

Research Article

An Analysis of Robustness Approaches for the Airport Baggage Sorting Station Assignment Problem

Amadeo Ascó

School of Computer Science, University of Nottingham, Nottingham NG8 1BB, UK

Correspondence should be addressed to Amadeo Ascó; a.asco@bocaditos.co.uk

Received 18 December 2015; Revised 11 April 2016; Accepted 10 May 2016

Academic Editor: Włodzimierz Ogryczak

Copyright © 2016 Amadeo Ascó. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the Airport Baggage Sorting Station Assignment Problem (ABSSAP), the Baggage Sorting Stations (BSSs) are assigned to flights for the period of time necessary to perform their service for a given flights' schedule. But the flights schedule may change on the day of operation which may deem the original assignment of some flights to BSSs infeasible. These changes may create conflicts between those flights whose schedules have changed and may not be restricted to those flights but propagating to the other flights for different reasons. Conflicts depend on the original assignments for the real arrival and departure flight times on the day of operation. It is therefore desirable to consider potential delays on the day of operation when generating the original flight assignments to BSSs, such that the final flight assignments differ little or do not differ at all from the original assignments on the day of operation. The term robustness is here used to give an indication of the degree to which this has been achieved. Some existing approaches originally presented in the Airport Gate Assignment Problem (AGAP) are adapted to the ABSSAP, other approaches are suggested for generating assignments which take account of potential perturbations on the day of operation for the ABSSAP, and all of them are then compared. It is shown that the suggested approaches by themselves do not perform better than the other considered approaches but when combined they enhance the result further compared to when each approach is used alone.

1. Introduction

Flight delays are caused by many factors like airport security, weather conditions, unavailability of required resources (mechanical breakdown), delayed propagation, airport congestion, etc.

The ABSSAP corresponds to the assignments of flights to BSSs, whereas the BSSs are the end elements in the baggage system at the ground of the airport terminal where the baggage is collected from and sorted by the baggage handlers or to where the baggage from arrived flights are placed to enter the baggage system. Figure 1 shows an overall simple view of the airport baggage handling system.

Once the baggage has been loaded into the aircraft hold, the BSSs assigned to that flight are ready for use on the next scheduled flights, and any delay on a flight at the gate will not necessarily have repercussions on these. However, such delays may affect the BSSs assigned to the flight scheduled for that gate next if such a flight is held waiting for the gate to

become free. BSSs will then need to hold the baggage longer, thus potentially affecting the following assignment in turn.

Delays on the day of operation may render some assignments infeasible which need to be reassigned. It is therefore desirable to account for potential delays on the day of operation when generating the flight assignments to BSSs in the planning stage, such that the final flight assignments differ little or do not differ at all from the original assignments on the day of operation. The degree to which this is achieved is an indication of the solution robustness, so a solution which requires fewer reassignments is said to be more robust than those solutions requiring more reassignments. Robustness is the ability of assignments to resist changes consequence of perturbations by reducing or removing the need to reassign current assignments. Mulvey et al. [1] characterise the desirable properties of a solution to model robustness.

There are different ways of increasing robustness depending on the intended effect. One of the most simple and widely used methods is the introduction of a buffer time between

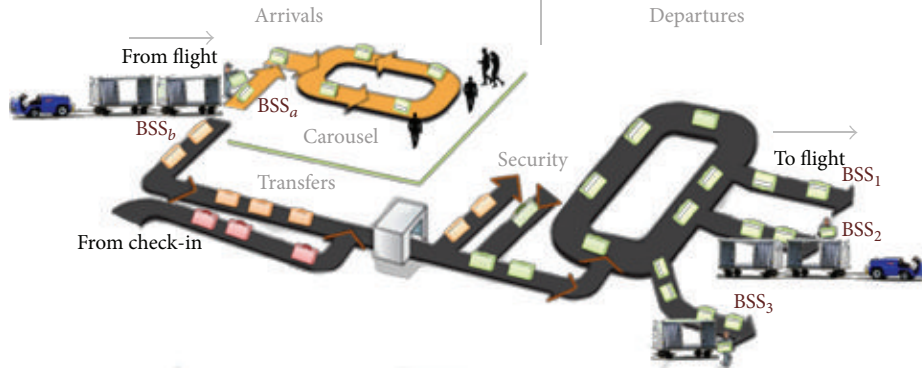


FIGURE 1: Simple view of an airport baggage handling system with the BSSs.

assignments which allows the absorption of small disturbances; implementation of this approach for the ABSSAP is presented in [2–5]. The amount of buffer time may take different contributory factors into account, which could perturb the schedules of handlers, airline, airport origin, destination, and flight. The size of the flight is normally related to the travel distance, longer distances presenting a higher probability of disturbances, which may accumulate generating a higher level of delays. Shorter distances present less chance of disturbances. The location of the origin and destination airports has a direct effect on potential disruptions, given that they place a constraint on the permitted routes possible and certain circumstances applicable to them, such as weather patterns during summer and winter or on the equator. Whereas the buffer time approach considers a predefined period of time, where its reduction is considered as a measure of robustness, the “idle time” refers to the time between two consecutive assignments to the same BSS, from the end time of one activity to the base starting time of the following assignment, also called gap. Other approaches for improving the robustness make use of the distribution of “idle time” and the reduction of the number of reassignments of the disrupted schedules.

Buffer time has been used in the scheduling of BSSs and airport gates, which may be applied between two consecutive flights to the same BSS in order to absorb small disturbances in the real system behaviour and was studied in [1, 6]. It was used in the ABSSAP in [2, 3, 5, 7] where it was used in the optimization of the aircraft turnaround process. The AGAP has some characteristics similar to the ABSSAP, and buffer times have been commonly considered for the AGAP in [7–11]. Yan and Huo [12] provided a sensitivity analysis for the AGAP buffer time, noting that the length of buffer time significantly influences the gate assignment process, so a reasonable minimum value should be used. Yan et al. [8] looked at the suitability of Flexible Buffer Times (FBTs) where, given low delays, short FBTs usually improve real-time objectives, such as the reassigning of an incoming aircraft at a minimum distance. Wei and Liu [13] showed the feasibility and effectiveness of using a fuzzy model in conjunction with fixed buffer times for the AGAPs. Ascó et al. [4] used buffer times to cope with small perturbations in

the ABSSAP, and several constructive algorithms were also studied. Wu and Caves [7, 14] showed the significance of a correct use of scheduled buffer time in maintaining schedule punctuality and performance by balancing the trade-offs between schedule punctuality and aircraft utilisation.

2. Problem under Consideration

The ABSSAP considered in this paper may be summarised as the assignment of available BSSs to flights which have already been scheduled. In the ABSSAP, 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, such as to airline preferences. This problem was described and studied in [2–5]. In this paper, we concentrate on finding solutions that comply with the constraints and take into account the objectives concentrating on the performance of different robustness approaches described in Section 3.

2.1. Overall Problem. Aircraft are usually parked at their allocated stand, around an airport terminal. Two layouts are considered in this paper which were presented in [2].

A list of some of the constants for this problem is presented in Table 1 and a list of the decision variables is shown in Table 2.

The aim is to find y_{ijp} values such that the objective function (3) is maximised, subject to the constraints expressed by (1) and (2) and the different robustness approaches studied in this paper which are presented in Section 3.

$$\sum_{i=0}^N y_{ijp} = P_j \quad \forall j \in \{1, \dots, M\}, \quad \forall p \in \{1, \dots, P_j\}. \quad (1)$$

For any pair of different flights where service times overlap, if the overlap in service times is greater than the maximum reduction allowed (B_{lq} for activity q of flight l), then both flight activities cannot be assigned to the same BSS. Thus, inequality (2) applies to any such pair of flights, j and l ($j \neq l$), where $t_{lq} < e_j \leq e_l$ and $(e_j - t_{lq}) > B_{lq}$.

$$y_{ijp} + y_{ilp} \leq 1. \quad (2)$$

TABLE 1: The constants for the model.

Constant	Description
N	Denotes the number of BSSs.
M	Denotes the number of flights.
P_j	The total number of activities to be serviced by baggage service stations for a given flight j , which also equates to the total number of sorting stations required to fully service flight j , $P_j > 0$.
i	Denotes a BSS where BSS 0 represents the dummy sorting station to which flights are assigned, if they cannot be assigned to a real BSS, $i \in \{0, \dots, N\}$.
j	Denotes a flight, $j \in \{1, \dots, M\}$.
T_j	Denotes the required service time for flight j (1 hour for short haul and 1.33 hours for long haul).
B_j	Denotes the desired buffer time for flight j —the time for which its BSS should be idle prior to this flight being serviced. B_j is considered to be 15 minutes for short haul and 30 minutes for long haul flights.
e_j	Denotes the end service time for flight j .
t_j	Denotes the target starting service time for flight j ($t_j = e_j - T_j - B_j$).
τ_j	Denotes the the base starting service time for flight j ($\tau_j = e_j - T_j$).

TABLE 2: List of the decision variables used in this ABSSAPs' model.

Name	Description
y_{ijp}	Specifies the assignment of flights to sorting stations. $y_{ijp} = 1$ if baggage sorting station $i \in [1, \dots, N]$ is allocated to flight $j \in [1, \dots, M]$ for $p \in [1, \dots, P_j]$ and 0 otherwise. If each flight only requires one activity, which means that each flight only requires one BSS, then this variable can be expressed as y_{ij} .
r_{jp}	Specifies the necessary reduction in service time for activity $p \in [1, \dots, P_j]$ of flight $j \in [1, \dots, M]$, given the service starting time allocated, s_{jp} .
s_{jp}	The service starting time allocated to activity $p \in [1, \dots, P_j]$ of flight $j \in [1, \dots, M]$ and given that a sorting station can only service one flight at a time. s_{jp} can be determined from r_{jp} since $s_{jp} = t_j - r_{jp}$.

Equation (1) states that each flight can be assigned either to exactly one real BSS or to the dummy sorting station. Inequality (2) states that flights cannot be assigned to the same real BSS if their service times overlap. If all flights only require one BSS, then the variables are reduced to y_{ij} , r_j , and s_j .

Although this is inherently a multiobjective problem, the importance of ensuring maximal assignment of flights to BSSs (top priority) and the relative importance of obtained robust solutions (second priority) allow these objectives to be combined into a single compound objective (3) with weights W_1 , W_2 , and W_3 chosen to implement these priorities.

$$\begin{aligned}
 & \underbrace{W_1 * \sum_{i=1}^N \sum_{j=1}^M \sum_{p=1}^{P_j} y_{ijp}}_{\text{assignment objective}} - \underbrace{W_2 * \text{Robustness}}_{\text{robustness objective}} \\
 & - \underbrace{W_3 * \sum_{j=1}^M \sum_{p=1}^{P_j} \left(C_j \cdot \sum_{i=1}^N y_{ijp} \cdot d_{ij} \right)}_{\text{distance objective}}.
 \end{aligned} \tag{3}$$

The first element in (3) aims to maximise the assignment of flights to BSSs, the second aims to increase the robustness, and the third aims to minimise some distance cost associated with the assignments, where C_j is a factor related to the amount of baggage for flight j (assumed to be 1 in all cases

for this paper) and d_{ij} is a measure of the distance or cost incurred from assigning BSS i to flight j . This aims to ensure that flights are allocated to appropriate BSSs.

The following two points were originally defined in [4] and will be observed to be useful later when defining the new suggested robustness approaches and when interpreting the results for the ABSSAP.

The *Lower Maximum Assignment Point* (LMAP) is the minimum number of resources required to service a certain number of activities when the service starting time (s_{jp}) coincides with the base starting service time (τ_j).

The *Upper Maximum Assignment Point* (UMAP) is the minimum number of resources required to service a certain number of activities when the service starting time (s_{jp}) coincides with the target starting service time (t_{jp}).

Two examples of both points are shown in Figure 2, where the maximum values of each diagram correspond to the LMAP and UMAP.

