Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 22 (2017) 469–478

Transportation Research Procedia

www.elsevier.com/locate/procedia

19th EURO Working Group on Transportation Meeting, EWGT2016, 5-7 September 2016, Istanbul, Turkey

# The comparison of the metaheuristic algorithms performances on airport gate assignment problem

Abdullah Aktel[a,*], Betul Yagmahan[b], Tuncay Özcan[c], M. Mutlu Yenisey[c], Engin Sansarcı[d]

[a] *Turkish Institute of Management Sciences Kocaeli, Turkey*
[b] *Industrial Engineering Department, Uludağ University, School of Engineering, Bursa, Turkey*
[c] *Industrial Engineering Department, İstanbul University, School of Engineering, İstanbul, Turkey*
[d] *Management Engineering Department, İstanbul Technical University, Faculty of Management, İstanbul, Turkey*

**Abstract**

The airport gate assignment problem (AGAP) is an important research area in air transportation planning and optimization. In this paper we study the airport gate assignment problem where the objectives are to minimize the number of ungated flights and the total walking distances. In order to solve the problem, we proposed a new tabu search (TS) algorithm which uses a probabilistic approach as an aspiration criterion. We compared two metaheuristics, namely, TS, and simulated annealing (SA). A greedy algorithm used as a benchmark. We compared the performances of the algorithms and analyzed at different problem sizes. Experimentations showed that the new proposed metaheuristic algorithm gave promising results.


Peer-review under responsibility of the Scientific Committee of EWGT2016.

*Keywords:* Gate assignment; Metaheuristics; Tabu search; Simulated annealing

* Corresponding author. Tel.:+905428968210
*E-mail address:* abdullah.aktel@tubitak.gov.tr


Peer-review under responsibility of the Scientific Committee of EWGT2016.
10.1016/j.trpro.2017.03.061

## 1. Introduction

Airport management includes very complex issues. One of the important tasks in the airports is to assign flights to available gates. The more efficient gate assignment plan results in lower flight delays, better customer services, and higher utilization rates for the usage of ground facilities. Clearly, this leads lower operating costs. Additionally, passenger satisfaction increases.

The Airport Gate Assignment Problem (AGAP) can be defined as finding feasible flight-to-gate assignments which optimize a performance measure. Mostly, the AGAP minimizes total passenger walking distances. The walking distances considered in airports are: (1) the distance from check-in counters to gates, (2) the distance from gates to baggage claim areas, and (3) the distance from one gate to another one for transfer passengers. Sometimes the number of flights exceeds the number of available gates. In this case, airport planners also aims to minimize the number of ungated flights.

Almost all kind of modelling approaches, i.e. integer, binary or mixed integer, linear or nonlinear models, are applied to AGAP in the studies in literature. Different solution techniques are also proposed to solve the models. Mainly researchers want to find an optimal or at least a good-quality solution in a reasonable time. The solution methods to the AGAP can be classified as either exact or heuristic methods. Exact solution algorithms assure finding optimal solution while heuristic algorithms do not guarantee the optimal solution. However, recent studies mostly focus on heuristics to solve AGAP because of the complexity of the problem.

The airport gate assignment problem has been extensively studied in the literature since the early 1980s. Babic et al. (1984) and Bihr (1990) proposed a 0-1 integer programming model, and solve by Branch and Bound (B&B) algorithm. Mangoubi and Mathaisel (1985) developed a linear relaxation of an integer program formulation and solved the problem by a heuristic algorithm. Bolat (1999, 2000) developed a mathematical model formulation and solved the problem by B&B algorithm and a constructive heuristic algorithm. Yan and Huo (2001) formulated a multiple objective 0-1 integer model and used the column generation and B&B approaches.

