4OR-Q J Oper Res
DOI 10.1007/s10288-009-0112-1

INDUSTRY

A gate reassignment framework for real time flight delays

Ching-Hui Tang · Shangyao Yan · Yu-Zhou Hou

Received: 9 June 2009 / Revised: 4 November 2009
© Springer-Verlag 2009

Abstract Flight delays (such as early or late arrivals and late departures) are a frequent occurrence in actual day to day airport operations and it is often not possible to assign such flights to their original gates. Flight delay information may also vary with time. As a consequence, the airport authority needs to reassign flights to different gates in real-time. The traditional manual flight reassignment method is neither efficient nor effective in cases with variable flight delay information, and the reassignment is frequently time constrained. Consequently, in this study the goal is to develop a gate reassignment framework, and a systematic computerized tool, for repeatedly handling gate reassignments given varied flight delay information. The results of a test case related to Taiwan international airport operations, show that the proposed framework performs well.

Keywords Airport operations · Real-time · Gate reassignment · Flight delays

MSC classification (2000) 90B06 · 90B80 · 90C90

1 Introduction

In airport operations, flight delays are a common occurrence, forcing airport authorities to often make flight to gate reassignments to meet changes in departure/arrival times in real-time. Since these departure/arrival times vary and are updated frequently, the

C.-H. Tang
Department of Air Transportation, Kainan University, Taoyuan 33857, Taiwan

S. Yan (✉) · Y.-Z. Hou
Department of Civil Engineering, National Central University, Chungli 32001, Taiwan
e-mail: t320002@cc.ncu.edu.tw

Published online: 02 December 2009

reassignment process must be immediately and repeatedly performed. Complicated and time-consuming methods are not appropriate for solving gate reassignment problems dynamically under time constraints in real-time operations.

In practice, the airport authority often needs to update and preprocess several relevant elements when there are changes in departure/arrival times. Reassignment results are based on these updated elements. In Taiwan, the reassignment process currently consists of two dependent phases: (1) a preprocessing phase; and (2) a reassignment construction phase. In the first phase, flight departure/arrival times are updated and other related elements, such as the number of available gates, the available time window for each gate, the number of disrupted flights, and the flight schedules, are pre-processed. Then, using the planned assignment and the above updated elements as a basis, reassignment is performed during the reassignment construction phase.

At Taiwan Taoyuan International Airport (TTY Airport, a major international airport), gate reassignment is currently carried out by airport staff. The preprocessing and reassignment construction phases described above are both performed manually. The method is neither efficient nor effective in terms of system optimization, and often leads to an inferior solution. This manual method is also difficult to apply to cases with varied flight delay information, and the reassignment process frequently needs to be performed under time constraints in real-time. It becomes desirable to have systematic computerized tools to carry out an automatic decision-making process, combining the reprocessing and reassignment construction phases together, for handling reassignment processes in real-time operations.

Gate assignment problems have already been studied by many researchers. In particular, a number of analytical gate flight assignment models, generally formulated as zero-one integer (linear or quadratic) programs, mixed integer programs or network flow problems, have been developed. For example, see Braaksma (1977), Babic et al. (1984), Mangoubi and Mathaisel (1985), Vanderstraetan and Bergeron (1988), Bihr (1990), Zhang et al. (1994), Yan and Chang (1998), Haghani and Chen (1998), Bolat (1999, 2000), Yan and Huo (2001), Ding et al. (2005), and Dorndorf et al. (2007). In addition to the above analytical models, Yan and Tang (2007) developed an integrated framework where they combined the planning and real-time stages together to solve for planned gate assignments sensitive to stochastic flight delays. However, their focus was on the production of a good planned assignment rather than a good reassignment. The differences between this current study and the recent developments introduced by Dorndorf et al. (2007) and Yan and Tang (2007) will be discussed in more detail in Sect. 3.3.

Some studies have also focused on airport gate reassignment or assignment problems under uncertain information. For example, see Hamzawi (1986), Brazile and Swigger (1988, 1991), Gosling (1990), Srihari and Muthukrishnan (1991), Su and Srihari (1993), Jo et al. (1997), Cheng (1997), and Gu and Chung (1999). Yan et al. (2002) also proposed a simulation framework with which to evaluate the interrelationship between the planned and the real-time gate assignments necessary to meet the stochastic flight delays that occur in real operations. However, they did not go further, to solve for gate reassignment under real-time flight delays.

In this study, the gate reassignment problem is considered from the airport authority's perspective. The focus is not on the development of complicated and time-consuming mathematical models or solution approaches which are not suitable for real-time operations, rather we aim to develop a gate reassignment framework for real time flight delays (GRFRTFD) which is embedded in a systematic and efficient computerized tool, where the preprocessing and reassignment construction (incorporating a flight delay gate reassignment model) phases are integrated together. The GRFRTFD is compiled and the two phases are linked using the C computer language, coupled with the CPLEX 10.0 mathematical programming solver. Unlike the traditional manual method, in the GRFRTFD, the reprocessing and reassignment construction phases are linked together, so that gate reassignments can be repeatedly handled in accordance with varied flight delay information.

The goal of this research is to develop the GRFRTFD to be an improvement over the current limited manual process. The practicality of its performance is mainly demonstrated by a comparison between it and the current manual method used for TTY Airport operations. A lower bound solution is also developed to help understand the theoretical performance of the GRFRTFD. In addition, we only address disturbances that are due to flight delays that occur in regular operations, rather than other types of disturbances, such as aircraft malfunctions, airport closures or unexpected airport congestion. However, adopting the model to fit other types of disturbances could be a direction of future research.

The remainder of the paper is organized as follows. In Sect. 2, we discuss the context of the problem. In Sect. 3, we introduce the GRFRTFD. In Sect. 4, a numerical test, based on TTY Airport operations, is performed. Finally, some conclusions are offered in Sect. 5.

2 Context of the problem

We first discuss a number of pertinent elements (based on TTY Airport operations), which include the buffer time, starting and ending times, gate assignment constraints, maximum delay time, objective, as well as the inconsistency value. Our GRFRTFD is constructed based on TTY Airport operations. Certainly, it can be similarly applied to other airports with their own specific needs.