2.2. Robustness. A scheduled assignment is said to be in conflict if the completion of its service time is greater than the commencement of the service time of the next assignment at the same BSS. When a delayed flight affects the assignment of subsequent flights to the BSS, then there are two ways it can be corrected: either reassign the conflicting flight or reassign the subsequent flight to the conflicting one. When reassigning a conflicting flight or subsequent flights

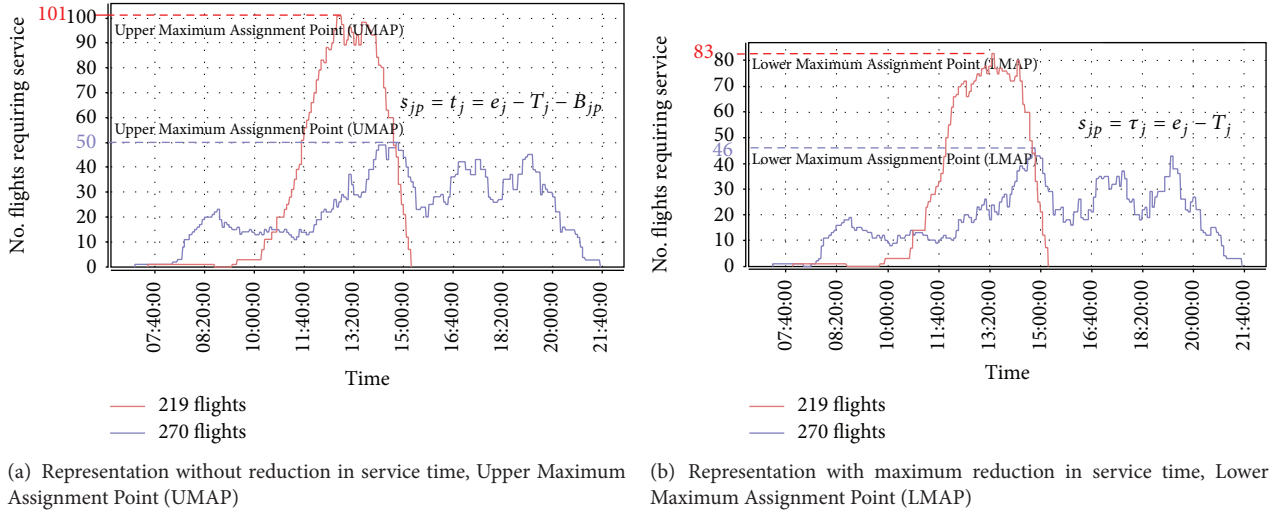


FIGURE 2: Flights service distribution for the two days' data sets obtained from the website of British Airports Authority (BAA).

to another BSS, a situation may arise where the reassigned flight is in conflict with the subsequent flight at the new BSS. Some reassignment may therefore have a downstream effect on the overall schedule, producing further conflicting flights requiring further reassignments, thus potentially increasing the problem difficulty later on.

In the ABSSAP, those flights which are late in arriving at their assigned stand are considered to be a perturbation, since baggage cannot be loaded into the aircraft at the scheduled time, and it needs to be held longer at the BSS. Any extension of an aircraft's stay on its assigned gate should not have an effect on its assigned BSS, as the baggage should have already been loaded into the aircraft, such that the BSS is free for use in its next assignment. This means that not all aircraft delays will affect their assigned BSS.

The main objective of this is to reduce the number of BSSs which have to be reassigned on the day the schedule is put into practice. In the ABSSAP, e_j is the end of the service time of flight j , τ_j is the base starting service time of flight j , and the variable y_{ij} has a value of 1 if flight j is assigned to BSS i , or y_{ij} is zero otherwise, as described in Section 2.1. A new decision variable x_{jk} is introduced with a value of 1 if flights j and k are assigned to the same BSS (i.e., if $y_{ij} = y_{ik} = 1$ for $i \in [1, \dots, N]$ and $j, k \in [1, \dots, M]$), or x_{jk} is 0 otherwise. On the day the schedule is implemented, the real times for the flights j and k correspond to $e'_j, \tau'_j, e'_k,$ and τ'_k for flights j and k , respectively, where $\tau'_j \geq \tau_j$ and $\tau'_k \geq \tau_k$, and the flights ordered by their base starting times, for $j < k$ and $j, k \in [1, \dots, M]$; then $\tau_j \leq \tau_k$. A conflict occurs when two flights j and k with $j < k$ and $j, k \in [1, \dots, M]$ originally assigned to the same BSS ($x_{jk} = 1$) have overlapping service times; that is, $e'_j > \tau'_k$ and $\tau'_j < e'_k$, as shown in Figure 3. Conflicts depend on the original assignments for the real arrival and departure flight times on the day of operation. It is therefore desirable to consider potential delays on the day of operation when generating the original flight assignments to BSSs, such

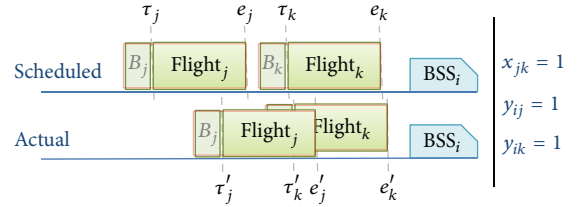


FIGURE 3: Example of conflict between two flights originally assigned to the same BSS.

that the final flight assignments differ little or do not differ at all from the original assignments on the day of operation. B_j refers to the buffer time originally assigned to the flight j , which may be absorbed by perturbations without raising the need for a reassignment. But overlapping, $\tau'_k < e'_j$, will deem the assignment infeasible and require reassigning it.

3. Robustness Approaches

In this section, some robustness approaches are presented and others are suggested. These are then studied and compared in the following section.

3.1. Minimise Reduction in Service Time. It may be possible to gain robustness by reordering assignments between BSSs so that “idle time” between flights consecutively assigned to the same BSS is greater, as shown in Figure 4.

Figure 4 shows two potential solutions with different robustness. While any delay to flight “a” in the “less robust” solution will certainly affect flight “b,” which will in turn have to be reassigned to another BSS, in the “more robust” solution a delay in flight “a” will not affect flight “b.” So the “more robust” solution is preferable to that of the “less robust” solution.

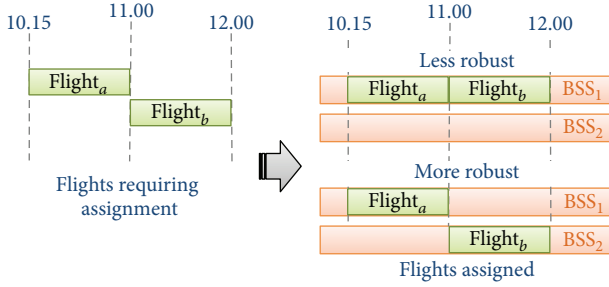


FIGURE 4: Simple example of two schedules, with the same flights, where one is obviously more robust in respect of perturbations than the other.

Mangoubi and Mathaisel [15] proposed the use of “buffer times” between two flights which are consecutively assigned to the same gate in order to obtain robust assignments, defining the reduction in service as that part of buffer time which overlaps with the previous assignment to the same gate. Given the detrimental effects that the reduction in service time has on the robustness of the assignment as against real-life delays, it is advisable to minimise the total reduction in service time, thus maximising buffer times. This objective,

$$u_{ijp} = \begin{cases} \arctan\left(\frac{\tau_j - e_k}{B_j}\right) - \frac{\Pi}{2} & \text{if } k < j, y_{ijp} = y_{ikq} = 1, \sum_{l=k+1}^j \sum_{p=1}^{P_l} y_{ilp} = 1, \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

$$\max \sum_{i=1}^N \sum_{j=1}^{M-1} \sum_{p=1}^{P_j} u_{ijp}. \quad (6)$$

Formula (5) defines the penalty u_{ijp} for assigning activity p of flight j to sorting station i , with the total penalty being represented by formula (6). There is only a penalty for consecutive assignments to the same sorting station, between the assignment and the previous assignment to the same sorting station. The reason for using arctangent is based on the properties of this function which imposes stronger penalties around the point at which the flights are assigned service time without any buffer time to the point at which all the buffer time is retained (by dividing the member of arctangent by B_k). The effect is reduced as the time separation between the assignments increases, but the contribution of any time separation is always considered negative. The constant $\pi/2$ is used so that there is always a cost associated with assigning two flights to the same sorting station; otherwise it would be a benefit. Considering the arctangent for each flight increases the individual effect on the overall objective, in contrast to using the arctangent of the sum of all of the flight contributions.

total reduction in service time (TRS), can be expressed by formula (4).

$$\min \sum_{j=1}^M \sum_{p=1}^{P_j} r_{jp}. \quad (4)$$

If a fitness function is defined as a weighted sum of the different objectives, as used here, and the robustness uses the reduction in service time only for assigned flights, then particular weights may compromise the importance of the main objective (maximisation of the number of assignments), where solutions with lower numbers of assignments are favoured over those with higher numbers of assignments because of the robustness objective.

The use of formula (4) treats any reduction in service equally, so it does not make a distinction between reducing all the assignment buffer time and allocating it to another flight from another solution where both flights share the available “idle time,” as shown in Figure 5. However, flight 3 with maximum reduction in service in solution “b” (Figure 5) will be unable to absorb any delay on the day of operation, although the same flight in the alternative solution “a” (Figure 5) will be able to do so, making it the preferable choice.

A nonlinear penalty function is required to take account of this, such as formulas (5) and (6).

The robustness objective represented by formula (6) also takes account of the objective of “minimising the service reduction,” as being the sum of the reduction in buffer time of each flight assigned.

3.2. Distribute Idle Time. Bolat [16] proposed the distribution of “idle time” uniformly amongst gates for the AGAP. In the case of the ABSSAP, the distribution of the “idle time” uniformly amongst BSSs can be considered, where “idle time” is calculated as the time between the start of service time on a flight’s assignment to BSS and the end of service time of the flight assigned immediately before the same baggage sorting station, as shown in formula (7). Whereas the buffer time implies preference for a particular gap size between consecutive assignments to the same sorting station, an “idle time” does not. When using the “idle time” it is normally intended that the gap size should be as large as possible. The reason for this is to increase the probability that, even with a delay, completion of service time for flight presently assigned

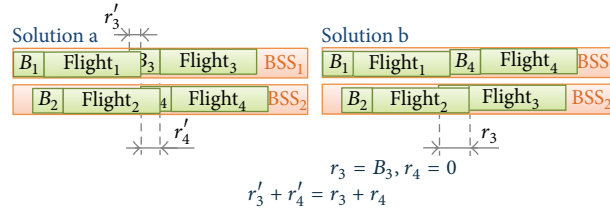


FIGURE 5: Simple example of a more robust schedule of four flights and two sorting stations using the reduction in buffer time.

will still be earlier than the start of service time for the next flight assigned to the same BSS.

$$\Delta t_j = (s_j - e_k) \quad (7)$$

for $j < k$, $y_{ijp} = y_{ikq} = 1$ where $\exists l \in [k, \dots, M]$ with $\tau_l < e_k$; $j, k \in [1, \dots, M]$, $p \in [1, \dots, P_j]$, $q \in [1, \dots, P_k]$, $i \in [1, \dots, N]$,

$$v^+ = \max \{ \Delta t_j \mid j \in [1, \dots, M] \}, \quad (8)$$

$$v^- = \min \{ \Delta t_j \mid j \in [1, \dots, M] \}, \quad (9)$$

$$\min (v^+ - v^-). \quad (10)$$

Formulas (8) and (9) refer to the maximum and minimum “idle time” for a solution, respectively, while formula (10) represents the objective as the difference between both the maximum (v^+) and the minimum (v^-) “idle times” (Δt_j , in (7)) for the same solution. Figure 6 shows a simple example of the robustness of two solutions, where solution “a” is more robust than solution “b.” In solution “a” the delay of flight 3 will not affect flight 4 and needs to be considerably larger to affect flight 5, whereas in solution “b” small delays in flight 3 will affect flight 4.

Formula (10) assumes that all flights within a solution have been assigned, which may not be the case in certain circumstances, where the maximum possible assignment is lower than a full assignment (simulation) or where the initial solution(s) has some flights remaining unassigned, and an example of this is shown in Figure 7(a). It should also be observed that this objective may conflict with the maximum assignment objective (first objective), as shown in Figure 7(b), where solution “f” has a smaller Δv compared to solution “e,” which is based on formula (10). This means that solution “f” is considered more robust, but solution “e” would be preferable because it achieves more assignments. Thus, it will be necessary to select the objective weights appropriately, where the fitness is a weighted sum of the different objectives, in order to ensure the correct selection of the solution.