Although the AGAP is a NP-hard problem (Ding et al., 2004, 2005; Lim et al., 2005), research efforts in recent years have focused on developing the heuristic and metaheuristic algorithms. The studies which use a problem specific heuristic approach are Bolat and As-Saifan (1996), Haghani and Chen (1998), Yan and Tang (2007), Dorndorf et al. (2008, 2012), and Genç et al. (2012). Finally, mixed integer programming based heuristic algorithm was presented by Yu et al. (2016). There are several studies in literature using metaheuristic approaches to solve the AGAP. Bolat (2001) proposed a genetic algorithm after discussing five different models for gate assignment. Xu and Bailey (2001) and Ding et al. (2004) formulated the gate assignment problem as a quadratic assignment problem and used tabu search metaheuristic. Ding et al. (2005) gave a 0-1 integer linear programming model. Authors firstly used a greedy algorithm to exchange the moves and then they applied simulated annealing and a hybrid approach of simulated annealing and tabu search. Lim et al. (2005) used a time shift algorithm then they applied tabu search and memetic algorithm. Hu and Paolo (2007) provided genetic algorithm for the multi-objective AGAP. Drexl and Nikulin (2008) addressed the problem similar way in Ding et al. (2005) and they used pareto simulated annealing approach. Cheng et al. (2012) compared three metaheuristics; genetic algorithm, tabu search, simulated annealing, and a hybrid approach made of simulated annealing and tabu search. Şeker and Noyan (2012) introduced the uncertainty into problem structure and developed a stochastic programming model with robustness measure. They employed tabu search algorithm to obtain assignments of reasonable quality. Marinelli et al. (2015a) introduced the bee colony optimization to solve the AGAP. A hybridized biogeography-based with bee colony optimization metaheuristic was proposed by Marinelli et al. (2015b) for solving AGAP, too.

Due to nature of the problem, stochastic models are considered as a tool to solve in the literature for AGAP. Cheng (1998) introduced a rule-based simulation method for the activities of aircraft on gates in apron control. Yan et al. (2002) analysed the effects of stochastic flight delays on static gate assignments, and evaluated flexible buffer times and real-time gate assignment rules. Narciso and Piera (2015) proposed a simulation-based experimental approach to evaluate the impact of AGAP policies.

The most recent survey on AGAP is by Dorndorf et al. (2007) and Bouras et al. (2014) where most of the existing literature is reviewed.

Many different objectives have been considered in the literature for AGAP. The most widely used objective is the minimization of the passenger walking distance (Babic et al., 1984; Bihr, 1990; Mangoubi and Mathaisel, 1985; Yan and Huo, 2001; Ding et al., 2004, 2005; Lim et al., 2005; Hu and Paolo, 2007; Yan and Tang, 2007; Drexl and Nikulin, 2008; Cheng et al., 2012; Marinelli et al., 2015a, 2015b). The objectives commonly used in the literature include the following:

- Gate idle time: Bolat and As-Saifan (1996), Şeker and Noyan (2012)
- Waiting time: Yan and Huo (2001)
- Connection time: Xu and Bailey (2001), Ding et al. (2005)
- Ungated flights: Ding et al. (2004, 2005), Drexl and Nikulin (2008), Dorndorf et al. (2008, 2012)
- Baggage transport distance, Aircraft waiting time on the apron: Hu and Paolo (2007)
- Assignment preference score, The number of tows, The robustness of the resulting schedule: Dorndorf et al. (2008, 2012)
- The total duration of un-gated flights: Genç et al. (2012)
- The number of conflicts: Şeker and Noyan (2012)
- Buffer times: Şeker and Noyan (2012)
- Remote gate usage: Marinelli et al. (2015a, 2015b)

In this study we developed an efficient algorithm for the AGAP. We considered optimization of two objectives for the selected airport gate assignment problems. We tried to minimize the ungated flights and the total walking distances together. To solve the problem, we proposed a new TS which uses a probabilistic tabu demolish criterion. We compared the performances of TS and SA algorithms on different problem sizes. We also used a greedy algorithm as a benchmark. The proposed TS algorithm differentiate the current metaheuristics in the literature by solution generation scheme and tabu demolish criteria. The proposed TS algorithm chromosome structure and solution generation mechanism always produce feasible solutions. The proposed tabu demolish criteria evaluate a move according to the solution quality and age in the algorithm so that intensification and diversification can be obtained together.

The remaining of the paper is organized as follows. In the Section 2, we briefly define the model used in the paper. This model is taken from Ding et al. (2005) and is a 0-1 Integer Programming Model. Section 3 introduces the TS and SA algorithms which are used to solve AGAP. We explained our study in Section 4. Finally, we concluded our finding and discussed the future research in Section 5.

