flights assigned to different sorting stations are preferred to those assigned to the same sorting station the arctangent needs to be offset by a constant $U = \pi/2$ in order to reflect this preference. A plot of the resulting function is given in Figure 4.

This robustness objective can be considered to handle the objectives of 'Minimising the Service Reduction' and spreading the reductions in service which were presented in Ascó et al (2011).

2.7.4 Fair Workload

The fairness objective corresponds to the minimisation of the total deviation of the actual usage of each baggage sorting station from the mean usage of all baggage sorting stations. This is expressed by Formula 10, where $e_j - s_j$ corresponds to the actual service time for the flight j, which is the usage time of the baggage sorting station. This objective aims to find a fairer assignment across sorting stations, as discussed in Abdelghany et al (2006).

$$min \sum_{i=1}^{N} \left| \underbrace{\sum_{j=1}^{M}(y_{ij} \cdot (e_j - s_j))}_{\text{workload for station i}} - \underbrace{\frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(y_{ij} \cdot (e_j - s_j))}{N}}_{\text{mean workload over all stations}} \right| \tag{10}$$

2.7.5 Preferred Piers

Flights may have preferred piers and these should be considered when assigning baggage sorting stations. In this paper, we consider that it is preferable to assign to each flight those sorting stations which are on the same pier. This objective is correlated to the distance minimisation objective (Formula 6) and is not, therefore, considered separately. The assignment of baggage sorting stations to preferred piers is considered in different ways by the different sorting station assignment algorithms which are described in Section 3.2.1, and which differ in whether they first consider pier preferences or avoid buffer time reductions.

2.7.6 Other Objectives

It is preferable that flights from the same carrier to the same destination be assigned to the same baggage sorting station, so that, for example, any delayed baggage could be transported on the next flight. However, flights would also normally be assigned to stands according to carrier, and potentially according to destination (or at least long-haul vs. short-haul).

Other objectives are also possible, such as a reduction in the number of open sorting stations (to reduce the number of baggage handlers required), however, these are in direct conflict with equity and reduction in service considerations. These are not considered in this paper for reasons of space, although there are some observations made about them in Section 3.2.2, in the Last In First Out (LIFO) discussion.

3 Algorithms

The constructive algorithms considered in this paper assign baggage sorting stations to flights one at a time until no more assignments are possible. Flights are first ordered according to one of the flight ordering methods discussed. A sorting station is then selected for each in turn. The flight ordering and baggage sorting station assignment problems are considered below.

3.1 Flight Ordering Methods

The flight ordering method determines the order in which flights are selected for assignment. The following different sorting approaches are considered here:

1. **Order by Starting Time (OST)**. This orders flights into ascending order by their t_j values. From the algorithm pseudo code presented therein, this appears to be what was previously used in Abdelghany et al (2006).
2. **Order by Departure Time (ODT)**. This was previously used by Ding et al (2005) for the Airport Gate Assignment Problem (AGAP). This orders flights into ascending order of their e_j values. When two flights have the same service end times, this will implicitly sort them by their target starting time t_j. When service time reductions are not permitted, sorting by service end times provides maximum assignments when using Last In First Out (LIFO) baggage sorting station selection and not constraining the set of sorting stations from which to select (see Section 3.2.2).

3.2 Baggage Sorting Station Assignment

Once the flight to assign has been identified, the next stage is to determine which sorting station to assign it to. Baggage Sorting Station Assignment involves two stages. The first decision is upon which sets of baggage sorting stations to consider for assignment and in what order. In particular, whether only those for the same pier should be considered first, and whether service time reductions should be considered. The second decision involves the ranking of baggage sorting stations within each set, to enable the selection of an individual baggage sorting station for assignment.

3.2.1 Baggage Sorting Station Assignment Algorithms

The baggage sorting station assignment algorithm determines which sets of baggage sorting stations (for example only those on the same pier, or on all piers) are considered, in which order, and at what point reductions in service times are considered within each set. The baggage sorting stations within each set are then considered according to a selection priority given in Section 3.2.2. We note that these algorithms were previously presented in Ascó et al (2011), although algorithms 'A', 'B' and 'C' were named 'A', 'C' and E respectively.

Algorithms 'A' to 'C' express different priorities. Algorithm 'A' will attempt to assign all aircraft to their own piers before considering assigning any aircraft to other piers. Algorithm 'B' is similar to 'A' but considers alternative piers or reductions in service for the current aircraft prior to considering the next aircraft, giving a much weaker preference overall. Algorithm 'C' will not impose any restriction.

Algorithm 'A': Baggage Sorting Station Assignment Algorithm A (Pier Preference)

```
Order flights using the flight ordering method (Sec. 3.1).
forall flights in order, assign to sorting station
    On own pier without service time reductions.
    Otherwise, on own pier allowing reduced service.
end
forall unassigned flights in order, assign to sorting station
    From anywhere, without service time reductions.
    Otherwise, from anywhere, allowing reduced service.
end
```

Algorithm 'B': Baggage Sorting Station Assignment Algorithm B (Partial Pier Preference)

```
Order flights using the flight ordering method (Sec. 3.1).
forall flights in order, assign to sorting station
    On own pier without service time reductions.
    Otherwise, on own pier allowing reduced service.
    Otherwise, from anywhere, without reduced service.
    Otherwise, from anywhere, allowing reduced service.
end
```

Algorithm 'C': Baggage Sorting Station Assignment Algorithm C (No Pier Preference)

```
Order flights using the flight ordering method (Sec. 3.1).
forall flights in order, assign to sorting station
    From anywhere, without service time reductions.
    Otherwise, from anywhere, allowing reduced service.
end
```

In each case, once the algorithm has determined the set of sorting stations for consideration, the appropriate sorting station to assign from amongst those available at the time is determined by the baggage sorting station selection method which is being used (see Section 3.2.2).