(1) The buffer time, starting time and ending time

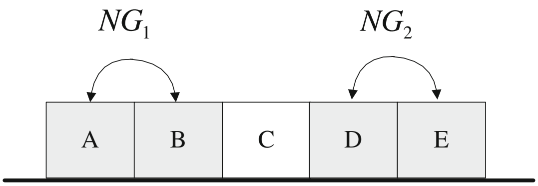
The buffer time is the period of time needed at the gate between two consecutive flights. The flight's starting time (a time point) is set to be when the gates are ready for the associated aircraft to use, and the ending time (a time point) is when the gates are released from use and ready for another. Generally, the time window between the starting and ending times comprises the time block for which a flight is assigned to a gate. For simplicity, the starting and ending times can be calculated as in Table 1. In addition, in current TTY Airport operations, the buffer time is usually 30 min; the ground service time for each flight is usually 30, 60 or 90 min according to the type of aircraft, number of passengers, and so on. Thus the buffer time and the ground service time are relative at TTY Airport and thus stochastic buffer times and ground service times are not considered in this research.

Table 1 Starting and ending times for each flight

	Starting time	Ending time
Arriving flight	Actual arrival time minus half of the buffer time	Actual arrival time, plus the passenger or package deplaning time, plus half of the buffer time
Departing flight	Actual departure time minus ground service time necessary for checking the aircraft prior to departure, for fueling and passenger or package boarding, and half of the buffer time	Actual departure time plus half of the buffer time
Transferring flight	Actual arrival time minus half of the buffer time	Actual departure time plus half of the buffer time

Note: A transferring flight forms a flight pair. There is a short connection time between a flight pair, so the pair needs, in practice, to be assigned to the same gate. Hence, the starting and ending times for a transferring flight can be calculated as above. For example, at TTY a short connection time is usually one hour. Otherwise, the staff classify the flight as a departing flight and an arriving flight, which need not be assigned to the same gate.

Fig. 1 Example of adjacent gate pairs



(2) The gate assignment constraints

Two classes of constraints are commonly required in gate assignment problems: (1) flight constraint: every flight must be assigned to one and only one gate; and (2) gate constraint: no two flights can be assigned concurrently to the same gate. In addition to the two constraints, here, we also consider an adjacency constraint. This means that a specific flight pair (for convenience, let us call them conflicting flights hereafter) cannot be assigned concurrently to two adjacent gates, due to various factors (for example, the wing span of the aircraft). To do this, we design a set N_q which indicates the set of gates for the qth adjacent gate pair. Then, conflicting flights cannot be concurrently assigned to gates which are elements of N_q, for each adjacent gate pair q. Figure 1 shows that flights 1 and 2 are conflicting. Let us also suppose that gates A and B, as well as gates D and E, are adjacent gate pairs (i.e., $q = 1$, $N_1 = \{A, B\}$; $q = 2$, $N_2 = \{D, E\}$). Thus, flights 1 and 2 cannot be concurrently assigned to gates A and B or gates D and E.

(3) The maximum delay time

In addition to the reassignment of flights to alternative gates, if there are not enough gates (especially likely during peak hours) the airport authority may have to delay the starting times of some flights (i.e., hold an aircraft to wait for a gate). To lessen the impact on the timetable, the passenger waiting time and the level of service, there is usually a maximum allowable delay time, determined by the airport's own considerations. The maximum delay time at TTY Airport is 30 min.

(4) Objective

If an aircraft cannot be assigned to its planned gate or time point due to flight delays, then the deviation of reassignment is considered inconsistent. The amount that the current assignment deviates from the reference can be regarded as an objective (Dorndorf et al. 2007). In practice, this inconsistency has an influence on service and/or system performance. For example, holding an aircraft to wait for an available gate will increase passenger waiting time. For most airport authorities (including TTY Airport), this type of inconsistency should be considered during the reassignment process and needs to be reduced to a minimum. Therefore, our objective is to minimize total inconsistencies brought about by reassignment.

(5) The inconsistency value

To design an objective function for minimizing total inconsistencies, the types of inconsistencies and the associated inconsistency values first have to be defined. We consider two types of inconsistencies, a space inconsistency and a time inconsistency. A space inconsistency occurs when a flight is reassigned to a gate different than the original one. A time inconsistency occurs when a flight's starting time is altered. However, in practice, these two inconsistency values are not easy to evaluate, since they are related to complicated factors as the feelings of passengers, inconvenience to airport and airline staff, the efficient use and flexible adjustment of ground service personnel, and of course, the airport's level of service. For ease of modeling, the space inconsistency value is represented by a disturbance time for the reassignment of a flight to an alternate gate. After repeated discussion with TTY Airport staff and based on their suggestions, the disturbance time for all flights reassigned to alternate gates is set to be 30 min. The time inconsistency value is also estimated as a time variation (i.e., the difference between the reassigned and planned starting times) for each flight.

Figure 2 shows an example of the calculation of the space and time inconsistency values. In the planned assignment, Flight 1 is originally assigned to Gate 2 at a starting time of 8:20, indicated by the dotted line. For simplicity, we take three possible reassignments as examples. The inconsistency values for Flight 1 are shown in italics. Both time and space inconsistencies occur for Reassignment 3 (Flight 1 is reassigned to Gate 3 at 8:40). Hence, the inconsistency values for Flight 1 are $20 + 30 = 50$. Note that the trade-off between the space and time inconsistencies should be determined by airport authorities according to their own characteristics and considerations. This will be discussed in more detail in Sect. 4.4.

It should be mentioned that each stage in the model used in the reassignment construction phase needs to be solved with an updated flight arrival/departure time (this will be discussed in greater detail later). To prevent developing a complicated model, with difficult GRFRTFD processes, the objective function is designed as a linear function based on the inconsistency values acceptable to the TTY Airport. The resultant model used in the reassignment construction phase becomes a linear integer program which can be efficiently solved using a mathematical programming solver, such as CPLEX. If the inconsistency values are adjusted to incorporate varied ones, then a

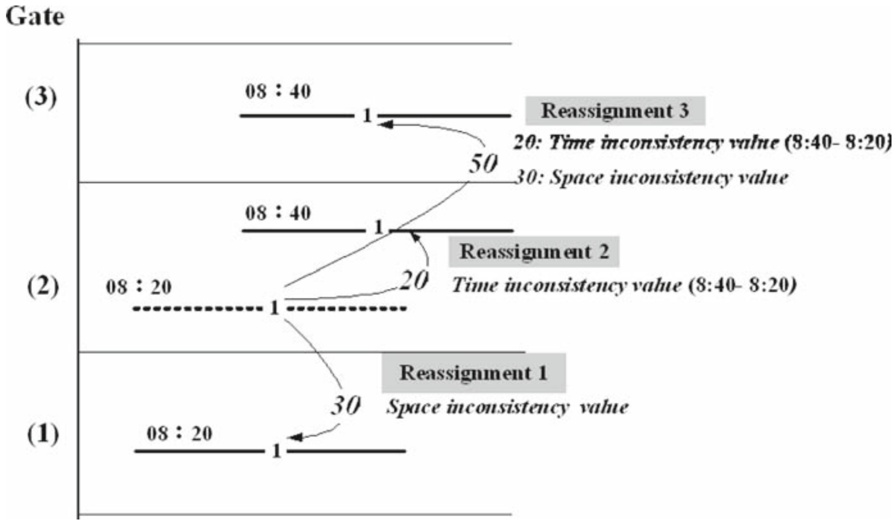


Fig. 2 Calculation of space and time inconsistency value

complicated inconsistency function (typically a nonlinear function) should be developed. The problem then becomes a complicated nonlinear integer program, which would be significantly more time-consuming to solve than the current one. It would also be difficult to put to practical use given the time constraints that characterize real time operations. This issue should be kept in mind when developing a practical gate reassignment framework.