## 2. AGAP Model Definition

The aim of the AGAP is assigning each flight to an available gate while both minimizing total passenger walking distances and maximizing gate utility. Three types of passengers are considered: arriving passengers, departing passengers, and transfer passengers. Three types of walking distances are considered: the distance from check-in to gates, the distance from gates to baggage claim areas, and the distance from gate to gate for transfer passengers.

In this study, we considered the model proposed by Ding et al. (2005). That model assigns flights to the gates while minimizing the number of ungated flights, and the walking distances of passengers.

The notation in the model are defined as follows:

$N$ set of flights arriving at (and/or departing from) the airport

$M$ set of gates available at the airport

$n$ total number of flights

$m$ total number of gates

$a_i$ arrival time of flight $i$ ($1 \le i \le n$)

$d_i$ departure time of flight $i$ ($1 \le i \le n$)

$w_{k,l}$ walking distance for passengers from gate $k$ to gate $l$ ($1 \le k, 1 \le m$)

$f_{i,j}$ number of passengers transferring from flight $i$ to flight $j$ ($1 \le i, j \le n$)

Additionally, two dummy gates are defined; 0 for the entrance and/or exit of the airport and $m$+1 for the apron. The gate $m$+1 was used for the aircrafts which could not assign to any gate because of unavailability. $w_{k,0}$ represent the distance between gate $k$ and the airport entrance or exit, $f_{0,i}$ represents the number of originating departing passengers of flight $i$; and $f_{i,0}$ represents number of the disembarking passengers of flight $i$. Clearly, the decision variable $y_{i,k} = 1$ stands for the assignment of flight $i$ to gate $k$ ($0 < k \le m + 1$), otherwise $y_{i,k} = 0$ .

The authors presented a 0-1 integer programming model to solve the gate assignment problem. The objective of the model is to minimize the number of flights assigned to the apron and to minimize the total passenger walking distance. Based on the notation defined above, the AGAP model is formulated as follows:

$$Minimize \quad \sum_{i=1}^{n} y_{i,m+1} \tag{1}$$

$$Minimize \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m+1}\sum_{l=1}^{m+1} f_{i,j} w_{k,l} y_{i,k} y_{j,l} + \sum_{i=1}^{n} f_{0,i} w_{0,i} + \sum_{i=1}^{n} f_{i,0} w_{i,0} \tag{2}$$

Subject to

$$\sum_{k=1}^{m+1} y_{i,k} = 1 \qquad 1 \le i \le n \tag{3}$$

$$y_{i,k} y_{j,k} (d_j - a_i)(d_i - a_j) \le 0 \qquad 1 \le i, \quad j \le n, \quad k \ne m+1 \tag{4}$$

$$y_{i,k} \in \{0,1\} \qquad 1 \le i \le n, \qquad 1 \le k \le m+1 \tag{5}$$

Objective function (1 and 2) minimizes the number of flights assigned to the apron and total distances. Constraint (3) requires every aircraft to be assigned to one and only one gate. Constraint (4) requires that no two flights are assigned to the same gate at the same time. Constraint (5) denotes that all the decision variables to 0 or 1.

## 3. Metaheuristics

In this section, we give the details of the metaheuristics for solving AGAP.

### *3.1. Tabu Search for AGAP*

In the proposed TS algorithm, a solution candidate is generated in two steps. Firstly, a chromosome is defined as a permutation of flights.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Flight5 | Flight2 | Flight1 | Flight3 | Flight4 | Flight9 | Flight7 | Flight6 | Flight8 | Flight10 |

Fig. 1. A chromosome

Fig. 1. shows a ten flights gate assignment problem solution representation. After flight permutations are generated, the solution permutation is decoded to establish gate assignment of each flights. Flights are assigned to the gates according to the permutation order. Gate utilizations want to be maximized. While flights assigning to the gates according to the permutation order gate availabilities are also considered. Therefore a flight is assigned to a gate so that to minimize the difference between the arrival time of a flight and the earliest available time of the gates in the airport. The initial permutation is generated by sorting the flights according to the ascending order of departure times. Neighborhoods are generated by swapping the generated solution permutation flight orders. In Fig. 2. a neighborhood is created by swapping Flight2 and Flight7 that given in Fig. 1.

