



# Robustness Algorithms for the Airport Gate Assignment Problem

By Amadeo Ascó Signes

*University of Nottingham*

**Abstract-** Assigning commercial flights to available airport gates can have a major impact on the efficiency of flight schedules as well as on the level of passenger satisfaction with the service. These assignments also depend on the service requirements of flights and the capacity of stand facilities. Unexpected changes also called perturbations, like those due to air traffic delays, severe weather conditions, or equipment failures, may disrupt the initial assignments and increase the difficulty of maintaining smooth operations, which will detrimentally affect customer satisfaction.

The provision of solutions which reduce the potential detrimental effect of perturbations in the stands already assigned on the day of operation is desirable and some approaches are presented here, and compare between them to help identify their performance and trends.

**Keywords:** *robustness, airport gate assignment problem, scheduling, heuristics, evolutionary algorithms.*

**GJCST-C Classification:** *F.2.2*



ROBUSTNESSALGORITHMSFORTHEAIRPORTGATEASSIGNMENTPROBLEM

*Strictly as per the compliance and regulations of:*



# Robustness Algorithms for the Airport Gate Assignment Problem

Amadeo Ascó Signes

**Abstract-** Assigning commercial flights to available airport gates can have a major impact on the efficiency of flight schedules as well as on the level of passenger satisfaction with the service. These assignments also depend on the service requirements of flights and the capacity of stand facilities. Unexpected changes also called perturbations, like those due to air traffic delays, severe weather conditions, or equipment failures, may disrupt the initial assignments and increase the difficulty of maintaining smooth operations, which will detrimentally affect customer satisfaction.

The provision of solutions which reduce the potential detrimental effect of perturbations in the stands already assigned on the day of operation is desirable and some approaches are presented here, and compare between them to help identify their performance and trends.

**Keywords:** robustness, airport gate assignment problem, scheduling, heuristics, evolutionary algorithms.

## I. INTRODUCTION

Aircraft depart from an airport and arrive at their destination airport, from which the aircraft may again depart to yet another airport, and this may be repeated many times a day for each aircraft. During the time between arrival and departure, while the aircraft is still at the airport, it needs to have a space allocated at a stand on the airport airside, where some operations may need to be performed before it is ready to continue its cycle of departure and arrival. The stands next to the airport gates are scarce and expensive resources which must be used efficiently and be assigned to aircraft effectively. The gate assigned to an aircraft arrival may not be the same as that assigned to the same aircraft for departure, and the intermediate parking operation if any is required, between arrival and departure assignments may also be at a different stand, Figure 1.

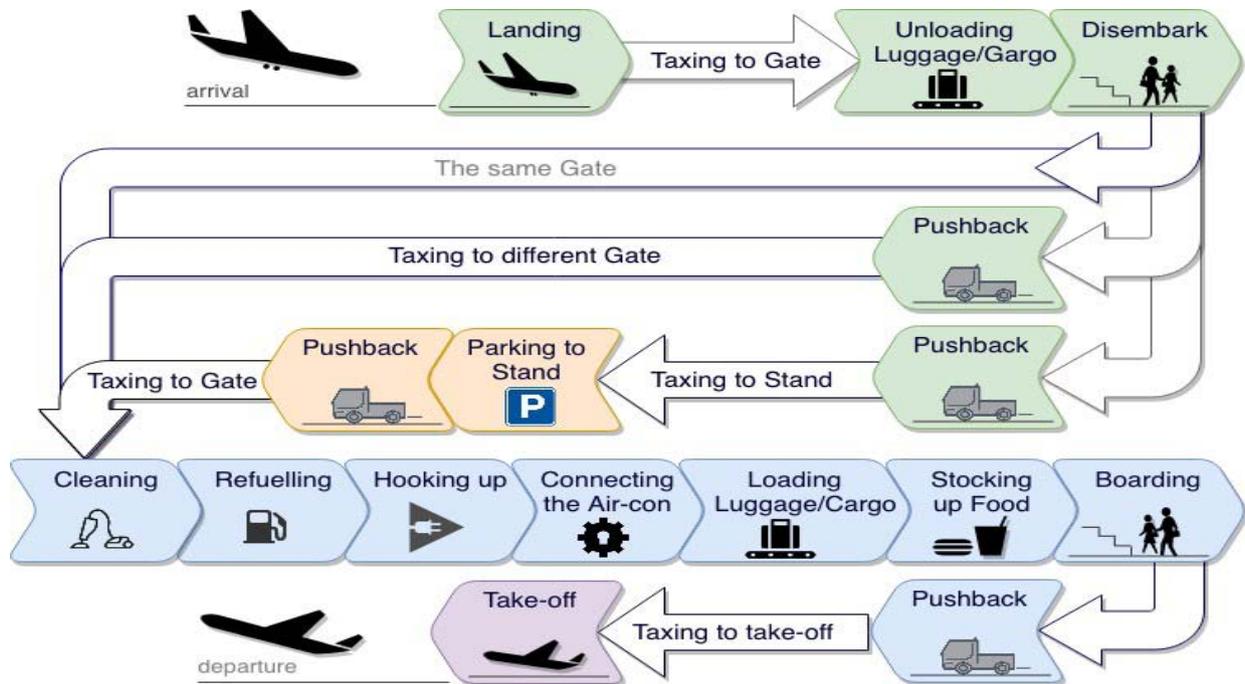


Fig. 1: Simplified overall process of an aircraft in an airport

The Airport Gate Assignment Problem (AGAP) relates to the assignment of stands to flights already scheduled. The flights have a scheduled arrival and departure time, between which the aircraft is located on either one or several stands sequentially. The movement of an aircraft between stands normally requires the use of tugs (towing trucks) which add extra cost to the aircraft's operations, which airlines would prefer to avoid

whenever possible. When an aircraft is assigned to different stands in the same arrival/departure period, to disembark passengers from one stand and embark them on another, then the aircraft must be moved from its assigned stand to the stand assigned to it next until it is located on the stand assigned for its departure. This operation may be executed either by using the aircraft's engines or by tugs, Figure 2.

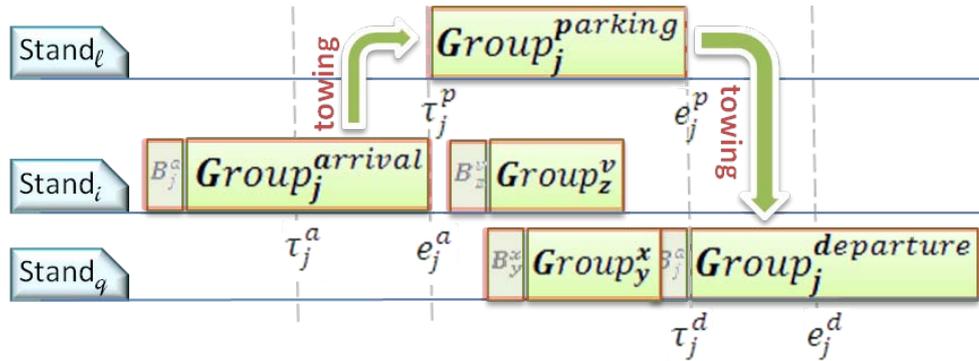


Fig. 2: Assignment of flights to stands when towing is required

With the increase in passenger traffic volumes and number of flights, the complexity of this task and the number of factors to be considered have increased significantly, and efficient gate utilization has received considerable attention in passed years, e.g. Ascó (2019), Kim and Feron (2013), Seker and Noyan (2012), Jaehn (2010), Li(2009), Hu and Di Paolo (2007). AGenetic Algorithm (GA) was used by Lim et al (2005) and Ghazouani et al (2015).