3 The GRFRTFD

In this study we develop a computerized method, which integrates the preprocessing and reassignment construction phases together, to solve for gate reassignments in accordance with varied flight delay information. We first address the assumptions used in the GRFRTFD, based on TTY Airport practices. Then we describe the process followed by a discussion of the GRFRTFD application. Finally, the position of the method with respect to current research is discussed.

We make the following assumptions based on actual TTY Airport practices:

(1) The peak flow will not be more than the airport capacity. Hence, we assume that under normal conditions any flight can be assigned to a gate. That is, the cancellation strategy is not considered in our reassignment.
(2) In the real world airlines may prefer to use certain gates. To prevent developing a complicated flight-gate preference function, which is not the focus of this study, airline gate assignment preference is not considered here. Note that this assumption does not exclude the constraint that some flights may need to be assigned to certain gates for other reason, such as an agreement between the airline and the airport, the aircraft type, and so on. In other words, each flight has feasible gates to which it can be assigned, and the assignment of each flight to a feasible gate is regarded as identical.

(3) For ease of modeling, a gap of 5 min is used in the reassignment; flights are usually scheduled in units of 5 or 10 min. Thus, a flight has 7 candidate time points (an allowable delay time from 0 to 30 min) from which to choose a suitable starting time.

3.1 The GRFRTFD process

Before performing the GRFRTFD process, we need to determine the planning time for the first stage, which is the point in time when the airport authority must start to plan the reassignments. After this, the planning times of other stages can be determined accordingly, with an incremental time gap. In practice, a lead time is usually needed to handle the reassignment process. The first stage's planning time is set as when daily airport operations begin (or the starting time of the first daily gate assignment), preceded by a lead time. Here, the planning time is set to be 1:00 AM. Both the lead time and time gap are set to be 30 min.

The process of the GRFRTFD is described below and shown in Fig. 3. The planned gate assignments at Stage 1 serve as the basis of reassignments. The relevant elements, including flight delay information, the available gates and the considered flights (which will be discussed in more detail later) are updated in the preprocessing phase. Then, in the reassignment construction phase, the flight delay gate reassignment model is solved using CPLEX to obtain a new reassignment, based on the updated elements. The preprocessing phase is again performed at later stages, with the reassignments obtained from the previous stages serving as the basis. The reassignment construction phase is then performed again to obtain new gate reassignments. The process is designed to automatically and chronologically handle the preprocessing and reassignment construction phases for each stage, until the end of daily operations, or until the last flight in daily gate assignments is considered. Note that the objective value obtained for each stage is not the real objective value for daily operations. The real objective value has to be estimated after the GRFRTFD is stopped. To do this, calculate the time and space inconsistency values for each flight, by comparing its original planned assignment with its final reassignment results. After this, the real objective value can be estimated by summing up all flight times and space inconsistency values. The two phases are discussed in greater detail below.

(1) Preprocessing phase

As described above, we need to update the relevant elements in the preprocessing phase. In particular, in addition to the flight delay information, the available gates and the considered flights are major elements that need to be updated at each stage.

a. The available gates are defined as those which have an available idle time, to which a flight can be assigned for each stage. The available idle time will be different for each gate due to variations in flight delays, the reassignment results from the previous stage, and so on.
b. The considered flights are those considered for reassignment at each stage. Taking the lead time described above into account, the flights considered are those that have starting times later than that stage's planning time plus the lead time (30 min).

