This paper can be cited as "Liu, S, Chen, WH, Liu, J (2016) Robust assignment of airport gates with operational safety constraints, International Journal of Automation and Computing, 13(1), pp.31-41."

International Journal of Automation and Computing X(X), X X, X-X
DOI: XXX

Robust Assignment of Airport Gates with Operational Safety Constraints

Shuo Liu[1] Wen-Hua Chen[1] Jiyin Liu[2]

[1]Department of *Aeronautical and Automotive Engineering*, [2]School of *Business and Economics*, Loughborough University, Loughborough LE11 3TU,UK

Abstract: This paper reviews existing approaches to the airport gate assignment problem (AGAP) and presents an optimization model for the problem considering operational safety constraints. The main objective is to minimize the dispersion of gate idle time periods (to get robust optimization) while ensuring appropriate matching between the size of each aircraft and its assigned gate type and avoiding the potential hazard caused by gate apron operational conflict. Genetic algorithm is adopted to solve the problem. An illustrative example is given to show the effectiveness and efficiency of the algorithm. The algorithm performance is further demonstrated using data of a terminal from PEK airport.

Keywords: Gate assignment problem, operational safety constraints, robust optimization, conflict avoidance, genetic algorithm.

1 Introduction

Airport Gate Assignment Problem (AGAP) focuses on flight-to-gate allocation at airports, which is a critical decision problem in the daily operations of modern air transportation industry. As the global traffic volume increases dramatically, the demand for air transportation grows apace as well [1]. Therefore, airports are increasingly facing capacity pressure. However, expanding airport capacity by planning and building new terminals is a very time-consuming process and cannot easy the capacity pressure in short term. Hence, airport operators or terminal managers have to utilize the limited gates available more effectively to guarantee safe and smooth daily operations at airports.

Optimizing the operation of an airport involves interaction among all airport partners working as a team. Gate assignment is a key activity and most other ground operations are then performed based on its results. Airport gate assignment involves the task of assigning a given set of flights from different airlines to a fixed number of gates available at airport while satisfying some operational requirements and specific constraints [2]. The flights have specified arrival and departure times and other important information including the sizes and types of the serving aircrafts, the numbers of passengers, etc. As a typical hub airport usually handles hundreds of domestic and international flights in each day [3], unreasonable assignments may result in flight delays, poor customer satisfaction, disproportion of gate utilization, ground congestion and safety issues with potential hazards caused by aircraft push-back or taxi conflicts around adjacent gate areas, and even extra cost of fuel for both arriving and departing aircrafts. Extra fuel consumption may increase the exhaust emission as well, especially when the airport capacity is nearly saturated by its present configuration.

As a combinatorial optimization problem, the gate assignment problem is easy to understand but difficult to solve. It is affected by a wide range of different resources running on the airport ground [3], including aircrafts, gates, gate facilities, and various types of service vehicles (cargo, food, fuel, de-icing vehicles and towing tug, etc.) with ground crews. Thus, any decision making for the usage of these interdependent resources will bring different degrees of influence on each section of the overall operation. Moreover, although gate assignment is a static and predetermined process, it has to handle some temporary changes (flight delays and emergency flights) and unexpected events (mechanical fault of aircrafts, manual operating errors and severe weather conditions) under the dynamic and uncertain environment of the airport in the last-minute phase. For instance, a significant delayed arrival of one specific flight may generate a series of problems and lead to a 'domino effect' or traffic standstill throughout the corresponding section of airport operation [4, 5]. In this case, from a practical point of view, an optimal or more efficient gate assignment should be flexible for compensating the minor delays or temporary changes subject to uncertainties.

Due to the uncertainties in real-time operations, another very important element that must be taken into account in gate assignment is operational safety. The present operations rely heavily on the ground crew to observe the movements of aircrafts and other vehicles on the ground and to ensure safety. However, it is not easy for the operators to control every section of the whole operation accurately all the time. Carelessness or other human errors may potentially cause safety hazards in airport daily activities. Meanwhile it is also difficult to solve the safety problem in an exact and efficient manner once conflict or collision happens. Some measures have been taken to improve safety of the airport operations in both theoretical and practical aspects [6, 7]. For example, setting a buffer time between two consecutive flights assigned to the same gate is a recognized measure for providing safety protection and avoiding conflicts at gates.

Safety problems in airport operation may be caused by

Regular paper
Manuscript received date; revised date

various different conflicts, including basic gate occupation conflicts, aircraft ground movement conflicts near the terminal gate area, as well as conflicts on main taxiways in the taxiing sequence problem [6, 8]. Most existing work considering safety issues only deals with the problem after the incident happens and further analyzes the results, rather than tries to avoid conflicts beforehand. In our work, we try to address the issue of conflicts in advance by including operational safety constraints (no two flights with overlapping ground movement times are assigned to adjacent gates) in the model. In this way the problem of ground movement conflicts near the terminal gate area will be avoided in the first place (pre-assignment stage). As can be seen from above, AGAP is more complicated than many traditional scheduling problems to some extent because it involves additional safety constraints apart from time and resource constraints.

The rest of the paper is organized as follows: A literature review is first presented in the next section. Section 3 then gives a brief description of the problem and the details of the safety requirements. In Section 4, an integer programming formulation is provided. The basic idea of the genetic algorithm used to solve the problem is introduced in Section 5. In Section 6, the significance and influence of whether considering safety issues of AGAP is discussed and demonstrated using an illustrative example and the method is further tested using real airport data. Conclusions are given in Section 7.

2 Literature review

In previous research, many mathematical models and techniques have been developed with different objectives and corresponding practical rules and restrictions (either hard or soft) for AGAP. A basic version is modelled as a quadratic assignment problem and proved to be NP-hard [9]. The objectives used for AGAP can be classified into either passenger-oriented or airport-oriented [5, 10, 11, 12, 13]. For the purpose of increasing passengers' satisfaction, AGAP with the first type of objectives are mainly focusing on minimizing the total walking distance of both arriving and departing passengers of the flights and minimizing the number of flights assigned to remote apron stand (un-gated area far away from the terminal building). The basic model with objective of minimizing the overall walking distance considers distance between check-in counter and boarding gate, between disembarking gate and baggage claim, and between arriving and departing flights for transferring passengers. As the basic model becomes mature, researchers turn to focus on using different methods to improve the computational efficiency in solving the problem [3, 4, 14, 15, 16].