Bolat [9, 17] extended the model by minimising the variance of the idle times.

$$\min \sqrt{\sum_{j=1}^M \left(\Delta t_j - \frac{\sum_{j=1}^M \Delta t_j}{M} \right)^2} \quad (11)$$

mean “idle time”

3.3. Reduce Reassignment on Disruption. The ability to reassign all flights directly affected by a disruption is desirable, without the need to reassign other flights. The intention here is to generate schedules which take account of this objective, allowing such reassignment to be performed more frequently.

One way to achieve this objective would be to count the number of assignments between which a reassignment could be placed when necessary. Whether the reassigned flights are on the same pier/side, as well as how many reassignments could be absorbed by a pair of assigned flights, must all be taken into account. Figure 8 shows the ability of the “idle time” between the two flights 1 and 8 to accommodate flights 3, 4, and 5, should one of them be delayed. Its reassignment to BSS₁ may be sufficient, thus avoiding transfer of any delay to other assignments.

The following model is proposed where the capacity to absorb reassignments may be achieved by weighting each reassignment by the Inverse of 1 plus the distance between the BSSs (d'_{in} , presented in Section 2.1), given that such distance (d'_{in}) may be zero, where all flights are ordered by their base start time (τ_j), as shown in (12) and (13).

The intention is to use the number of flights which could be reassigned between two flights already consecutively assigned to the same BSS, without the need to reassign either of these flights in order to achieve this. Equation (12) states that φ_{jkl} is equal to 1 if flight k 's base service duration does not overlap with the base service duration of both flights j and l ($j < l$), both of which are assigned to BSS _{i} , and there is

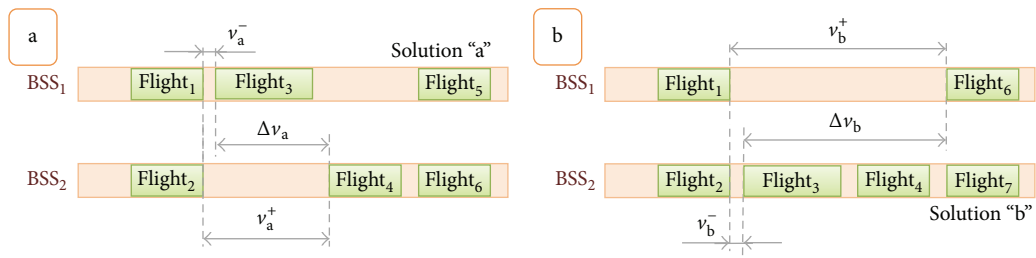
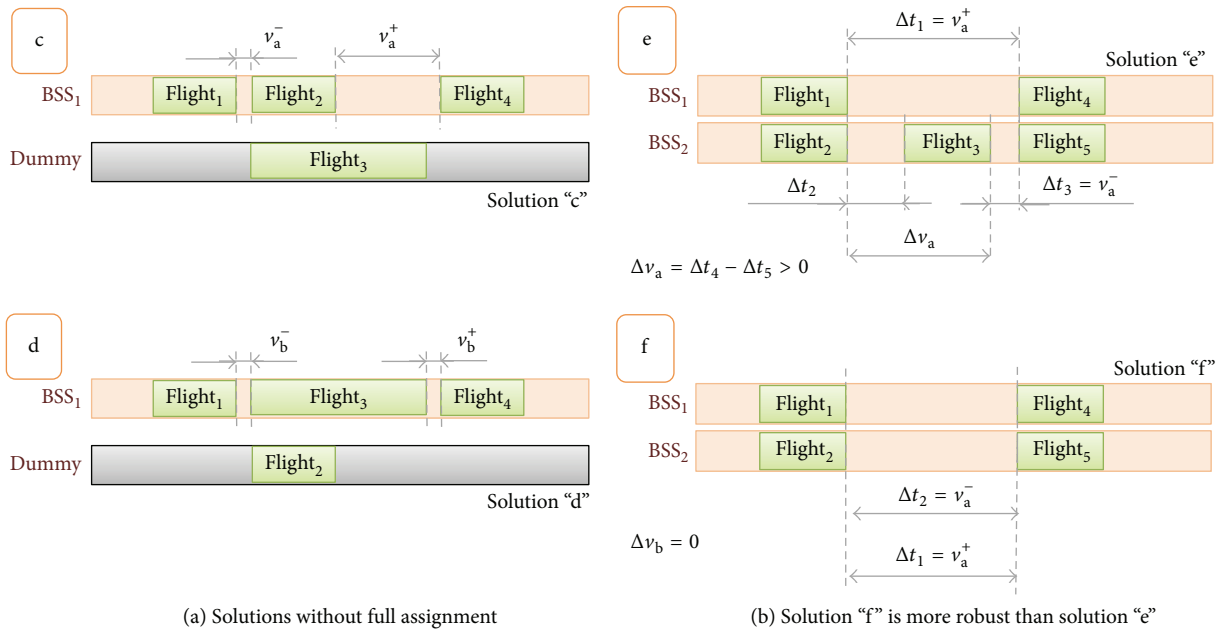


FIGURE 6: Examples of a range of idle times for different solutions, where solution “a” is clearly better than solution “b” for formula (10).



(a) Solutions without full assignment

(b) Solution “f” is more robust than solution “e”

FIGURE 7: Examples of a range of idle times for different solutions using formula (10).

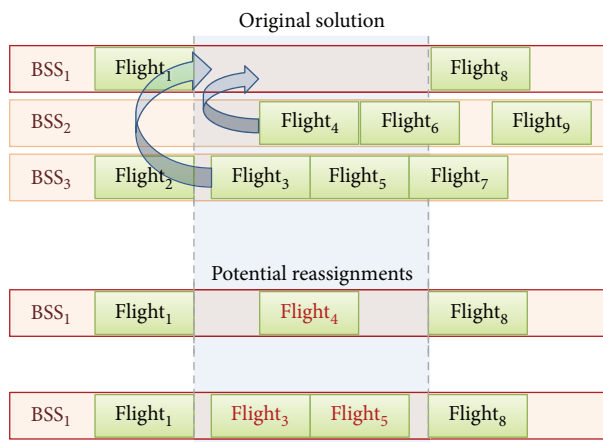


FIGURE 8: An example of the capacity to absorb reassignments.

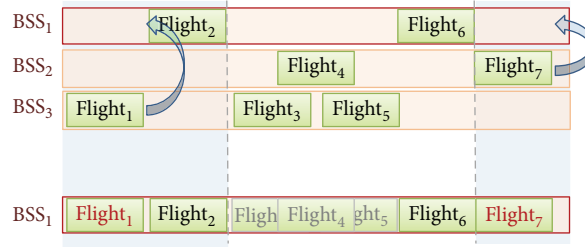


FIGURE 9: An example of border assignments in baggage sorting stations.

no other flight r between these ($j < r < l$) already assigned to the same BSS as flight j ; otherwise φ_{jkl} is zero, as shown in

$$\varphi_{jkl} = \begin{cases} 1 & \text{if } j < k < l, e_j \leq \tau_k, e_k \leq \tau_l \forall j, k, l \in [1, \dots, M] \nexists r \in (j, \dots, l), \\ & \text{with } y_{ijp} = y_{irq} = 1 \text{ for any } p \in [1, \dots, P_j], q \in [1, \dots, P_r], \\ & y_{ikz} = 0 \forall z \in [1, \dots, P_k], \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

The objective is to maximise formula (13), which weights the contribution of each potential reassignment, based on which BSS the reassigned flight was originally assigned to. The underlying idea is that closer reassignments are preferred to more distant ones, but the ability to reassign without affecting other assignments is preferable.

$$\sum_{i=1}^N \sum_{j=1}^M \sum_{p=1}^{P_j} \sum_{l=j+1}^M \sum_{q=1}^{P_l} \sum_{k=j+1}^{l-1} y_{ijp} * y_{ilq} * \varphi_{ikl} * \underbrace{\sum_{n \neq i}^N \sum_{z=1}^{P_k} \frac{y_{nkz}}{d_{in}' + 1}}_{\text{effect of the BSS assigned}}. \quad (13)$$

The representation in formula (13) also needs to include the border cases relating to the first and last assignments in BSS as shown in Figure 9.

The previous formula could be extended to cover the border cases by assigning two extra dummy flights to all available BSSs; first, $j = 0$, with the end time being the start of the time period studied, and the second flight, $j = M + 1$, with the start time being the completion of the time period studied, which for our time period would be $e_0 = 0$ and $(e_{M+1} - T_{M+1}) = 24 \text{ hr}$, with $y_{i0} = y_{i(M+1)} = 1 \forall i \in [1, \dots, N]$, as shown in

$$\sum_{i=1}^N \sum_{n \neq i}^N \left(\frac{\sum_{j=0}^M \sum_{p=1}^{P_j} \sum_{l=j+1}^{M+1} \sum_{q=1}^{P_l} \sum_{k=j+1}^{l-1} \sum_{z=1}^{P_k} y_{ijp} * y_{ilq} * y_{nkz} * \varphi_{ikl}}{d_{in}^{\text{BSSs}} + 1} \right). \quad (14)$$

3.4. Area of Reduction in Service (ARS). Bolat [16] examined the minimisation of the range of idle time and the difference between the maximum and minimum idle times for the AGAP, which was later extended in [9] to consider both minimisation of the idle time range and minimisation of the idle time variance (Section 3.2). However, these do not take account of the influence or effect which the disruptions have on the schedule due to the time of their occurrence, as all reductions in service time are treated as being the same, irrespective of the time period considered in the whole “planned schedule” for the given set of departures. It is anticipated that the more congested time periods in

the “planned schedule” will also represent periods where disruptions are more likely to occur and propagate, extending their effect and further increasing costs.

LMAP and UMAP values are an indication of the difficulty of the problem, and these may be obtained from the distribution of the number of flights requiring service over time, as shown in Figures 10 and 11.

Delay during a high flight density period is more likely to propagate given that fewer resources will be available to absorb any reassignment without repercussions on other flights. At the same time, it is these cases where it is most difficult to keep a sufficiently large gap between assignments

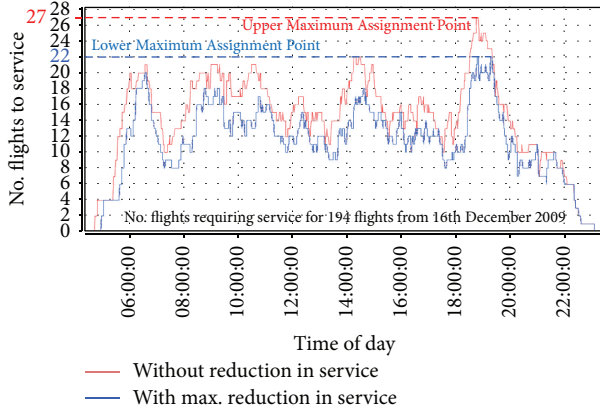


FIGURE 10: Flight distributions with LMAP and UMAP for 194 flights on 16th December 2009 at Terminal 1 of London Heathrow airport (identified as HIT091216).

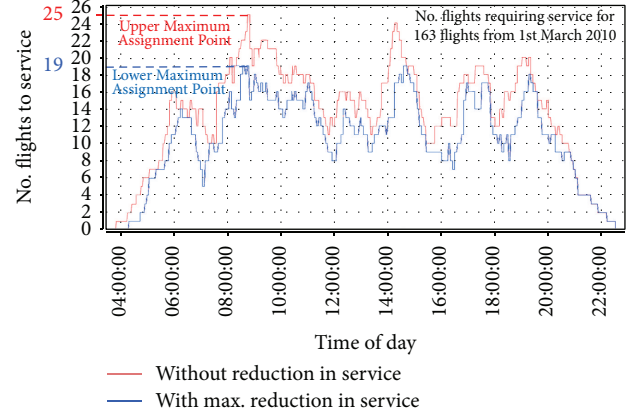


FIGURE 11: Flight distributions with LMAP and UMAP for 163 flights on 1st March 2010 at Terminal 1 of London Heathrow airport (identified as HIT100301).

to the same BSS. In assessing the importance of the time of day when service reduction is performed, it is suggested that it would be advisable to increase penalisation of flight assignments with reduced service time when there is a higher flight density. This can be accomplished by calculating the required number of BSSs at different times of the day, as shown in Figures 10 and 11.