The provision of solutions which reduce the potential detrimental effect of perturbations in the resources already assigned on the day of operation is desirable and was previously studied for the Airport Baggage Sorting Station Assignment Problem (ABSSAP) by Ascó (2016, 2013), and it was studied for the AGAP by van Schaijk and Visser (2017), Ascó (2013), Yan and Tang (2007). But this is achieved at the expense of the optimality, although this is far from easy, as the perturbations are not known in advance. It would also be advantageous if the disrupted assignments have no knock on effect, or if any, only a minor one.

A flight is said to be in conflict if the departure time of the flight is greater than the arrival time of the next flight at a gate. Reassigning conflicting flights caused by flight delays is one of the major daily tasks for the airport scheduling team. When a delayed flight affects the flight schedule of the subsequent flight then there are two ways it may be corrected: either to reassign the conflicting flight or else reassign the subsequent flight to the conflicting one. A situation may arise when reassigning conflicting flights or the subsequent flight to another gate, where the reassigned flight is interfering with the subsequent flight at the new gate. Thus some reassignments may, therefore, have a

downstream effect on the overall schedule, producing more conflicting flights requiring further reassignments, and potentially increasing the difficulty of the problem at a later stage.

The model used for the AGAP is based on that proposed in Dorndorf (2002), which considers the problem as a resource constrained project scheduling problem, originally presented in Dorndorf et al (2000), used in Ascó (2013) Chapter 7 and 8 with the Steady State Evolutionary Algorithm (SSEA), and Ascó (2019). The robustness approaches presented and studied for the ABSSAP by Ascó (2016) are adapted to the AGAP and are next presented and studied with an adapted implementation of the SSEA for the AGAP.

## II. ROBUSTNESS APPROACHES

The approaches considered here take account of the potential disruptions on the day of operation, and are: Total Reduction in Service Time (TRS), Area of Reduction in Service (ARS), Sub-Area of Reduction in Service (SARS), Unsupervised Estimated Stochastic Reduction in Service (UESRS), Reduction in the Number of Conflicts (RNC) and Probability of Conflict Based on the Gap (PCBG), which are described below and more details can be found in Ascó (2016), and Ascó et al (2013).

### a) Total Reduction in Service

Mangoubi and Mathaisel (1985) 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 reallife delays, it is advisable to minimise the total reduction in service time, thus maximising buffer times.

The arrival and departure flights correspond to the arrival and departure activities respectively. The time between the scheduled arrival time at the stand and the time at which the flight is scheduled to leave is called the base service duration. A predefined period of time, called buffer time, the value of which depends on the flight, is preappended to the flight base starting service time, so that such buffer time may be reduced to allow other assignments to be placed before this flight, but the base service duration must not be affected, Figure 3. The use of buffer service time implies a preference for a greater predetermined service time for each flight, and this buffer time may be obtained from historical information. A reduction in the buffer time for the arrival and departure of aircraft  $j$  has been named  $\tau_j^a$  and  $\tau_j^d$  respectively, and the sum of these constitutes the reduction in service cost, so this objective can be expressed as  $-\sum_{j=1}^M \sum_x^{a,d} r_j^x * \sum_{i=1}^N y_{ij}^x$  where  $y_{ij}^x$  is one if flight  $j$  is assigned to activity  $i$  for activity  $x$  (arrival, parking, or departure), or zero if flight  $j$  is not assigned to stand  $i$  during activity  $x$ .

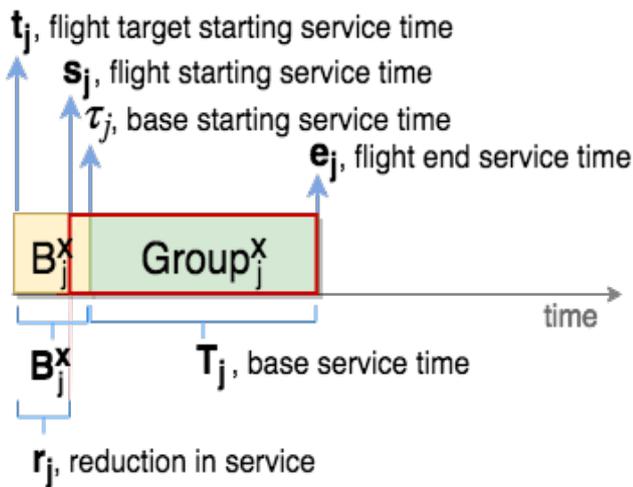


Fig. 3: Overview of the flight times

If the remote parking activity is assigned to the same stand as the departure activity, then the reduction in service for the departure flight is zero. This is a consequence of both activities referring to the same aircraft.

b) *Distribute Idle Time*

Different approaches aim to spread the time between the end of one assignment and the start of the next one at the same gate. This time is normally called 'idle time'. This may be achieved by using a penalty function, such as the arctangent function. Bolat (1999) proposed the distribution of 'idle time' uniformly amongst gates.

A disadvantage of this approach and many others, i.e. UESRS, PCBG and RNC, is that they normally treat all assignments equally, whereas disturbance of a schedule is more likely to have disruptive consequences for the assignment at the time of day when service flight density is greater, which is taken into account by the next two approaches.

c) *Area Reduction in Service*

The effect of service reduction is not the same throughout the day, but depends on the time of day. It is more likely that disruptions will occur during periods when the flight density is higher than when fewer flights require servicing, i.e. delay during high flight density is more likely to propagate given that less 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 to the same Airport Gate (gate).

Lower Maximum Assignment Point with Parking (LMAPP) is the number of resources required to service a certain number of activities (full assignment is achievable for any other higher number of gates for the same number of flights) when the service starting time ( $s_j$ ) coincides with the target starting service time ( $t_j$ ), which it is an adaptation for the AGAP of the Lower Maximum Assignment Point (LMAP) initially introduced in Ascó et al (2013) for the ABSSAP.

Upper Maximum Assignment Point with Parking (UMAP<sub>p</sub>) is the number of resources required to service those activities when the service starting time ( $s_j$ ) coincides with the base starting service time ( $\tau_j$ ), which it is an adaptation for the AGAP of the Upper Maximum Assignment Point (UMAP) initially introduced in Ascó et al (2013) for the ABSSAP. When considering the extra constrain of parking it may potentially be required extra resources to service the same number of flights in comparison to when there is not parking to consider.

The effect of assigning two activities to the same stand, namely too close to each other, has a more detrimental effect in circumstances where there are fewer gates able to accommodate flights. An indication of gate availability on time is provided by the activity density distribution, where no reduction in the target service start time is allowed, see Figure 4. To take account of this, use is made of the area between the reduced service time of the flight density distribution function divided by the Average Assignment Point (AAP) as presented in Ascó (2016, 2013).