On the contrary, the airport-oriented objectives in AGAP concentrate on improving gate utilization and the robustness of the assignment for dealing with sudden changes such as flight delays. Unexpected disruptions including early or late arrivals and late departures have a major impact on the smooth performance of pre-determined AGAP plan. Therefore, instead of using inherent random input parameters to represent the stochastic disruptions, some concepts of achieving robustness of gate assignment are proposed in literatures [17, 18, 19], such as idle time, buffer time, gate conflict, etc. Mangoubi and Mathaisel [12] state that if only considering minimization of the total passenger walking distance in real-time gate assignment problem, then the highly utilized gates may have the weakest performance on absorbing early or late arrival aircrafts, which may also lead to the gate conflict problem for every flight pre-assigned to the same gate with estimated gate occupation time. Yan and Chang [20] also argue the importance of adding a buffer time between flights into the model and demonstrate that it is useful in improving the punctuality of robust schedule between the consecutive flights assigned to the same gate. Alternatively, Bolat [19] considers the objective of minimizing the variance of the idle time. The purpose of his approach is to improve the possibility of uniform distribution for gate utilization, while maintaining robustness in gate assignment at the same time.

Most previous work on airport gate assignment problem do not take into account the safety issues firmly or only touch upon the safety elements separately from the gate assignment problem itself. As a matter of fact, considering safety issues should be given higher priority than efficiency and economic considerations in airport traffic management (ATM). Therefore, different from the most common objectives, some other approaches turn to focus on solving the conflict problems in relevant airport operations [6, 8, 21, 22, 23, 24]. Cheng [6] first defines push-out conflicts and makes an overall analysis of the influence on both ground movement operations (on gate apron taxiways) and gate assignment operations simultaneously. A network-based simulation model is proposed to support the analysis. With the inspiration of his achievement, some following works also consider avoiding the potential hazard in push-out (or push-back) conflicts by using either the ground holding strategies or simulating the taxiway information on the basis of real-time process. For instance, Kim et al. [8, 23] apply the gate-holding departure control strategies and use a queuing model to simulate the aircraft departure process. They then try to predict and reduce the operation time in order to minimize the ramp congestion. Moreover, Atkin and Burke [22] and Newman and Atkin [24] demonstrate using various test data that the major problem of this push-back conflict will affect further steps of taxi operation in ATM. They also evaluate the final allocation results by introducing a novel towing constraint, which can solve the problem of ground movement conflicts.

Some gate holding strategies pay more attention to simulating the taxi process of arriving aircrafts in different speeds and positions, then use the simulation results to help determine whether the taxiing aircrafts will block the relevant towing area for other departing aircrafts. However, in practice this kind of ground holding strategy will reduce gate utilization and may sometimes cause further delays. Moreover, parameters used in the simulation may affect the results. Therefore, considering real time operation uncertainties, we try to rule out possible hazards in advance and ensure that operations based on the gate assignment plan will not lead to conflicts. This means that conflicts in aircraft ground movements near gate area will be avoided by considering the operational safety constraints in the gate allocation stage.

It is worth noting that little work has been done in existing studies on improving both safety and efficiency in AGAP. Thus, the main goal of this paper is to apply robust assignment to improve the gate utilization efficiency, while considering the safety issues concurrently.

3 Problem description

According to the description of push-back conflict in previous studies [6, 21], taxi-in or taxi-out refers to the movement of each aircraft into or out of the allocated parking position by its own engine power, whereas the push-back means that when an aircraft is ready for departing, it will be pushed out of the gate area or apron stand by a towing tractor. For small aircraft (around 60 passengers), taxi-out operation may be applied. For most of the medium and large aircrafts, push-back arrangement is commonly used in terms of nose-in parking. However, the nose-in parking requires manual operation and guidance from ground handlers, which may cause conflicts on the taxi lane near the gate ramp area.

Hence, we try to consider conflicts of three types: conflict between push-back and taxi-in, conflict between taxi-ins, and conflict between taxi-outs.

Push-back & taxi-in conflict: In Fig. 1, there is an overlapping ground movement time between two aircrafts, one departing and one arriving, that are assigned to adjacent gates 3 and 4. In this case, there is a high potential of conflict between the two aircrafts. Obviously this condition cannot be allowed in our optimization model.

Conflict between taxi-ins: When two arriving aircrafts with overlapping ground movement time are assigned to adjacent gates 2 and 3 in Fig. 2, there may be a high potential of conflict under this circumstance.

Conflict between push-outs: Similarly, in Fig. 3, when both departing aircrafts with overlapping ground movement time are assigned to adjacent gates 2 and 3, there is also a high risk of potential conflict between the two flights.

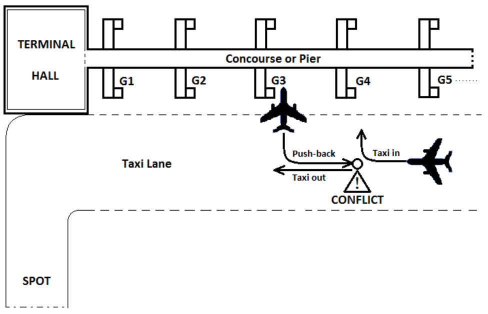


Fig. 1 Conflict between push-back and taxi in.

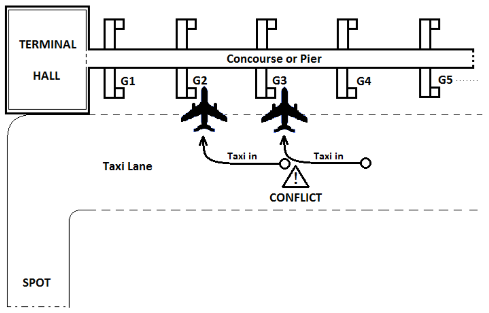


Fig. 2 Conflict between taxi-ins.

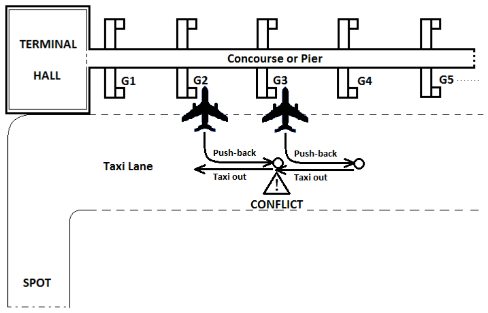


Fig. 3 Conflict between push-out.