3.2.2 Baggage Sorting Station Selections

The Baggage Sorting Station Selection method determines which of the baggage sorting stations in the current set should be assigned to the current flight. The following methods are considered:

1. **First In First Out (FIFO)**: The baggage sorting station with the earliest free service time amongst all the baggage sorting stations in the set under consideration is selected. This will initially keep opening new service stations, while they exist, since a new one would always be the least recently used. This is useful for meeting the fairness objective expressed by Formula 10.
2. **Last In First Out (LIFO)**: The most recently used baggage sorting station amongst those in the set is selected. This selection reduces the number of baggage sorting stations in use at any time, since a new baggage sorting station is only opened when the previous ones cannot be assigned to the flight. When flights are ordered by their departure times, service time reductions are not permitted and assignment Algorithm 'C' is used (so that all sorting stations are considered, rather than only those on the preferred pier), this selection method guarantees the maximum assignments (maximising the objective expressed by Formula 5), by minimising the wasted/idle time between flights, Ding et al (2004) and Cormen et al (2001).
3. **Closest**: The baggage sorting station with the least distance to the current flight is selected from those in the set under consideration. This considers both new and previously used service stations. This method is useful for meeting the distance reduction objective expressed by Formula 6. Using the distance measure used in this paper, this objective will ensure that flights are assigned to sorting stations on their own pier by preference. Where sorting stations

have the same distance, a LIFO method is used to break the ties.

3.3 Lookahead and Improvement

Haralick and Elliott (1980) considered the concept of "Lookahead and anticipate the future in order to succeed in the present" and "Lookahead to the future in order not to worry about the past". A type of lookahead was also used in Voß et al (2005). The Order by Departure Time (ODT) flight ordering method could potentially perform badly on the maximisation of assignments. The aim of the Order by Departure Time Lookahead and Improvement (ODTLI) look-ahead is to keep the ODT flight ordering but to look ahead when assigning sorting stations, thus improving the assignment objective. The developed ODTLI algorithm maintains a list of available sorting stations for this flight. Rather than immediately making a selection from this list using the ordering method, it first looks ahead to find whether the selection of any of the available sorting stations may deem a future flight infeasible. If this is the case, and there are other available sorting stations from which to select, this sorting station will be removed from the list. At the improvement stage of the process, sorting stations which have been removed will be reconsidered and may be exchanged for a station in the list if this improves the current selection method used.

4 Results

This section details the experiments which were performed to evaluate the differences between the algorithms described in Section 3 and the ways in which these depend upon the number of sorting stations available for assignment to flights. The behaviour was studied both when there are too few sorting stations as well as when sorting stations are plentiful.

4.1 Problem Data

Since it would be unrealistic to assume that baggage from a flight at a stand in one terminal is serviced by a baggage sorting station in another terminal (e.g. passengers usually go through security and board flights from the same terminal at which their baggage was checked in), it was decided to centre the analysis on a single terminal.

Following the work in Ascó et al (2011), NATS Ltd (NATS) provided us with more detailed data for London Heathrow airport, which contained the details of the assignments of flights to stands. It was composed of 194 flights for 16th December 2009 and 163 flights for 1st March 2010 and only considered flights which were departing from Terminal 1. In this work we will present only the results for the 1st March 2010 data, since the 16th December 2009 results are very similar. We will point out any cases where the results differed between the data sets. Although we executed experiments considering both three and four pier topologies, we will only present the results for the three pier topology since the results were similar for both and the three pier topology was used in Ascó et al (2011), thus allowing a comparison of the results in the relevant papers. Ascó et al (2011) used information from the British Airports Authority (BAA) website, which did not include stand allocation information. Despite this, many of the results here are identical to those from Ascó et al (2011) and comparisons between these results and the earlier results in Ascó et al (2011) will be made where this is useful.

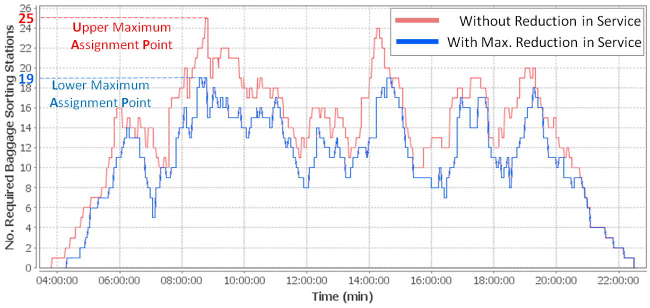

Fig. 5 Minimum number of required baggage sorting stations throughout the day.

For the moment, we will assume that reductions in service time can only reduce the buffer time rather than the base service time, so $R_j = B_j$. Service times were set based upon the flight destination so that $T_j = 1$ hour and $B_j = 15$ minutes for European flights, and $T_j = 1\frac{3}{4}$ hours and $B_j = 30$ minutes for non European (longer haul) flights, since these are usually larger flights with more baggage to process and often with a requirement to check-in earlier. Figure 5 shows the total number of flights requiring service at different times of the day when full buffer times are used (i.e. where there is no service time reduction allowed) and when no buffer times are included. It is possible to draw the following conclusions:

1. With a limited number of baggage sorting stations, the maximum line heights shown in Figure 5 can be considered as an indication of the difficulty of the assignment problem.

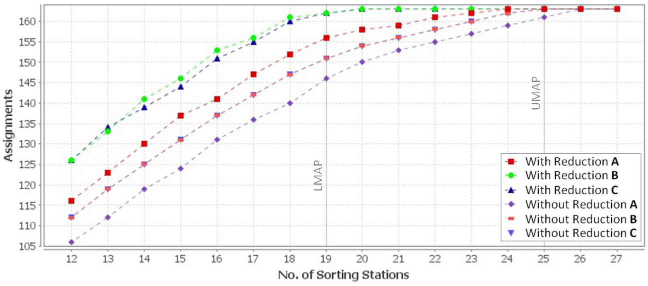

Fig. 6 Number of sorting stations assigned, OST ordering and LIFO selection method.

2. Fewer sorting stations are required when buffer times are not included, so the peaks are lower, but the absence of buffer times would result in less robust solutions.

All of the experiments described in this section were executed using a single threaded Java application, running on a 3GHz MS Windows XP SP3 PC. The number of sorting stations was varied from 3 to 54. For the purpose of the distance reduction objective, a distance of one unit was assumed between different sides of a pier and a distance of two units was assumed between different piers, so that it is preferable to use the other side of the same pier before considering sorting stations at other piers.

