

Review Article

The Airport Gate Assignment Problem: A Survey

Abdelghani Bouras, Mageed A. Ghaleb, Umar S. Suryahatmaja, and Ahmed M. Salem

Industrial Engineering Department, College of Engineering, King Saud University, P.O. Box 800, Riyadh 11421, Saudi Arabia

Correspondence should be addressed to Abdelghani Bouras; bouras@ksu.edu.sa

Received 6 August 2014; Accepted 2 September 2014; Published 20 November 2014

Academic Editor: Dehua Xu

Copyright © 2014 Abdelghani Bouras et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The airport gate assignment problem (AGAP) is one of the most important problems operations managers face daily. Many researches have been done to solve this problem and tackle its complexity. The objective of the task is assigning each flight (aircraft) to an available gate while maximizing both conveniences to passengers and the operational efficiency of airport. This objective requires a solution that provides the ability to change and update the gate assignment data on a real time basis. In this paper, we survey the state of the art of these problems and the various methods to obtain the solution. Our survey covers both theoretical and real AGAP with the description of mathematical formulations and resolution methods such as exact algorithms, heuristic algorithms, and metaheuristic algorithms. We also provide a research trend that can inspire researchers about new problems in this area.

1. Introduction

The complexity of airport management has increased significantly. Flight delays or accidents might happen if operations were not handled well, and domino effect might happen to influence the whole operations of airport. In airports, the tasks related to gate assignment problem (AGAP) are one of the most important daily operations many researches have been published on with the aim of solving the problem in spite of its complexity. The objective of the task is assigning each flight (aircraft) to an available gate while maximizing both conveniences to passengers and the operational efficiency of airport. Large airlines typically need to manage different gates across an airport in the most efficient way in a dynamic operational environment. This requires a solution that provides the ability to change and update the gate assignment data on a real time basis. It should also provide robust and efficient disruption management, while maintaining safety, security, and cost efficiency.

Numerous methods have been developed to solve this problem since 1974. Steuart [1] proposed simple stochastic model to find the efficiency use of the gate positions. The research interest in this field was slow in development because there were less than 15 publications within 25 years. However,

after 2000, the interest to develop solutions for this problem increased, until nowadays, though with small growth. The objective of this problem varied and depended on the point of view. The first is as an airport owner, which is the government. The objectives are to maximize the utilization of the available gates and terminal [1–4], minimize the number of gate conflicts [5], minimize the number of ungated flights [3, 6–9], and minimize the flights delay [10]. Another point of view is as an airlines owner. Their goals were to increase the customer satisfaction with minimizing the passenger walking distance between gates [3, 6, 7, 11–18] and minimizing the travelling distance from runway to the gate [19].

Dorndorf et al. [20] divided the objectives into five parts, which are reducing the number of the procedures for the costly aircraft towing, minimizing the passengers total walking distance, minimizing the deviations in the schedules, minimizing the number of ungated aircrafts, and maximizing the preferences (i.e., certain aircrafts should go for particular gates). They also defined three usually used constraints, which are the fact that only one aircraft can be gated in a defined amount of time, the fulfillment of the space restriction and service requirements, and the assurance of getting a minimum time between sequent aircrafts and a minimum ground time.

The solution approaches and the solving techniques are varied with no methods, until nowadays, that provide a robust technique for such problem. This study focuses on assessing the trend of solving gate assignment problem in light of the preceding four points. Specifically, this study will address the following research questions. (1) Is this problem NP-hard? (2) What formulation can be defined for such problem? (3) How effective are the recent methods and techniques to solve the problem? (4) What recommendation can be made based on the current findings with regard to research trends?

From a mathematical view, AGAP has been formulated as integer, binary, or mixed integer, general linear or nonlinear models. Specific formulation as binary or mixed binary quadratic models has also been suggested. Other well-known related problems in combinatorial optimization such as quadratic assignment problem (QAP), clique partitioning problem (CPP), and scheduling problem have been used to formulate AGAP. However, few publications on AGAP tackled stochastic or robust optimization.

While the goal of combinatorial optimization research is to find an algorithm that guarantees an optimal solution in polynomial time with respect to the problem size, the main interest in practice is to find a nearly optimal or at least good-quality solution in a reasonable amount of time. Many approaches to solve the GAP have been proposed, varying from Brand and Bound (B&B) to highly esoteric optimization methods. The majority of these methods can be broadly classified as either “exact” algorithms or “heuristic” algorithms. Exact algorithms are those that yield an optimal solution. As discussed in Section 3.1 different exact solution techniques have been used to solve the GAP and in some research, the authors used some optimization programming languages like CPLEX and AMPL.