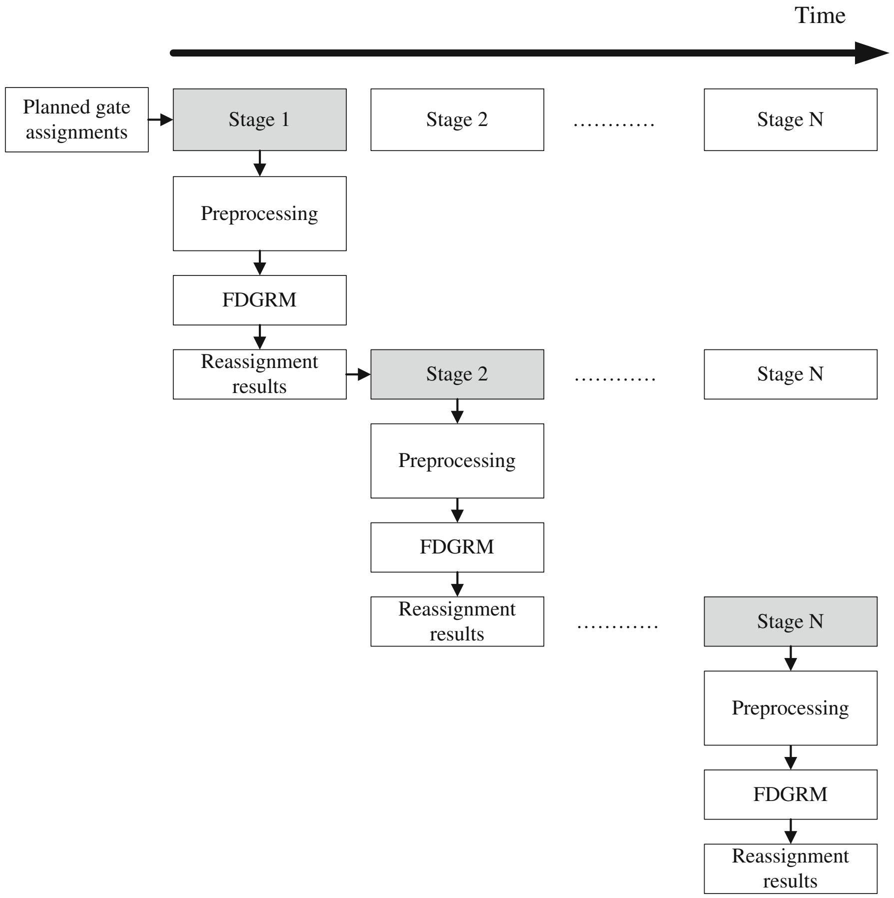


Fig. 3 GRFRTFD process

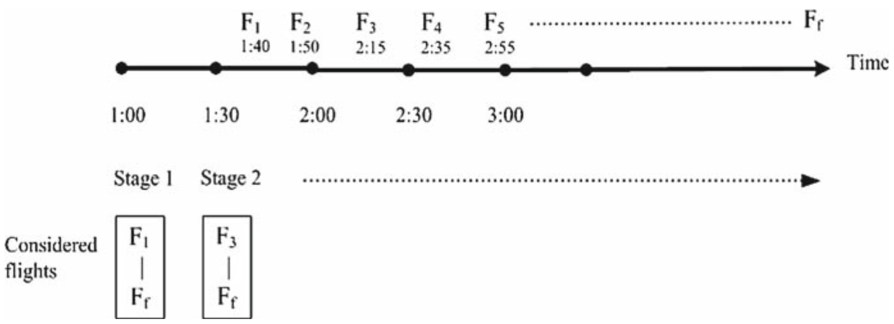


Fig. 4 Demonstration of considered flights

For example, as can be seen in Fig. 4, the planning times for stages 1 and 2 are 1:00 A.M. and 1:30 A.M., respectively. Suppose that there are f flights, F_1 to F_f. Thus, all f flights are considered at stage 1; while F_3 to F_f are considered at stage

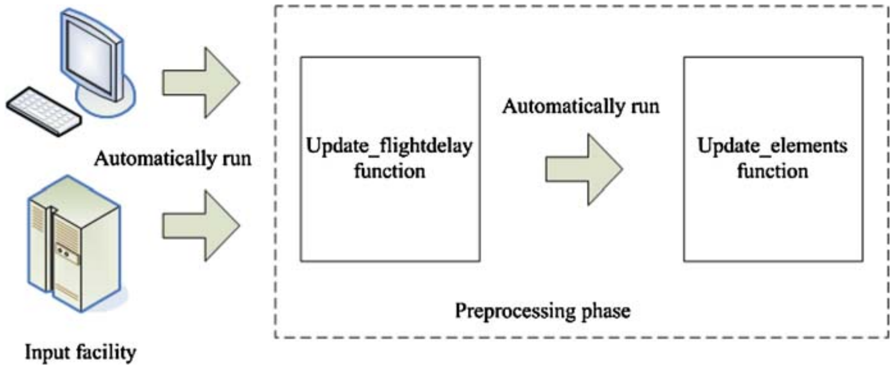


Fig. 5 Manipulation of the two functions in the preprocessing phase

2. The considered flights at other stages can be determined accordingly. Note that, in this case, F_1 and F_2 act according to the reassignments obtained from stage 1.

The two computer functions of the preprocessing phase, update_flightdelay and update_elements, are programmed (using the C computer language). The update_flightdelay function is used to save and to update flight delay data (i.e., flight departure/arrival times). An input facility, such as a flight departure or arrival time report system, is used to connect the update_flightdelay function. The flight delay data will be automatically compiled and saved in the array format used in the computer program. In addition, the update_elements function is used to update relevant elements, especially for available gates and considered flights. The available gates and flights considered at each stage will be automatically updated, according to the other elements, such as the planning time, the variations in flight delays (from the update_flightdelay function), and the reassignments for the previous stage. The manipulations of the two functions are shown in Fig. 5. Note that the two functions are designed to run automatically without any manual calculation, when the framework is put to practical use.

(2) Reassignment construction phase

The flight delay gate reassignment model based on the updated elements is solved in the reassignment construction phase to obtain a new gate reassignment using the CPLEX 10.0 Callable Library Code, including a series of computer functions (ILOG CPLEX 2006). Because the preprocessing and the reassignment construction phases are integrated and linked in a do-while loop, the model can be automatically and repeatedly solved at each stage. Before introducing the model formulation, the notations and symbols used are listed.

Decision variables:

x_{ijk}: assignment variable; 1 if the ith flight is assigned to the jth gate at the kth time point (starting time); 0 otherwise;

Parameters and sets:

d_{ik}: the time inconsistency value that indicates that the ith flight is assigned at the kth time point (starting time); if k is equal to the original time point, then $d_{ik}=0$;

w_{ij}: the space inconsistency value that indicates that the ith flight is assigned to the jth gate; if j equals to the original gate, then $w_{ij} = 0$;
I: the set of considered flights;
G: the set of available gates;
E_i: the set of gates that the ith flight can be assigned;
D_{ij}: the set of time points that the ith flight can be assigned to the jth gate;
F_{js}: the set of flights that can be assigned to the jth gate so that their time windows will cover the sth time point;
S: the set of all time points (i.e., the time points from the planning time at each stage to the end of daily operations);
H_{is}: the set of time points (starting times) assigned to the ith flight, where the resulting time windows cover the sth time point;
T_q : the set of conflicting flight pairs for the qth adjacent gate pair;
L_{tq}: the set of flights included in the tth conflicting flight pair for the qth adjacent gate pair;
Q: the set of adjacent gate pairs.

The model is formulated as a zero-one integer program as follows:
Min

$$Z = \sum_{i \in I} \sum_{j \in E_i} \sum_{k \in D_{ij}} d_{ik} x_{ijk} + \sum_{i \in I} \sum_{j \in E_i} \sum_{k \in D_{ij}} w_{ij} x_{ijk} \tag{1}$$

s.t.

$$\sum_{j \in E_i} \sum_{k \in D_{ij}} x_{ijk} = 1, \quad \forall i \in I, \tag{2}$$

$$\sum_{i \in F_{js}} \sum_{k \in H_{is}} x_{ijk} \le 1, \quad \forall j \in G, \forall s \in S, \tag{3}$$

$$\sum_{i \in L_{tq}} \sum_{j \in N_q} \sum_{k \in H_{is}} x_{ijk} \le 1, \quad \forall t \in T_q, \forall q \in Q, \forall s \in S, \tag{4}$$

$$x_{ijk} = 0 \; or \; 1, \quad \forall k \in D_{ij}, \forall j \in E_i, \forall i \in I. \tag{5}$$

The objective function (1) is designed to minimize the total inconsistency values for each stage. The first and second terms are the total time and space inconsistency values, respectively. Constraint (2) is the flight constraint, indicating that every flight is exactly assigned to a gate. Constraint (3) is the gate constraint, ensuring that every gate is assigned to at most one flight at any time. Constraint (4) is the gate adjacency constraint, denoting that two conflict flights cannot be concurrently assigned to an adjacent gate pair. Constraint (5) indicates that the assignment variables are either zero or one.