4.2 Initial Observations

Experiments were executed for different numbers of baggage sorting stations, using each of the sorting station assignment algorithms and sorting station selection methods. Two cases were considered: firstly without allowing reductions in service times (i.e. requiring full buffer times) and secondly allowing reductions in service times (i.e. allowing buffer times to be reduced).

Figure 6 shows the number of sorting stations which could be assigned to flights for the Order by Starting Time (OST) flight ordering method with Last In First Out (LIFO) sorting station selection method, for various numbers of available sorting stations, comparing the situation when reduction in service is and is not allowed. We had originally planned to use Order by Departure Time (ODT), but it was shown in Ascó et al (2011) that the Order by Starting Time (OST) ordering method provided better assignments than Order by Departure Time Lookahead and Improvement (ODTLI) when reductions in service were allowed and the number of sorting stations was close to, or above, the lower maximum assignment point (LMAP). This persuaded us to use OST in our initial observations here. Figure 6 gives an idea of the behaviour of the algorithms as the number of baggage sorting stations changes. Comparing the results for the different baggage sorting station assignment algorithms the following can be observed:

1. As expected, allowing reductions in service times allows more flights to be serviced since smaller service times may allow a flight to sit between two other flights where this would otherwise be impossible.
2. Regardless of whether reductions in service are permitted, sorting station assignment Algorithm ‘A’ achieves fewer assignments than the other algorithms. This is a consequence of Algorithm ‘A’ assigning aircraft to their own pier by preference when an assignment to a different pier may have allowed more flights to be packed in.
3. When reductions in service are not permitted, the performance of Algorithm ‘C’ was exactly the same as Algorithm ‘B’ in this case. The results in Ascó et al (2011) show that in general Algorithm ‘C’ always performed at least as well as Algorithm ‘B’, and sometimes better, as we would expect since the pier preference can sometimes conflict with maximising the allocations. Interestingly, this was not always the case when reductions in service were permitted, and there are instances when the preference for the same pier actually means that more flights can be assigned. Similar results were obtained in Ascó et al (2011) for different data sets with random allocation of stands to flights, indicating that the advantages of Algorithm ‘C’ when reductions are not permitted no longer apply when reductions are allowed.
4. When reduction in service was not permitted, Algorithm ‘C’ achieved the maximum assignment at the upper maximum assignment point (UMAP) from Figure 6, when there are 25 baggage sorting stations. We note however, that this is only actually guaranteed when ordering flights by departure times with LIFO selection method.

It was also noted that, in each case, these results corroborate those from Ascó et al (2011), where different data sets were used and that both sets of results indicate that the better assignment method will depend upon the ratio of flights to sorting stations. Furthermore, we note that the counts of the minimum number of sorting stations which are needed with and without reductions in service time, shown in Figure 5, provide a simple method to determine whether the available sorting stations are sufficient or not for avoiding reductions in service times.

In order to determine the maximum sorting station assignments when reduction in service time is permitted, experiments were executed with the buffer times removed (equivalent to maximal service time reduction), OST ordering method, Algorithm ‘C’ and LIFO selection method. The results are shown in Figure 7, for the 163 flight problem. More flights can be assigned when reductions are permitted, as expected, until sufficient sorting stations are available to assign the maximum number of flights even without needing reductions. In most cases, allowing reduction is almost as good as using maximum reductions.

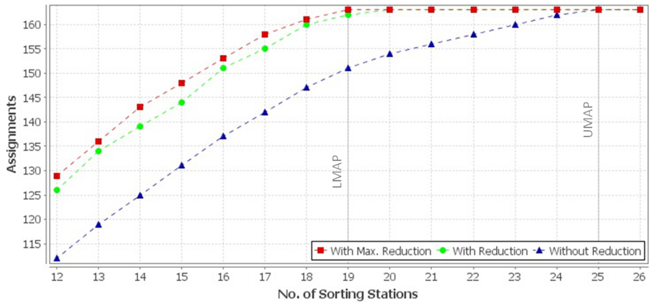

Fig. 7 Number of assignments, OST ordering method, Algorithm ‘C’ and LIFO selection method.

With maximum reductions (i.e. no buffer times), the maximum assignment occurs when there are 19 baggage sorting stations for the 163 flight problem. These values are the same as the theoretical minimum (the lowest maximum assignment point, LMAP) shown in Figure 5. Here, OST (Order by Starting Times) is achieving maximal number of assignments at the theoretical minimum points (LMAP for maximal reductions and UMAP for no reductions in service times), even though it gives no guarantee of doing so (unlike ODT, Order by Departure Times).

Figure 6 can also be used to compare the performance of Algorithms ‘A’, ‘B’ and ‘C’ in terms of the number of assignments which are achieved when reduction in service time is permitted, using the OST ordering method and the LIFO selection method. This shows that Algorithm ‘A’ provides the lowest number of assignments, as was also seen in Ascó et al (2011). Both algorithms ‘B’ and ‘C’ provided a similar number of assignments, with Algorithm ‘B’ providing slightly more than ‘C’ in some cases.

Since reductions in service time have obvious benefits, the remaining experiments consider the cases where these are permitted and evaluate the differences between Algorithms ‘A’, ‘B’ and ‘C’ and also between the different flight ordering and baggage sorting station selection methods.

4.3 Comparison of Assignments With Service Reduction

Figure 8 compares the ODTLI and OST flight ordering methods, showing the number of sorting station assignments which were made with the LIFO selection method. This shows that the ODTLI flight ordering method provided a better assignment when there were fewer sorting stations (between 13 and 16 sorting stations), but at some point, as the number of sorting stations increases, the difference decreases and as it approaches the number necessary for optimal assignment (LMAP), the OST flight ordering actually improves upon ODTLI.