An incident is considered an accident when there is a loss of life or severe damage [25]. Normally, there were no casualties of passengers reported in the conflicts between the pushing back and taxiing of aircrafts, because the speed restrictions will limit the operation of aircraft taxiing or towing on the airport ground. However, this kind of low speed collision will bring tremendous cost of aircraft damage. A major reason for such collision is improper gate assignment. If aircrafts with time overlapping in their taxi-in, push-back or taxi-out operations are not assigned to the adjacent gates, then the possibility of potential hazards in these operations will be reduced markedly.

From most of the former studies [3, 4], the classic constraints can be recognized as either hard or soft (particular airport layout, airline specified gate area, priority of gate occupation for emergency flight, etc.). Normally, there are two hard constraints:

Flight-to-gate unicity: Every flight must be assigned to one and only one feasible gate.

Gate conflict avoidance: No two flights with overlapping gate occupation times are assigned to the same gate.

In our model, we also consider the safety constraints (aircraft conflict avoidance) as a hard one:

Aircraft conflict avoidance: No two flights with overlapping taxi-in or push-back times are assigned to adjacent gates.

The gate assignment problem we study is then to assign a set of flights to gates to minimize the dispersion of idle times of the gates while satisfying the above hard constraints.

4 Problem formulation

The gate assignment problem can be viewed as a resource-constrained assignment problem where gates are considered as limited resources and aircrafts play the role of resource consumers. We formulate the problem as an integer programming model in this section.

4.1 Notation

The notation used for formulating the model is shown as follows.

Problem parameters:
$N = \{1, 2, ..., |N|\}$: the set of flights arriving at and/or departing from the airport in the planning day;

$M = \{1, 2, ..., |M|\}$: the set of gates available at the airport;

A_i: arrival time of flight i;

D_i: departure time of flight i; $D_0 = 0$;

α: minimum conflict avoidance time;

β: buffer time between two consecutive flights assigned to the same gate;

u_i: parameter indicating the type of aircraft of flight i; $u_i = 1$ if flight i is a large aircraft, else $u_i = 0$.

v_k: parameter indicating the type of gate k; $v_k = 1$ if gate k is a large one, else $v_k = 0$.

T: the fixed closing time of all gates after daily use.

θ: penalty for assigning a flight to a remote apron stand.

Decision variables:
S_{ik}: the idle time of gate k before flight i;

x_{ik}: binary decision variable; $x_{ik} = 1$ if flight i is assigned to gate k, else $x_{ik} = 0$.

$x_{i,|M|+1} = 1$, if flight i is assigned to a remote apron stand, else $x_{i,|M|+1} = 0$;

z_{ijk}: binary variable; $z_{ijk} = 1$ if both flight i and flight j are assigned to gate k, and flight i is followed by flight j, else $z_{ijk} = 0$.
$z_{0jk} = 1$, if flight j is the first flight assigned to gate k, else $z_{0jk} = 0$;

τ_k: the departure time of the last flight of gate k.

4.2 Objective function and model formulation

The main objective is to minimize the dispersion of idle time periods while avoiding mismatch between flight size and gate type as well as satisfying safety requirements. The original objective function can be denoted as

$$min \quad F = \sum_{k=1}^{|M|}\sum_{i=1}^{|N|}(S_{ik} - \bar{S})^2$$

Since the total available time of gates and the ground time of flights are known as a constant, whereas the specific slack time for each gate is independent of the way that flights are assigned, the total idle time for all available gates at the airport in one day is fixed as well. In this case, the function can be substituted by the form of $\sum_{k=1}^{|M|}\sum_{i=1}^{|N|} S_{ik}^2$.

In general, there should be an immediately preceding idle time before each aircraft arrives. While for each gate, the final idle time of a day should be considered as well, which usually refers to the duration between the time when the last aircraft leaves or is towed away for maintenance and the time the gate closes. Consequently, the total number of idle times can be concluded as $|N| + |M|$. In addition, at busy hours there may be more flights than the gates can handle. In this case some flights will be assigned to remote apron stands. We add a penalty for every such flight in the objective function to minimize the number of flights assigned to remote apron stands. Therefore, the objective function can be expressed as the variance of idle times plus these penalties:

$$min \quad F = \sum_{k=1}^{|M|}\sum_{i=1}^{|N|} S_{ik}^2 + \sum_{k=1}^{|M|}(T - \tau_k)^2 + \sum_{i=1}^{|N|}\theta x_{i,|M|+1}$$

The related constraints are formulated as follows.

$$\sum_{k\in M\bigcup\{|M|+1\}} x_{ik} = 1, \quad i \in N \tag{1}$$

$$x_{ik} + x_{jk} \leqslant 1, if(D_j - A_i)(D_i - A_j) > 0, \ i, j \in N, \ k \in M \tag{2}$$

$$\sum_{k\in M}\sum_{i\in N\bigcup\{0\}} z_{ijk} = 1, \quad j \in N \tag{3}$$

$$x_{ik} + x_{jk} - 2z_{ijk} \geqslant 0, \quad i, j \in N, \ k \in M \tag{4}$$

$$x_{jk} - z_{0jk} \geqslant 0, \quad j \in N, \ k \in M \tag{5}$$

$$S_{jk} \geqslant \beta(1 - z_{0jk}), \quad j \in N, \ k \in M \tag{6}$$

$$(u_i - v_k)x_{ik} \leqslant 0, \quad i \in N, \ k \in M \tag{7}$$

$$S_{jk} \leqslant A_j - D_i + (1 - z_{ijk})T, \ i \in N \cup 0, \ j \in N, \ k \in M \tag{8}$$

$$S_{jk} \geqslant A_j - D_i + (z_{ijk} - 1)T, \ i \in N \cup 0, \ j \in N, \ k \in M \tag{9}$$

$$\tau_k = \max_{i\in N}\{D_i x_{ik}\}, \quad k \in M \tag{10}$$

$$\begin{cases} |D_i - D_j| \geqslant \alpha x_{ik}x_{j,k+1}, \\ |D_i - A_j| \geqslant \alpha x_{ik}x_{j,k+1}, \\ |D_j - A_i| \geqslant \alpha x_{ik}x_{j,k+1}, \\ |A_i - A_j| \geqslant \alpha x_{ik}x_{j,k+1}, \end{cases} \quad i, j \in N, \ k, k+1 \in M \tag{11}$$