Average Assignments Point. The function $f_u(t)$ refers to the distribution of flights over time with t_s being the schedule starting time and t_e being the schedule end time, as shown in Figure 12. The Average Assignment Point (AAP) is here defined as the number of BSSs for which the distribution of flights would be uniform, which can be calculated by

$$AAP = \frac{\int_{t_s}^{t_e} f_u(t) dt}{t_e - t_s}. \quad (15)$$

A_j is the density distribution area for the time period from the target service time of flight j (t_j) to the end of service time for the previous flight assigned to the same BSS; for example, in Figure 12 $A_{18} = \int_{t_{18}}^{e_9} f_u(t) dt$. AAP_j is here defined as the mean number of flights over the target start time for flight j and the end of service time for the previous flight assigned to the same BSS as flight j ; for example, $AAP_{18} = A_{18}/(e_9 - t_{18})$. So the contribution to the objective for assignment j is AAP_j/AAP times the reduction in service previously considered in Section 3.1. This corresponds to values greater than one for dense flight regions of the schedule and less than one for underused regions.

The approach proposed intends to penalise more those reductions in flight service time during time periods where more flights require servicing compared to those periods with less service load by means of the AAP and the distribution of flights over time, as shown in (16). The idea is that flights which require servicing during congested periods are more likely to have a knock-on effect compared to those in less congested periods. It is therefore preferable not to reduce the

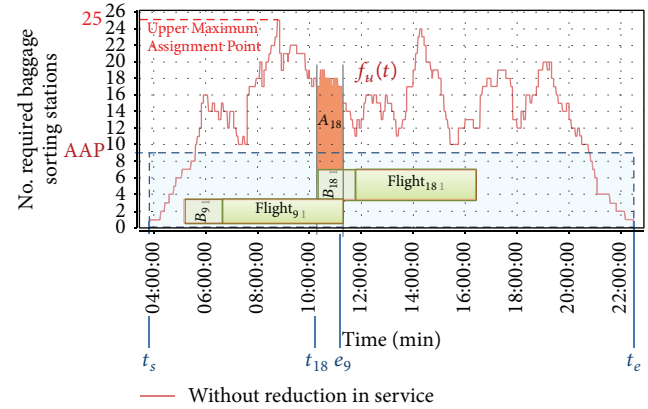


FIGURE 12: Distribution of flights over time and Area Reduction in Service Objective.

service time of flights at more congested times so much, in order to limit the effect of potential delays:

$$f_2 = -\frac{1}{AAP} * \sum_{i=1}^N \sum_{j=1}^M \sum_{p=1}^{P_j} y_{ijp} * A_j. \quad (16)$$

This approach is compared with some of the other approaches described here in Section 4.

3.5. Subarea of Reduction in Service (SARS). In this case, the objective is based on the area between both flight densities “without reduction in service time” and “with reduction in service time” shown in Figure 13, being here called a subarea. The flight density subarea corresponds to the area from the flight target start time (described in Section 2.1) to the previously assigned flight’s end service time, which lies between both flight density distributions (f_u and f_i), where a reduction in service time is not permitted and when all

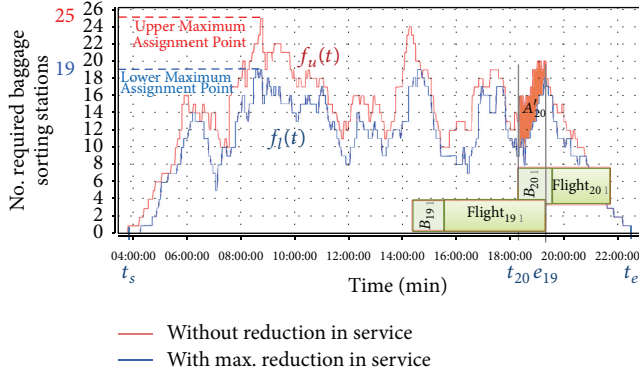


FIGURE 13: Subarea Reduction in Service Objective.

the buffer time has been reduced (considering only the base service duration, T).

$$A'_j = \int_{t_j}^e (f_u(t) - f_l(t)) dt. \quad (17)$$

The approach now proposed has a fitness which covers the subarea divided by the difference between the UMAP and the LMAP for all the assigned flights.

$$f_2 = - \sum_{i=1}^N \sum_{j=1}^M \sum_{p=1}^{P_j} y_{ijp} * \frac{A'_j}{\text{UMAP} - \text{LMAP}}. \quad (18)$$

Similarly to the Area Reduction in Service (Section 3.4), the main idea is to penalise more heavily those reductions in flight service time which occur in regions with high flight density, as these are more likely to further disrupt the schedule in case of delays. Adding both approaches together with the TRS will be seen in Section 4 to increase the robustness as compared to using each approach individually.

Nevertheless, in cases where the LMAP is equal to the UMAP, the SARS approach cannot be used; an example is shown in Figure 14. In these cases, the area between both flight densities could be used, an approach that is here named Base Subarea Reduction in Service (BSARS). As will be seen in Section 4.1, this approach significantly increases the robustness and widens the range of the quantity of BSSs, in which it performs better when compared to the ARS, TRS, and SARS.

These approaches are compared with some of the other approaches introduced in this paper in Section 4.

3.6. Unsupervised Estimated Stochastic Reduction in Service (UESRS). Lim and Wang [18] proposed a stochastic programming model for the AGAP with a robustness cost of conflicts, which is estimated by a function, $v(j, k)$. Flights are ordered by their base starting service time, so the gap between two flights j and k , $l(j, k)$, assigned to the same BSS, where $j < k$ and $j, k \in [1, \dots, M]$, is the difference between flight k 's target service time and the prior assigned flight j 's end service time (19), where $l(j, k) = -r_j$ for $l(j, k) < 0$, as shown in Figure 15. $v(j, k)$ is used to estimate the mean

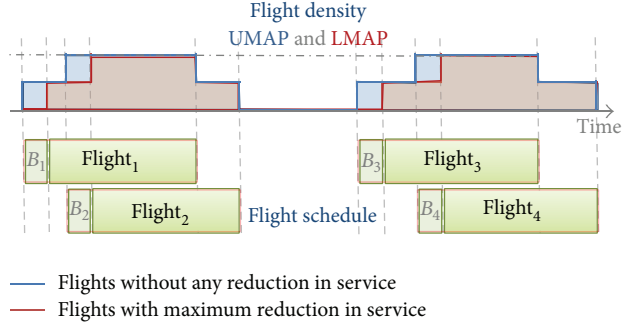
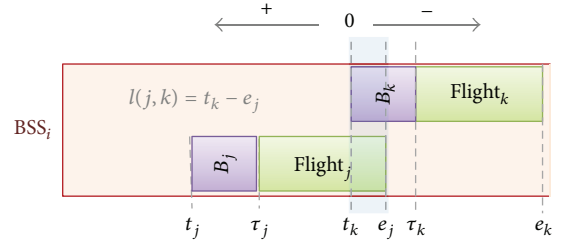


FIGURE 14: Example of distributions with the same LMAP and UMAP.

FIGURE 15: Overlap between two flights j and k assigned to the same BSS.

conflict probability between flights j and k assigned to the same BSS, which is a function of the gap $l(j, k)$, where larger gaps between assignments to the same BSSs result in lesser probability of real flight conflicts. $v(j, k)$ is normalised in (20).

$$l(j, k) = t_k - e_j \quad (t_k = \tau_k - B_k), \quad (19)$$

$$E(p(j, k)) = \sum_{i=1}^N \left(y_{ij} * y_{ik} * \frac{v(j, k) - v_{\min}(j, k)}{v_{\max}(j, k) - v_{\min}(j, k)} \right), \quad (20)$$

$$f_2 = - \sum_{j=1}^{M-1} \sum_{k=j+1}^M E(p(j, k)). \quad (21)$$

The definition of $v(j, k)$ comes from the application domain, in the absence of historical data; some unsupervised estimation functions were introduced in [18]. Figure 16 shows the penalty (y -axis) incurred for different unsupervised estimation functions as a function of the gap (x -axis). Negative values refer to reductions in service time between two assignments to the same BSS, which are heavily penalised as they may require reassignment should delays occur, whereas positive gaps are penalised less. Wider gaps between two assignments reduce the need to reassign delayed flights, given that the delay has to be larger than the gap in order to affect the following assignment to the same BSS. Similarly, starting the service earlier may not require the flight to be reassigned because the duration of earliness has to be lower than the gap in order to affect the previous assignment. Both earliness and delay probabilities decrease as the gap increases. Sufficiently large gaps may also be used on the day of operation by

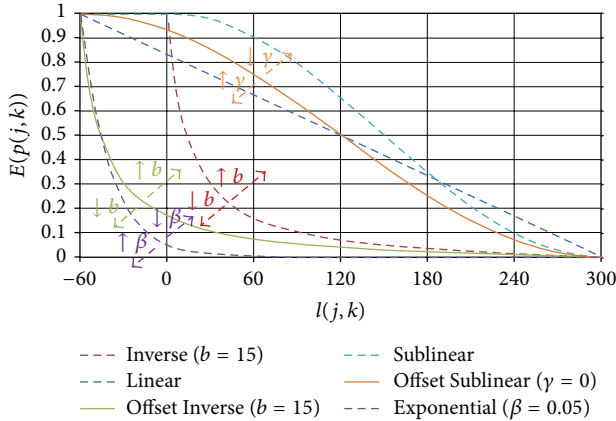


FIGURE 16: Penalty for different unsupervised estimation functions based on the gap between assignments.

disrupted flights which need to be reassigned, such that the detrimental effect of disruptions on that day is reduced.

The unsupervised estimation functions introduced in [18] are presented below and are shown in Figure 16:

- (1) Linear estimation is as follows.

$$v(j, k) = -l(j, k). \quad (22)$$

- (2) Exponential estimation is as follows.

$$v(j, k) = e^{-\beta * l(j, k)}. \quad (23)$$

- (3) Inverse estimation is as follows.

$$v(j, k) = \begin{cases} \frac{b}{l(j, k) + b} & \text{if } l(j, k) > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (24)$$

The value of the constant “ b ” changes the penalisation as shown in Figure 16, so a higher “ b ” increases the penalisation and a lower “ b ” decreases it. An appropriate value should be selected to properly weight the influence of the potential conflicts. Lim and Wang [18] used $b = 15$ minutes, which proved to provide rather poor results when compared with the exponential estimation function, which may partly be caused by the fixed cost when $l(j, k) < 0$ (dark red dash line, Figure 16), whereas in the exponential estimation function (purple dash line, Figure 16) this is not the case. The value used for “ b ” may be too great, and a lower value would make this estimation function provide values closer to those provided by the exponential estimation function which provided fitter solutions in the results presented in [18]. Consequently, a value $b = 6$ was seen in the experiments studied in Section 4.1 to provide better results than when $b = 15$. In general, an even lower value did appear to perform better in some instances but not as well as $b = 6$, as shown in Section 4.1.

The Inverse estimation function as considered in [18] treats all gaps smaller than the buffer time equally, which does not represent a real case since smaller gaps between flights are more likely to result in conflicts than larger ones on the day of operation. Given this and that the exponential estimation function performs best and treats all gaps differently, it is proposed that all of the gaps be treated differently, as shown by the modified version which is herein named “Offset inverse” (25), which is shown in Figure 16 for $b = 15$ (green line).

$$v(j, k) = \frac{b}{l(j, k) - \min\{l(j, k)\} + b}. \quad (25)$$

- (4) Sublinear estimation is as follows.

$$v(j, k) = \begin{cases} \cos\left(\frac{\pi * l(j, k)}{l_{\max}}\right) & \text{if } l(j, k) > 0, \\ 1 & \text{otherwise.} \end{cases} \quad (26)$$

This estimation also suffers from the same problem as the Inverse estimation and may be improved by offsetting its value so that the maximum penalisation corresponds to l_{\max} and the minimum corresponds to l_{\min} (27), which is shown in Figure 16 for $\gamma = 0$.

$$v(j, k) = \cos\left(\frac{\pi * (l(j, k) - l_{\min} + \gamma)}{l_{\max} - l_{\min} + \gamma}\right). \quad (27)$$

The gap definition used takes account of the buffer time, as the target service duration is the base service duration (T_k) plus the buffer time (B_k) for the flight. This makes the estimation functions dependent on the buffer time of each flight, as shown in Figure 17 for two buffer times of 30 and 15 min each. When the buffer time is the same irrespective of the flights, $B_k = B \forall k \in [1, \dots, M]$, as considered in [18], the cost is the same irrespective of the flight, depending only on the separation between consecutive flight assignments, but this is not the situation when the buffer time depends on the flight, namely, long, medium, or short distance flights, which are the cases studied here.