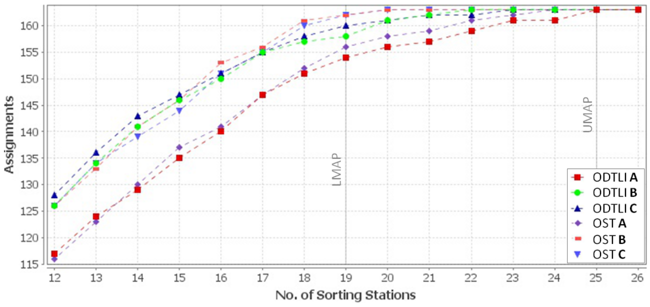

Fig. 8 Number of assignments for LIFO selection method with different algorithms and ordering methods.

Comparison of some resulting assignments showed that, perhaps counter intuitively, not only was ODTLI (Order by Departure Time Lookahead and Improvement) failing to assign more flights at these times, but the flights which were not assigned had longer service times than those which OST (Order by Start Time) failed to assign. Indeed, there were cases where every aircraft which OST failed to assign was a short-haul flight and every aircraft which ODTLI failed to assign was a long-haul flight. The order of consideration of flights appears to be important in this case.

The key to understanding this behaviour is to consider the size of the remaining gaps. Since the ODT and ODTLI methods order the flights by their departure times, where flights have similar service starting times, preference will be given to flights with shorter service times (i.e. earlier departure times). On the other hand, the OST choice of flights could be regarded as preferring flights with longer service times (for similar departure/end of service times). By assigning long-haul flights first, the OST algorithm was able to fit short-haul flights into the remaining gaps (with appropriate service time reductions). However, by assigning short-haul flights first the ODTLI was then unable to schedule the long-haul flights which remained, resulting in fewer assignments. When there are few sorting stations, the ability of the ODTLI choice to minimise the gaps is a useful one and results in more sorting station assignments than the OST ordering method. However, as the number of sorting stations increases, the remaining gaps start to become large enough to accommodate short-haul aircraft, and OST performs better.

Further experiments showed that this behaviour was not restricted to the LIFO selection method, but also occurred for the First In First Out (FIFO) and 'Closest' selection methods, and did so at the same number of sorting stations.

4.4 Comparison of Distances With Service Reduction

Figure 9 shows the results as far as the distance reduction objective (expressed by Formula (6)) is concerned. These show the total distance between the assigned baggage sorting stations and the stands at which the flights are located. Results are shown for the three sorting station selection algorithms 'A', 'B' and 'C', with the 'Closest' selection methods and the ODTLI and OST flight ordering methods.

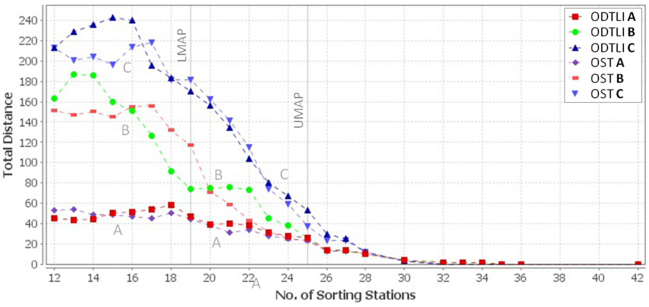

Fig. 9 Total distance for 'Closest' selection method.

The distance basically measures the number of flights which could not be assigned to sorting stations on their preferred pier. It can be observed that the total distance decreases as the number of sorting stations is increased, since more sorting stations become available on the preferred piers. Even after all flights have been assigned to sorting stations, the distances can be positive, since the availability of a sorting station at the terminal does not imply that it is on the correct pier for the flight.

As expected, since Algorithm 'A' first attempts to assign flights to the same pier and considers applying a service time reduction before considering other piers, Algorithm 'A' performs better than algorithms 'B' and 'C' when there is a shortage of piers. For similar reasons, Algorithm 'B' performs better than Algorithm 'C'. However, Algorithm 'C' assigned more flights to sorting stations, and unassigned flights are here assumed to have no distance, so we also need to take this into account. Figure 10 shows the mean distance per assigned flight to avoid the problem of unassigned flights, and it can clearly be seen that Algorithm 'A' attained the lower mean distance.

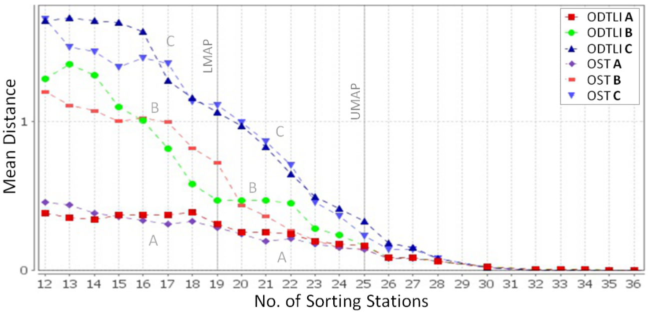

Fig. 10 Mean distance, ODTLI and OST ordering methods and 'Closest' selection method.

It is possible to conclude from this that, in the cases where Algorithm 'B' achieves at least the same number of assignments as Algorithm 'C', Algorithm 'B' would be the preferable choice since the distances would be lower. On the other hand, by the time that Algorithm 'A' would be achieving maximal assignment (which is usually considered to be the primary objective); there is no distance benefit from using Algorithm 'A' rather than Algorithm 'B'.

4.5 Fair Workload With Reduction in Service

As a measure of fairness we considered the deviation of total usage times of the sorting stations from the mean usage time. This corresponds to the fairness objective which was expressed by Formula (10).

Figure 11 compares the results for the 'Closest', FIFO and LIFO sorting station selection methods and the ODTLI and OST flight ordering methods, showing the total deviation in seconds from the mean usage across all baggage sorting stations, using sorting station assignment Algorithm 'C'.