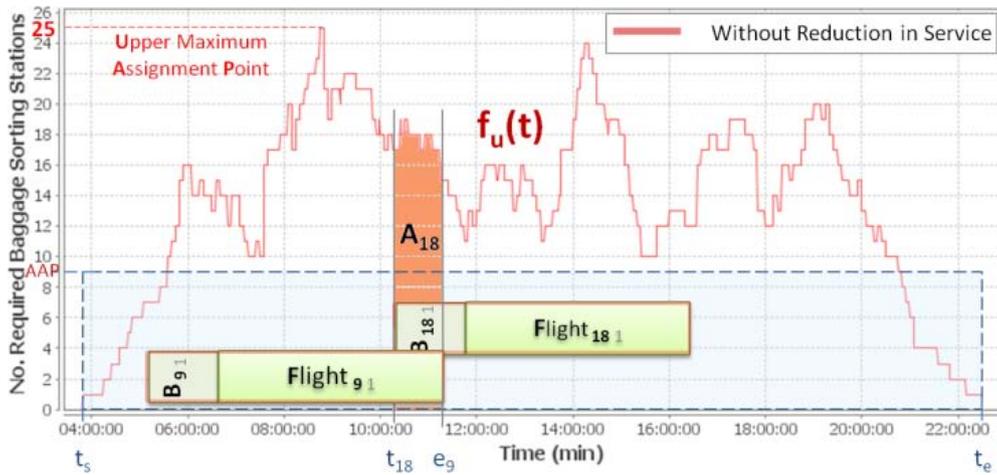


Fig. 4: Distribution of flights over time and Area Reduction in Service Objective,  $A_{18}$

$$AAP = \frac{\int_{t_s}^{t_e} f_u(t) dt}{t_e - t_s} \quad (1)$$

$A_j$  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 gate, for example in Figure 4,  $A_{18} = \int_{t_{18}}^{e_9} f_u(t) dt$ .  $AAP_j$  is here defined as the mean number of flights over the target start time for flight  $j$ ,  $t_{18}$ , and the end of service time for the previous flight assigned to the same gate as flight  $j$ ,  $e_9$ , Equation 1.

d) *Sub-Area Reduction in Service*

This is based on the same principle as the ARS, but uses the two activity density distributions, firstly

where no buffer time is used and secondly when the full buffer time is used (buffer time may not be reduced). The cost is that area between both distributions divided by a factor, Figure 5. The factor was estimated to be equal to the UMAP less the LMAP, which cannot be used if both have the same value, that is to say, LMAP equal to UMAP. To avoid this predicament the un-factored subarea is taken, i.e. using the sub-area itself. This will be the approach considered here, as known as Base Sub-Area Reduction in Service (BSARS), Equation 2.

$$A' = \int_{t_s}^{t_e} (f_u(t) - f_i(t)) dt \quad (2)$$

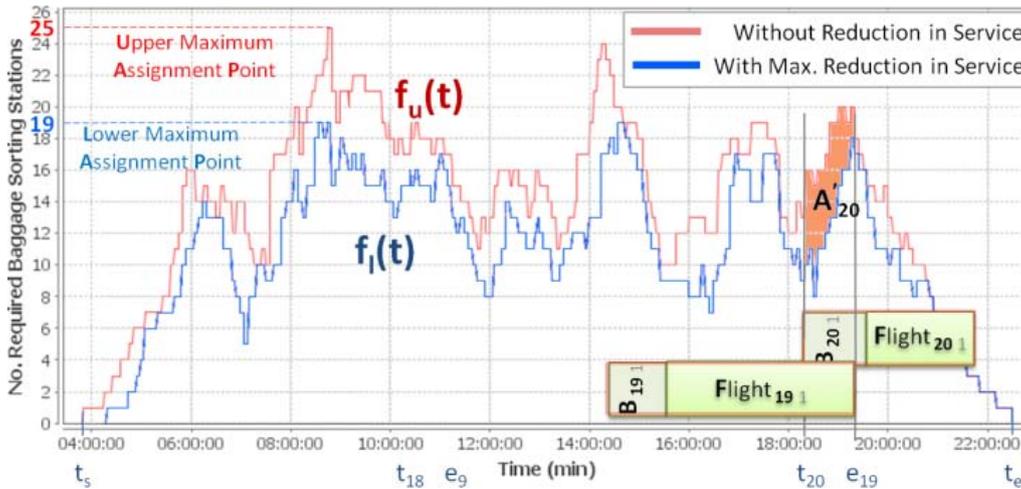


Fig. 5: Sub-Area Reduction in Service Objective,  $A_{20}$

e) *Reduction in the Number of Conflicts*

A similar approach was used in Yan and Tang (2007) 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 semi deviation risk measure (Ruszczynski and Shapiro (2003)) for all those delay scenarios.

The Reduction in the Number of Conflicts (RNC) is a stochastic approach which uses perturbed schedules to account for the conflict potential in consequence of schedule perturbations on the day of execution.

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

Considering 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 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 Equation 3, which is a measure of the solution robustness.

$$f_2 = \frac{-1}{|S|} * \sum_{s \in S} \sum_{j=1}^m c_{js} \tag{3}$$

When reassigning conflicting flights or subsequent flights to other gates, a situation can arise where the reassigned flight interferes with the subsequent flight at the new gate, 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 gate 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 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 Equation 4 where the constant,  $\alpha$ ,  $0 \leq \alpha \leq$

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 Equation 3, 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. An  $\alpha > 1$  refers to the cases where more importance is given to the repercussions of a conflict on assignments other than the conflict itself.

$$f_2 = \frac{-1}{|S|} * \left( \sum_{s \in S} \sum_{j=1}^m (c_{js} + \alpha * c'_{js}) \right) \tag{4}$$

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.

f) *Probability of Conflicts Based on the Gap*

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 the probability of having a conflict in a given 'idle time' is used for each flight. This can be easily obtained if the delay distribution is known. In the case of a normal folded distribution (no negative numbers and with zero mean) and independent delays for flights, the probability of a conflict for different standard deviations is shown in Figure 6. This could also be extended to other distributions and to non-zero means.

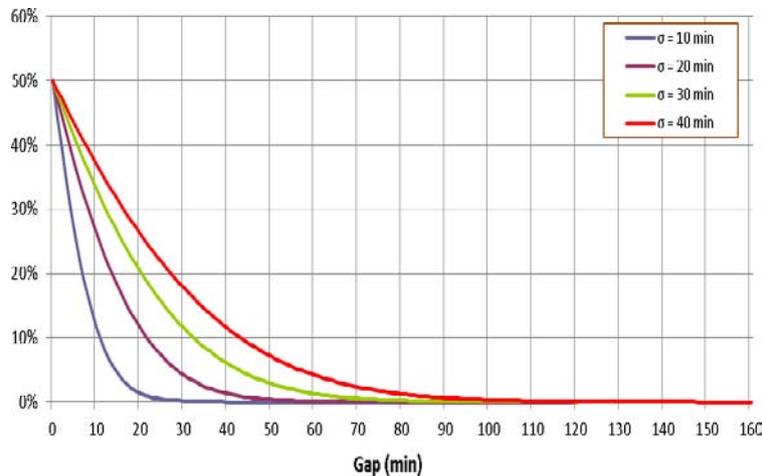


Fig. 6: The probability of a conflict for two consecutive flights assigned to the same gate based on the intervening gap modeled with Folded Normal distributions of a zero mean and standard deviation  $\sigma$ .

The PCBG is another stochastic approach which uses the predicted delay in activities probability distribution functions to anticipate the delay and take account of it in the assignments. In the case of a normal folded distribution (no negative numbers and with zero mean) and independent delays for flights, the probability of a conflict for different standard deviations.

Independent delays are considered initially, where a conflict between two consecutive flights is independent of other flights assigned to the same gate.

The assignments to different gates 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 gate, and the probability of the next consecutive assignment to the same gate not being sufficiently delayed, as shown in Figure 7, where  $\eta_j(t)$  is the probability density function for flight  $j$  and  $t_0 = \tau_k - e_j$ .