3.7. Reduction in the Number of Conflicts (RNC). A stochastic approach for improving schedule performance is described here, when disruptions occur on the day of operation. A similar approach was used in [19] where random delay scenarios are generated in the “planning stage” which are used to account for the potential disruptions in the schedule on the day of implementation by means of calculating the expected semideviation risk measure [20] for all those delay scenarios.

This approach is based on reducing the number of conflicts on the day of operation. Given that the real perturbed conditions will not be available until the day the schedule is implemented, these perturbed conditions are simulated by examining a set of perturbed base cases, S , which may be obtained in different ways, such as randomly, for example,

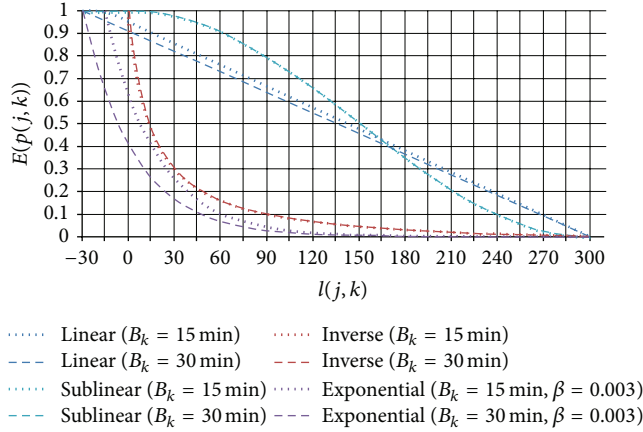


FIGURE 17: Penalty for different unsupervised estimation functions based on the gap between assignments for different buffer times.

from historical data or calculated using known distribution(s) from information available at the time of generating the assignments.

It is considered a set of perturbed schedules S , which simulate the perturbations on the day of operation. A new variable is introduced c_{js} , which for a given solution of assignments has the value of 1 if flight j is in conflict with another flight in the perturbed schedule $s \in S$ or zero otherwise. The average number of conflicts in the set of perturbed schedules S is calculated by (28), which is a measure of the solution robustness.

$$f_2 = \frac{-1}{|S|} * \sum_{s \in S} \sum_{j=1}^M c_{js}. \quad (28)$$

When reassigning conflicting flights or subsequent flights to other BSSs, a situation can arise where the reassigned flight interferes with the subsequent flight at the new BSS, a so-called secondary conflict. Some reassignment may therefore have a downstream effect on the overall schedule, producing more conflicting flights, in turn requiring further reassignments, thus potentially increasing the difficulty of the problem later on.

The above version considers all the conflicts to be of the same importance, but it is preferable to have conflicts which do not have repercussions later, that is, can be reassigned to another BSS without affecting any of the assignments already in existence. To account for this situation, a new variable c'_{js} is defined which takes the value of 1 if the reassignment of conflicting flight j in a perturbed schedule “ s ” affects other assignments already in existence or zero otherwise. The objective is presented as (29) where the constant, α , $0 \leq \alpha \leq 1$, denotes the importance of the conflicting flight repercussions on other assignments; $\alpha = 0$ corresponds to the case where no account is taken of any repercussion on other assignments, which corresponds in turn to (28), and $\alpha = 1$ corresponds to the cases in which both the conflicting flights and their repercussions on other assignments are considered to be of the same importance. $\alpha > 1$ refers to the cases where

more importance is given to the repercussions of a conflict on assignments other than the conflict itself.

$$f_2 = \frac{-1}{|S|} * \left(\sum_{s \in S} \sum_{j=1}^M (c_{js} + \alpha * c'_{js}) \right). \quad (29)$$

Calculation of the conflicts is time consuming and even more so if the effect of the conflict repercussions is also calculated, which is further aggravated by the need to use a large number of schedules in the perturbed set S in order to achieve a good representation of all the potential situations. The execution time is one of the disadvantages of using this approach as was experienced when executing the experiments presented in Section 4.

3.8. Probability of Conflict Based on the Gap (PCBG). The previous approach would normally require a large number of perturbed data sets, which makes its application very slow. Given that we are still interested in reducing the number of conflicts, but without the heavy cost in speed imposed by the RNC approach, then it is proposed to use the probability of having a conflict in a given “idle time” for each flight. This can be easily obtained if the delay distribution is known. In the ABSSAP, the early arrival of a flight does not normally affect the assignment to the Baggage Sorting Station Selection (BSSS) as this does not extend the time in which the BSS is required for servicing the flight, but transportation of the baggage already in the BSS to the flight may start earlier, so no earlier arrival is considered here, in the case of a normal folded distribution (no negative numbers and with zero mean) and independent delays for flights. This could also be extended to other distributions and to nonzero means.

Independent delays are considered initially, where a conflict between two consecutive flights is independent of other flights assigned to the same BSS. The assignments to different BSSs are independent from the point of view of conflicts. The probability of two consecutive flights having overlapping service times (a conflict) corresponds to the sum of the product between the probability of a sufficiently large delay on the part of the previous flight assigned to that BSS and the probability of the next consecutive assignment to the same BSS not being sufficiently delayed, as shown in Figure 18, where $\eta_j(t)$ is the probability density function for flight j and $t_0 = \tau_k - e_j$.

Equation (30) is the probability of a conflict, which is equal to the probability of exceeding the gap between both assignments, multiplied by the probability of the following assignment not being sufficiently delayed to avoid conflicts. $p(j, k)$ is the probability of a conflict between two flights j and k assigned consecutively to the same BSS with a gap between them of $t_0 = \tau_k - e_j$ given their respective probability density distribution of delay $\eta_j(t)$ and $\eta_k(t)$.

$$p(j, k) = \int_{\tau_k - e_j}^{\infty} \eta_j(t) * \left(1 - \int_0^{t - (\tau_k - e_j)} \eta_k(x) dx \right) dt. \quad (30)$$

A simplification of the conflict probability is represented in (31), which uses the “Riemann integral” approach for the range of delays between $(\tau_k - e_j)$ and four times the delay

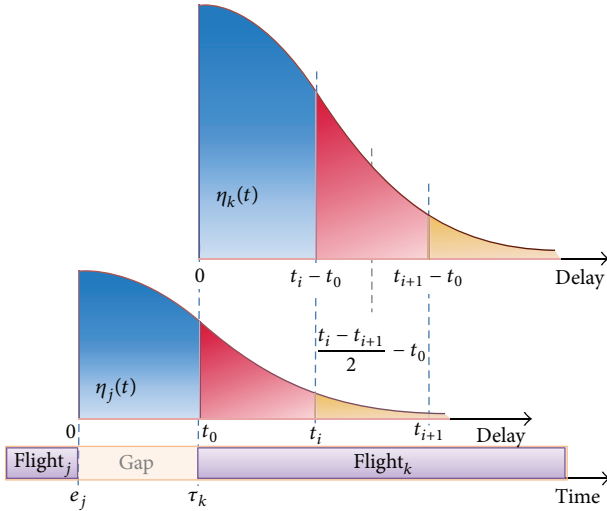


FIGURE 18: Probability of a conflict between two consecutive flights based on the intervening gap.

distribution standard deviation of flight j , σ_j , and a time increment of Δt .

$$p(j, k) = \sum_{i=0}^{(4*\sigma_j - (\tau_k - e_j))/\Delta t} \eta_j \left(\tau_k - e_j + \left(i + \frac{1}{2} \right) * \Delta t \right) * \left(1 - \sum_{l=0}^i \eta_k \left(\left(l + \frac{1}{2} \right) * \Delta t \right) * \Delta t \right) * \Delta t. \quad (31)$$

4. Results

This section looks at the performance from the point of view of the robustness by measuring the number of conflicts for a given set of perturbed schedules, for all the approaches presented in Section 3: firstly when they are used alone, secondly when combined with the TRS approach, and finally for the combinations with the approaches which consider the flight densities. The comparison made between the results obtained, when applying the different approaches, uses the Mann-Whitney test to establish the statistical significance of the different approaches, presented in the summary result tables as the number of instances in each range of the number of BSSs which can be said to have no statistically significantly higher numbers of conflicts compared to any of the other approaches compared. Regarding an airport, N referring to the number of BSSs available, three ranges of the number of BSSs have been defined, based on the LMAP and UMAP, where the first is for $N < \text{LMAP}$, the second is for $\text{LMAP} \leq N < \text{UMAP}$, and the third is for $\text{UMAP} \leq N$. These are shown within brackets and separated by a comma in the following tables. Any approach achieving full coverage of a range of the number of BSSs is presented in bold font and the approaches with higher numbers for a range, covering the highest number of BSSs in the range compared, are presented in underlined font to assist in the interpretation of the summary result tables.

The robustness approaches described are applied to the ABSSAP using the Steady State Evolutionary Algorithm (SSEA) from [3] and their results are compared and analysed in this section using the data sets obtained from NATS for London Heathrow airport Terminal 1, which were also used in [2, 3].

To compare the performance of each of the robustness approaches introduced in the previous sections, three sets of perturbed schedules were generated using a folded normal distribution with a zero mean and 10-, 20-, and 30-minute standard deviations. These sets are used to calculate the average number of conflicts for each robustness approach, where a lower value represents a more robust solution compared to those with higher values. Each experiment is repeated at least 30 times. To calculate the number of conflicts within a solution in the perturbed schedule, each flight in the original solution is assigned to the same BSS as in the original solution where possible; otherwise it is assigned to the dummy. This is repeated until all of the flights are assigned to BSS or the dummy. The number of flights assigned to the dummy represents the number of conflicts. If a solution does not achieve the maximum assignment possible, then the number of extra unassigned flights may be accounted for with a higher contribution, given that they are less desirable solutions. Nevertheless, all of the solutions having the highest fitness in the following experiments achieve maximum assignments, which simplifies comparison of the different robustness approaches, based on the described measure. This is possible since the measure only depends on assignments to the BSSs and not on any other objective, such as those upon which the fitness depends.

The number of possible perturbed schedules depends on the number of flights, and this accounts for an extremely large number of possible combinations, making it impossible to consider them all. The number of combinations for 194 flights corresponds to $194! \approx 1.3291 * 10^{361}$ which is far greater than 10^{82} atoms estimated to exist in the observable universe. The number of perturbed schedules necessary to calculate the quality of a solution should therefore be as large as possible to account for as many potential combinations as possible. However, as the number of perturbed schedules increases, so does the time required to perform the calculations, and the memory requirements also increase likewise: moreover, this is further multiplied by the number of solutions which will be used in the comparison. However, not all combinations are likely to represent a valid schedule. So, 10,000 perturbed schedules were used based on the number of solutions to be processed, the time available, and the memory required.

The stochastic robustness approach RNC is time consuming when compared with the other approaches reviewed. It would be desirable to use an approach which provides solutions closer to, or better than, those provided by the stochastic robustness approach, without the heavy cost of the time required. With the aim of assessing the difference in performance when the number of perturbed schedules is reduced, two sets of 1,000 and 25 perturbed schedules used with the RNC were also considered. Initial experiments were conducted using SSEA with $\ell = 1$ and the RNC

and Multiexchange between a Fixed Number of 3 Resources (MEFNR3) for 25 perturbed schedules and 800,000 total iterations which required an average execution time of 52 min per instance. These, when extrapolated to 1,000 perturbed schedules, provide an execution time of around 34 hours per instance, whereas the other approaches require no more than two minutes to complete the full set of iterations. Both applications of RNC required too long an execution time for the number of iterations specified, so they were run with a time limit of 30 min, as the other approaches required less than 2 min.