$$x_{ik}, z_{0jk}, z_{ijk} \in \{0, 1\}, \quad i, j \in N, \ k \in M \tag{12}$$

$$S_{ik}, \tau_k \geqslant 0, \quad i \in N, \ k \in M \tag{13}$$

Constraints (1) indicate that every flight must be assigned to only one gate or a remote apron stand. Constraints (2) ensure that one gate can serve at most one aircraft at a time. Constraints (3) and (4) give an exact description of variable z_{ijk}. Constraints (5) define z_{0jk} variables for the special case where a flight is the first one allocated to a gate. Constraints (6) stipulate that the idle time between the departure of a flight and the arrival of the next flight assigned to the same gate must be at least the required buffer time. Constraints (7) avoid mismatch between flights and gates. Constraints (8) and (9) calculate the idle time of each gate before each flight. Constraints (10) obtain the departure time of the last flight of each gate, which is used to calculate the last idle time of each gate in the objective function. Constraints (11) guarantee the minimum time between the arrival and departure of two flights assigned to adjacent gates to avoid conflict. Constraints (12) and (13) are binary and non-negativity constraints for the variables.

5 Solution method

As seen from literature review, both exact and heuristic methods have been proposed to find optimal or near-optimal solutions for the AGAP. Due to the problem complexity, exact algorithms such as branch and bound algorithm [11] can only solve small scale problems. In most of the major city airports, there are usually over 50 gates [3], and the exact methods are unable to solve the realistic problems of this large size. Therefore, most of the previous research solves the AGAP using meta-heuristic methods (e.g., genetic algorithm, tabu search, simulated annealing, swarm intelligence and their hybrid approaches). In this study we use genetic algorithm to solve the problem and to check the effect of adding safety constraints to the original gate assignment problem. In general, GA performs well for global searching and we expect it to be effective for this problem.

5.1 Encoding and initialization

Using an integer string to present the chromosome is a direct way to express the flight-to-gate relations. The length of the string is $|N|$ and each bit corresponds to a flight, while the specific number in that gene bit represents the gate number this flight assigned to. For example, string 5164532 represents a solution of assigning seven flights to six gates successively, where flight 1 and flight 5 are both assigned to gate 5.

As one of the ideas in evolutionary algorithms, genetic algorithm also concerns about how to make individuals consistently updating and reproducing strong fitness in the population. Literally, genetic algorithm maintains a population of chromosomes or individuals for each generation. Each chromosome represents a solution to the problem at hand. The GA process needs an initial population to start the evolution. We generate the solutions in the initial population randomly to ensure diversity. Randomly generated solutions may be infeasible because the AGAP is highly constrained. For each gene in the chromosome corresponding to a flight, we randomly select a gate among those which can accommodate the flight. This avoids the obvious infeasibility of flight-to-gate mismatching in the generated solution. Other types of infeasiblity will be resolved in the decoding procedure described in the next subsection.

5.2 Decoding and fitness calculation

For any chromosome in the GA process, we need to construct a corresponding gate assignment plan and calculate its objective value. We have developed a procedure to check the feasibility of each chromosome and if it is infeasible, a revised feasible solution is generated. This procedure is applied to all chromosomes throughout the whole GA evolution process, i.e., the chromosomes generated in the initial population as well as those generated in further genetic operations. An outline of the procedure is as follows.

1) Assign the flights to the gates according to the codes in the chromosome being checked. Check the feasibility of the assignment.

2) If the assignment is not feasible, revise it to make it feasible. This is done by checking the flights one-by-one in ascending order of their arrival times. If it is not feasible for the flight to use the gate assigned, move it to another gate which does not cause infeasibility to earlier flights. In case there are more than one gate feasible, choose the one with maximum idle time after the previous flight assigned to the same gate. Repeat this until all flights are checked. If the flight cannot be assigned to any gate feasibly in this way, the fight is then assigned to a remote apron stand.

3) Calculate and return the objective value of the feasible chromosome.

5.3 Genetic operations

Fig. 4 illustrates the major steps of the solution process we have used in this paper. The population evolves and the fitness improves in general from generation to generation. After certain number of generations, the results are more likely to approach an optimal or near optimal solution.

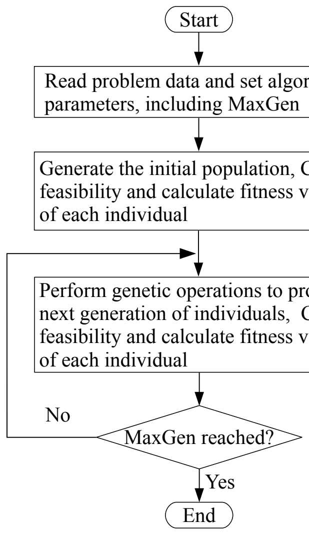

Start
Read problem data and set algorithm parameters, including MaxGen
Generate the initial population, Check feasibility and calculate fitness value of each individual
Perform genetic operations to produce next generation of individuals, Check feasibility and calculate fitness value of each individual
No
MaxGen reached?
Yes
End

Fig. 4 Flow chart of the solution method.

Selection: Selection provides the driving force in a genetic algorithm. The Roulette Wheel method is adopted as the fitness proportionate selection operator here, such that chromosomes with better fitness (smaller objective value for our minimization problem) will have higher chance to be selected. Using this method two chromosomes are selected each time to produce offspring through crossover and mutation for the next generation. Repeating the process, we will obtain a new population of chromosomes. To prevent best chromosomes from being destroyed at crossover and mutation, some best chromosomes in the current population are chosen to substitute the worst chromosomes in the offspring population. The number of best chromosomes chosen for this is relatively small to avoid them dominating the selection process.

Crossover: Crossover and mutation are common GA operators. Crossover operates on two chromosomes at a time and generates offspring by combining both chromosomes' features. One-point crossover method is adopted in the paper. A random point i is first generated where $i < N$. Then the parts to the right of bit i of the two parents are exchanged to generate two offspring chromosomes.

Mutation: A random exchange method is adopted to implement the operator. Each chromosome has a small probability (mutation rate) for mutation. If a random generated number is smaller than the predetermined mutation rate, then the current chromosome will participate in the mutation operation, otherwise it will not. Moreover, the specific mutating genes are randomly chosen from the chromosome and their values will be replaced by a different gate number randomly chosen from those feasible for that flight.