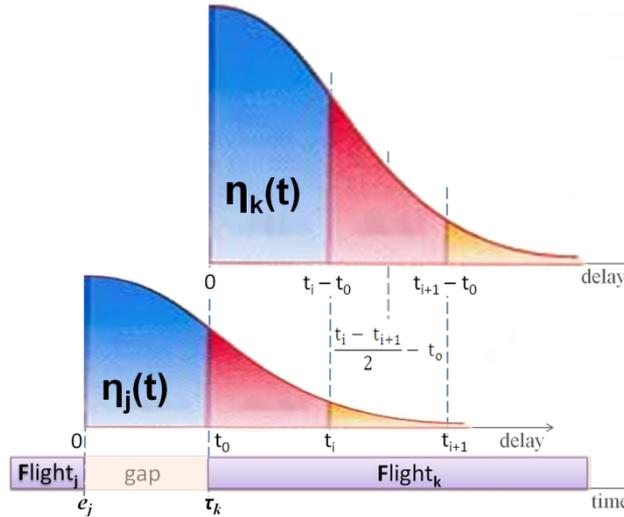


Fig. 7: Probability of a conflict between two consecutive flights based on the intervening gap

g) *Unsupervised Estimated Stochastic Reduction in Service*

Lim and Wang (2005) 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 gate, 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, Equation 5,

where  $l(j, k) = -r$  for  $l(j, k) < 0$ , as shown in Figure 8.  $v(j, k)$  is used to estimate the mean conflict probability between flights  $j$  and  $k$  assigned to the same gate, which is a function of the gap  $l(j, k)$ , where larger gaps between assignments to the same gates result in lesser probability of real flight conflicts.  $v(j, k)$  is normalised in Equation 6.

$$l(j, k) = t_k - e_j \quad (t_k = \tau_k - B_k) \tag{5}$$

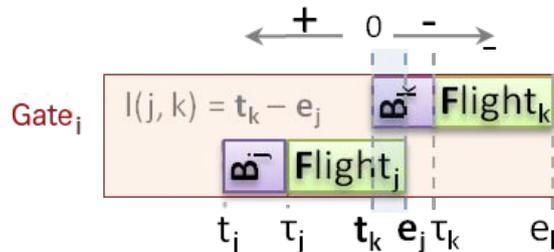


Fig. 8: Overlap between two flights  $j$  and  $k$  assigned to the same gate

$$E(p(j, k)) = \sum_{i=1}^N \left( y_{ij} * y_{ik} * \frac{v(j, k) - v_{min}(j, k)}{v_{max}(j, k) - v_{min}(j, k)} \right) \tag{6}$$

$$f_2 = - \sum_{j=1}^{M-1} \sum_{k=j+1}^M E(p(j, k)) \tag{7}$$

The definition of  $v(j, k)$  comes from the application domain, in the absence of historical data; some unsupervised estimation functions were introduced in Lim and Wang (2005). Figure 9 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 baggage sorting

station (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, to start 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 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 Lim and Wang (2005) are presented below and are shown in Figure 9:

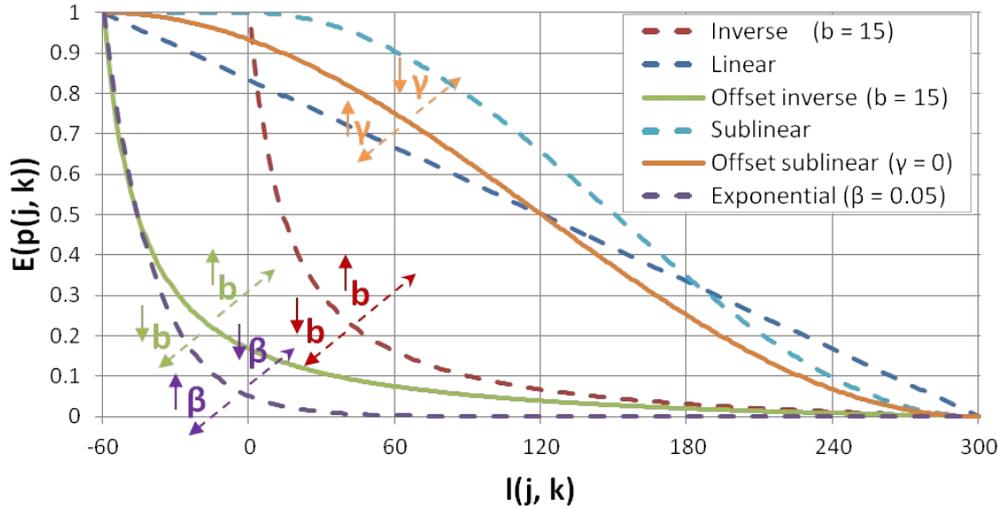


Fig. 9: Penalty for different unsupervised estimation functions based on the gap between assignments

1. Linear estimation:

$$v(j, k) = -l(j, k) \tag{8}$$

2. Exponential estimation:

$$v(j, k) = e^{-\beta * l(j, k)} \tag{9}$$

3. Inverse estimation

$$v(j, k) = \begin{cases} \frac{b}{l(j, k) + b} & \text{if } l(j, k) > 0 \\ 1 & \text{otherwise} \end{cases} \tag{10}$$

The value of the constant ‘b’ changes the penalization as shown in Figure 9, 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 (2005) 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 9), whereas in the exponential estimation function (purple dash line, Figure 9) this is not the case. The value used for ‘b’ may be too great, and a lower value would make this estimation function provides values closer to those provided by the exponential estimation function which provided fitter solutions in the results presented by Lim and Wang (2005). Consequently, a value  $b = 6$  was seen in the experiments studied for the ABSSAP in Ascó (2016) 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 Ascó (2016).

The inverse estimation function as considered by Lim and Wang (2005) 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’, Equation 11, which is shown in Figure 9 for  $b = 15$  (green line).

$$v(j, k) = \frac{b}{l(j, k) - \min\{l(j, k)\} + b} \tag{11}$$

4. Sublinear estimation:

$$v(j, k) = \begin{cases} \cos\left(\frac{\pi * l(j, k)}{l_{max}}\right) & \text{if } l(j, k) > 0 \\ 1 & \text{otherwise} \end{cases} \tag{12}$$

This estimation also suffers from the same problem as the Inverse estimation, and may be improved by offsetting its value so that the maximum penalization corresponds to  $l_{max}$  and the minimum to  $l_{min}$ , Equation 13, which is shown in Figure 9 for  $\gamma = 0$ .

$$v(j, k) = \cos\left(\frac{\pi * (l(j, k) - l_{min} + \gamma)}{l_{max} - l_{min} + \gamma}\right) \tag{13}$$

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 10 for two buffer times of 30 min 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 Lim and Wang (2005), 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.

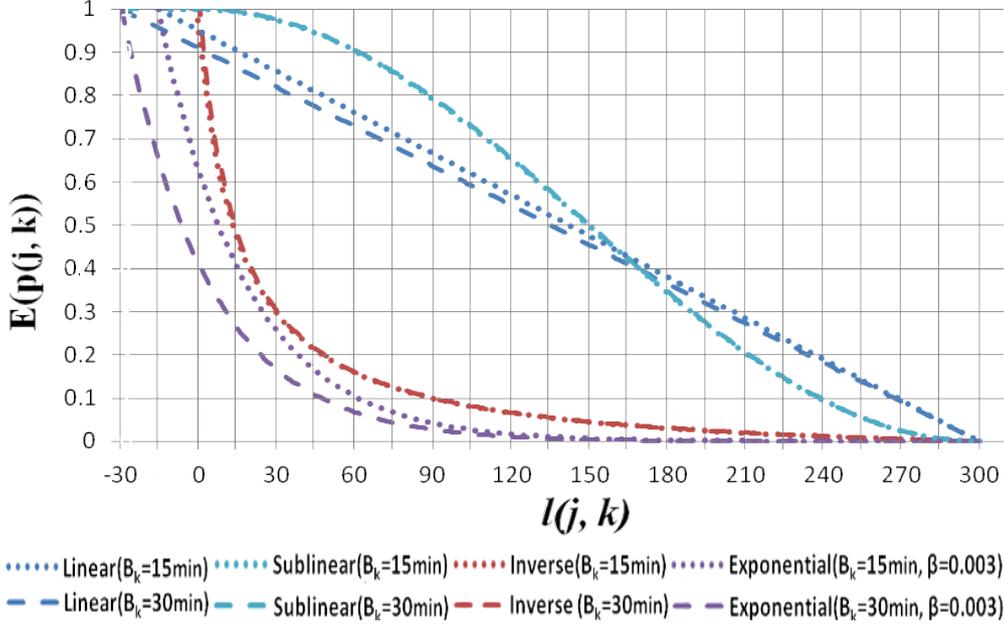


Fig. 10: Penalty for different unsupervised estimation functions based on the gap between assignments for different buffer times

To penalise all of the flights similarly it is necessary to adapt the previous estimation formulas for  $v_{j,k}$  as follow:

1. Inverse:

$$v(j, k) = \frac{b}{l(j, k) + B_k + b} \tag{14}$$

2. Sub-linear:

$$v(j, k) = \cos\left(\frac{\pi * (l(j, k) + B_k + \gamma)}{l_{max} + D_k + \gamma}\right) \tag{15}$$

In addition in the case of the two consecutive flights  $j$  and  $k$  the  $\max\{v(j, k)\}$  corresponds to the  $v(j, k)$  for  $\min\{l(i, k)\} = -B_k$ .