It should be mentioned that the above model is used to develop a lower bound to the solution. This helps to evaluate the theoretical effectiveness of the GRFRTD. It is assumed that all flight departure/arrival times can be realized in advance, the so-called "perfect information" situation. The solution obtained based on perfect information provides a lower bound to the problem. In this study, the realization of all flights' departure/arrival times at the end of daily operations is regarded as perfect information. In other words, we directly solve the above model, using such flights' updated departure/arrival times as input, to obtain the lower bound solution.

3.2 GRFRTFD application issues

Our GRFRTFD is designed based on TTY Airport operations. However, with suitable modifications it is expected to become more general and thus applicable for other airports:

(1) The objectives can be suitably modified for other considerations. Since the standard formulation of the model is used in the GRFRTFD, the objective function can be easily changed to include typical objectives, such as the total passenger walking distance, the number of aircraft towing procedures, the variance in idle gate times, the airline preference, or a combination of the above.

(2) The cancellation strategy is not considered here. However, if such a strategy needs to be applied at other airports, then it can be incorporated into the GRFRTFD. Artificial gates with infinite inconsistency values can be added to the model in the reassignment construction phase. If an optimal solution contains flight assignments to these artificial gates, this means that the number of gates is insufficient in which case the cancellation strategy needs to be used (e.g., to cancel the flights assigned to artificial gates) to solve such gate shortages.

(3) If airport authorities consider the robustness issue in the reassignments, and then a robust model can be incorporated into the GRFRTFD. For example, the semi-deviation risk measure (SRM, Ruszczynski and Shapiro 2003) could be added to the objective function, which is represented below. Certainly, other types of robustness concepts can also be applied. The interested reader can refer to Mulvey et al. (1995), Ruszczynski and Shapiro (2003) and Dorndorf et al. (2007) for a more detailed description.

$$
\begin{aligned}
\mathrm{SRM} &= E[(z^s - E[z^s])_+] = E\left[\max(0, z^s - E[z^s])\right] \\
&= E\left[\max\left(0, \sum_{i\in I}\sum_{j\in E_i}\sum_{k\in D_{ij}} d_{ik}x^s_{ijk} + \sum_{i\in I}\sum_{j\in E_i}\sum_{k\in D_{ij}} w_{ij}x^s_{ijk}\right.\right. \\
&\qquad \left.\left. -E\left[\sum_{i\in I}\sum_{j\in E_i}\sum_{k\in D_{ij}} d_{ik}x^s_{ijk} + \sum_{i\in I}\sum_{j\in E_i}\sum_{k\in D_{ij}} w_{ij}x^s_{ijk}\right]\right)\right]
\end{aligned} \tag{6}
$$

Where:

s: the sth stochastic flight delay scenario;

x^s_{ijk}: assignment variable; 1 if the ith flight is assigned to the jth gate at the kth time point (starting time), for the sth stochastic flight delay scenario; 0 otherwise;

z^s: the objective value for the sth flight delay stochastic scenario. In this study z^s can be represented as $\sum_{i \in I} \sum_{j \in E_i} \sum_{k \in D_{ij}} d_{ik} x^s_{ijk} + \sum_{i \in I} \sum_{j \in E_i} \sum_{k \in D_{ij}}^{w_{ij} x^s_{ijk}}$;

$E\left[z^s\right]$: the expected objective value of all stochastic flight delay scenarios.

(4) The flexibility of the GRFRTFD process makes it easy to design the planning time, lead time, and time gap to meet an airport's practical needs or considerations. For example, a variable lead time or time gap can be set by taking into consideration peak or non peak hours.

(5) The design of considered flights and available gates at each stage can be adjusted for other airports, so that they can handle and control the problem size at each stage according to their own considerations, requirements, and practical needs. Note that a reduction in number of considered flights and available gates results in a reduction in problem size which increases solution efficiency and applicability for real-time operations.

3.3 Position of the GRFRTFD

To better understand the position of the GRFRTFD with respect to other recent research, we discuss differences between it and recent methods developed by Dorndorf et al. (2007) and Yan and Tang (2007). We found that most other research has been focused on the planned gate assignment rather than the gate reassignment as is looked at here. In general, there are several differences between gate reassignment and planned gate assignment problems:

(1) In real time operations gate reassignments are typically time constrained. An optimal but time-consuming framework or model is difficult to apply for reassignment problems. However, there is usually sufficient time, at least one day before the operating day, to handle the daily planned gate assignment.

(2) There is usually no pre-assignment information required when designing a planned assignment (e.g., see Mangoubi and Mathaisel (1985); Yan and Tang (2007)). However, to perform gate reassignment, a planned assignment is needed. It serves as the measure or basis, to avoid or lessen cascading downstream impact and disturbances to the original planned assignment. In other words, a planned assignment usually serves as the input to a reassignment.

(3) A reassignment is performed to solve perturbations that occur in "current" operations, meaning that it cannot be pre-preformed before these perturbations occur. In contrast, a planned assignment is completed in advance for "future" operations.

In addition to the above differences between reassignment and planned assignment problems, other differences as noted in Dorndorf et al. (2007) and Yan and Tang (2007) are pointed out below.

(1) Most past studies have focused on the development of mathematical models and solution techniques for planned gate assignment problems (Dorndorf et al. 2007). The GRFRTD, which is a computerized decision process, has been developed to automatically and repeatedly solve gate reassignment problems that occur as a result of flight delays in real time environments.

(2) Dorndorf et al. (2007) formulated two multiple objective models, in accordance with the strategies usually applied in American and European airports. The two objective functions considered the minimization of both the number of flights which are not assigned to terminal gates and total passenger walking distances, as well as the maximization of the assignment preference plus the minimization of the number of towing operations and the deviation from the reference gate assignment. However, in our GRFRTD, after repeated discussion with TTY Airport staff, the single objective is designed to minimize the time and space inconsistency values, which is more suitable for the current TTY Airport practices.

(3) In Dorndorf et al. (2007) review, one of the mainstream directions for gate assignment problems is simulation analysis and expert systems construction. However, most of these approaches did not incorporate optimization models, and therefore, are not the same as the approach introduced in this paper.

(4) Yan and Tang (2007) only used a simple manual reassignment rule to perform their reassignment problems embedded in their planning gate assignment framework. Unlike our study, they did not develop an automatic decision process (coupled with an optimization model) suitable for the reassignment situation.