6 Testing results

In this section, the genetic algorithm is first used to solve an illustrative example to show the effectiveness of the algorithm and the effect of considering the safety constraints in the model. The algorithm performance is further tested using data of one terminal from PEK airport.

6.1 An illustrative example and test on problems of different sizes

Table I shows an example set of test data containing 40 flights to be assigned to 10 gates (7 large gates and 3 medium gates) and an extra un-gated apron stand in one day operation between 8:00am and 8:00pm. However, in the initial stage, the test focuses on the effect of the safety constraints and the over-constrained situation is not considered [26], which means that the gate resources are enough for all the flights considered in the assignment without using the remote apron stand. In practice remote apron stand often needs to be used, and in literature [13, 14], the un-gated area is regarded as one point with unlimited capacity whose use should be minimized rather than being considered with its own configuration. Different resource constraint levels will be considered in later tests.

The buffer times α and β are chosen as 5 and 15 minutes, respectively. The parameters of GA are set as follows:

Population size: 20

Crossover probability: 0.9

Mutation probability: 0.05

Maximum generation: 200

Table I Flight information for the illustrative example

FlightNo.	ArrivalTime(min)	DepartureTime(min)	FlightType
1	0	55	M
2	8	72	L
3	24	96	L
4	35	110	M
5	48	108	M
6	66	135	M
7	87	152	M
8	104	164	S
9	115	182	M
10	137	191	M
11	144	210	M
12	156	227	M
13	160	220	L
14	168	225	M
15	168	253	L
16	183	302	M
17	192	278	M
18	224	289	M
19	230	295	M
20	252	309	M
21	268	348	M
22	276	385	L
23	293	359	M
24	320	387	M
25	332	395	L
26	347	402	M
27	360	429	S
28	369	435	M
29	384	447	M
30	411	480	M
31	425	489	M
32	436	500	M
33	461	543	M
34	489	540	M
35	495	599	M
36	535	620	M
37	528	599	M
38	550	645	M
39	560	677	L
40	620	700	L

The results of flight-to-gate assignment with and without considering the safety constraints are presented in Fig. 5 and Fig. 6, respectively. According to the test data, flights 35 and 37 have exactly the same departure time. In Fig. 5, these two flights are assigned to the adjacent gates 2 and 3 without considering safety constraint: the time difference of their push-outs is obviously less than the minimum conflict avoidance time α. The results also show another potential hazard of push-out and taxi-in conflict between flights 31 and 34 with complete overlapping ground movement time at time point 489. Similarly, this kind of conflict may also happen on the assignments of flights 6 and 10, flights 18 and 23, flights 10 and 17, as well as flights 23 and 27, due to the overlapping of their arrival or departure times shown in the table above.

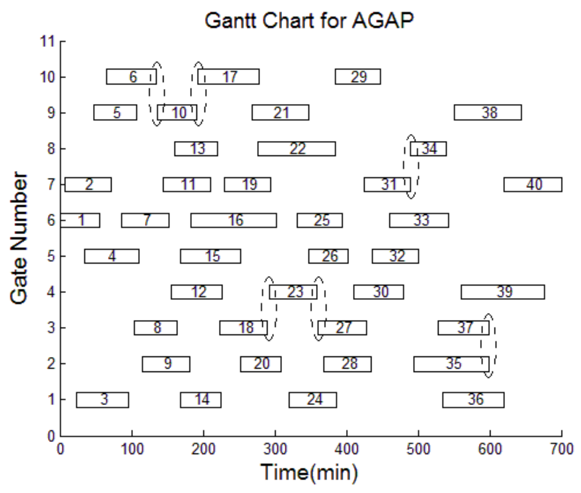


Fig. 5 Gantt chart of gate assignment without safety constraints.

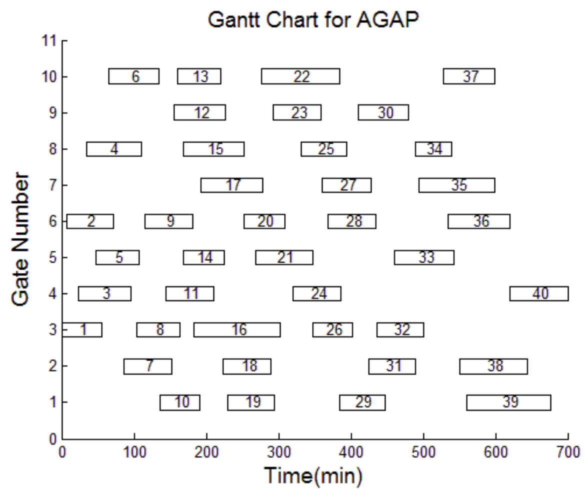


Fig. 6 Gantt chart of gate assignment with safety constraints.

The result in Fig. 6 has avoided flight conflict by the strict constraints in the model: no two flights with overlapping ground movement times are assigned to adjacent gates. We can notice that because of this constraint, the gate assignment of many other flights is also different.

After the initial stage of demonstrating the effect of considering safety constraints in our model, we use the algorithm to solve more test problems with different sizes (numbers of gates and flights) and different resource constraint levels (gates to flight ratios) as shown in Table II. For each scenario we solve the problem 10 times and the experimental results are also shown in Table II. The test is implemented in MATLAB R2013a and run on a computer with an Intel (R) Core i7 2.4 GHz CPU and 16 GB memory. Most of the parameters involved are the same as for the illustrative example. The objective value of each group represents the minimum objective value (dispersion of idle times and penalty for using the remote apron stand), while the computation time is the average of 10 runs.

The results in Table II show that the GA can solve the test problems efficiently in resource sufficient conditions. For the problems with the same number of gates, as the number of flights increases, the gate resources become more constrained and the objective value in the result increases as well. This reflects the fact that when the gate resources are constrained, some of the flights have to be assigned to the remote apron area due to the gate insufficiency, resulting high penalties. As expected, with the problem size increasing, the average computation time increases as well.