### III. DATA

A week of records of flight assignments to stands was provided by London Heathrow airport for terminal four, composed of schedules from the 6<sup>th</sup> to the 12<sup>th</sup> September 2010 (H4T1009dd). Some details are shown in Table 2, which were generated from the data supplied. Using the data summarised in Table 2, tables were generated showing the preferences of each airline, and these were used in the 'Maximise Airline

Preferences' objective, shown in Section 3.1. Also, a table was generated showing the preferences of each handler, which is used in the 'Maximise Handler Preferences' objective, shown in Section 3.2.

#### i. Airlines Gate Preferences

The following is the data collected from 1<sup>st</sup> September 2011 to 31<sup>st</sup> August 2012, which it was provided by Heathrow Airport Limited (HAL), and a summary of the data is shown on Table 1.

Figure 11 shows the overall number of flights assigned to each gate, to each airline for the period from 6<sup>th</sup> September 2010 to 12<sup>th</sup> September 2010 for London Heathrow airport Terminal 4.

#### ii. Handlers Gate Preferences

Figure 12 shows the overall number of flights assigned to each gate, to each handler for the period from 6<sup>th</sup> September 2009 to 12<sup>th</sup> September 2009 for London Heathrow airport Terminal 4.

### IV. ROBUSTNESS RESULTS

In this section, some experiments are conducted using the robustness approaches summarised in Section 2 for the respective weights shown in Table 3 using the SSEA previously presented in Ascó (2013), Ascó (2018).



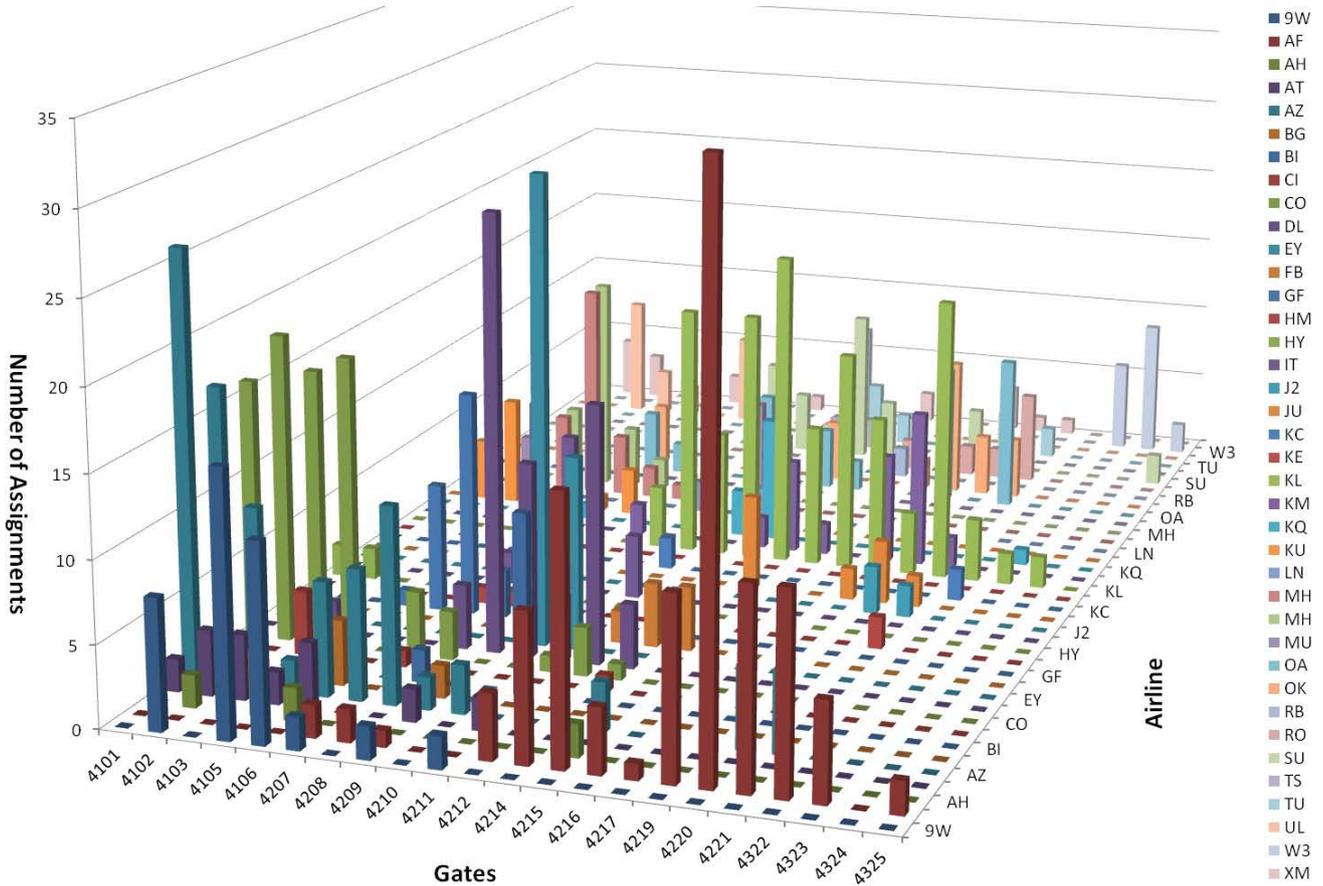


Fig. 11: Airlines preference at London Heathrow airport Terminal 4

Table 2: Data set information provided by HAL for London Heathrow airport Terminal 4.

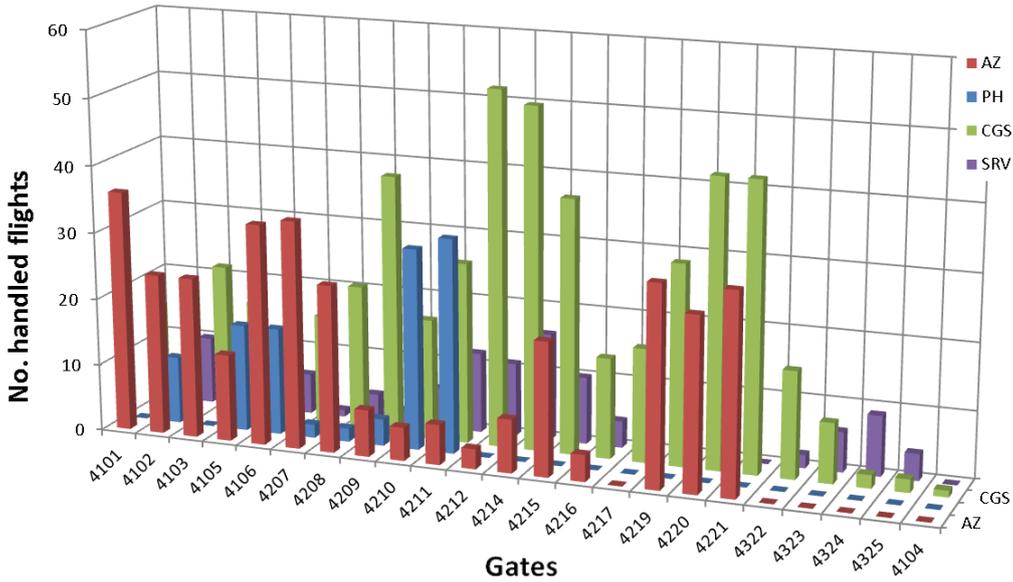
ID	Date	LMAP	UMAP	LMAP $\rho$	UMAP $\rho$	No. Activities	No. Parking Activities
H4T100906	6 Sept 2010	8	10	17	19	118	15
H4T100907	7 Sept 2010	11	14	18	20	120	15
H4T100908	8 Sept 2010	7	10	16	18	119	16
H4T100909	9 Sept 2010	8	10	18	20	119	15
H4T100910	10 Sept 2010	9	12	15	18	120	15
H4T100911	11 Sept 2010	9	10	16	16	110	11
H4T100912	12 Sept 2010	11	11	18	19	117	15