The objective importance presented in Section 2.1 is used in the following experiment where the most important objective is to achieve maximum assignment, with the second in importance being to maximise robustness and the third objective being minimisation of the distance between flights and their assigned BSS being the last objective considered. The fitness function used to guide the search in the SSEA is a weighted sum of the different objectives evaluated, introduced previously in Section 2.1, whose weights are those calculated in [3]. Both approaches UESRS and PCBG need their robustness weight (W_2) to be recalculated. Thus, given a maximum distance between a flight and its assigned BSS, D_{\max} , which depends on the airport topology, which for the topologies studied here is $D_{\max} = 9$, a distance of one unit is assumed between different sides of a pier and a distance of two units was assumed between different piers. A new assignment between two previously assigned flights may incur a service reduction for the new assignment and next flight, which is used to obtain the decrease in robustness (the second objective), which for UESRS and PCBG cannot be greater than 1 for each flight, thus totalling 2 in this case. Finally, using the objective priorities the following relations can be established: $W_1 * 1 > |W_2| * 2 + |W_3| * D_{\max}$ and $|W_2| * 2 > |W_3| * D_{\max}$. The original conditions for the weights, when the ‘‘minimise reduction in service time’’ objective was used (Section 2.1), are $W_3 = -1$ and $W_1 > 23.4$, which together with the objective priorities obtain $|W_2| > (1 * 9)/2 = 4.5$ and $|W_2| * 2 + |W_3| * 9 < 23.4$ giving $|W_2| < 14.4/2 = 7.2$. The value used for W_2 is -7.2 as $W_1 = 90 > 23.4$. The value used for W_1 is greater than the value originally used to calculate W_2 , so a value of -10 was also used.

The unsupervised estimation functions introduced in Section 3.6 were used for the same parameter values as those used in [18] and with Offset Inverse $b = 6$, Offset Inverse $b = 15$, Offset Sublinear $\gamma = 0$, and Offset Sublinear $\gamma = 1000$. A summary of the robustness approaches studied and their parameter values is shown in Table 3. In Table 3, the first column contains the name of the robustness approaches considered, all of which were introduced in Section 3; the second column shows the weights for each approach, and the subsequent columns show the name and value of the parameters for the corresponding robustness approach. The SSEA with $\ell = 1$, operator MEFNR3 from [2, 3], a population size of 10, the replacement strategy being Index Selection with Elitist Selection and a group size of 1 (ISIES), and a maximum of 800,000 iterations for two data sets from London Heathrow airport Terminal 1 (HIT091216: 16th December 2009 and HIT100301: 1st March 2010) were used to obtain the solutions for comparison. The fitness is the sum of the total number

TABLE 3: Robustness approaches used with their parameter values.

Approach	Weight (W_2)	Parameters Name: values
TRS	0.008	<i>Buffer time:</i> 30 min. long-haul and 15 min. others
ATRS	0.008	<i>Buffer time:</i> 30 min. long-haul and 15 min. others
ARS	0.008	<i>Buffer time:</i> 30 min. long-haul and 15 min. others
BSARS	0.008	<i>Buffer time:</i> 30 min. long-haul and 15 min. others
SARS	0.008	<i>Buffer time:</i> 30 min. long-haul and 15 min. others
PCBG	7.2 and 10	<i>Std. deviation:</i> 10, 20 and 30 min.
RNC	10 and 14	<i>Std. deviation:</i> 10, 20 and 30 min. <i>Num. schedules:</i> 25 and 1000 <i>Max. execution time:</i> 30 min.
UESRS	7.2 and 10	<i>Estimation function:</i> Exp 0.03, Exp 0.05, Inverse 6, Inverse 15, Linear, Offset Inverse 6, Offset Inverse 15, Offset Sublinear 0, Offset Sublinear 1000, and Sublinear <i>Buffer time:</i> 30 min. long-haul and 15 min. others

TABLE 4: Number of instances with significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with a delay $\sigma = 10$ min for ARS, BSARS, SARS + TRS, and TRS for a significance level of 0.05.

Approach	HIT091216 Max. (9, 5, 3)		HIT100301 Max. (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
ARS	(1, 2, 2)	(2, 3, <u>3</u>)	(3, 3, 4)	(1, 2, 1)
BSARS	(<u>9</u> , <u>5</u> , <u>3</u>)	(8, 2, <u>3</u>)	(4, <u>6</u> , <u>5</u>)	(4, <u>5</u> , <u>5</u>)
SARS + TRS	(3, 4, 2)	(3, <u>5</u> , <u>3</u>)	(3, 3, 4)	(0, 2, 1)
TRS	(0, 1, 1)	(1, 0, 0)	(2, 3, 4)	(1, 2, 0)

of assignments with weight of 90, less the distance between assigned flights and their assigned BSS with a weight of 1, less the robustness approach with the appropriate weight, all of which are shown in Table 3.

The results presented in the following sections were summarised for simplicity and clarity by considering the average number of times an approach achieves statistically significantly lower conflicts, or at least no worse, than the other approaches, in the different regions of numbers of BSSs, based on the LMAP and UMAP, which divide the range of BSSs studied into three areas corresponding to $N < \text{LMAP}$, $\text{LMAP} \leq N < \text{UMAP}$, and $\text{UMAP} \leq N$. The values between brackets correspond to the number of times the approach provides significantly statistically solutions not worse than the other approaches used, for each of the BSS ranges. Table 4 shows that the ARS approach for the data set of HIT091216 as well as a 4-pier topology has the values (2, 3, 3) showing that it achieves a statistically significant number of conflicts not

TABLE 5: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with a delay $\sigma = 10$ min and all the approaches alone and combined with TRS for a significance level of 0.05.

Approach	HIT091216		HIT100301	
	Max. (9, 5, 3)		Max. (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
BSARS	(2, 0, 0)	(1, 0, 0)	(0, 0, 0)	(3, 0, 0)
TRS + BSARS	(3, 0, 0)	(3, 0, 0)	(0, 0, 0)	(4, 0, 0)
PCBG				
7.2	(5, 2, 1)	(3, 1, 0)	(1, 1, 0)	(5, 2, 0)
10	(7, 4, 0)	(8, 1, 0)	(2, 4, 0)	(5, 4, 0)
TRS + UESRS				
Exp 0.03				
7.2	(2, 1, 0)	(0, 1, 0)	(2, 1, 1)	(1, 2, 0)
10	(2, 1, 0)	(1, 4, 3)	(4, 2, 4)	(2, 1, 1)
Inverse 6				
7.2	(0, 0, 2)	(0, 1, 0)	(0, 1, 1)	(0, 0, 0)
10	(0, 0, 2)	(0, 1, 2)	(0, 1, 4)	(0, 2, 4)

higher in two instances for the range of $N < \text{LMAP}$, three for the range $\text{LMAP} \leq N < \text{UMAP}$, and three for $N \geq \text{UMAP}$. Thus, the larger the number between parenthesis, the better the performances in respect of robustness.

Bold font is used to identify those cases where the robustness approach achieves good results for all numbers of BSSs in a range; for example, in the previous example the approach performs well for all numbers of BSSs in the range of $N \geq 27$ (UMAP). Underlining is used to identify those cases where the robustness approach performs well for greater numbers of BSSs in a range. The maximum quantity of instances of number of BSSs in a range is presented between brackets at the top of the table for each of the ranges discussed, preceded by the word ‘‘Max.’’; for example, in Table 4 the cell in the second column and second row of the header shows that the first range contains nine instances of numbers of BSSs for $N < \text{LMAP}$, five for $\text{LMAP} \leq N < \text{UMAP}$, and three for $N \geq \text{UMAP}$. For simplicity and clarity, those approaches which do achieve statistically significantly higher number of conflicts on average than any of the other approaches in all three BSSs regions are not shown in the tables which followed.

The next sections present the experimental results for the robustness approaches presented.

4.1. Results for All Robustness Approaches Considered. In this section, we look at the performance from the point of view of robustness for all the approaches presented in Section 3; first when they are used alone and then when combined with the TRS approach.

The BSARS provides more robust solutions than those obtained by the other proposed approaches, that is, TRS, ARS, SARS, and TRS + SARS, as shown in Table 4. Furthermore, BSARS will be seen to perform even better when combined with TRS in Tables 5, 6, and 7. ARS and SARS were similarly seen to improve robustness when combined with TRS.

The results which are summarised in Tables 5, 6, and 7 show that the PCBG does not gain any advantage when

TABLE 6: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with a delay $\sigma = 20$ min and all the approaches alone and combined with TRS for a significance level of 0.05.

Approach	HIT091216		HIT1003010	
	Max. (9, 5, 3)		Max. (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
BSARS	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(2, 0, 0)
TRS + BSARS	(0, 0, 0)	(4, 0, 0)	(0, 0, 0)	(3, 0, 0)
PCBG				
7.2	(5, 1, 0)	(2, 0, 0)	(3, 2, 0)	(3, 0, 0)
10	(7, 2, 1)	(8, 5, 1)	(1, 5, 1)	(5, 6, 1)
TRS + UESRS				
Exp 0.03				
7.2	(2, 1, 0)	(1, 0, 0)	(2, 1, 0)	(1, 0, 0)
10	(5, 3, 1)	(2, 0, 3)	(5, 1, 4)	(4, 0, 1)
Inverse 6				
10	(1, 1, 3)	(0, 2, 2)	(0, 1, 2)	(2, 2, 5)

TABLE 7: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with a delay $\sigma = 30$ min and all the approaches alone and combined with TRS for a significance level of 0.05.

Approach	HIT091216		HIT100301	
	Max. (9, 5, 3)		Max. (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
BSARS	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(1, 0, 0)
TRS + BSARS	(0, 0, 0)	(2, 0, 0)	(0, 0, 0)	(2, 0, 0)
PCBG				
7.2	(4, 2, 0)	(0, 0, 0)	(1, 4, 0)	(1, 0, 0)
10	(1, 4, 3)	(5, 4, 3)	(2, 6, 5)	(3, 6, 5)
TRS + UESRS				
Exp 0.03				
7.2	(3, 0, 0)	(1, 0, 0)	(3, 0, 0)	(1, 0, 0)
10	(9, 1, 0)	(4, 1, 0)	(5, 1, 0)	(2, 0, 0)
Inverse 6				
7.2	(1, 0, 0)	(1, 0, 0)	(1, 0, 0)	(2, 0, 0)
10	(3, 3, 1)	(2, 3, 0)	(0, 0, 0)	(3, 0, 0)

combined with the TRS, but it appears to be detrimental, as PCBG alone performs better throughout the whole ranges of the numbers of BSSs compared to when used combined with TRS. Even with the RNC for 25 and 1,000 instances taking account of the distribution of delays, the other approaches provide a statistically significantly lower number of collisions, but with a much lower running time. This could be due to the fact that the number of iterations which it is possible to execute in the 30 minutes is too low to find promising solutions with a lower number of collisions compared to those obtained by the other approaches. Around 33,000 to 70,000 iterations were executed for the experiments conducted here (the numbers depending mainly on the data set and the number of BSSs).

TABLE 8: Terminal 1 with algorithm SSEA1, operator MEFNR3, and population size of 2,000 for perturbations of 0 min average and $\sigma = 10$ min deviation, Mann-Whitney test, and significance level of 0.05 and for the TRS, UESRS, and the combination of TRS with UESRS.