Table II Objective values and computation times in different scenarios

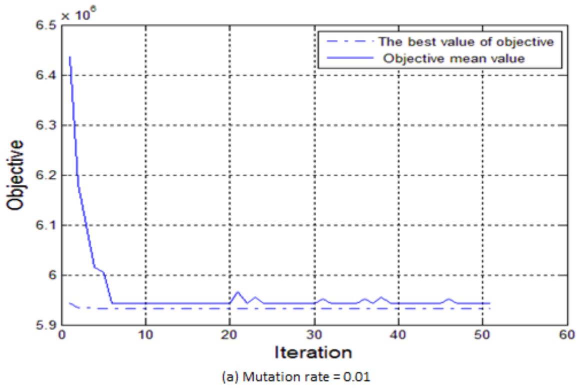
Gate/Flight	Population	ObjectiveValue(min)	ComputationTime(s)
10/30	50	180306	200.8
10/40	50	254127	390.2
10/50	50	493382	462.5
15/45	50	91480	343.3
15/60	50	164508	937.3
15/75	50	272241	1017.6
20/60	100	115013	1040.7
20/80	100	151447	1856.2
20/100	100	175621	2016.9

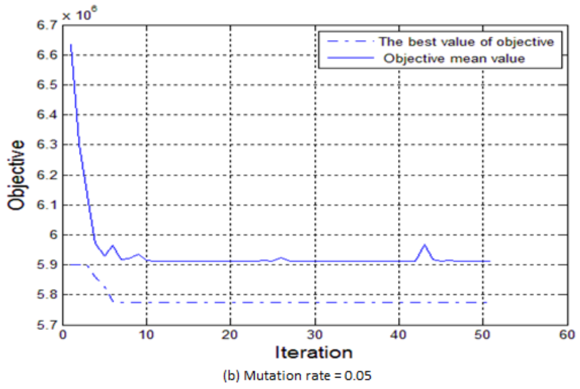
6.2 Further testing of algorithm performance

The experiment in this subsection is to test the algorithm performance using real data. We collected flight data from one terminal area of Beijing Capital International Airport (PEK) in China. There are more than 110 flights arriving and/or departing from the specific terminal area in a day. The flights can be classified into either the overnight flights or normal turn around ones during the day time. The major types of aircraft operating at the terminal include A319, A333, A332, A320, B738, B763, B788, which can be classified as types C, D and E (small, medium and large) according to the aircraft wingspan and the distance between the outer edge of the main gear wheels [27]. There are 13 gates connected to the terminal building, only one suitable for large aircraft as this is an old terminal. We use the flight information as well as the real terminal configuration in our experiments. According to the airport authority, the buffer time is set to 30 minutes. Each part of the testing in this experiment is also run for 10 times.

We first use small numbers of populations and max-generation (20 and 50 respectively) to obtain the results quickly and to observe the influence of the mutation rate on the algorithm performance. Fig. 7 shows the test results using different mutation rates. The solid line in each graph shows the average objective value in each generation, while the dashed line presents the best objective value up to each generation. Clearly, the solid curves in all the three graphs of Fig. 7 have presented a declined trend at the initial iterations. Then, the result in graph (a) shows a fast convergence. However, the best value achieved in (a) is still higher than those in (b) and (c). When the mutation rate is equal to 0.1, it can be seen that the best objective value still improves after many generations in part (c). However, the evolution process is not very stable.

In general, the mutation operation in genetic algorithm can help maintain the diversity of solutions and help avoiding premature convergence. Too small mutation rate may not achieve the purpose. However, if the mutation rate is set too large, it will be difficult for the features of good solutions to be passed to the next generations and solutions generated will be quite random. Based on the result, we set the mutation rate to 0.05 in further experiment.





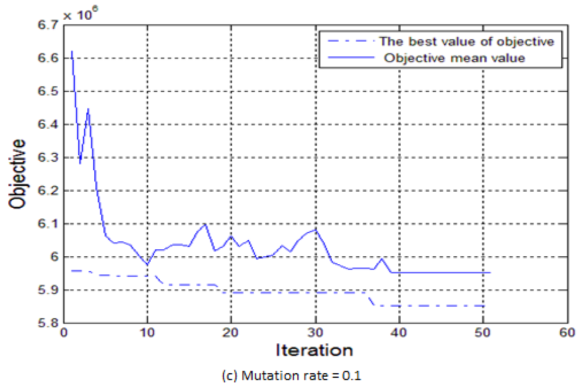
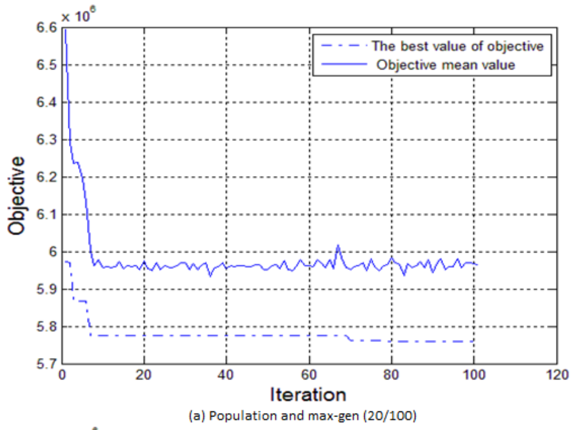
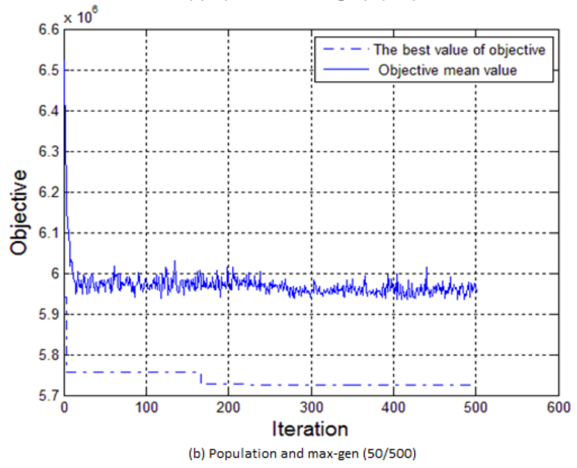


Fig. 7 The process of GA with different mutation rates.

We then enlarge the population size and max-generation providing sufficient time for the algorithm to converge. The evolution processes with different values of these parameters are shown in Fig. 8 (a)-(c). With the parameters set large, the result in Fig. 8(c) has the minimum objective value among the three, and the diagram also indicates that the result may be further improved if iteration continues. However, with this parameter setting the algorithm also requires significantly longer time to run. In application, when choosing an algorithm and its parameter setting, both computational efficiency and solution quality have to be considered.