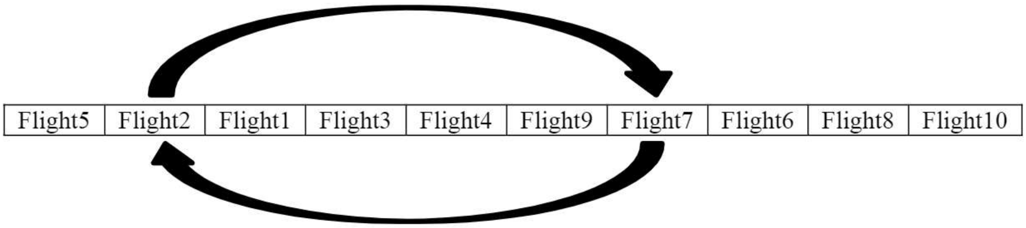


Fig. 2. A swap operation

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Flight5 | **Flight7** | Flight1 | Flight3 | Flight4 | Flight9 | **Flight2** | Flight6 | Flight8 | Flight10 |

Fig. 3. A neighbourhood

Total passenger walking distance is used as a fitness function, i.e. the smaller value of fitness represents a better solution. A probabilistic approach is used as an aspiration criterion. A tabu demolish probability is defined for each tabu move which uses the expiry time of that tabu, target objective value of the move, mean and standard deviation of candidate solutions in the neighborhood. Suppose $x$ represents the objective value of a candidate solution, $\bar{x}$ represents the average objective value of all possible candidates in the neighborhood and $s$ represents the corrected sample standard deviation. If we assume that the objective values in the neighborhood is distributed normally, then $\phi(Z)$ becomes the cumulative distribution function of the standard normal distribution, where

$$Z = (x - \bar{x})/s. \tag{6}$$

Demolish probability ($P$) of any move then becomes

$$P = (t - t_r)(1 - \phi(Z)/((t - t_r)(1 - \phi(Z) + t_r.\phi(Z)) \tag{7}$$

where $t$ and $t_r$ represent tenure time and expiry time of that move. With this formulation, moves that observed very recently are unlikely to be chosen again while the probability increases as the target objective value of that move decreases. The TS framework is shown in Fig. 4. Iteration number (max_iter) is chosen as 250, and 50 different neighbourhood solutions are generated in every iteration.

***Step 1:*** Generate an initial solution $S_{now}$ by sorting the flights according to the ascending order of departure times. Add $S_{now}$ to best solution pool ($S_{best}$) and iter=0.

***Step 2:*** If iter > max_iter, terminate with $S_{optimum}$ by taking minimum of $S_{best;}$ else continue.

***Step 3:*** Create neighbourhoods. Evaluate neighbourhoods by using fitness function ($f$) choose best neighbour.

***Step 4:*** Check best neighbour is a tabu move. If best neighbour is a tabu move. Calculate tabu demolish probability.

***Step 5:*** Generate a random number (RN). If (RN < tabu demolish probability) add best neighbour to $S_{best}$ pool.

Fig. 4. TS structure

### 3.2. Simulated Annealing for AGAP

In the SA, the solution structure and initial solution is created by using the same way as in the TS algorithm. A solution is defined as a permutation of flights. The solution permutations are decoded to generate gate assignments of each flight. While solution candidates are being evaluated, gate utilizations want to be maximized. Hence, a flight is assigned to a gate to minimize the difference between the arrival time of a flight and the earliest available time of the gates. Neighborhoods are generated by swapping the generated solution permutation flight orders. A probabilistic acceptance criterion is used as a move acceptance. In the cooling mechanism, the temperature is updated according to the geometric schedule. SA framework is shown in Fig. 5. Initial temperature $T$ = 500, and cooling function is determined as $T = 0.99*T$

***Step 1:*** Generate initial solution $S_{now}$ by sorting the flights according to the ascending order of departure times. Set $S_{now}$ to best solution ($s_0$).