Basically the GAP is a QAP and it is an NP-hard problem as shown in Obata [21]. Since the AGAP is NP-hard, researchers have suggested various heuristic and metaheuristics approaches for solving the GAP. With heuristic algorithms, theoretically there is a chance to find an optimal solution. That chance can be remote because heuristics often reach a local optimal solution and get stuck at that point. But metaheuristics or “modern heuristics” introduce systematic rules to deal with this problem. The systematic rules avoid local optima or give the ability of moving out of local optima. The common characteristic of these metaheuristics is the use of some mechanisms to avoid local optima. Metaheuristics succeed in leaving the local optimum by temporarily accepting moves that cause worsening of the objective function value. Sections 3.2 and 3.3 addressed the heuristic and metaheuristics approaches for solving the GAP. Some papers presenting good overviews as well as annotated bibliographies on the topic of GAP and a good literature on the AGAP and the use of metaheuristics for AGAP are Dorndorf et al. [20, 22] and Cheng et al. [23].

This paper surveys a large number of models and techniques developed to deal with GAP. In Section 2, we detail the models formulations of the problem. In Section 3, we addressed the resolution methods used to solve the problem. We conclude in Section 4, and we represent the research trends.

2. Formulations of AGAP and Related Problems

Many researchers formulated the AGAP as an integer, binary, or mixed integer linear or nonlinear model and some of them formulated it as binary or mixed binary quadratic models, whereas some of the researchers have formulated the AGAP as well-known related problems in combinatorial optimization such as quadratic assignment problem (QAP), clique partitioning problem (CPP), and scheduling problem or even as a network representation. However, some of the researchers formulated the AGAP as a robust optimization model. In this section, according to the way of how the researchers deal with the gate assignment problem, a classification for the AGAP has been made as follows.

2.1. Selected AGAP Formulations

2.1.1. Integer Linear Programming Formulations (IP). Lim et al. [24] formulated the AGAP as an integer programming model and developed two models with time windows. The first model was devoted to minimization of the passenger walking distance (travel time)

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^m \sum_{l=1}^m f_{ij} w_{kl} z_{ijkl} + \sum_{i=1}^n p_i (c_i - a_i) \quad (1)$$

while the second model optimized the gate assignments with cargo handling costs:

$$\begin{aligned} \text{Minimize } & \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^m \sum_{l=1}^m [f_{ij}^A (c_{kl} + c^A) + f_{ij}^B (c_{kl} + c^B)] z_{ijkl} \\ & + \sum_{i=1}^n p_i (c_i - a_i). \end{aligned} \quad (2)$$

Both of these objectives put a penalty function due to a delay. These two objectives used the constraints, as follows:

$$\sum_{k=1}^m x_{ik} = 1, \quad 1 \leq i \leq n, \quad (3)$$

$$z_{ijkl} \leq x_{ik}, \quad 1 \leq i, j \leq n, 1 \leq k, l \leq m, \quad (4)$$

$$z_{ijkl} \leq x_{jl}, \quad 1 \leq i, j \leq n, 1 \leq k, l \leq m, \quad (5)$$

$$x_{ik} + x_{jl} - 1 \leq z_{ijkl}, \quad 1 \leq i, j \leq n, 1 \leq k, l \leq m, \quad (6)$$

$$c_i \geq a_i, \quad 1 \leq i \leq n, \quad (7)$$

$$c_i \leq b_i - d_i, \quad 1 \leq i \leq n, \quad (8)$$

$$(c_i + d_i) - c_i + y_{ij} M > 0, \quad 1 \leq i, j \leq n, \quad (9)$$

$$(c_i + d_i) - c_i - (1 - y_{ij}) M \leq 0, \quad 1 \leq i, j \leq n, \quad (10)$$

$$y_{ij} + y_{ji} \leq z_{ijkl}, \quad 1 \leq i, j \leq n, i \neq j, 1 \leq k \leq m, \quad (11)$$

where x_{ik} , y_{ij} , and z_{ijkl} are binary and c_i is integer.

Constraint (3) ensures that each flight must be assigned to exactly one gate. Constraints (4)-(5) state that a binary variable z_{ijkl} can be equal to one if flight i is assigned to gate k ($x_{ik} = 1$) and flight j is assigned to gate l ($x_{jl} = 1$). Constraint (6) further specifies the necessary condition that z_{ijkl} must be equal to one if $x_{ik} = 1$ and $x_{jl} = 1$. Constraints (7) and (8) ensure that the flight must land and depart within the specified time window. Constraint (9) indicates that $y_{ij} = 1$ if $(c_i + d_i) \leq c_j$, which means $y_{ij} = 1$ when flight i departs before or right at the time when some gate opens for flight j . Constraint (10) states that $y_{ij} = 0$ if $(c_i + d_i) > c_j$, which means $y_{ij} = 0$ when flight i departs after some gate opens for flight j . Constraint (11) specifies that one gate cannot be occupied by two different flights simultaneously.