The FIFO selection method can be considered to take fairness into account, only reusing a sorting station once the others have been used, and indeed it consistently performs better than LIFO and 'Closest' for both flight ordering methods. However, although the FIFO selection method will cycle through the sorting

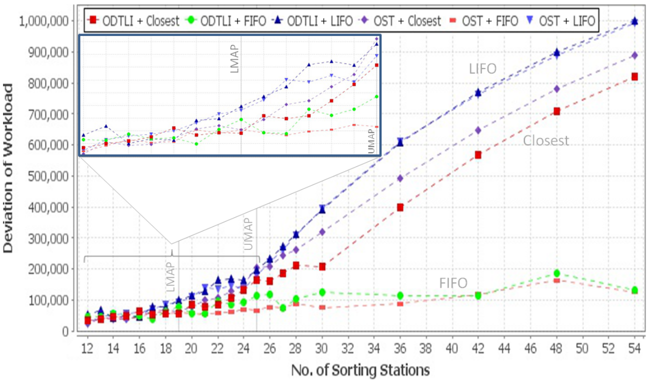

Fig. 11 Fair workload for different ordering methods, selection methods and Algorithm 'C'.

stations, giving a more equitable number of flights to each sorting station, long-haul and short-haul flights are treated identically. This can result in differences in the total service times. These differences will depend upon how many of the long-haul flight assignments coincide so that they are assigned to the same service stations. As the number of sorting stations is increased, a cyclic-type behaviour is observed.

Conversely, the LIFO selection method will continue to re-use the same sorting stations wherever possible, so increasing the number of sorting stations will further increase the inequity, as can be observed in Figure 11.

The 'Closest' method takes no explicit account of equity or sorting station reuse frequency, and instead will tend to follow the flight assignment. It can be observed that this results in an inequity almost as great as for the LIFO method.

4.6 Reduction in Service

Figure 12 shows the total reduction in service time (expressed by Formula (7)) for all assigned flights, with differing numbers of baggage sorting stations, using Algorithm 'C', comparing the performance of ODTLI and OST flight ordering methods and 'Closest', LIFO and FIFO sorting station selection methods.

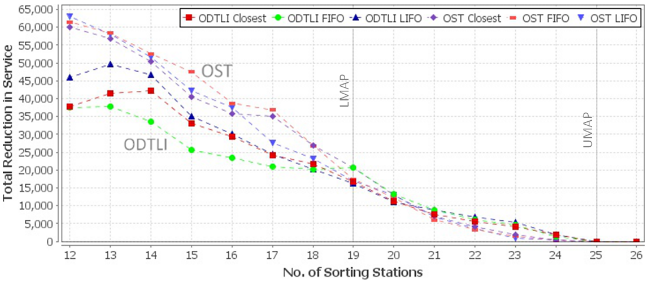

Fig. 12 Total reduction in service time and Algorithm 'C'.

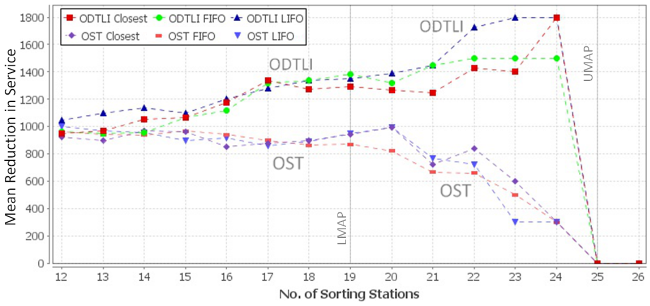

Fig. 13 Mean reduction in service time and Algorithm 'C'.

Figure 12 shows the total reduction in service time. Figure 13 shows the mean reduction in service time for the flights which have had a reduction in service time. When comparing Figures 12 and 13 we observe that, for the ODTLI ordering method, as the number of sorting stations increases so the reduction in service time initially increases, as more assignments are achieved, although the mean reduction in each case increases either more slowly or not at all. Figure 8 shows that the number of assignments increases at this point. This indicates that the increased number of sorting stations available plus the ability to reduce the service time are both contributing to an increase in the number of flights assigned at that time.

As the number of sorting stations increases further, a point is soon reached where the total reduction decreases, but the mean reduction per reduced sorting station goes up. This indicates that more and more flights are being assigned without reduction in service. This situation continues until the number of sorting stations is sufficient to allow all of the assignments to be made, at which point the total reduction in service decreases, until all flights can eventually be assigned without a reduction in service time. This was also observed in Ascó et al (2011).

In the OST ordering method, the mean reduction is relatively stable, whereas the total reduction is constantly decreasing, indicating that the number of sorting stations with reduced service is decreasing over the same period. Soon after the LMAP, when there are 20 sorting stations, a point is reached where the number of sorting stations is sufficient to allow all of the assignments to be made (see Figure 8), with ever decreasing reductions in service time as the number of sorting stations is further increased, so that the total reduction in service and the mean reduction in service times both decrease, until eventually all flights can be assigned with no reduction in service time.

4.7 Comparison between Constructive Solutions and CPLEX

To compare the quality of the solutions obtained when applying the different constructive algorithms with other approaches presented in this and the following sections, a weighted sum of the three main objectives (Maximise Assignment of Baggage Sorting Stations 2.7.1, Minimise Distance 2.7.2 and Minimise Reduction in Service 2.7.3) was considered, Equation 11. The weights used are the same as those presented in Ascó et al (2012) with the weights of $W_1 = 90$, $W_2 = 1$ and $W_3 = 0.008$ respectively, which give an appropriate balance between these objectives.

$$f = W_1 \cdot \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} - W_2 \cdot \sum_{j=1}^{M} r_j - W_3 \cdot \sum_{i=1}^{N} \left(C_j \cdot \sum_{j=1}^{N} y_{ij} \cdot d_{ij} \right) \tag{11}$$

For an objective quality assessment, the solutions obtained when applying the constructive algorithms were compared both with the Upper Bound and the best solutions obtained from applying CPLEX to the Integer Linear Programming (ILP) representation of the Airport Baggage Sorting Station Problem (ABSSP), presented in Ascó et al (2012), running on a 2.4GHz Windows 7, 64bit machine with 4GB RAM for 1 hour for each group of baggage sorting stations (BSSs) for a 3-pier topology. Initially, the study was conducted for the two original data sets presented in Ascó et al (2011) composed of 219 flights and 270 flights respectively using CPLEX. The results are presented in this and the next sections. Then the study was extended to the data sets presented in Ascó et al (2012) and was compared against the best solution obtained by a Genetic Algorithm (GA), the results of which are presented in Sections 4.9 and 4.10.