***Step 2:*** Determine initial temperature ($T$).

***Step 3:*** Determine cooling function ($\alpha$)

***Step 4:*** If ($T$ > 0.1) generate a neighbourhood solution ($s$) determine $\delta = f(s) - f(s_0)$

  ***Step 4.1:*** If ($\delta < 0$), $s = s_0$

  ***Step 4.2:*** If ($\delta \geq 0$) generate a random number (RN); if (RN < $exp(-\delta/T)$), $s = s_0$

  ***Step 4.3:*** $T = \alpha(T)$

Fig. 5. SA structure

## 4. The Computational Experience

Firstly, we design three example test problems to check the performances of the metaheuristics. Table 1. shows the characteristics of the test problems.

Table 1. Example test problems

| Problem No | Number of flights | Number of gates |
|---|---|---|
| 1 | 15 | 6 |
| 2 | 25 | 9 |
| 3 | 184 | 23 |

We developed the generic 0-1 integer programming model then we solved the first problem LINGO14.0. After that we run the TS and SA algorithms for test purpose. The TS and SA algorithms find the optimum with the fitness value 196,490 of the first problem as a total walking distance. Since our algorithms find the optimum solution we decided to solved the other test problems. After we solved the 25x9 test problem, we tested our approach for a large-scale problem with 184 flights and 23 gates. The data in that problem was approximately generated based on the distribution of Istanbul Ataturk Airport (LTBA/IST) daily flight schedule. Istanbul Ataturk Airport is the busiest airport of Turkey. Currently, it is 11$^{th}$ in the rank of the 2015 world's busiest airports list according to passenger traffic. The generated data reflects the daily average air traffic in Istanbul Ataturk Airport. We tested the algorithm with three data sets including that one. The results were summarized in Table 2.

Table 2. The results obtained by metaheuristic algorithms

| **Algorithm** | **(15,6)** | **(25,9)** | **(184,23)** |
|---|---|---|---|
| TS | 196,490 | 97,200 | **1,779,750** |
| SA | 196,490 | 97,500 | 1,802,675 |

Finally, we compared our algorithms against by a greedy algorithm (Greedy-A) which was proposed in the work Ding et al. (2005). A structure of the Greedy-A is shown in Fig. 6.

***Step 1:*** Sort the flights of $S = \{i \mid i=1,\ldots,n\}$ according to the ascending order of departure time ($d_i$).

***Step 2:*** Let $g_k$ ($1 \leq k \leq m$) represents the earliest available time (the departure time of last fight) of gate $k$.

Set $g_k = -1$ for all $k$.

***Step 3:*** For successive $i \in S$, find gate $k$ such that $g_k < a_i$ and $g_k$ is maximized. Update $g_k = d_i$. If $k$ does not exist, assign flight $i$ to the apron.

***Step 4:*** Compute the total walking distance using Equation (2) for the final solution $\{y_{ik}\}$.

Fig. 6. Greedy-A structure

To compare our algorithm performances against greedy algorithm we design six more test problems (Table 3). The test problem data set was generated by using uniform distribution. The walking distances between gates fits U(1,100), total number of arriving, departing and transfer passengers fits U(1,50), arrival times of flights fits U(1,100) and departure time of flights fits arrival times + U(1,30). We also compute the TS algorithm which is only uses fitness values as a tabu demolish criteria. A tabu move is accepted only if that tabu fitness value is better than the solutions fitness values generated in that time.

The comparison results were summarized in Table 4. In Table 4, fitness1 (F1) shows total ungated flights, fitness2 (F2) shows total walking distance.

Table 3. The size of test problems

| Problem # | Number of flights | Number of gates |
|---|---|---|
| 1 | 100 | 16 |
| 2 | 160 | 20 |
| 3 | 220 | 24 |
| 4 | 280 | 28 |
| 5 | 340 | 32 |
| 6 | 520 | 44 |