In the first model and according to the linearity of the objective function and constraints, they used a standard IP solver (CPLEX) to find the optimal solution, whereas in the second model authors used several heuristic algorithms, namely, the "Insert Move Algorithm," the "Interval Exchange Move Algorithm," and a "Greedy Algorithm" to generate solutions. The generated solutions then have been improved using a tabu search (TS) and memetic algorithm. The results showed that the used heuristics performed better than the IP solver (CPLEX) in both CPU time and solutions quality.

Diepen et al. [25, 26] formulated the AGAP as integer linear programming model with a relaxation for the integrality. After relaxing the integrality, the resulting relaxed LP was exploited to obtain solutions of ILP by using column generation (CG). The problem was divided into two phases, planning and attaching. The first phase was the planning section and it was easier to model and calculate. Their objective is to minimize the cost of a gate plan. They proposed the following model:

$$\text{Minimize } \sum_{j=1}^n c_j x_j \tag{12}$$

subject to

$$\sum_{j=1}^n g_{ij} x_j + U_i = 1 \quad \text{for } i = 1, \dots, m, \tag{13}$$

where

$$g_{ij} = \begin{cases} 1, & \text{if flight } i \text{ is in gate plan } j, \\ 0, & \text{otherwise,} \end{cases} \tag{14}$$

$U_i \geq 0$ for $i = 1, \dots, m$, U_i is a penalty variable.

This constraint defined the high price penalty (U_i) when the flights were not assigned to the gates. This penalty appeared since the planner should do the assignment manually. In addition, they added another constraint regarding the assignment since there was possibility that a long stay flight could be split into two parts. The extra flights i_a and i_b that refer to

the arrival and departure of flight i were added to the previous constraints:

$$\sum_{j=1}^n (g_{ij} + g_{iA,j}) x_j + U_{iA} = 1, \tag{15}$$

$$\sum_{j=1}^n (g_{ij} + g_{iB,j}) x_j + U_{iB} = 1,$$

$$\sum_{j=1}^n t_{jh} x_j = T_h \quad \text{for } h = 1, \dots, H, \tag{16}$$

$$x_j \in \{0, 1\} \quad \text{for } j = 1, \dots, n, \tag{17}$$

where

$$x_j = \begin{cases} 1, & \text{if gate plan } j \text{ is selected,} \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

$$t_{jh} = \begin{cases} 1, & \text{if gate plan } j \text{ is type } h, \\ 0, & \text{otherwise,} \end{cases}$$

$$\sum_{j=1}^n \sum_{i=1}^m \sum_{h=1}^H p_{ihr} t_{jh} g_{ij} x_j \geq P_r \quad \text{for } r = 1, \dots, R, \tag{19}$$

where

$$p_{ihr} = \begin{cases} 1, & \text{if flight } i \text{ has preference on gate type } h \\ & \text{in preference } r, \\ 0.5, & \text{if the unsplit version of flight } i \text{ has} \\ & \text{preference on gate type } h \text{ in preference } r, \\ 0, & \text{otherwise,} \end{cases} \tag{20}$$

and R denotes the total number of preferences.

This constraint defined the flight preferences; for example, a flight should be assigned to the same gate due to the ownership or security. The coefficient 0.5 refers to the extra flight defined in constraint (15).

They checked the solution's optimality using pricing problem (minimum reduced cost) since they had dual multipliers π_i , λ_h , and ψ_r for constraints (15), (16), and (19), respectively:

$$c_j - \sum_{h=1}^H t_{jh} \lambda_h - \sum_{i=1}^m \left(g_{ij} \pi_i + \sum_{r=1}^R \sum_{h=1}^H (g_{ij} t_{jh} p_{ihr} \psi_r) \right). \tag{21}$$

The second phase was a matter of assignment in physical gate. They made the rules to solve this phase as follows.

- (i) Sort the gates based upon the quality.
- (ii) Sort the gate plans from the highest on the total number of departing passengers that are on the flights in that gate plan.
- (iii) Assign the gate plan to the best gate considering the highest number of departing passengers, assign the next gate plan to the next-best gate, and so on.

In [26], Diepen et al. used the solution obtained from their assignment of gates as an input to solve the bus-planning problem in the same airport.