Table 3: Weights for the different robustness approaches considered with SSEA1.

Approach	Weight	Parameters	
		Name	Values
ARS	0.00025	Buffer Time	15 min long-haul and 10 min others
ATRS	0.00025	Buffer Time	15 min long-haul and 10 min others
BSARS	0.00025	Buffer Time	15 min long-haul and 10 min others
PCBG	0.225 and 0.3125	Std. deviation	10, 20 and 30 min
TRS	0.00025	Buffer Time	15 min long-haul and 10 min others
UESRS	0.225 and 0.3125	Estimator	Exp 0.03, Exp 0.05, Inverse 6, Inverse 15, Linear, Offset Inverse 6, Offset Inverse 15, Offset-Sub linear 0, Offset Sub linear 1000 and Sub linear
		Buffer Time	15 min long-haul and 10 min others

These weights are smaller than the weights used in the ABSSAP because they have been normalised, Equation 16.

$$\sqrt{W_1^2 + W_2^2 + W_3^2 + W_4^2 + W_5^2} = 1 \tag{16}$$



	4101	4102	4103	4105	4106	4207	4208	4209	4210	4211	4212	4214	4215	4216	4217	4219	4220	4221	4322	4323	4324	4325	4104
AZ	36	24	24	13	33	34	25	7	5	6	3	8	20	4	0	30	26	30	0	0	0	0	0
PH	0	10	0	16	16	2	2	4	30	32	0	0	0	0	0	0	0	0	0	0	0	0	0
CGS	16	20	23	18	2	17	22	39	18	27	53	51	38	15	17	30	43	43	16	9	2	2	1
SRV	2	10	11	6	6	1	4	2	6	12	11	16	10	4	0	4	4	0	2	6	9	4	0

Fig. 12: Handlers preference at London Heathrow airport Terminal 4

The results are summarised in tables which only show the robustness approaches which, at least in one instance of the disruptions for a given standard deviation, provide statistically significantly less collisions than other approaches evaluated, and cannot be said to be statistically worse than any of the approaches considered. The tables show for each standard deviation the number of times an approach cannot be said to be statistically significantly worse than any of the other approaches. The last column provides the sum of each result for each of the standard deviations. The case where all instances in a given standard deviation cannot be said to be statistically worse than any other are shown in bold text, and in underlined text for those cases which provide the highest number of all the approaches considered.

The results for the different robustness approaches, when applied to the data sets in Table 2 (data sets from HAL for London Heathrow airport Terminal 4) are summarised in Table 4. There is no apparent statistical difference between them for short disruptions (10 min standard deviation). For longer disruptions it is the UESRS with exponential unsupervised estimation function with  $\beta = 0.03$ , weight of 0.3125, with and without TRS approach which performs best for each of the similar disruptions considered. These results correspond to data sets where there is a sufficient number of gates for assignment to all of the activities ( $N < UMAP_p$ ). No

general gain is shown by combining the base approach with TRS. Nevertheless, there seems to be no detriment in combining with TRS either. The approaches ARS and BSARS do not perform well in any instance when either used alone or combined with TRS, which has also been observed when the rate of activities per gate increases (Table 6). These results also corroborate those presented in Lim and Wang (2005), namely, when the number of gates is greater than the UMAP, the exponential unsupervised estimation function performs better, but only when compared with the other unsupervised estimation functions.

Table 6 shows the summary results for the new data sets with an extra 37 groups for the same number of gates (a summary of data sets is shown in Table 5). These data sets are equivalent to a reduction in the number of gates available per group, representing more activities for the same number of resources. The UESRS approaches alone or in combination with TRS still perform well for low disruptions (particularly with the exponential estimation function with  $\beta = 0.05$ ), and is even better than the PCBG(x), but PCBG(x) subsequently performed better for longer disruptions. The ARS and BSARS also achieved solutions with statistically significantly less collision when they were used together with the TRS than when used alone, but not when compared to UESRS and PCBG(x).

Table 4: Summary of statistical significance of AGAP robustness (significance level 0.05) using perturbed schedules generated from normal distributions of 10, 20 and 30 min standard deviations (x), SSEA1 for data sets H4T1009dd.

Approach	Standard Deviation (x) in min			Total
	10	20	30	
0.225PCBG(x)+0.00025TRS(2)	<u>7</u>	1	1	9
0.3125PCBG(x)+0.00025TRS(2)	<u>7</u>	1	4	12
0.225PCBG(x)	<u>7</u>	0	2	9
0.3125PCBG(x)	<u>7</u>	1	4	12
0.225UESRS(E0.03)+0.00025TRS(2)	<u>7</u>	4	5	16
0.225UESRS(E0.05)+0.00025TRS(2)	<u>7</u>	1	0	8
0.225UESRS(I4)+0.00025TRS(2)	<u>7</u>	1	0	8
0.225UESRS(I6)+0.00025TRS(2)	<u>7</u>	3	0	10
0.3125UESRS(E0.03)+0.00025TRS(2)	<u>7</u>	<u>6</u>	<u>6</u>	<u>19</u>
0.3125UESRS(E0.05)+0.00025TRS(2)	<u>7</u>	2	2	11
0.3125UESRS(I4)+0.00025TRS(2)	<u>7</u>	3	1	11
0.3125UESRS(I6)+0.00025TRS(2)	<u>7</u>	2	1	10
0.225UESRS(E0.03)	<u>7</u>	5	<u>6</u>	15
0.225UESRS(E0.05)	<u>7</u>	3	1	11
0.225UESRS(I4)	<u>7</u>	1	0	8
0.225UESRS(I6)	<u>7</u>	3	0	10
0.225UESRS(I15)	<u>7</u>	3	3	12
0.3125UESRS(E0.03)	<u>7</u>	<u>6</u>	<u>6</u>	<u>19</u>
0.3125UESRS(I4)	<u>7</u>	1	0	8
0.3125UESRS(I6)	<u>7</u>	3	2	12
0.3125UESRS(I15)	<u>7</u>	<u>6</u>	5	18

Table 5: Generated data sets information with an extra 37 groups.