Table 4. Computational results of algorithms

| Prob # | SA | | | TS (Probabilistic approach) | | | TS (Fitness based approach) | | | Greedy-A | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | F1 | F2 | Run time | F1 | F2 | Run time | F1 | F2 | Run time | F1 | Fs2 | Run time |
| 1 | 11 | 8600194 | 21,77 | 11 | 8604069 | 21,63 | 11 | 8605125 | 17,29 | 17 | 12068397 | 0,12 |
| 2 | 35 | 36232166 | 42,32 | 35 | 36143860 | 39,16 | 35 | 36132065 | 41,49 | 40 | 40580009 | 0,2 |
| 3 | 41 | 61397892 | 79,35 | 41 | 61307957 | 77,84 | 41 | 61216993 | 79,93 | 58 | 78916549 | 0,26 |
| 4 | 62 | 112469947 | 118,87 | 64 | 114396340 | 123,16 | 64 | 114447588 | 118,89 | 67 | 120548304 | 0,34 |
| 5 | 88 | 184071214 | 158,36 | 88 | 184321438 | 164,93 | 88 | 184268714 | 162,19 | 89 | 188769638 | 0,42 |
| 6 | 161 | 489854517 | 388,45 | 161 | 489649284 | 387,06 | 162 | 491421745 | 385,59 | 164 | 495389260 | 0,52 |

The performance of an algorithm on a test problem is evaluated by calculating the relative percentage increase. The relative percentage increase is given as follows:

$$\Delta(F_i) = \left( \frac{F_i - F_i^{\min}}{F_i^{\min}} \right) \tag{8}$$

Two different fitness functions is aggregated by taking equally weighted averages of the fitness1 and fitness2. The total relative percentage increase is given as follows:

$$\Delta(F) = w_1 \Delta(F_1) + w_2 \Delta(F_2) \tag{9}$$

The solutions obtained by all algorithms (SA, TS and Greedy-A) are presented in Table 5.

Table 5. Performance solutions of algorithms

| Problem # | SA | TS (Probabilistic approach) | TS (Fitness based approach) | Greedy-A |
|---|---|---|---|---|
| 1 | 0.0000 | 0.0002 | 0.0003 | 0.4744 |
| 2 | 0.0014 | 0.0002 | 0.0000 | 0.1330 |
| 3 | 0.0015 | 0.0007 | 0.0000 | 0.3519 |
| 4 | 0.0000 | 0.0247 | 0.0249 | 0.0762 |
| 5 | 0.0000 | 0.0007 | 0.0005 | 0.0184 |
| 6 | 0.0002 | 0.0000 | 0.0049 | 0.0152 |
| ***Avg.*** | ***0,0005*** | ***0,0044*** | ***0,0051*** | ***0,1782*** |

## 5. Conclusion and Future Research

The experimental results showed that the proposed metaheuristic algorithms are efficient for providing good results at a reasonable time for large-sized gate assignment problems. SA gives best results for problem 1, 4, 5, the TS algorithm which uses probabilistic tabu demolish criteria gives best results for problem 6, the TS algorithm which uses fitness based approach gives best result for problem 2 and 3. The solutions in Table 5 show that SA algorithm has yielded the best performance on average. As a result we can say that the proposed probabilistic TS algorithm works very well at bigger size problems. The proposed probabilistic TS algorithms chromosome solution generation scheme prevents infeasible solutions and the probabilistic tabu demolish criteria mechanism obtains both intensification and diversification so that to continue search with better solution without stuck in local optima As a future research, a hyper-heuristic can be developed to choose more effective metaheuristic automatically according to the implemented problem.

## References

Babic, O., Teodorovic, D., Tošic, V. 1984. Aircraft Stand Assignment to Minimize Walking. Journal of Transportation Engineering 110(1), 55–66.

Bihr, R. A. 1990. A Conceptual Solution to the Aircraft Gate Assignment Problem Using 0,1 Linear Programming. Computers & Industrial Engineering 19(1-4), 280–284.

Bolat, A., As-Saifan, K. 1996. Procedures for Aircraft Gate Assigment. Mathematical & Computational Applications 1(1), 9–14.

Bolat, A. 1999. Assigning Arriving Flights at an Airport to the Available Gates. Journal of the Operational Research Society 50, 23–34.

Bolat, A. 2000. Procedures for Providing Robust Gate Assignments for Arriving Aircrafts. European Journal of Operational Research 120(1), 63–80.

Bolat, A. 2001. Models and a Genetic Algorithm for Static Aircraft-Gate Assignment Problem. Journal of the Operational Research Society 52(10), 1107–1120.

Bouras, A., Ghaleb, M. A., Suryahatmaja, U. S., Salem, A. M. 2014. The Airport Gate Assignment Problem: A Survey. The Scientific World Journal 2014.