Figure 14 shows the improvement percentage in the fitness of the results for different constructive algorithms compared to the solution obtained from CPLEX with a 1 hour run. Different numbers of BSSs were assessed. The scale is from the worst constructed solution (f_w, 0%) to the Upper Bound obtained by CPLEX (f_{UB}, 100%) and the values are calculated using Equation 12. The best constructed solution is also shown for comparison, labelled 'Best', which refers to the best solution of all of the solutions obtained by application of the constructive algorithms described in this paper for the given number of BSSs. It should be noted that 100% improvement represents solutions which reach the upper bound for their specific number of baggage sorting stations, whereas 0% corresponds to no improvement over the worst initial (constructed) solution. We note that application of the constructive algorithms to obtain the solutions requires no more than 9 milliseconds per solution whereas CPLEX was run for 1 hour.

$$\%Fitness = \frac{f - f_w}{f_{UB} - f_w} \cdot 100 \tag{12}$$

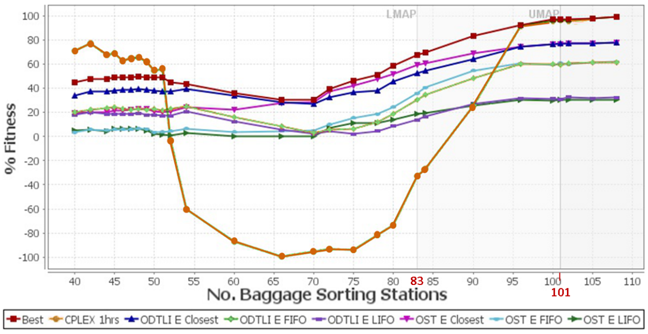

Fig. 14 Percentage improvement in fitness for 219 flights.

Three areas can be clearly identified with different comparative fitness between the constructive algorithms and CPLEX. For a very low number of BSSs ($N \ll$ Lower Maximum Assignment Point (LMAP)) the solutions obtained by CPLEX after one hour are better that any of the solutions obtained by applying the different constructive algorithms. There is then a region from a low number of BSSs (~ 51, below the LMAP) to a higher number (~ 92, between the LMAP and Upper Maximum Assignment Point (UMAP)) where CPLEX does not do as well as any of the constructive algorithms (for a mere 9 milliseconds run against 1 hour for CPLEX solutions). Finally, for a number of BSSs near the UMAP, both methods provide solutions with similar or equal fitness.

4.8 Using Constructive Solutions for CPLEX

Figure 15 shows the effect of the final solution on the fitness (Equation 11) from CPLEX when using different types of initial solution. Where 'Best' and 'Worst' are defined in the same way as for Figure 14. These results show that feeding CPLEX with any of the solutions obtained by application of the constructive algorithms improves the final solution obtained by CPLEX, with the exception of the worst generated solution, which when fed to CPLEX may not assist in finding better solutions any earlier, as can be seen in the case of 102 BSSs. In the two areas where CPLEX performs well when compared with the constructive algorithms, the use of these constructive solutions does not in some cases seem to help CPLEX find fitter solutions.

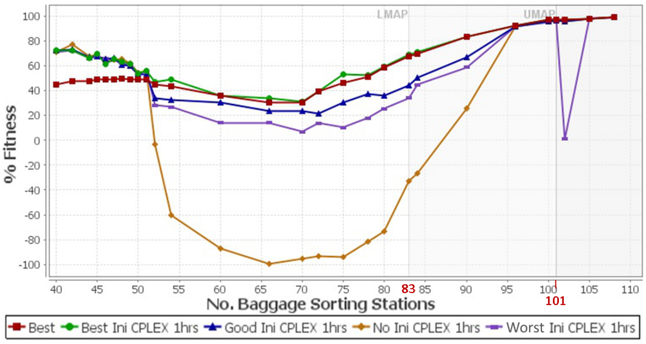

Fig. 15 Percentage improvement in fitness for 219 flights.

It is also interesting to examine the effect of using the different solutions obtained from the constructive algorithms in reaching the final solution as CPLEX progresses in the search, as can be seen in Figure 16. The initial solution fed to CPLEX appears to make a huge difference to CPLEX, allowing it to reach better solutions earlier, and in many cases improving the fitness achieved by the final solution after an hour. It can also be seen that some of the constructive solutions provide very good solutions, which CPLEX is unable to improve upon, as is shown in Figure 15.

Fig. 16 CPLEX progress fitness for 83 BSSs (LMAP) and 219 flights.

4.9 Constructive Solutions and an Evolutionary Algorithm

Evolutionary Algorithms (EAs) are population based algorithms, part of the group of metaheuristics which use the solutions within a population to guide the search, hopefully to the optimal solution(s). For the purpose of assessing the potential value of the constructive solutions obtained here, new experiments were designed and executed for an implementation of the Canonical Genetic Algorithm (CGA) using the Evolutionary Computation Java library (ECJ) (Java-based Evolutionary Computation research system, reviewed in Wilson et al (2004)). The data sets used were those provided by NATS for London Heathrow airport Terminal 1. The operators used are 1-point random crossover and random mutation.

An integer encoding of the Airport Baggage Sorting Station Assignment Problem (ABSSAP) was also used with randomly generated initial solutions and a population size of 1,000. The average fitness from the solutions obtained by the GA implemented together with the constructive algorithm solution fitness is shown in Figures 17 and 18. The fitness for the GA in such figures corresponds to the average fitness for all of the best solutions found amongst the thirty instances run. These results show that the constructive algorithms which were used generally provided better solutions than the CGA throughout all of the ranges of numbers of baggage sorting stations.