(5) Yan and Tang (2007) framework incorporated a complicated stochastic flight delay gate assignment model which is more time-consuming to solve than ours. It is difficult to put their model to use in our GRFRTD given the time constraints in real time operations. In comparison, the model in our GRFRTD is designed to be compact and computationally efficient (based on operational needs), thus making it applicable for real-time operations.

4 Numerical tests

To test the performance of the GRFRTFD, we performed numerical tests based on TTY Airport operations. The tests were performed on an Intel P4 3.2G with 1 GB RAM in the environment of Microsoft Windows XP.

4.1 Data analyses and test results

The planned gate assignment data used for the testing were based on the operating day, Sep. 24, 2005 and included 171 flights and 14 aircraft types, which could be classified into two main types, large (specifically, B747s) and wild (all other aircraft types), corresponding to the two types of gates at the TTY airport. There were 49 arrival flights (5 served by large aircraft), 50 departure flights (8 served by large aircraft), and 72 transferring flights (24 served by large aircraft). There were 23 gates in operation, of which 7 were remote gates. 7 of the 23 gates were wild type gates, while the others were large type gates. In addition, to simulate flight delay situations during daily operations, we used the flight delay distributions, obtained from the TTY airport data, to chronologically generate the arrival or departure time for each flight at each stage. Specifically, 88.39% of the departing flight delays ranged from 0 to 20 min while for arriving flights, 75.56% of the delays ranged from −9 (early) to 20 min.

Table 2 Problem scales for stages 1, 20, and 40

Stage	1	20	40
Number of variables	14,692	10,316	3,232
Number of constraints			
Constraint (2)	171	118	36
Constraint (3)	6,486	3,864	1,104
Constraint (4)	1,974	1,176	336
Constraint (5)	14,692	10,316	3,232

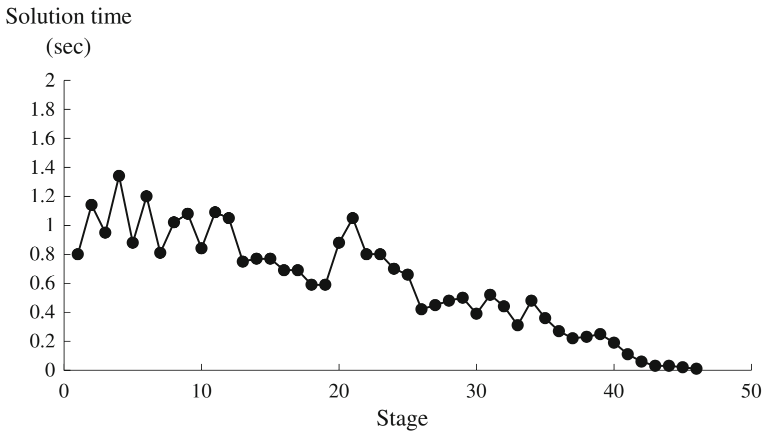


Fig. 6 Solution times for each stage

At each stage for solving the flight delay gate reassignment model, the CPLEX gap between the objective value and the best node (the best objective function value of all the unexplored nodes in the branch-and-bound tree, which can serve as the lower bound of the problem) was set to be 0%. According to the timetable used for the test, the planning period was set from 1:00 A.M. to 24:00 P.M., resulting in 46 stages in the GRFRTFD. Due to the fact that the number of flights considered tended to decrease as the stage number increased, the problem scales at each stage decreased accordingly. For example, all 171 flights were considered in stage 1; however, there were only 118 and 36 flights considered in stages 20 and 40, respectively. Table 2 shows the problem scale for stages 1, 20, and 40. As a result, as shown in Fig. 6, the solution time for each stage also tended to decrease as the stage number increased. In general, the solution time for each stage was about 1 s; even the largest time was still within 1.34 s (stage 4). The results show that the GRFRTFD is efficient for solving gate reassignment problems and could be applied in real time.

We also used the current TTY manual method to solve the problem. The procedure is not described in detail here; the interested reader can refer to Yan and Tang (2007) for a more detailed description. The GRFRTFD's performance was compared with

Table 3 Comparison between the GRFRTFD and the manual method

	GRFRTFD	Manual method
Objective value	390	420
Total space inconsistency value	240	390
Total time inconsistency value	150	30
Number of flights reassigned to alternative gate	8	13
Number of flights reassigned to remote gates	5	8
Number of flights with delayed starting times	7	2

that of the manual method. The improvement percentage (IP) is defined as follows:

$$\text{IP}\ (\%) = \frac{\text{Objective value of manual method} - \text{Objective value of GRFRTFD}}{\text{Objective value of GRFRTFD}} \tag{7}$$

As shown in Table 3, we found that the number of flights with delayed starting times was greater for the GRFRTFD (7 flights) than for the manual method (2 flights), meaning that there was a larger time inconsistency value for this method. However, the number of flights reassigned to alternative gates was less for the GRFRTFD (8 flights, of which 5 were reassigned to remote gates) than for the manual method (13 flights, of which 8 were reassigned to remote gates), meaning that there was a smaller space inconsistency value. We also found that the objective value was better for the GRFRTFD than the manual method; the IP (%) was about $7.69\%(=\frac{420-390}{90})$. These results indicate that in comparison with the manual method, the delaying of these flights in the GRFRTFD avoids the production of a larger space inconsistency value due to the reassignment of flights to other alternative gates, which can help to reduce the total inconsistency value (i.e., provide a better objective value). The comparison also shows that, the GRFRTFD produces a better reassignment plan and would, therefore, be a great improvement in real-time operations. Note that the test results were based on the space and time inconsistency values suggested by the TTY Airport authority (as mentioned in Sect. 2). The consequences would vary with different inconsistency values for other airports. Although the test results show the potential usefulness of the GRFRTFD to handle gate reassignments with flight delays, more tests should be performed for other airports, with other space and time inconsistency values. This would help to better evaluate the GRFRTFD and to understand its characteristics and limitations for other applications.