ID	Date	LMAP	UMAP	LMAP <sub>p</sub>	UMAP <sub>p</sub>	No. Activities	No. Parking Activities
N4T100906	6 Sept 2010	17	20	23	26	164	21
N4T100907	7 Sept 2010	21	23	25	28	160	19
N4T100908	8 Sept 2010	18	20	23	25	169	24
N4T100909	9 Sept 2010	21	21	28	28	168	22
N4T100910	10 Sept 2010	19	20	20	21	164	21
N4T100911	11 Sept 2010	19	21	21	21	154	15
N4T100912	12 Sept 2010	19	21	23	24	167	22

The empirical results show, when comparing the results of Tables 3 and 4, that combining the approaches with TRS helps to reduce the number of collisions where there is a lower number of gates per activity. These results suggest that when fewer resources (gates) are available the increase in the influence of the buffer time is advantageous, given that there is more chance of future disruptions as there is less 'idle time' available for the overall problem. It is therefore anticipated that combining both UESRS and PCBG(x) with other approaches using the buffer time, such as ARS and BSARS, should also further improve the results. The ARS and BSARS are tailored to take account of the influence of the flights distribution over

time, so increasing the penalty in periods where there is a higher demand for gates, which the experiments indicate improved results.

a) Robustness

The summary of the statistical significance of the different robustness approaches is shown in Tables 7 and 8, where the PCBG uses the same standard deviation as the normal distribution which was used to generate the perturbed schedules. The LMAP, UMAP, LMAP<sub>p</sub> and UMAP<sub>p</sub> from Table 2 are shown between brackets in the table heading for convenience as (LMAP, UMAP, LMAP<sub>p</sub>, UMAP<sub>p</sub>). The table only presents those approaches which either alone or combined with others

provide solutions with statistically significantly fewer collisions than other approaches studied and cannot be said to have more collisions than any of the other

operators studied when used alone or in combination, which are shown with a tick. Only those approaches having at least one tick are shown.

**Table 6:** Summary of statistical significance of AGAP robustness (significance level = 0.05) using perturbed schedules generated from normal distributions of 10, 20 and 30 min standard deviations (x), SSEA1 for new data sets N4T1009dd with 37 extra groups each.

Approach	Standard Deviation (x) in min			Total
	10	20	30	
0.225PCBG(x)+0.00025TRS(2)	3	3	5	11
0.3125PCBG(x)+0.00025TRS(2)	2	<u>6</u>	<u>6</u>	<u>14</u>
0.225PCBG(x)	2	4	<u>6</u>	12
0.3125PCBG(x)	3	5	5	13
0.225UESRS(E0.03)+0.00025TRS(2)	5	0	0	5
0.225UESRS(E0.05)+0.00025TRS(2)	<u>6</u>	1	0	7
0.225UESRS(I4)+0.00025TRS(2)	3	0	0	3
0.225UESRS(I6)+0.00025TRS(2)	5	0	5	5
0.3125UESRS(E0.03)+0.00025TRS(2)	<u>6</u>	2	1	9
0.3125UESRS(E0.05)+0.00025TRS(2)	<u>6</u>	4	2	12
0.3125UESRS(I4)+0.00025TRS(2)	5	0	0	5
0.3125UESRS(I6)+0.00025TRS(2)	4	1	0	5
0.225UESRS(E0.03)	3	0	0	3
0.225UESRS(E0.05)	4	1	0	5
0.225UESRS(I4)	4	0	0	4
0.225UESRS(I6)	3	0	0	3
0.225UESRS(I15)	4	1	0	5
0.3125UESRS(E0.03)	5	2	0	7
0.3125UESRS(I4)	5	0	0	5
0.3125UESRS(I6)	4	1	0	5
0.3125UESRS(I15)	3	1	0	4

To speed up execution of the PCBG robustness approach instead of using the density function for the distribution (folded normal distribution), a pre-generated table of the accumulative probabilities was used for up to four times the standard deviation.

It should be noted that given that the PCBG used considers standard deviations equal to those used to build the perturbed data sets it may be considered biased and be expected to perform better. However the results obtained for data sets with a sufficient number of gates to assign all the activities shows that the UESRS performs better for different unsupervised functions than PCBG for all the disruption standard deviations considered.

## V. CONCLUSIONS

Different algorithms and their parameters were studied to find characteristics which could be used to identify the algorithm and parameters most appropriate to the AGAP. Both the model and algorithms are derived by modifying those presented in previous work, and are based on the specific characteristics of the problem.

The time an aircraft spends parked at a gate has a considerable effect on the operations which take place up stream in the overall airport operation, especially when some of the resources required, such as gates, are limited. Delays in starting the departure sequencing may have important effects on the departure itself, which in turn may also require other aircraft to extend the time during which they are held at the gates. This could well affect other flights arriving which have had the same gates assigned to them. It would be therefore advisable to account for the effect of potential disturbances in the assignment plan and so some approaches were considered. The number of conflicts in perturbed schedules were used as a means of comparing the performance of different approaches. Different conditions were considered from when there are too few gates to assign to all flights, represented by LMAPP, when there are sufficient gates to service all the flight but at the expenses of a reduction on buffer time, represented by UMAPP, and when there are sufficient gates to assign all flight to the available gates without having to reduce the buffer time. It was concluded that

the empirical results indicate that the PCBG did not provide such good results as the UESRS regarding those conflicts where there are plentiful gates to which to assign activities. PCBG performance improved as the number of gates available to service the activities is reduced. Furthermore, it was noted that the close relationship between the PCBG approach and the perturbed base schedule used to calculate the conflicts, and which provides some bias in favour of the PCBG,

may be reduced or removed if the buffer times, considered for the other robustness approaches, are modified accordingly. The combination of UESRS and PCBG with TRS provides good solutions, and there is still some potential for combining UESRS and PCBG with other approaches, such as ARS and BSARS, which take account of other problem characteristics which both UESRS and PCBG do not, so potentially further improving the robustness of the solutions reached.

Table 7: Summary AGAP robustness statistical significance (significance level 0.05) using perturbed schedules generated from normal distributions of 10, 20 and 30 min standard deviations and SSEA1.

Approach	H4T100906 (8, 10, 17, 19)			H4T100907 (11, 14, 18, 20)			H4T100908 (7, 10, 16, 18)			H4T100909 (8, 10, 18, 20)			H4T100910 (9, 12, 15, 18)			H4T100911 (9, 10, 16, 16)			H4T100912 (11, 11, 18, 19)				
	Standard Deviation (x) in min																						
	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30		
0.225PCBG(x)+0.00025TRS(2)	✓	✓		✓			✓			✓			✓			✓			✓			✓	
0.3125PCBG(x)+0.00025TRS(2)	✓	✓	✓	✓		✓	✓			✓	✓	✓	✓			✓			✓			✓	✓
0.225PCBG(x)	✓			✓			✓			✓	✓	✓	✓			✓			✓			✓	✓
0.3125PCBG(x)	✓	✓	✓	✓		✓	✓			✓	✓	✓	✓	✓	✓	✓			✓			✓	
0.225UESRS(E0.03)+0.00025TRS(2)	✓	✓	✓	✓			✓	✓	✓	✓	✓		✓	✓	✓	✓			✓	✓	✓	✓	✓
0.225UESRS(E0.05)+0.00025TRS(2)	✓	✓		✓			✓			✓			✓			✓			✓			✓	
0.225UESRS(I4)+0.00025TRS(2)	✓	✓		✓			✓			✓			✓			✓			✓			✓	
0.225UESRS(I6)+0.00025TRS(2)	✓	✓		✓			✓			✓	✓		✓			✓			✓			✓	✓
0.3125UESRS(E0.03)+0.00025TRS(2)	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0.3125UESRS(E0.05)+0.00025TRS(2)	✓	✓	✓	✓			✓			✓			✓			✓			✓			✓	✓
0.3125UESRS(I4)+0.00025TRS(2)	✓	✓		✓			✓			✓			✓			✓	✓		✓	✓	✓	✓	✓
0.3125UESRS(I6)+0.00025TRS(2)	✓	✓		✓			✓			✓			✓			✓			✓			✓	✓
0.225UESRS(E0.03)	✓	✓	✓	✓			✓	✓	✓	✓	✓		✓	✓		✓			✓	✓	✓	✓	✓
0.225UESRS(E0.05)	✓	✓		✓			✓			✓	✓		✓			✓			✓			✓	✓
0.225UESRS(I4)	✓	✓		✓			✓			✓			✓			✓			✓			✓	
0.225UESRS(I6)	✓	✓		✓			✓			✓	✓		✓			✓			✓			✓	✓
0.225UESRS(I15)	✓	✓					✓			✓	✓	✓	✓	✓		✓			✓	✓	✓	✓	✓
0.3125UESRS(E0.03)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓
0.3125UESRS(I4)	✓	✓		✓			✓			✓			✓			✓			✓			✓	
0.3125UESRS(I6)	✓	✓	✓	✓			✓			✓			✓	✓		✓			✓			✓	✓
0.3125UESRS(I15)	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓

**Table 8:** Summary AGAP robustness statistical significance (significance level = 0.05) using perturbed schedules generated from normal distributions of 10, 20 and 30 min standard deviations and SSEA1 for new data sets with 37 extra groups each.