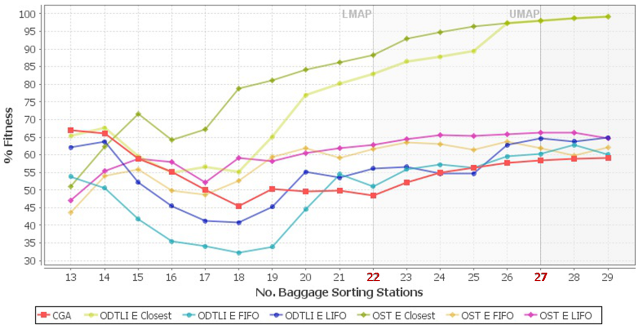

Fig. 17 Fitness for 194 flights, a 3-pier topology and 48 stands for CGA and some constructive algorithms.

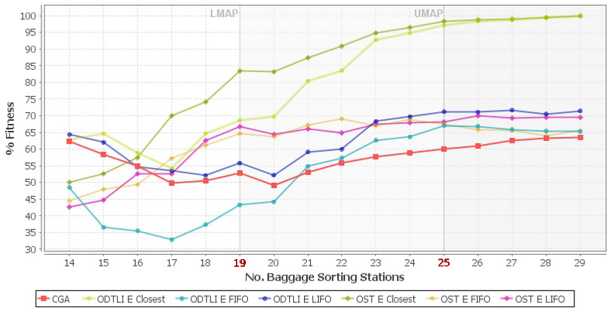

Fig. 18 Fitness for 163 flights, a 3-pier topology and 48 stands for CGA and some constructive algorithms.

4.10 Using Constructive Solutions for the Evolutionary Algorithm

Further experiments were run to identify whether the use of the constructive solutions as part of the initial population for the CGA may help the algorithm to reach fitter solutions. The CGA was run thirty times for each number of BSSs, for a population size of 1,000, using both an initial population of random solutions and the best solutions obtained from applying the constructive algorithm 2,000 times. The 'Best' refers to the best solution amongst all of those generated using the constructive algorithms described. The results show that this approach is not detrimental to the algorithm, and in some cases it is seen to help the CGA to reach fitter solutions, as shown for number of BSSs lower than 17 in Figures 19 and 20.

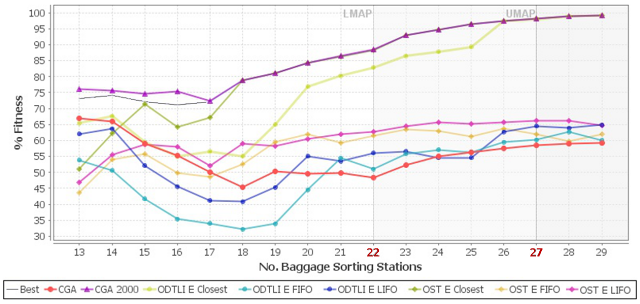

Fig. 19 Fitness for 194 flights, a 3-pier topology and 48 stands for some constructive algorithms and CGA for different initial population.

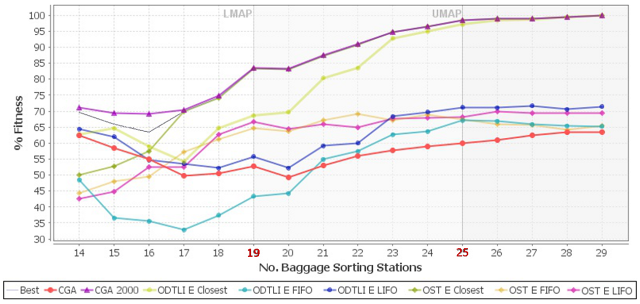

Fig. 20 Fitness for 163 flights, a 3-pier topology and 48 stands for some constructive algorithms and CGA for different initial populations.

The solutions provided by these constructive algorithms have also been used in Ascó et al (2012) to initialise some metaheuristics, and were shown to help in finding fitter solutions.

5 Conclusions

It was observed that the behaviour of the assignment methods (flight ordering, sorting station assignment algorithm and selection method) depends upon the relationship between the number of flights and the number of sorting stations. It was also observed that the different methods have different effects and can prefer different objectives. It was previously observed in Ascó et al (2011) that a data set with higher flight density (the number of flights requiring service at any time of the day) but fewer flights was more problematic than one with more flights but lower density which, as expected, implied that the flight density was more important than the total number of flights when determining the number of sorting stations required throughout the day. Some points have been identified in this paper, in terms of the number of baggage sorting stations, where the performance of the algorithms changes, and it has been noted that these depend upon the distribution of the flights over time. It has also been noted that the choice of whether or not to allow reductions in service time can affect the relative efficacy of the algorithms. In particular, if reductions in service time are to be permitted, then it may be better to select an algorithm which will not minimise the gap sizes, since these are then less likely to be available for use by other flights after application of the service time reductions. Together, these effects show that the appropriate algorithm for use depends not only upon the objective which is under consideration but also upon the problem characteristics and the relative flight density in relation to the number of sorting stations available.

Given the differences between Order by Starting Time (OST) and Order by Departure Time Lookahead and Improvement (ODTLI), the use of a generalisation of the 'Flight Ordering Methods' which uses a point in time between the starting and end flight service times to order them, namely Order Between Times (OBT), as expressed in Equation 13, is also suggested for future study. $\alpha = 0$ is equivalent to the OST, whereas $\alpha = 1$ is equivalent to the Order by Departure Time (ODT), thus by changing the α it is possible to cover the 'Flight Ordering Methods' presented in this paper and others.

$$t_j^{OBT} = \alpha * (e_j - t_j),\ \alpha \in [0 \ldots 1] \qquad (13)$$

The aim of this research was not to identify the perfect constructive algorithm which would meet all objectives, but to gain insights into the differing behaviour of the algorithms, particularly when service time reductions are permitted. Further experiments showed the potential of using these solutions to aid other solution approaches. Subsequent research, Ascó et al (2012), has also used these insights to generate better initial solutions for use with perturbative algorithms (particularly other Genetic Algorithms, Tabu Search and other Metaheuristics), improving the quality of the solutions which can be generated within very limited search times. The ability to quickly generate a variety of solutions which have different trade-offs between the objectives has also been particularly useful.

Acknowledgements We are grateful to NATS Ltd (NATS) and EPSRC for providing funding for this project, and especially John Greenwood (NATS) for his continuous support.