4.2 Different evaluation days

In theory, the scenario used in the above tests is one possible set of realizations that can occur in actual operations. Here, we also evaluate the performance of the proposed GRFRTFD and the current TTY manual method using a different set of 35 evaluation days. We randomly generate each flight's arrival/departure time for each

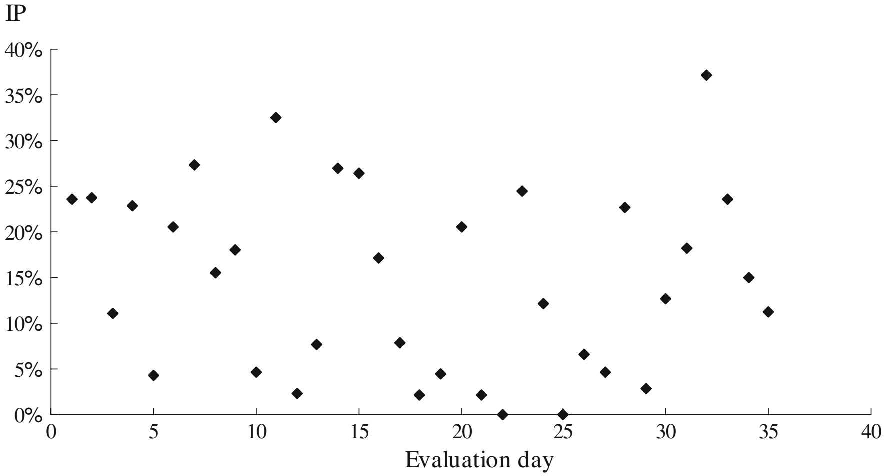


Fig. 7 IPs for different evaluation days

evaluation day. Note that the flight's arrival/departure times for these 35 evaluation days differed from that for the scenario discussed in the above section. As shown in Fig. 7, all the IPs (%) for the 35 evaluation days were positive, ranging from 0 to 37.14%. The average IP for the 35 evaluation days was about 14.67%. In particular, the IPs for 22 and 13 evaluation days were larger than 10 and 20%, respectively. This result shows that the GRFRTFD also significantly out-performs the current manual method for different evaluation days and would thus be helpful to enhance airport operations.

In addition, to measure the theoretical effectiveness of the GRFRTFD, we used the lower bound solution, as discussed in Sect. 3.1, to estimate a worst error gap (WEG) for the obtained solution. The WEG is defined as follows:

$$\text{WEG (\%)} = \frac{\text{Objective value of GRFRTFM or manual method} - \text{Lower bound solution}}{\text{Lower bound solution}} \tag{8}$$

The results are shown in Fig. 8. For the GRFRTFD, there were 11 evaluation days where the WEGs were smaller than 3%, although there were 4 evaluation days with a WEG larger than 15% (evaluation days 3, 5, 12, and 22). In particular, the WEGs of evaluation days 16 and 31 are 0%, showing that the GRFRTFD obtained the optimal solution, from a theoretical point of view. These results also show that the method has the potential to be used for effectively and efficiently solving the problem. However, such findings could not be obtained for the manual method. There was no evaluation day with a WEG smaller than 3% (the smallest WEG was still high, 4.44%). As well, the average WEG of the evaluation days obtained by the GRFRTFD (7.63%) is much better than that obtained by the manual method (23.23%). The results again show that the superiority of the performance of our method to that of the manual method, regardless of the evaluation days.

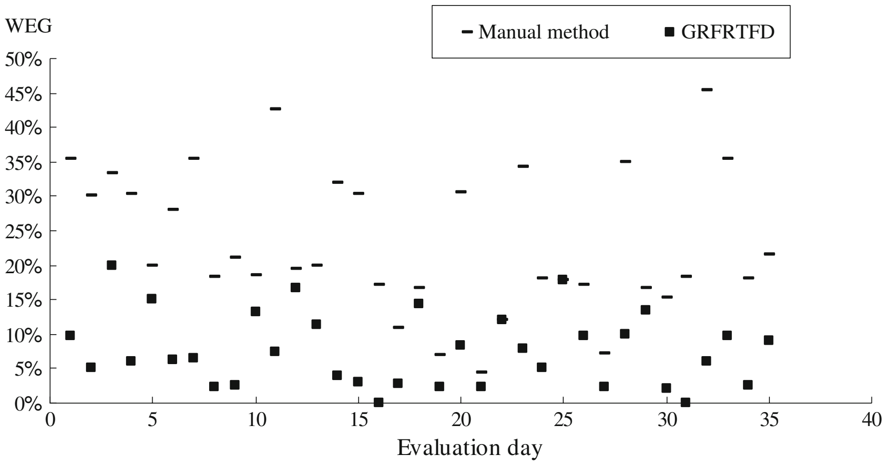


Fig. 8 WEGs for different evaluation days

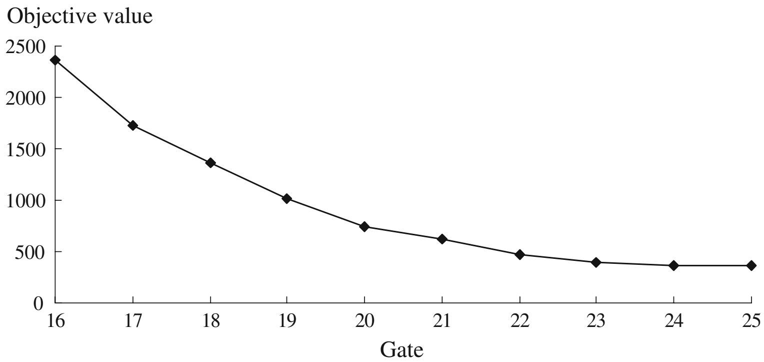


Fig. 9 Objective values for different numbers of gates

4.3 The number of gates

We performed sensitivity analysis of the number of gates (from 16 to 25), resulting in 10 scenarios. As shown in Fig. 9, as the number of gates decreased, the objective values increased nonlinearly. For example, when the number of gates decreased from 24 to 23, the incremental percentage of the objective value was only 5.41% (i.e., the objective value increased from 370 to 390). However, when the number of gates decreased from 17 to 16, the incremental percentage of the objective value was as high as 36.42% (i.e., objective value increased from 1,730 to 2,360). In addition, we found the objective value to be the same (370) after 24 gates, implying that when the number of gates was more than 24, the reassignments could not be improved due to excess gates. All these results are useful as reference material for airport authorities to understand the usage and boundary of the number of gates.