SSEA1 and population size 5 Approach	H4T100906 (17, 20, 23, 26)			H4T100907 (21, 23, 25, 28)			H4T100908 (18, 20, 23, 25)			H4T100909 (21, 21, 28, 28)			H4T100910 (19, 20, 20, 21)			H4T100911 (19, 21, 21, 21)			H4T100912 (19, 21, 23, 24)								
	Standard Deviation (x) in min																										
	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30			
0.225PCBG(x)+0.00025TRS(2)			√	√	√	√			√	√	√	√			√	√						√					
0.3125PCBG(x)+0.00025TRS(2)		√		√	√	√			√	√	√	√	√		√	√			√			√	√				
0.225PCBG(x)			√	√	√	√			√	√	√	√			√				√			√	√				
0.3125PCBG(x)		√	√	√					√	√	√				√	√			√	√		√	√				
0.225UESRS(E0.03)+0.00025TRS(2)				√				√							√			√				√					
0.225UESRS(E0.05)+0.00025TRS(2)	√			√				√							√			√				√	√				
0.225UESRS(I4)+0.00025TRS(2)	√							√														√					
0.225UESRS(I6)+0.00025TRS(2)	√							√							√			√				√					
0.3125UESRS(E0.03)+0.00025TRS(2)				√				√		√	√				√			√	√			√	√				
0.3125UESRS(E0.05)+0.00025TRS(2)		√		√	√	√	√				√		√	√	√			√	√			√	√				
0.3125UESRS(I4)+0.00025TRS(2)	√							√							√			√				√					
0.3125UESRS(I6)+0.00025TRS(2)	√							√							√			√	√								
0.225UESRS(E0.03)				√				√							√												
0.225UESRS(E0.05)								√							√			√	√			√					
0.225UESRS(I4)	√							√							√							√					
0.225UESRS(I6)								√							√							√					
0.225UESRS(I15)								√							√			√	√			√					
0.3125UESRS(E0.03)	√							√							√			√	√			√	√				
0.3125UESRS(I4)	√							√							√			√				√					
0.3125UESRS(I6)								√							√			√	√			√					
0.3125UESRS(I15)								√							√			√	√								

**ACKNOWLEDGEMENTS**

This work was supported by the Engineering and Physical Sciences Research Council through a Doctoral Training Award, and NATS Ltd. I am grateful to the University of Nottingham for giving me the opportunity to take part in this project.

**REFERENCES RÉFÉRENCES REFERENCIAS**

1. Ascó A (2013) Constructive and evolutionary algorithms for airport baggage sorting station and

gate assignment problems. PhD thesis, School of Computer Science.  
 2. Ascó A (2016) An analysis of robustness approaches for the airport baggage sorting station assignment problem. Journal of Optimization 2016, DOI 10.1155/2016/1213949.  
 3. Ascó A (2018) An evolutionary algorithm and operators for the airport baggage sorting station problem. Soft Computing DOI 10.1007/s00500-018-3561-7, URL https:// doi.org/10.1007/s00500-018-3561-7.

4. Ascó A (2019) Steady state evolutionary algorithm and operators for the airport gate assignment problem. *Int J Adv Robot Automn* 4(1): 24, DOI 10.15226/2473-3032/4/1/00139, URL <https://symbiosisonlinepublishing.com/robotics-automation/robotics-automation39.php> # References.
5. Ascó A, Atkin JAD, Burke EK (2013) An analysis of constructive algorithms for the airport baggage sorting station assignment problem. *Journal of Scheduling* DOI 10.1007/s10951-013-0361-x.
6. Bolat A (1999) Assigning arriving flights at an airport to the available gates. *Journal of the Operational Research Society* 50(1):23 – 34.
7. Dorndorf U (2002) Project Scheduling with Time Windows: From Theory to Applications. *Contributions to Management Science*, Physica Verlag.
8. Dorndorf U, Pesch E, Phan-Huy T (2000) Constraint propagation techniques for the disjunctive scheduling problem. *Artificial Intelligence* 122 (1–2): 189–240, DOI 10.1016/S0004-3702(00)00040-0, URL <http://www.sciencedirect.com/science/article/pii/S0004370200000400>.
9. Ghazouani H, Hammami M, Korbaa O (2015) Solving airport gate assignment problem using genetic algorithms approach. In: 2015 4th International Conference on Advanced Logistics and Transport (ICALT), pp 175–180, DOI 10.1109/ICAdLT.2015.7136615.
10. Hu X, Di Paolo E (2007) An efficient genetic algorithm with uniform crossover for the multiobjective airport gate assignment problem. In: *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pp 55–62, DOI 10.1109/CEC.2007.4424454.
11. Jaehn F (2010) Solving the flight gate assignment problem using dynamic programming. *Z Betriebswirtsch* 80:1027-1039, DOI 10.1007/s11573-010-0396-9.
12. Kim SH, Feron E (2013) Impact of gate assignment on gate-holding departure control strategies. *CoRR abs/1306.3429*.
13. Li C (2009) Airport gate assignment – a hybrid model and implementation. *CoRR abs/0903.2528:5*, URL <http://arxiv.org/abs/0903.2528>.
14. Lim A, Wang F (2005) Robust airport gate assignment. In: *Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on*, pp 8 pp. –81, DOI 10.1109/ICTAI.2005.110.
15. Lim A, Rodrigues B, Zhu Y (2005) Airport gate scheduling with time windows. *Artificial Intelligence Review* 24(1):5 – 31, DOI 10.1007/s10462-004-7190-4.
16. Mangoubi RS, Mathaisel DFX (1985) Optimizing gate assignments at airport terminals. *TRANSPORTATION SCIENCE* 19:173 – 188, DOI 10.1287/trsc.19.2.173.
17. Ruszczyński A, Shapiro A (2003) Stochastic programming, *Handbooks in Operations Research and Management Science*, vol 10. Elsevier
18. Van Schaijk ORP, Visser HG (2017) Robust flight-to gate assignment using flight presence probabilities. *Transportation Planning and Technology* 40(8): 928–945, DOI 10.1080/03081060.2017.1355887, URL <https://doi.org/10.1080/03081060.2017.1355887>.
19. Seker M, Noyan N (2012) Stochastic optimization models for the airport gate assignment problem. *Transportation Research Part E: Logistics and Transportation Review* 48(2): 438 – 459, DOI 10.1016/j.tre.2011.10.008, URL <http://www.sciencedirect.com/science/article/pii/S136654511001335>.
20. Yan S, Tang CH (2007) A heuristic approach for airport gate assignments for stochastic flight delays. *European Journal of Operational Research* 180(2): 547 – 567, DOI 10.1016/j.ejor.2006.05.002, URL <http://www.sciencedirect.com/science/article/pii/S037721706003316>.