2.1.2. *Binary Integer Programming.* In 2009, Tang et al. [27] formulated the AGAP as a binary integer programming model as below. The output model was used to generate a lower bound to their original problem:

$$\text{Minimize } Z = \sum_{i \in I} \sum_{j \in E_i} \sum_{k \in D_{ij}} d_{ik} x_{ijk} + \sum_{i \in I} \sum_{j \in E_i} \sum_{k \in D_{ij}} w_{ij} x_{ijk} \quad (22)$$

subject to

$$\sum_{j \in E_i} \sum_{k \in D_{ij}} x_{ijk} = 1, \quad \forall i \in I, \quad (23)$$

$$\sum_{i \in F_{js}} \sum_{k \in H_{is}} x_{ijk} \leq 1, \quad \forall j \in G, \quad \forall s \in S, \quad (24)$$

$$\sum_{i \in L_{tq}} \sum_{j \in N_q} \sum_{k \in H_{is}} x_{ijk} \leq 1, \quad \forall t \in T_q, \quad \forall q \in Q, \quad \forall s \in S, \quad (25)$$

$$x_{ijk} = \begin{cases} 1, & \text{if flight } i \text{ is assign to gate } j \\ & \text{at the time point (starting point) } k \\ 0, & \text{otherwise} \end{cases} \quad (26)$$

$$\forall k \in D_{ij}, \quad \forall j \in E_i, \quad \forall i \in I.$$

Parameter Variables

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

w_{ij} : space inconsistency value indicating that the i th flight is assigned to the j th gate; if j equals the original gate, then $w_{ij} = 0$.

The following sets have been defined:

I : considered flights;

G : available gates;

E_i : gates that the i th flight can be assigned to;

D_{ij} : time points in which the i th flight can be assigned to the j th gate;

F_{js} : flights that can be assigned to the j th gate so that their time windows will cover the s th time point;

S : all time points (i.e., the time points from the planning time at each stage to the end of daily operations);

H_{is} : time points (starting times) assigned to the i th flight, where the resulting time windows cover the s th time point;

T_q : conflicting flight pairs for the q th adjacent gate pair;

L_{tq} : flights included in the t th conflicting flight pair for the q th adjacent gate pair;

Q : adjacent gate pairs.

Equation (23) is the flight constraint, indicating that every flight is exactly assigned to a gate. Equation (24) is the gate constraint, ensuring that every gate is assigned to at most one flight at any time. Constraint (25) is related gate adjacency, denoting that two conflicting flights cannot be concurrently assigned to an adjacent gate pair. Constraint (26) indicates that the assignment variables are either zero or one.

Kumar et al. [18] presented a binary integer programming model that produced a feasible gate plan in the light of all the business constraints:

$$x_{ik} = \begin{cases} 1, & \text{if turn } i \text{ is assigned to gate } k, \\ 0, & \text{otherwise,} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{if turn } i \text{ is not assigned to any gate,} \\ 0, & \text{otherwise,} \end{cases} \quad (27)$$

$$w_t = \begin{cases} 1, & \text{if long turn } t \text{ is towed,} \\ 0, & \text{otherwise,} \end{cases}$$

$$\text{Maximize } \sum_{i \in T} \sum_{k \in K} C_{ik} x_{ik} - C_1 \sum_{t \in L} w_t - C_2 \sum_{i \in T} y_i$$

subject to

$$\sum_{k \in K} x_{ik} + y_i = 1, \quad i \in T, \quad (28)$$

$$\sum_{k \in K; e_i \in E_k} x_{ik} + y_i = 1, \quad i \in T, \quad (29)$$

$$\sum_{i \in T} y_i \leq \xi, \quad (30)$$

$$y_{ik} + y_{jk} \leq 1, \quad i, j \in T; \quad k \in K: a_i < b_j + \alpha, \quad (31)$$

$$a_j < b_i + \alpha, \quad i \neq j$$

$$y_{ik} + y_{jl} \leq 1, \quad i, j \in T; \quad l \in K;$$

$$(k, l) \in J: a_i < b_j, \quad a_j < b_i, \quad (32)$$

$$i \neq j, \quad e_i \in E_k^1, \quad e_j \in E_l^1$$

$$y_{ik} + y_{jl} \leq 1, \quad i, j \in T; \quad k, l \in K;$$

$$(k_{LiFo}^F, k_{LiFo}^R) \in LF: a_j \leq a_i \leq b_j, \quad (33)$$

$$i \neq j, \quad e_i \in E_{k_{LiFo}^F}, \quad e_j \in E_{l_{LiFo}^R}$$

$$y_{ik} + y_{jl} \leq 1, \quad i, j \in T; \quad k, l \in K;$$

$$(k_{LiFo}^F, l_{LiFo}^R) \in LF: a_j \leq b_i \leq b_j, \quad (34)$$

$$i \neq j, \quad e_i \in E_{k_{LiFo}^F}, \quad e_j \in E_{l_{LiFo}^R}$$

$$y_{ik} + y_{jl} \leq 1, \quad i, j \in T; \quad k, l \in K;$$

$$(k^{PB}, l^{PB}) \in J^{PB}: b_i - \beta < b_j \leq b_i + \beta, \quad (35)$$

$$i \neq j, \quad e_i \in E_k, \quad e_j \in E_l$$

$$w_t \leq \tau, \quad t \in T_L \tag{36}$$

$$y_{i_1 k} - y_{i_2 k} \leq w_t, \quad i_1, i_2 \in T, \quad t \in T_L, \tag{37}$$