Approach	H091216 17 (9, 5, 3)		H100301 17 (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
0.008 × TRS + 10 × UESRS (Exp 0.03)	<u>14</u> (<u>9</u> , 4, 1)	<u>15</u> (<u>7</u> , <u>5</u> , <u>3</u>)	<u>14</u> (<u>6</u> , <u>5</u> , <u>3</u>)	<u>11</u> (<u>6</u> , 4, 1)
0.008 × TRS + 10 × UESRS (Inverse 2)	1 (0, 0, 1)	2 (0, 2, 0)	1 (0, 0, 1)	0 (0, 0, 0)
0.008 × TRS + 10 × UESRS (Inverse 4)	2 (0, 0, <u>2</u>)	3 (0, 2, 1)	3 (0, 1, 2)	6 (0, 2, <u>4</u>)
0.008 × TRS + 10 × UESRS (Inverse 6)	3 (0, 1, <u>2</u>)	4 (0, 2, 2)	4 (0, 1, <u>3</u>)	6 (0, 2, <u>4</u>)
0.008 × TRS + 7.2 × UESRS (Exp 0.03)	8 (7, 1, 0)	5 (4, 1, 0)	9 (4, 4, 1)	8 (4, <u>4</u> , 0)
0.008 × TRS + 7.2 × UESRS (Inverse 2)	1 (0, 0, 1)	1 (0, 1, 0)	0 (0, 0, 0)	0 (0, 0, 0)
0.008 × TRS + 7.2 × UESRS (Inverse 4)	1 (0, 0, 1)	1 (0, 1, 0)	2 (0, 1, 1)	1 (0, 1, 0)
0.008 × TRS + 7.2 × UESRS (Inverse 6)	2 (0, 0, <u>2</u>)	1 (0, 1, 0)	2 (0, 1, 1)	0 (0, 0, 0)
10 × UESRS (Exp 0.03)	0 (0, 0, 0)	1 (1, 0, 0)	2 (0, 0, 2)	0 (0, 0, 0)

TABLE 9: Terminal 1 with algorithm SSEA1, operator MEFNR3, and population size of 2,000 for perturbations of 0 min average and $\sigma = 20$ min deviation, Mann-Whitney test, and significance level of 0.05 and for the TRS, UESRS, and the combination of TRS with UESRS.

Approach	H091216 Max. 17 (9, 5, 3)		H100301 Max. 17 (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
0.008 × TRS + 10 × UESRS (Exp 0.03)	<u>12</u> (<u>9</u> , 3, 0)	<u>13</u> (<u>7</u> , 3, <u>3</u>)	<u>14</u> (<u>6</u> , <u>4</u> , <u>4</u>)	9 (5, 3, 1)
0.008 × TRS + 10 × UESRS (Inverse 2)	2 (2, 0, 0)	4 (4, 0, 0)	0 (0, 0, 0)	2 (2, 0, 0)
0.008 × TRS + 10 × UESRS (Inverse 4)	5 (2, 1, 2)	5 (2, 3, 0)	1 (0, 0, 1)	5 (2, 1, 2)
0.008 × TRS + 10 × UESRS (Inverse 6)	5 (0, 2, <u>3</u>)	10 (3, <u>5</u> , 2)	4 (0, 1, 3)	<u>12</u> (3, <u>4</u> , <u>5</u>)
0.008 × TRS + 7.2 × UESRS (Exp 0.03)	4 (3, 1, 0)	3 (3, 0, 0)	4 (3, 1, 0)	4 (2, 2, 0)
0.008 × TRS + 7.2 × UESRS (Inverse 2)	1 (0, 1, 0)	1 (1, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)
0.008 × TRS + 7.2 × UESRS (Inverse 4)	1 (1, 0, 0)	1 (1, 0, 0)	1 (0, 1, 0)	0 (0, 0, 0)
0.008 × TRS + 7.2 × UESRS (Inverse 6)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 1, 0)	1 (0, 1, 0)
10 × UESRS (Exp 0.03)	0 (0, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)

The PCBG provides statistically significantly lower number of conflicts through a wider range of BSSs, and such a range also includes the range of BSSs used in the real problems; that is, $N \geq \text{UMAP}$. Nevertheless, this result could be regarded as biased, given that the PCBG considers a normal folded distribution of the same standard deviation as that from which the perturbed schedules were generated. The PCBG could consider different standard deviations and distributions depending on the aircraft, season, route, destination, and time of the day which should further improve the results in real situations. The running time is also of the same magnitude as that for the other approaches, with the exception of the RNC, which has a much higher running time for the same number of iterations.

There is a possibility of combining the different approaches, in particular either ARS or BSARS with UESRS studied below. There is also a question about the preference as to how these approaches are combined, either as a sum of each individual with the TRS, as studied in here, or as a product of their individual contributions.

4.2. Other Combinations of These Approaches. In this section we study the combination of the UESRS with each of the ARS, BSARS, and TRS. It may be noted that the SARS is equal to

BSARS divided by (UMAP-LMAP), which it is equivalent to thus changing the weight (W_2) in the fitness function. As seen previously in Table 4, BSARS provides more robust solutions compared to when using SARS alone for the same weight (W_2) or when combined with TRS, and these differences between both similar approaches are even more pronounced when BSARS is used in combination with TRS.

The combination of TRS and UESRS with exponential estimation function overall performs better for $N < \text{LMAP}$, as shown in Tables 8, 9, and 10, which correspond to the situation when there are no sufficient resources for the number of flights to service. The results which are summarised in Tables 11, 12, and 13 and which correspond to the cases when two robustness approaches are combined show that the combination of ARS with UESRS for exponential estimation function overall performs better for $\text{LMAP} \leq N < \text{UMAP}$, which refers to the situation when there are just sufficient resources for the number of flights to service, as, in this case, it is particularly important to consider the distribution of the flights requiring service which ARS takes into account. Also the combination of UESRS with BSARS in general improves the performance more than when using ARS for all the ranges. Similarly, the combination of BSARS with UESRS with Inverse estimation function overall performs

TABLE 10: Terminal 1 with algorithm SSEA1, operator MEFNR3, and population size of 2,000 for perturbations of 0 min average and $\sigma = 30$ min deviation, Mann-Whitney test, and significance level of 0.05 and for the TRS, UESRS, and the combination of TRS with UESRS.

Approach	H091216		H100301	
	Max. 17 (9, 5, 3)		Max. 17 (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
0.008 × TRS + 10 × UESRS (Exp 0.03)	<u>13</u> (<u>9</u> , 3, 1)	<u>12</u> (<u>6</u> , 3, <u>3</u>)	<u>15</u> (<u>6</u> , <u>5</u> , <u>4</u>)	6 (3, 2, 1)
0.008 × TRS + 10 × UESRS (Inverse 2)	5 (5, 0, 0)	3 (3, 0, 0)	0 (0, 0, 0)	4 (<u>4</u> , 0, 0)
0.008 × TRS + 10 × UESRS (Inverse 4)	7 (5, 1, 1)	8 (5, 3, 0)	3 (1, 1, 1)	3 (3, 0, 0)
0.008 × TRS + 10 × UESRS (Inverse 6)	9 (3, <u>3</u> , <u>3</u>)	11 (5, <u>5</u> , 1)	10 (2, <u>5</u> , 3)	<u>12</u> (3, <u>4</u> , <u>5</u>)
0.008 × TRS + 7.2 × UESRS (Exp 0.03)	3 (3, 0, 0)	2 (2, 0, 0)	3 (3, 0, 0)	2 (1, 1, 0)
0.008 × TRS + 7.2 × UESRS (Inverse 2)	1 (1, 0, 0)	3 (3, 0, 0)	0 (0, 0, 0)	2 (2, 0, 0)
0.008 × TRS + 7.2 × UESRS (Inverse 4)	3 (3, 0, 0)	2 (2, 0, 0)	1 (0, 1, 0)	1 (1, 0, 0)
0.008 × TRS + 7.2 × UESRS (Inverse 6)	1 (1, 0, 0)	2 (2, 0, 0)	3 (2, 1, 0)	2 (2, 0, 0)
10 × UESRS (Exp 0.03)	0 (0, 0, 0)	0 (0, 0, 0)	2 (1, 0, 1)	0 (0, 0, 0)

TABLE 11: Conflicts statistical significance for combined robustness approaches for Terminal 1 with algorithm SSEA1, operator MEFNR3, and population size of 2,000 for perturbations of 0 min average and $\sigma = 10$ min deviation, Mann-Whitney test, and significance level of 0.05.

Approach	H091216		H100301	
	Max. 17 (9, 5, 3)		Max. 17 (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
0.008 × ARS + 10 × UESRS (Exp 0.03)	3 (1, 2, 0)	5 (0, <u>3</u> , 2)	4 (0, 1, 3)	3 (0, 2, 1)
0.008 × ARS + 10 × UESRS (Inverse 6)	1 (0, 0, 1)	4 (0, 1, <u>3</u>)	3 (0, 1, 2)	<u>6</u> (0, 1, <u>5</u>)
0.008 × ARS + 7.2 × UESRS (Inverse 6)	3 (0, 2, 1)	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)
0.008 × BSARS + 10 × UESRS (Exp 0.03)	<u>7</u> (4, 2, 1)	<u>6</u> (0, <u>3</u> , <u>3</u>)	3 (0, 0, 3)	<u>6</u> (3, 2, 1)
0.008 × BSARS + 10 × UESRS (Inverse 6)	<u>7</u> (1, <u>3</u> , <u>3</u>)	5 (0, 2, <u>3</u>)	<u>6</u> (0, <u>2</u> , <u>4</u>)	<u>6</u> (0, 2, <u>4</u>)
0.008 × BSARS + 7.2 × UESRS (Exp 0.03)	3 (2, 1, 0)	2 (2, 0, 0)	1 (0, 0, 1)	5 (<u>4</u> , 1, 0)
0.008 × BSARS + 7.2 × UESRS (Inverse 6)	5 (1, 2, 2)	0 (0, 0, 0)	2 (0, 1, 1)	0 (0, 0, 0)
0.008 × TRS + 0.008 × BSARS	0 (0, 0, 0)	2 (2, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)
0.008 × TRS + 10 × UESRS (Exp 0.03)	0 (0, 0, 0)	2 (0, 0, 2)	4 (<u>2</u> , 0, 2)	1 (0, 0, 1)
0.008 × TRS + 10 × UESRS (Inverse 2)	1 (0, 0, 1)	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)
0.008 × TRS + 10 × UESRS (Inverse 4)	1 (0, 0, 1)	0 (0, 0, 0)	2 (0, 0, 2)	4 (0, 1, 3)
0.008 × TRS + 10 × UESRS (Inverse 6)	2 (0, 0, 2)	1 (0, 0, 1)	3 (0, 0, 3)	5 (0, 1, 4)
0.008 × TRS + 7.2 × PCBG (10)	6 (<u>6</u> , 0, 0)	2 (2, 0, 0)	4 (<u>2</u> , 2, 0)	4 (2, 2, 0)
0.008 × TRS + 7.2 × UESRS (Inverse 4)	1 (0, 0, 1)	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)
0.008 × TRS + 7.2 × UESRS (Inverse 6)	1 (0, 0, 1)	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)
10 × PCBG (10)	5 (4, 1, 0)	5 (<u>5</u> , 0, 0)	6 (<u>2</u> , <u>4</u> , 0)	5 (2, <u>3</u> , 0)
7.2 × PCBG (10)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 1, 0)	3 (1, 2, 0)

TABLE 12: Conflicts statistical significance for combined robustness approaches for Terminal 1 with algorithm SSEA1, operator MEFNR3, and population size of 2,000 for perturbations of 0 min average and $\sigma = 20$ min deviation, Mann-Whitney test, and significance level of 0.05.

Approach	H091216		H100301	
	Max. 17 (9, 5, 3)		Max. 17 (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
0.008 × ARS + 10 × UESRS (Exp 0.03)	3 (1, 2, 0)	0 (0, 0, 0)	3 (0, 1, 2)	1 (0, 0, 1)
0.008 × ARS + 10 × UESRS (Inverse 6)	3 (0, 1, 2)	0 (0, 0, 0)	3 (0, 0, 3)	7 (0, 2, <u>5</u>)
0.008 × BSARS + 10 × UESRS (Exp 0.03)	7 (3, 2, 2)	0 (0, 0, 0)	5 (0, 1, <u>4</u>)	3 (<u>3</u> , 0, 0)
0.008 × BSARS + 10 × UESRS (Inverse 6)	<u>10</u> (3, <u>4</u> , <u>3</u>)	0 (0, 0, 0)	6 (0, 3, 3)	<u>9</u> (2, 3, 4)
0.008 × BSARS + 7.2 × UESRS (Exp 0.03)	3 (2, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	3 (<u>3</u> , 0, 0)
0.008 × BSARS + 7.2 × UESRS (Inverse 6)	2 (2, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)
0.008 × TRS + 0.008 × ARS	0 (0, 0, 0)	<u>17</u> (<u>9</u> , <u>5</u> , <u>3</u>)	0 (0, 0, 0)	0 (0, 0, 0)
0.008 × TRS + 10 × UESRS (Exp 0.03)	0 (0, 0, 0)	0 (0, 0, 0)	2 (1, 0, 1)	1 (0, 0, 1)
0.008 × TRS + 10 × UESRS (Inverse 6)	1 (0, 0, 1)	0 (0, 0, 0)	2 (0, 0, 2)	5 (0, 1, 4)
0.008 × TRS + 7.2 × PCBG (20)	7 (<u>7</u> , 0, 0)	0 (0, 0, 0)	<u>8</u> (<u>4</u> , <u>4</u> , 0)	3 (2, 1, 0)
10 × PCBG (20)	4 (3, 1, 0)	0 (0, 0, 0)	6 (1, <u>4</u> , 1)	<u>9</u> (2, <u>6</u> , 1)

TABLE 13: Conflicts statistical significance for combined robustness approaches for Terminal 1 with algorithm SSEA1, operator MEFNR3, and population size of 2,000 for perturbations of 0 min average and $\sigma = 30$ min deviation, Mann-Whitney test, and significance level of 0.05.