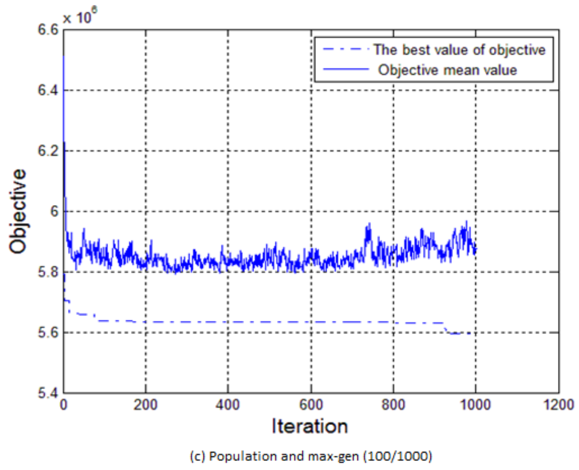


Fig. 8 The GA performance of different scenarios.

In the experiment, we assume that once an aircraft is assigned to a gate it stays there until it next departs or the end of the day. If the time between the arrival of an aircraft and its departure is very long, then towing it away from the gate to the remote apron area after disembarking and then towing it back to a gate before embarking could make the local gate area more efficiently utilized. However, the towing procedure may cause potential conflict problem. Therefore, in practice the airport often would rather let such aircraft stay at the pre-assigned gate than towing it away and then towing it back before it departs.

6.3 Allocation of parking positions at nearby apron area

Test results in section 6.1 show that when the number of flight is around 100, at least 20 gates will be needed to serve them. For the set of data collected and used in Section 6.2, there are more than 110 flights. The terminal is an old one with only 13 gates connected with the terminal building and only one of them can handle large aircrafts. Therefore, though the result in section 6.2 achieved high utilisation of the gates, a large number of flights still have to be allocated to apron stands. Fortunately, there are 10 parking positions near this terminal which are dedicated for flights at this terminal to use. Each of them also has a limit on the types of aircrafts that can park. These parking positions are next to each other and so the safety rules are also applicable to them. Therefore the allocation of aircrafts to each of these positions needs to be explicitly specified considering the safety rules.

After the allocation of aircrafts to the 13 gates, as a step further, we solve the problem of allocating the remaining aircrafts to these 10 nearby parking positions using the same method. Those that still cannot be handled will be allocated to remote apron area. The final allocation result is shown in Fig. 9.

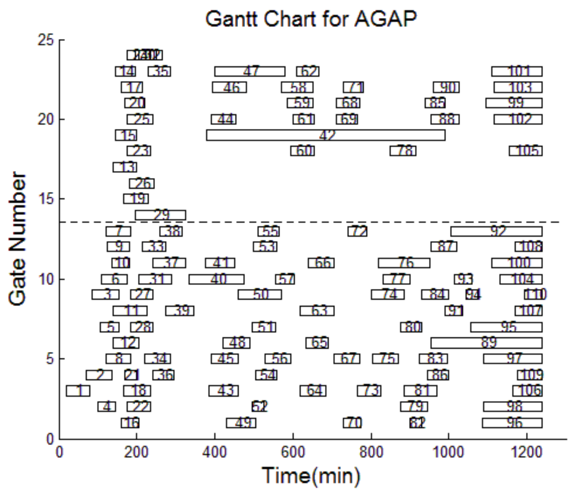


Fig. 9 Gantt chart showing the assignment of gates and nearby parking positions.

In Fig. 9, the 10 nearby parking positions are marked as gates 14-23. Gate 24 represents the remote apron area which has no capacity restrictions. As can be seen from the figure, the remote apron stand is used to deal with the peak time in the morning (0-400 min) for the overnight flights departing. It is worth noting that the aircraft for flight No. 42 with long stay time is actually of small type. Allocating it to a un-gated parking position frees normal gate resources for more flights. Meanwhile, the blank area showed in Fig. 9 in parking positions 14 to 17, actually illustrates the restriction of flight-to-gate mismatching. Although there are still three flights assigned to remote apron stands (marked as gate No. 24) which is far away from the terminal area based on the current airport configuration, this is due to the terminal's condition of insufficient resources. The assignment presented here already represents an improvement.

7 Conclusions

In this paper, we have studied the airport gate assignment problem and modelled it considering safety constraints. To achieve robust assignment and better utilize the gates, the objective function was set to minimize the dispersion of idle time periods and the number of flights assigned to remote apron stands. Genetic algorithm has been used to solve this gate assignment problem. A unified feasibility checking function is applied to decode each chromosome and generate a corresponding feasible solution in the whole solution process. An illustrative example is used to show the running of the algorithm and the effect of the safety constraints. Problem parameters were varied to observe the algorithm performance on the problem with different sizes and different resource constraint levels. The algorithm performance is further demonstrated using data of a terminal from PEK airport. The same method is also used to obtain detailed allocation of the nearby parking positions to the flights that could not be allocated to the gates.

References

[1] EUROCONTROL. Performance Review Report. Performance Review Commission. An assessment of Air Traffic Management in Europe during the calendar year 2013, May. 2014.

[2] M. Şeker, N. Noyan. Stochastic optimization models for the airport gate assignment problem. *Transp. Res. Part E Logist. Transp. Rev.*, vol. 48, no. 2, pp. 438–459, Mar. 2012.

[3] C. H. Cheng, S. C. Ho, C. L. Kwan. The use of meta-heuristics for airport gate assignment. *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12430–12437, Nov. 2012.

[4] U. Dorndorf, A. Drexl, Y. Nikulin, E. Pesch. Flight gate scheduling: State-of-the-art and recent developments. *Omega*, vol. 35, no. 3, pp. 326–334, Jun. 2007.

[5] U. Dorndorf, F. Jaehn, E. Pesch. Modelling Robust Flight-Gate Scheduling as a Clique Partitioning Problem. *Transp. Sci.*, vol. 42, no. 3, pp. 292–301, Aug. 2008.

[6] Y. Cheng. Solving push-out conflicts in apron taxiways of airports by a network-based simulation. *Comput. Ind. Eng.*, vol. 34, no. 2, pp. 351–369, Apr. 1998.

[7] S. Yan, C. M. Huo. Optimization of multiple objective gate assignments. *Transp. Res. Part A Policy Pract.*, vol. 35, no. 5, pp. 413–432, Jun. 2001.