$$k \in K: i_1 \neq i_2, \quad p_{i_1} = p_{i_2} = t$$

$$y_{i_1 k} - y_{i_3 k} - 1 \leq w_t, \quad i_1, i_2, i_3 \in T, \quad t \in T_L, \tag{38}$$

$$k \in K: a_{i_1} < a_{i_3}, \quad b_{i_2} < a_{i_3}, \quad p_{i_1} = l, \quad p_{i_2} = l$$

$$y_{i_1 k_1} - y_{i_3 k_2} - 1 \leq w_t, \quad i_1, i_2, i_3 \in T, \quad t \in T_L,$$

$$k_1, k_2 \in K; \quad j \in J: a_{i_1} < a_{i_3}, \quad b_{i_2} < b_{i_3}, \quad p_{i_1} = l, \tag{39}$$

$$p_{i_2} = l, \quad k_1 = q_j^1, \quad k_2 = q_j^2, \quad e_{i_1} \in E_j^1, \quad e_{i_2} \in E_j^2.$$

Constraint (28) ensures that turn i is assigned to at most one gate. Constraint (29) states that turn i is assigned to a gate only if its equipment type is among the types which the assigned gate can accommodate. Constraint (30) restricts the number of ungated turns to less than or equal to the allowed number ξ . Constraint (31) shows that, at any given time, at most one turn is assigned to one gate. Constraint (32) ensures that adjacency constraints are observed. Constraints (33)-(34) enforce LIFO restrictions. Constraint (35) guarantees that pushback restrictions are observed. Constraint (36) confirms that no turn is towed if towing is not allowed. Finally, constraints (37)–(39) certify that if a long turn t is towed, the w_t variable is set to be 1.

Mangoubi and Mathaisel [11] also developed a binary integer model to minimize the passenger total walking distance and proposed a heuristic method to find the solution. The heuristic method result has been compared with the results from a standard IP solver and the comparison results showed that the heuristic method was superior to the LP solver; the average walking distance using the LP is 527 feet while heuristic is 558 feet. The developed model is introduced as follows:

$$\text{Minimize } Z = \sum_{i=1}^M \sum_{j=1}^M (p_i^a d_j^a + p_i^d d_j^d + p_i^t d_j^t) x_{ij}, \tag{40}$$

where

$$x_{ij} = \begin{cases} 1, & \text{if flight } i \text{ is assigned to gate } j, \\ 0, & \text{otherwise.} \end{cases} \tag{41}$$

Transfer passenger walking distances are determined from a uniform probability distribution of all intergate walking distances. The expected walking distance if w_{jk} is the distance between gate j and gate k is

$$d_j^t = \frac{1}{N} \sum_{k=1}^N w_{jk} \quad \forall j = 1, \dots, N \tag{42}$$

subject to

$$\sum_{j=1}^N x_{ij} = 1 \quad \forall i = 1, \dots, M \tag{43}$$

$$\sum_{h \in L(i)} x_{hj} + x_{ij} \leq 1 \quad \forall i = 1, \dots, M, \quad \forall j = 1, \dots, N, \tag{44}$$

$$\sum_{h \in L(p+1)} x_{hj} + x_{p+1,j} = x_{p-3,j} + x_{p-2,j} \tag{45}$$

$$+ x_{p-1,j} + x_{pj} + x_{p+1,j} \leq 1,$$

$$L(p) \subset L(p+1) \subset \dots \subset L(p+k), \tag{46}$$

$$\sum_{z=1}^N \sum_{s=1}^N x_{gz} w_{zs} x_{hs} \leq D_{gh}^{\max}, \tag{47}$$

$$p_f^a d_s^a + p_f^d d_s^d + p_f^t d_s^t = \min_{j \in S} \{p_f^a d_j^a + p_f^d d_j^d + p_f^t d_j^t\}. \tag{48}$$

Constraint (43) shows that each flight is assigned to at most one gate. Constraint (44) ensures that no two planes are assigned to the same gate concurrently. Constraint (45) determines the conflict constraint for each gate j . Constraint (46) is written to consider only the constraint generated by the last plane of two or more flights arriving with no departure in between. Constraint (47) ensures that flights are assigned to nearby gates. Constraint (48) assigns flight f to gate $s \in S$, where s is the gate with the minimum total passenger walking distance for flight f .