Approach	H091216		H100301	
	Max. 17 (9, 5, 3)		Max. 17 (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
$0.008 \times \text{ARS} + 10 \times \text{UESRS}$ (Exp 0.03)	3 (2, 1, 0)	1 (0, 1, 0)	3 (2, 1, 0)	0 (0, 0, 0)
$0.008 \times \text{ARS} + 10 \times \text{UESRS}$ (Inverse 6)	2 (0, 1, 1)	1 (0, 1, 0)	0 (0, 0, 0)	1 (1, 0, 0)
$0.008 \times \text{ARS} + 7.2 \times \text{UESRS}$ (Exp 0.03)	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
$0.008 \times \text{ARS} + 7.2 \times \text{UESRS}$ (Inverse 6)	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)
$0.008 \times \text{BSARS} + 10 \times \text{UESRS}$ (Exp 0.03)	6 (3, 2, 1)	0 (0, 0, 0)	4 (2, 1, 1)	4 (<u>3</u> , 1, 0)
$0.008 \times \text{BSARS} + 10 \times \text{UESRS}$ (Inverse 6)	11 (6, 4, 1)	1 (0, 1, 0)	3 (1, 2, 0)	5 (<u>3</u> , 2, 0)
$0.008 \times \text{BSARS} + 7.2 \times \text{UESRS}$ (Exp 0.03)	2 (2, 0, 0)	4 (4, 0, 0)	1 (1, 0, 0)	3 (<u>3</u> , 0, 0)
$0.008 \times \text{BSARS} + 7.2 \times \text{UESRS}$ (Inverse 6)	2 (2, 0, 0)	5 (<u>5</u> , 0, 0)	1 (1, 0, 0)	2 (2, 0, 0)
$0.008 \times \text{TRS} + 10 \times \text{UESRS}$ (Exp 0.03)	1 (1, 0, 0)	1 (1, 0, 0)	3 (3, 0, 0)	1 (1, 0, 0)
$0.008 \times \text{TRS} + 10 \times \text{UESRS}$ (Inverse 6)	3 (1, 1, 1)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
$0.008 \times \text{TRS} + 7.2 \times \text{PCBG}$ (30)	<u>13</u> (<u>7</u> , <u>5</u> , 1)	5 (3, 2, 0)	<u>11</u> (<u>5</u> , <u>6</u> , 0)	5 (2, 3, 0)
$0.008 \times \text{TRS} + 7.2 \times \text{UESRS}$ (Exp 0.03)	2 (2, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
$10 \times \text{PCBG}$ (30)	5 (0, 2, <u>3</u>)	<u>11</u> (4, <u>4</u> , <u>3</u>)	9 (2, 2, <u>5</u>)	<u>13</u> (2, <u>6</u> , 5)

better for $\text{UMAP} \leq N$ and also for N lower than UMAP but closer to it. This would be the preferable approach for real problems. Finally, the ARS with UESRS with Inverse estimation function performance decreases for big delays for $N \geq \text{UMAP}$ and improves for $N \leq \text{LMAP}$.

Future studies could look at the dependency of the robustness objective weight on performance. If other robustness weights are to be considered, all of the weights taking part in the fitness function should be modified accordingly to maintain the order of importance of each objective. When the unassigned flights are not taken into account by the robustness measure used in the fitness function, then particular care has to be taken when selecting the appropriate weights for maximisation of the number of assignments (the first objective), since incorrect selection of this objective weight may sometimes deem solutions with a lower number of assignments to be fitter. This interdependency mainly between the maximum number of assignments' objective and the robustness could also be decoupled by penalising the unassigned flights in the robustness.

5. Conclusions

Several approaches were presented here to take account of solution robustness when applying the SSEA. The TRS approach provides solutions with a statistically significantly higher number of conflicts than those obtained by many of the other approaches considered, as shown in [21]. TRS does not consider the extra increase in conflict, as the service time is reduced between assignments, thus further penalising those assignments. Where the service reduction is higher the Arc Tangent Reduction in Service (ATRS) was also used, but the results were not very encouraging as some of the other approaches improved on it.

When looking at the overall schedule it is evident that the number of flights requiring assignment at each time is not uniformly distributed, as shown in Figure 2, such that when

the number of flights to be serviced at any one time increases, the potential for conflict also increases, when perturbed, such that there is potential for further improving the results when this is taken into account, so other approaches were proposed and studied. The approaches examined, which consider the distribution of flights in time, were the ARS, BSARS, and SARS. These approaches performed better for a very low number of BSSs but in general provided solutions inferior in respect of conflicts compared to the PCBG, as well as the UESRS with exponential and Inverse estimation functions. The advantage of the TRS approach in conjunction with some of the other approaches was anticipated, as was corroborated by results from the experiments conducted for the combination of TRS with the ARS, BSARS, SARS, and UESRS.

Several stochastic approaches were also considered; namely, UESRS was evaluated for different estimation functions and the exponential function with $\beta = 0.03$ provided the best results in respect of robustness for the measure considered. PCBG provides the best overall performance regarding robustness but did not seem to improve when combined with TRS. RNC appeared to provide solutions with a higher number of conflicts compared to the other approaches, but this could be a consequence of the low number of iterations which it was possible to execute given both the low speed, which is a characteristic of this approach, and the higher memory requirement. The UESRS approach also improved when combined with TRS, showing the Inverse function for $b = 6$ with TRS providing good results for high numbers of BSSs ($N \geq \text{LMAP}$).

When UESRS was combined with BSARS the performance improved to when these approaches were used alone or combined with TRS.

It would be interesting to apply these approaches to some data sets where the original flight schedules and final real schedules are known, to see if the solutions obtained by these approaches did cope well with the changes, but unfortunately

this was impossible at the time, given the unavailability of such data. Furthermore, if the data sets contain the real flight assignments to BSSs on the day of implementation, then it would be possible to quantify the actual improvement which could have been achieved by each of the approaches presented here, if they were implemented. There is also a question about the preference as to how these approaches are combined, either as a sum of each individual with the TRS, BSARS, and ARS, as studied here, or as a product of their individual contributions.

Future work should consider the use of multiple distributions, in general one per flight, based on the particular characteristics of each flight, such as aircraft type, airline, destination, route, and season. This could be applied similarly to the generation of the buffer times. This approach was not used, since such information was unavailable at the time this study was conducted. It is envisaged that the use of this information in the PCBG and RNC will improve their performance and may also be used to generate the disrupted schedules used to measure the solution quality provided by all of the approaches, thus assisting in the identification of the best approach for use in the specific problem.

More information about the problem studied here, the metaheuristic algorithms used, and the application of these approaches to the ABSSAP and AGAP can be found in [21].

Competing Interests

The author declares having no competing interests.

Acknowledgments

The author is grateful to NATS Ltd. and to EPSRC for providing the funding for the original project and to the University of Nottingham for giving the author the opportunity to take part in this project.

References

- [1] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios, "Robust optimization of large-scale systems," *Operations Research*, vol. 43, no. 2, pp. 264–281, 1995.
- [2] A. Ascó, J. A. D. Atkin, and E. K. Burke, "An analysis of constructive algorithms for the airport baggage sorting station assignment problem," *Journal of Scheduling*, vol. 17, no. 6, pp. 601–619, 2014.
- [3] A. Ascó, J. A. D. Atkin, and E. K. Burke, "An evolutionary algorithm for the over-constrained airport baggage sorting station assignment problem," in *Simulated Evolution and Learning: 9th International Conference, SEAL 2012, Hanoi, Vietnam, December 16–19, 2012. Proceedings*, L. Bui, Y. Ong, N. Hoai, H. Ishibuchi, and P. Suganthan, Eds., vol. 7673 of *Lecture Notes in Computer Science*, pp. 32–41, Springer, Berlin, Germany, 2012.
- [4] A. Ascó, J. A. D. Atkin, and E. K. Burke, "The airport baggage sorting station allocation problem," in *Proceedings of the 5th Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA '11)*, J. Fowler, G. Kendall, and B. McCollum, Eds., pp. 419–444, Phoenix, Ariz, USA, August 2011.
- [5] A. Abdelghany, K. Abdelghany, and R. Narasimhan, "Scheduling baggage-handling facilities in congested airports," *Journal of Air Transport Management*, vol. 12, no. 2, pp. 76–81, 2006.
- [6] Y. Nikulin, "Robustness in combinatorial optimization and scheduling theory: an extended annotated bibliography," Tech. Rep. 606, Institut für Betriebswirtschaftslehre, 2006.
- [7] C.-L. Wu and R. E. Caves, "Modelling and optimization of aircraft turnaround time at an airport," *Transportation Planning and Technology*, vol. 27, no. 1, pp. 47–66, 2004.
- [8] S. Yan, C.-Y. Shieh, and M. Chen, "A simulation framework for evaluating airport gate assignments," *Transportation Research Part A: Policy and Practice*, vol. 36, no. 10, pp. 885–898, 2002.
- [9] A. Bolat, "Procedures for providing robust gate assignments for arriving aircrafts," *European Journal of Operational Research*, vol. 120, no. 1, pp. 63–80, 2000.
- [10] S. Yan and C. M. Chang, "A network model for gate assignment," *Journal of Advanced Transportation*, vol. 32, no. 2, pp. 176–189, 1998.
- [11] M. I. Hassounah and G. N. Steuart, "Demand for aircraft gates," *Transportation Research Record*, no. 1423, pp. 26–33, 1993.
- [12] S. Yan and C.-M. Huo, "Optimization of multiple objective gate assignments," *Transportation Research Part A: Policy and Practice*, vol. 35, no. 5, pp. 413–432, 2001.
- [13] D.-X. Wei and C.-Y. Liu, "Fuzzy model and optimization for airport gate assignment problem," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS '09)*, pp. 828–832, Shanghai, China, November 2009.
- [14] C.-L. Wu and R. E. Caves, "Aircraft operational costs and turnaround efficiency at airports," *Journal of Air Transport Management*, vol. 6, no. 4, pp. 201–208, 2000.
- [15] R. S. Mangoubi and D. F. X. Mathaisel, "Optimizing gate assignments at airport terminals," *Transportation Science*, vol. 19, no. 2, pp. 173–188, 1985.
- [16] A. Bolat, "Assigning arriving flights at an airport to the available gates," *Journal of the Operational Research Society*, vol. 50, no. 1, pp. 23–34, 1999.
- [17] A. Bolat, "Models and a genetic algorithm for static aircraft-gate assignment problem," *Journal of the Operational Research Society*, vol. 52, no. 10, pp. 1107–1120, 2001.
- [18] A. Lim and F. Wang, "Robust airport gate assignment," in *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '05)*, pp. 74–81, November 2005.
- [19] S. Yan and C.-H. Tang, "A heuristic approach for airport gate assignments for stochastic flight delays," *European Journal of Operational Research*, vol. 180, no. 2, pp. 547–567, 2007.
- [20] "Stochastic programming," in *Handbooks in Operations Research and Management Science*, A. Ruszczyński and A. Shapiro, Eds., vol. 10, Elsevier, New York, NY, USA, 2003.
- [21] A. Ascó, *Constructive and evolutionary algorithms for airport baggage sorting station and gate assignment problems [Ph.D. thesis]*, School of Computer Science, 2013.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