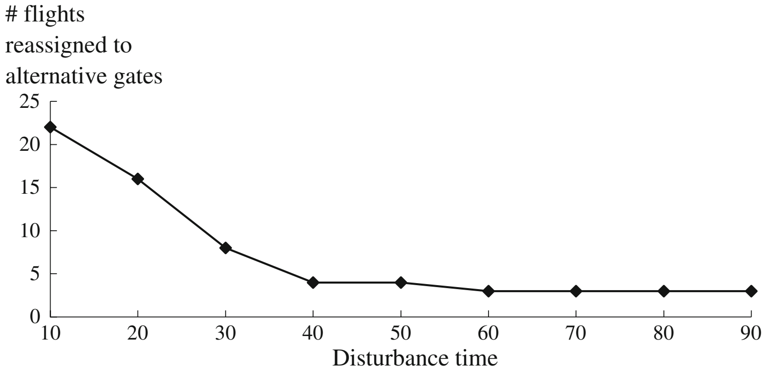


Fig. 10 Number of flights reassigned to alternative gates for different disturbance times

4.4 The disturbance time

The above test results indicate that if the disturbance time is set to be 30 min, this would lead to 8 flights being reassigned to alternative gates in daily operations. To evaluate the influence of different disturbance times on the solution, we tested 9 scenarios, starting from 10 to 90 min, in increments of 10 min. The results are shown in Fig. 10. In particular, when the disturbance times were larger than or equal to 40 min, the number of flights reassigned to alternative gates did not change much (only 3 or 4 flights, because of the limitation of the maximum delay time), showing that flight reassignment to an alternative gate becomes increasingly unfavorable. However, when the disturbance times were less than or equal to 30 min, the number of flights reassigned to alternative gates rose sharply, from 8 (30 min) to 22 flights (10 min). From this, we know that if the airport authority places greater consideration on the reassignments of flights to alternative gates, then the disturbance time should be set larger or equal to 40 min. On the other hand, if the airport authority considers that the delaying of flights to be undesirable, then a smaller disturbance time (or a smaller maximum delay time) could be set. In practice, users could evaluate their own most suitable disturbance time in a similar manner, according to their own problem characteristics and considerations.

5 Conclusions

In this study, we developed a gate reassignment framework, a systematic computerized tool, designed to handle repeated gate reassignments, in accordance with varied flight delay information. As needed for practical processes, the reprocessing and reassignment construction phases are linked together using the C computer language, coupled with the CPLEX 10.0 mathematical programming solver. Numerical tests, utilizing data from a Taiwan international airport's operations, were performed to evaluate the GRFRTFD. The results show that the method is not time-consuming and can be applied in real-time operations. As well, the GRFRTFD significantly out-performs the manual method and will be helpful to enhance airport operations. In consultation with the Taiwan airport staff, the GRFRTFD which has some advantages over the current manual method is qualified for use at that airport in future, although it is not currently applied in actual operations. In addition, several sensitivity analyses, including for the number of gates and the disturbance time, were also performed to gain greater understanding of the influence of different parameters on the solution.

One direction of future research is the development of a supporting interface designed to make the GRFRTFD easier to use in practice; cooperation with a computer system/language expert may be needed for this aspect of the research. Finally, in actual operations, some departing/arrival times might be stochastic and others deterministic. If the GRFRTFD is extended to incorporate such stochastic flight departing/arrival times, then a stochastic flight delay model, which incorporates both stochastic and determinist flight departing/arrival times, needs to be applied in the reassignment construction phase. How to incorporate them into the GRFRTFD, to make it more useful, is a worthy future research topic.

Acknowledgments This research was supported by a grant (NSC-96-2415-H-424-003) from the National Science Council of Taiwan. We thank the TTY Airport personnel for providing the test data and their valuable opinions. We also thank the editor and two anonymous referees for their helpful comments and suggestions on the presentation of this paper.

References

Babic O, Teodorovic D, Tosic V (1984) Aircraft stand assignment to minimize walking. J Transp Eng 110:55–66

Bihr RA (1990) A conceptual solution to the aircraft gate assignment problem using 0,1 linear programming. Comput Ind Eng 19:280–284

Bolat A (1999) Assigning arriving flights at an airport to the available gates. J Oper Res Soc 50:23–34

Bolat A (2000) Procedures for providing robust gate assignments for arriving aircrafts. Eur J Oper Res 120:63–80

Braaksma JP (1977) Reducing walking distance at existing airports. Airport Forum 135(145):135–145

Brazile R, Swigger K (1988) Expert gate scheduling. J IEEE Expert 3:33–39

Brazile R, Swigger K (1991) Generalized heuristic for the gate assignment problem. J Control Comput 19:27–32

Cheng Y (1997) A knowledge-based airport gate assignment system integrated with mathematical programming. Comput Ind Eng 32:837–852

Ding H, Lim A, Rodrigues B (2005) The over-constrained airport gate assignment problem. Compu Oper Res 32:1867–1880

Dorndorf U, Drexl A, Nikulin Y, Pesch E (2007) Flight gate scheduling: state-of-the-art and recent developments. Omega, Int J Manag Sci 35:326–334

Gosling GD (1990) Design of an expert system for aircraft gate assignment. Transp Res A 24:59–69

Gu Y, Chung CA (1999) Genetic algorithm approach to aircraft gate reassignment problem. J Transp Eng 125:384–389

Haghani A, Chen MC (1998) Optimizing gate assignments at airport terminals. Transp Res A 32:437–454

Hamzawi SG (1986) Management and planning of airport gate capacity: a microcomputer-based gate assignment simulation model. Transp Plann Technol 11:189–202

ILOG CPLEX (2006) User's manual. ILOG, France

Jo G, Jung J, Zang C (1997) Expert system for scheduling in an airline gate allocation. Expert Syst Appl 13:275–282

Mangoubi RS, Mathaisel DFX (1985) Optimizing gate assignment at airport terminals. Transp Sci 19:173–188

Mulvey JM, Vanderbei RJ, Zenios SA (1995) Robust optimization of large-scale systems. Oper Res 43:254–281

Ruszczynski A, Shapiro A (2003) Stochastic programming. Elsevier, Amsterdam
Srihari K, Muthukrishnan R (1991) An expert system methodology for aircraft-gate assignment. Comput Ind Eng 21:101–105
Su YY, Srihari K (1993) A knowledge based aircraft-gate assignment advisor. Comput Ind Eng 25:123–126
Vanderstraetan G, Bergeron M (1988) Automatic assignment of aircraft to gates at a terminal. Comput Ind Eng 14:15–25
Yan S, Chang CM (1998) A network model for gate assignment. J Adv Transp 32:176–189
Yan S, Shieh CW, Chen M (2002) A simulation framework for evaluating airport gate assignments. Transp Res A 36:885–898
Yan S, Huo CM (2001) Optimization of multiple objective gate assignments. Transp Res A 35:413–432
Yan S, Tang CH (2007) A heuristic approach for airport gate assignments for stochastic flight delays. Eur J Oper Res 180:547–567
Zhang SX, Cesarone J, Miller FG (1994) A comparative study of an aircraft assignment problem at a large aircraft. Int J Ind Eng 1:203–212