Vanderstraeten and Bergeron [28] formulated the GAP as a binary integer model but with the objective of minimizing the off-gate events and they developed a new heuristic, which is the "Affectation Directe des Avions aux Portes (ADAP)," to solve the developed model. A real case has been studied in an Air Canada terminal. A new heuristic was applied to real data at Toronto International Airport. The developed model was as follows:

$$\text{Maximize } Z = \sum_{i \in I} \sum_{j \in J} V_{ij} \tag{49}$$

subject to

$$\sum_{j \in J_i} V_{ij} \leq 1, \quad i \in I, \tag{50}$$

$$V_{ij} + \sum_{k \in A_{ij}} V_{kj} \leq 1, \quad i \in I, \quad j \in J_i, \tag{51}$$

$$V_{ij} + \sum_{k \in B_{ijr}} V_{kr} + \sum_{k \in B_{ijl}} V_{kl} \leq 1, \quad i \in I, \quad j \in J_i, \tag{52}$$

$$V_{ij} = 0, 1 \quad i \in I, \quad j \in J_i. \tag{53}$$

Constraint (50) ensures that each flight is assigned to at most one gate. Constraint (51) defines the occupation time at any gate. Constraint (52) determines the neighboring constraint. Constraint (53) expresses the binary constraint for all decision variables. The results showed that using the

developed method resulted in no more than 30 events ever being handled off gate while the manual procedure obtained events up to 50 of the 300 events being handled off gate.

Bihir [12] developed a binary integer model to minimize the passenger walking distance and applied this model to solve a sample problem using primal-dual simplex algorithm. As a result, he obtained a total walking distance of 22,640. The developed model is introduced as follows:

$$\text{Minimize } \theta = \sum C_{ij}X_{ij}, \quad i = 1, \dots, m, \quad j = 1, \dots, k, \quad (54)$$

where the C_{ij} are the elements of the matrix product of $PAX(i, j) * DIST(i, j)^T$ and

$PAX(i, j)$ = number of passengers arriving on flight i and departing from gate j ;

$DIST(i, j)$ = number of passengers – distance units from gate i to gate j ;

$$X_{ij} = 0 \text{ or } 1$$

subject to

$$\begin{aligned} \sum X_{ij} &= 1, \\ \sum X_{ji} &= 1. \end{aligned} \quad (55)$$

In 2002, Yan et al. [29] formulated the static GAP as a binary integer programming model to serve as a basis of real time gate assignments in a simulation framework developed to analyze the effects of stochastic flight delays on static gate assignments. The presented model is as follows:

$$\text{Minimize } Z = \sum_{i=1}^M \sum_{j=1}^N c_{ij}x_{ij} \quad (56)$$

subject to

$$\sum_{j=1}^N x_{ij} = 1 \quad \forall i, \quad (57)$$

$$\sum_{i \in L_s} x_{ij} \leq 1 \quad \forall s, \forall j, \quad (58)$$

$$x_{ij} = 0 \text{ or } 1 \quad \forall i, \forall j. \quad (59)$$

Constraint (57) ensures that each flight is assigned to at most one gate. Constraint (58) certifies that no two planes are assigned to the same gate concurrently. Constraint (59) is related to the binary constraint for all decision variables. Two greedy heuristics were used to solve the model and their results were compared with the insights of the optimization method. The simulation framework was tested to solve certain real case instances from CKS airport. The results of the used methods were 24,562,588 for the optimization model and 27,833,552 and 30,166,809 (meters) for the two greedy heuristics.

2.1.3. Mixed Integer Linear Programming (MILP). Bolat [30] formulated a mixed integer program for the AGAP with the objective of minimizing the range of slack times (slack time is an idle time between two successive utilizations of the gate). Certain instances, with more than 20 gates, have been considered according to airplane types, gate types, terminal types, and utilization levels:

$$\text{Minimize } v^+(\Gamma) - v^-(\Gamma) \quad (60)$$

subject to

$$v^+(\Gamma) = \max \{v_i(\Gamma)\}, \quad i = 1, \dots, N + M, \quad (61)$$

$$v^-(\Gamma) = \min \{v_i(\Gamma)\}, \quad i = 1, \dots, N + M, \quad (62)$$

$$v_i(\Gamma) = A_i - E_{i,\Gamma_i}, \quad i = 1, \dots, N, \quad (63)$$

$$\begin{aligned} &E_{i,\Gamma_i} \\ &= \begin{cases} D_{k^*}, & \text{where } k^* = \max \{k\}, \Gamma_k = \Gamma_i, k = 1, \dots, i - 1, \\ B_{\Gamma}, & \text{if no such } k \text{ exist,} \end{cases} \end{aligned} \quad (64)$$

$$v_{N+j}(\Gamma) = T_j - E_{N+1,j}, \quad j = 1, \dots, M. \quad (65)$$

The results related to expected average utilizations were, respectively, 88.54%, 67.13%, and 45.57% over heavily utilized, normally utilized, and underutilized problems. Concerning the average number of flights, results were 10%, 7.59%, and 5.15% per gate.