Cheng, Y. 1998. A Rule-based Reactive Model for the Simulation of Aircraft on Airport Gates. Knowledge-Based Systems 10(4), 225–236.

Cheng, C. H., Ho, S. C., Kwan, C. L. 2012. The Use of Meta-heuristics for Airport Gate Assignment. Expert systems with applications 39(16), 12430–12437.

Ding, H., Lim, A., Rodrigues, B., Zhu, Y. 2004. New Heuristics for Over-constrained Flight to Gate Assignments. Journal of the Operational Research Society 55, 760–768.

Ding, H., Lim, A., Rodrigues, B., Zhu, Y. 2005. The Over-constrained Airport Gate Assignment Problem. Computers & Operations Research 32(7), 1867–1880.

Dorndorf, U., Drexl, A., Nikulin, Y., Pesch, E. 2007. Flight Gate Scheduling: State-of-the-art and Recent Developments. Omega 35(3), 326–334.

Dorndorf, U., Jaehn, F., Pesch, E. 2008. Modelling Robust Flight-gate Scheduling as a Clique Partitioning Problem. Transportation Science 42(3), 292–301.

Dorndorf, U., Jaehn, F., Pesch, E. 2012. Flight Gate Scheduling with Respect to a Reference Schedule. Annals of Operations Research 194(1), 177–187.

Drexl, A., Nikulin, Y. 2008. Multicriteria Airport Gate Assignment and Pareto Simulated Annealing. IIE Transactions 40(4), 385–397.

Genç, H. M., Erol, O. K., Eksin, İ., Berber, M. F., Güleryüz, B. O. 2012. A Stochastic Neighborhood Search Approach for Airport Gate Assignment Problem. Expert Systems with Applications 39(1), 316–327.

Haghani, A., Chen, M. C. 1998. Optimizing Gate Assignments at Airport Terminals. Transportation Research Part A: Policy and Practice 32(6), 437–454.

Hu, X. B., Paolo, E. D. 2007. An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem, IEEE Congress on Evolutionary Computation, CEC 2007. (pp. 55–62). IEEE.

Lim, A., Rodrigues, B., Zhu, Y. 2005. Airport Gate Scheduling with Time Windows. Artificial Intelligence Review 24(1), 5–31.

Mangoubi, R. S., Mathaisel, D. F. 1985. Optimizing Gate Assignments at Airport Terminals. Transportation Science 19(2), 173–188.

Marinelli, M., Dell'Orco, M., Sassanelli, D. 2015a. A Metaheuristic Approach to Solve the Flight Gate Assignment Problem. Transportation Research Procedia 5, 211–220.

Marinelli, M., Palmisano, G., Dell'Orco, M., Ottomanelli, M. 2015b. Fusion of Two Metaheuristic Approaches to Solve the Flight Gate Assignment Problem. Transportation Research Procedia 10, 920–930.

Narciso, M. E., Piera, M. A. 2015. Robust Gate Assignment Procedures from an Airport Management Perspective. Omega 50, 82–95.

Şeker, M., Noyan, N. 2012. Stochastic Optimization Models for the Airport Gate Assignment Problem. Transportation Research Part E: Logistics and Transportation Review 48(2), 438–459.

Xu, J., Bailey, G. 2001. The airport gate assignment problem: mathematical model and a tabu search algorithm, Proceedings of the 34th Annual Hawaii International Conference on System Sciences. IEEE.

Yan, S., Huo, C. M. 2001. Optimization of Multiple Objective Gate Assignments. Transportation Research Part A: Policy and Practice 35(5), 413–432.

Yan, S., Shieh, C. Y., Chen, M. 2002. A Simulation Framework for Evaluating Airport Gate Assignments. Transportation Research Part A: Policy and Practice 36(10), 885–898.

Yan, S., Tang, C. H. 2007. A Heuristic Approach for Airport Gate Assignments for Stochastic Flight Delays. European Journal of Operational Research 180(2), 547–567.

Yu, C., Zhang, D., Lau, H. Y. K. 2016. MIP-based Heuristics for Solving Robust Gate Assignment Problems. Computers & Industrial Engineering 93, 171–191.