[8] S. H. Kim, E. Feron, J. Clarke. Assigning Gates by Resolving Physical Conflicts. *AIAA Guidance, Navigation and Control Conference*, Chicago, Illinois, 10–13 Aug. 2009.

[9] T. Obata. The quadratic assignment problem: Evaluation of exact and heuristic algorithms. *Tech. Report TRS-7901*, Rensselaer Polytechnic Institute, Troy, New York, 1979.

[10] J. P. Braaksma. Reducing walking distance at existing airports. *Airport Forum*, vol. 7, pp. 135–142, 1977.

[11] O. Babic, D. Teodorvic, V. Tosic. Aircraft stand assignment to minimize walking. *Journal of Transportation Engineering*, vol. 110, no. 1, pp. 55–66, 1984.

[12] R. Mangoubi, D. Mathaisel. Optimizing gate assignments at airport terminals. *Transp. Sci.*, vol. 19, no. 2, pp. 173–188, May. 1985.

[13] H. Ding, A. Lim, B. Rodrigues, Y. Zhu. The over-constrained airport gate assignment problem. *Comput. Oper. Res.*, vol. 32, no. 7, pp. 1867–1880, Jul. 2005.

[14] J. Xu, G. Bailey. The Airport Gate Assignment Problem: Mathematical Model and a Tabu Search Algorithm. In *Proceedings of the 34th Hawaii International Conference on System Sciences*, USA, pp. 1–10, 2001.

[15] C. M. Pintea, P. C. Pop, C. Chira, D. Dumitrescu. A Hybrid Ant-Based System for Gate Assignment. *Hybrid Artificial Intelligence Systems*, vol. 5271, pp. 273–280, 2008.

[16] H. M. Genç, O. K. Erol, İ. Eksin, M. F. Berber, B. O. Güleryüz. A stochastic neighborhood search approach for airport gate assignment problem. *Expert Syst. Appl.*, vol. 39, no. 1, pp. 316–327, Jan. 2012.

[17] A. Bolat. Assigning arriving flights at an airport to the available gates. *Journal of the Operational Research Society*, vol. 50, no. 1, pp. 23–34, 1999.

[18] A. Bolat. Procedures for providing robust gate assignments for arriving aircrafts. *European Journal of Operational Research*, vol. 120, no. 1, pp. 63–80, Jan. 2000.

[19] A. Bolat. Models and a genetic algorithm for static aircraft-gate assignment problem. *Journal of the Operational Research Society* vol. 52, no. 10, pp. 1107–1120, 2001.

[20] S. Yan, C. Y. Shieh, M. Chen. A simulation framework for evaluating airport gate assignments. *Transp. Res. Part A Policy Pract.*, vol. 36, no. 10, pp. 885–898, Dec. 2002.

[21] C. Y. Liu, N.J. Zhai. On the multi-objective optimization of airport gate assignment with push-out conflict avoidance. In *Proceedings of 29th Chinese Control Conference*, China, pp. 1802–1806, 2010.

[22] J. A. D. Atkin, E. K. Burke. A more realistic approach for airport ground movement optimisation with stand holding. In *Proceedings of the 5th multidisciplinary international scheduling conference (MISTA)*, Phoenix, Arizona, 2011.

[23] S. H. Kim, E. Feron. Impact of Gate Assignment on Gate-Holding Departure Control Strategies. In *Proceedings of IEEE/AIAA 31st Digital Avionics Systems Conference*, IEEE, 2012.

[24] U. M. Neuman, J. A. D. Atkin. Airport Gate Assignment Considering Ground Movement. *ICCL 2013, Computational Logistics*, vol. 8197, pp. 184–198, 2013.

[25] EUROCONTROL. Annual Safety Report. Safety Regulation Commission. ATM Safety Performance 2013, Jan. 2014.

[26] S. Liu, W. Chen, J. Liu. Optimizing Airport Gate Assignment with Operational Safety Constraints. In *Proceedings of the 20th International Conference on Automation & Computing*, Cranfield University, Bedfordshire, 12–13 Sep. 2014.

[27] AIRBUS S.A.S. A. Characteristics and M. Planning. *Technical Data Support and Services*, Blagnac Cedex, France, 2005.

Shuo Liu graduated from Kunming University of Science and Technology in China, 2008. She received her MSc. degree in Control Systems from The University of Sheffield, UK in 2010. She is currently a Ph.D. candidate at Department of Aeronautical and Automotive Engineering in Loughborough University.

Her research interests include operational research, optimisation and decision making support in air traffic management.

E-mail: S.Liu@lboro.ac.uk (Corresponding author)

Wen-Hua Chen received the M.Sc. and Ph.D. degrees from Northeast University, Shenyang China, in 1989 and 1991, respectively. From 1991 to 1996, he was a Lecturer and then Associate Professor with the Department of Automatic Control, Nanjing University of Aeronautics and Astronautics, Nanjing, China. From 1997 to 2000, he held a research position and then a Lecturer in control engineering with the Centre for Systems and Control, University of Glasgow, Glasgow, U.K. In 2002, he moved to the Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, U.K., as a Lecturer, where he was appointed as a Professor in 2012. He is a Senior Member of IEEE and a Fellow of IET.

His research interests include the development of advanced control strategies (Nonlinear Model Predictive Control, Disturbance Observer Based Control, etc) and their applications in aerospace engineering. Currently, much of his work has also involved in the development of Unmanned Autonomous Intelligent Systems.

E-mail: W.Chen@lboro.ac.uk

Jiyin Liu is Professor of Operations Management in the School of Business and Economics at Loughborough University, UK. He previously taught at Northeastern University and The Hong Kong University of Science and Technology in China. He received his Ph.D. in Manufacturing Engineering and Operations Management from The University of Nottingham in the UK and his B.Eng. in Industrial Automation and M.Eng. in Systems Engineering from Northeastern University in China. His research interests are in the areas of operations planning and scheduling in production and logistics, as well as in modeling, analysis, and solution of practical operations problems.

His research has been published in various academic journals such as European Journal of Operational Research, IEEE Transactions, IIE Transactions, International Journal of Production Research, Journal of the Operational Research Society, Naval Research Logistics, Operations Research, and Transportation Research.

E-mail: J.Y.Liu@lboro.ac.uk