In 2001, Bolat [31] presented a framework for the GAP that transformed the nonlinear binary models (it will be discussed in Section 2.1.4 according to our classification) into an equivalent linear binary model with the objective of minimizing the range or the variance of the idle times. The framework consists of five mathematical models, where two of the five models were formulated as a mixed integer linear programming and the others as a mixed integer nonlinear programming. Models P1 to P4 were defined for homogenous gate while model P5 was defined for heterogeneous gate:

$$\begin{aligned} X_{jk} &= \begin{cases} 1, & \text{if flight } j \text{ is assigned to gate } k, \\ 0, & \text{otherwise,} \end{cases} \\ P_{jk} &= \begin{cases} 1, & \text{if the assignment of flight } j \text{ to gate } k \text{ can} \\ & \text{satisfy all considerations,} \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (66)$$

Using the presented framework, nonlinear model P1 (model P1 will be discussed in Section 2.1.4 according to our classification) was transformed to the following mixed integer linear model, which is model P2.

Model P2. Consider

$$\text{Minimize } S_{\max} - S_{\min} \quad (67)$$

subject to

$$S_{\max} \geq S_{jk} \quad j = 1, \dots, N, \quad k = 1, \dots, M, \quad (68)$$

$$S_{\min} \leq S_{jk} + (1 - X_{jk})Z \quad j = 1, \dots, N, \quad k = 1, \dots, M, \quad (69)$$

$$S_{\max}, S_{\min} \geq 0, \quad (70)$$

$$\sum_{k=1}^M P_{jk} X_{jk} = 1 \quad j = 1, \dots, N, \quad (71)$$

$$E_{1k} = \text{Maximize} \{A_1 X_{1k}, B_k\} \quad k = 1, \dots, M, \quad (72a)$$

$$E_{jk} = \text{Maximize} \{A_j X_{jk}, L_{j-1,k}\} \quad j = 2, \dots, N, \quad (72b)$$

$$k = 1, \dots, M,$$

$$L_{jk} = E_{jk} + G_j X_{jk} \quad j = 1, \dots, N, \quad k = 1, \dots, M, \quad (73)$$

$$S_{1k} = E_{1k} - B_k \quad k = 1, \dots, M, \quad (74a)$$

$$S_{jk} = E_{jk} - L_{j-1,k} \quad j = 2, \dots, N, \quad (74b)$$

$$k = 1, \dots, M,$$

$$S_{N+1,k} = F_k - L_{Nk} \quad k = 1, \dots, M, \quad (75)$$

$$X_{jk} = 0 \text{ or } 1 \quad j = 1, \dots, N, \quad k = 1, \dots, M, \quad (76)$$

$$E_{jk}, L_{jk}, S_{jk}, S_{N+1,k} \geq 0 \quad j = 1, \dots, N, \quad k = 1, \dots, M. \quad (77)$$

Similarly, for model P3 (Section 2.1.4), the resultant model was model P4 that is a mixed binary model as in model P2, but with two additional real variables as follows.

Model P4. Consider

$$\text{Minimize } S_{\max} - S_{\min} \quad (78)$$

subject to

$$S_{\max} \geq I_{ij} Y_{ij} \quad i = 0, \dots, N, \quad j = i + 1, \dots, N + 1, \quad (79)$$

$$S_{\min} \leq I_{ij} Y_{ij} \quad i = 0, \dots, N, \quad j = i + 1, \dots, N + 1, \quad (80)$$

$$S_{\max}, S_{\min} \geq 0, \quad (81)$$

$$\sum_{i=0}^N \sum_{j=i+1}^{N+1} Y_{ij} = N + M, \quad (82)$$

$$\sum_{j=1}^{N+1} Y_{0j} \leq M, \quad (83)$$

$$\sum_{i=0}^N Y_{i,N+1} \leq M, \quad (84)$$

$$\sum_{i=0}^{j-1} Y_{ij} = 1 \quad j = 1, \dots, N, \quad (85)$$

$$\sum_{j=i+1}^{N+1} Y_{ij} = 1 \quad i = 1, \dots, N, \quad (86)$$

$$Y_{ij} = 0 \text{ or } 1 \quad i = 0, 1, \dots, N, \quad j = i + 1, \dots, N + 1. \quad (87)$$

Different instances have been studied according to the number of the gates: small (five gates), medium (10 gates), and large (20 gates). Instances with more than 20 gates were not considered. The results were as follows: average numbers of flight were 26.125, 52.25, and 105.417 and the average utilizations were 45.725, 66.548, and 88.871% according to the gate size, respectively.