References

Abdelghany A, Abdelghany K, Narasimhan R (2006) Scheduling baggage-handling facilities in congested airports. Journal of Air Transport Management 12(2):76 – 81, DOI 10.1016/j.jairtraman.2005.10.001

Air Transport Users Council (2009) Annual report 2008/09. Tech. rep., AUC

Ascó A, Atkin JAD, Burke EK (2011) The airport baggage sorting station allocation problem. In: Fowler J, Kendall G, McCollum B (eds) In proceedings of the 5th Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2011), 9-11 August 2011, Phoenix, Arizona, USA, MISTA, pp 419–444, URL http://www.schedulingconference.org/previous/publications/displaypub.php?key=2011-419-444-P&filename=mista.bib

Ascó A, Atkin JAD, Burke EK (2012) An evolutionary algorithm for the over-constrained airport baggage sorting station assignment problem. In: Bui L, Ong Y, Hoai N, Ishibuchi H, Suganthan P (eds) 9th Intl. Conf. on Simulated Evolution And Learning, SEAL2012, Lecture Notes in Computer Science, vol 7673, Springer Berlin Heidelberg, pp 32–41, DOI 10.1007/978-3-642-34859-4_4

Atkin JAD, Burke EK, Greenwood JS, Reeson D (2008) On-line decision support for take-off runway scheduling at london heathrow airport. Journal of Scheduling 11(5):323 – 346, DOI 10.1007/s10951-008-0065-9

Atkin JAD, De Maere G, Burke EK, Greenwood JS (2012) Addressing the pushback time allocation problem at heathrow airport. Transportation Science DOI 10.1287/trsc.1120.0446

Bolat A (2000) Procedures for providing robust gate assignments for arriving aircrafts. European Journal of Operational Research 120(1):63 – 80, URL http://www.sciencedirect.com/science/article/pii/S0377221798003750

Cormen TH, Leiserson CE, Rivest RL, Stein C (eds) (2001) Introduction to Algorithms. The Massachusetts Institute of Technology Press, URL http://www-2.cs.cmu.edu/afs/cs/academic/class/15451-s04/www/Lectures/shortestPaths.pdf

Diepen G (2008) Column generation algorithms for machine scheduling and integrated airport planning. PhD thesis, Utrecht University, URL http://dspace.library.uu.nl/handle/1874/29732

Ding H, Lim A, Rodrigues B, Zhu Y (2004) Aircraft and gate scheduling optimization at airports. In: System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on, pp 1530 – 1605, DOI 10.1109/HICSS.2004.1265219

Ding H, Rodrigues AL, Zhu Y (2005) The over-constrained airport gate assignment problem. Computers and Operations Research 32(7):1867 – 1880

Federal Aviation Administration (2010) Faa aerospace forecast fiscal years 20102030 - forecast highlights. Tech. rep., Federal Aviation Administration, URL http://www.faa.gov/data_research/aviation/aerospace_forecasts/2010-2030/

Haralick RM, Elliott GL (1980) Increasing tree search efficiency for constraint satisfaction problems. Artificial Intelligence 14:263 – 313, DOI 10.1016/0004-3702(80)90051-X

Hassounah MI, Steuart GN (1993) Demand for aircraft gates. Transportation Research Record 1423(1423):26 – 33, URL http://pubsindex.trb.org/view.aspx?id=390306

ICAO (2010) Icao news release - pio 06/10. Tech. rep., International Civil Aviation Organization, URL http://icaopressroom.files.wordpress.com/2010/07/pio-06-10-en.pdf

Mulvey JM, Vanderbei RJ, Zenios SA (1995) Robust optimization of large-scale systems. Operational Research 43:254 – 281, DOI 10.1287/opre.43.2.264

Nikulin Y (2006) Robustness in combinatorial optimization and scheduling theory: An extended annotated bibliography. URL http://www.optimization-online.org/DB_HTML/2004/11/995.html

Pitt M, Wai FK, Teck PC (2002) Technology selection in airport passenger and baggage systems. Facilities 20(10):314 – 326, DOI 10.1108/02632770210442992

Rijsenbrij JC, Ottjes JA (2007) New developments in airport baggage handling systems. Transportation Planning and Technology 30(4):417 – 430, DOI 10.1080/03081060701461899

US Department of Transportation (2010) Air travel consumer report. Tech. rep., U.S. Department of Transportation, URL http://airconsumer.dot.gov/reports/2010/April/2010AprilATCR.pdf

Voß S, Fink A, Duin C (2005) Looking ahead with the pilot method. Annals of Operations Research 136(1):285 – 302, DOI 10.1007/s10479-005-2060-2

Wei D, Liu C (2009) Fuzzy model and optimization for airport gate assignment problem. In: IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS, IEEE, Shanghai, vol 2, pp 828 – 832, DOI 10.1109/ICICISYS.2009.5358268

Wilson GC, Intyre AM, Heywood MI (2004) Resource Review: Three Open Source Systems for Evolving ProgramsLilgp, ECJ and Grammatical Evolution, vol 3, Kluwer Academic Publishers, chap Genetic Programming and Evolvable Machines, pp 103–105. DOI 10.1023/B:GENP.0000017053.10351.dc

Wu CL, Caves RE (2000) Aircraft operational costs and turnaround efficiency at airports. Journal of Air Transport Management 6(4):201 – 208, DOI 10.1016/S0969-6997(00)00014-4

Wu CL, Caves RE (2004) Modelling and optimization of aircraft turnaround time at an airport. Transportation Planning and Technology 27(1):47 – 66, DOI 10.1080/0308106042000184454

Yan S, Chang CM (1997) A network model for gate assignment. Journal of Advanced Transportation 32(2):176 – 189, DOI 10.1002/atr.5670320204

Yan S, Huo CM (2001) Optimization of multiple objective gate assignments. Transportation Research Part A: Policy and Practice 35(5):413 – 432, DOI 10.1016/S0965-8564(99)00065-8

Yan S, Shieh CY, Chen M (2002) A simulation framework for evaluating airport gate assignments. Transportation Research Part A: Policy and Practice 36(10):885 – 898, DOI 10.1016/S0965-8564(01)00045-3