Şeker and Noyan [9] formulate the GAP as a mixed integer program with the objective of minimizing the number of conflicts and at the same time minimizing the total semideviation between idle time and buffer time:

$$\text{Minimize } \sum_{i \in N^c \setminus \{0\}} \sum_{s \in S} R_{i,s} P_s + \Lambda \sum_{i \in N^c} \sum_{j \in N^c} \sum_{s \in S} c_{i,j,s} P_s. \quad (88)$$

Another model was developed as a mixed integer program for the same objective function. The model was the same as the previous model but with some differences:

$$\text{Minimize } \sum_{i \in N^c \setminus \{0\}} \sum_{s \in S} H_{i,s} P_s + \Lambda \sum_{i \in N^c} \sum_{j \in N^c} \sum_{s \in S} c_{i,j,s} P_s, \quad (89)$$

where

$$H_{i,s} = \begin{cases} 1, & \text{if idle time of flight } i \text{ is} \\ & \text{less than the buffer time} \\ 0, & \text{otherwise.} \end{cases} \quad (90)$$

These two models have the same constraints properties,

$$\sum_{k \in M} x_{i,k} = 1, \quad i \in N, \quad (91)$$

$$x_{0,k} = 1, \quad k \in M, \quad (92)$$

$$x_{n+k,k} = 1, \quad k \in M, \quad (93)$$

$$\sum_{j \in L_{i,s}} x_{j,k} + x_{i,k} \leq 1, \quad i \in N^c, \quad k \in M, \quad s = 0, \quad (94)$$

$$c_{i,j,s} \geq x_{i,k} + x_{j,k} - 1, \quad i \in N^c, k \in M, s \in S, j \in L_{i,s}, \tag{95}$$

$$R_{i,s} \geq b - I_{i,s}, \quad i \in N^c \setminus \{0\}, s \in S, \tag{96}$$

$$D_{i,s} \geq (x_{i,k} + x_{j,k} - 2)Z + d_{j,s}, \quad i \in N^c \setminus \{0\}, k \in M, s \in S, j \in Q_{i,s} \tag{97}$$

$$I_{i,s} \leq a_{i,s} - D_{i,s} + b \sum_{j \in N^c} c_{i,j,s}, \quad i \in N^c \setminus \{0\}, s \in S, \tag{98}$$

$$x_{i,k} \in \{0, 1\}, \quad i \in N^c, k \in M, \tag{99}$$

$$\text{All remaining variables} \geq 0, \tag{100}$$

while objective (89) has the following additional constraints:

$$H_{i,s}b \geq b - I_{i,s}, \quad i \in N^c \setminus \{0\}, s \in S, \tag{101}$$

$$H_{i,s} \in \{0, 1\}, \quad i \in N^c \setminus \{0\}, s \in S. \tag{102}$$

2.1.4. Mixed Integer Nonlinear Programming. Li [5] formulated the GAP as a nonlinear binary mixed integer model hybrid with a constraint programming in order to minimize the number of gate conflicts of any two adjacent aircrafts assigned to the same gate. The developed model has been solved using CPLEX software:

$$\text{Minimize } \sum_{i \in N} \sum_{j < i, j \in N} y_{i,j} * E(p(i, j)), \tag{103}$$

where

$$E(p(i, j)) = \frac{1}{a_i - d_j + 2b}, \tag{104}$$

where a : scheduled arriving time, d : scheduled departure time, and b : buffer time (constant). Consider

$$x_{i,k} = \begin{cases} 1, & \text{if and only if aircraft } f_i \text{ is assigned to gate } c_k, \\ 0, & \text{otherwise } (1 \leq i \leq n, 1 \leq k \leq c), \end{cases}$$

$$y_{i,j} = \begin{cases} 1, & \text{if } \exists k, x_{i,k} = x_{j,k} = 1 \quad (1 \leq k \leq c), \\ 0, & \text{otherwise } (1 \leq i, j \leq n). \end{cases} \tag{105}$$

In another work, Li [32] defined the objective as

$$\text{Minimize } \sum_{i,j \in N; i \neq j} \frac{y_{i,j}}{a_i - d_j + 2b}. \tag{106}$$

These two models have the same constraints; all constraints are as follows.

$$\sum_{i \in N} \sum_{k \in C} x_{i,k} = 1, \tag{107}$$

$$\sum_{i \in N} \sum_{j < i, j \in N} \sum_{k \in C} (x_{i,k} * x_{j,k}) = y_{i,j}, \tag{108}$$

$$y_{i,k} * y_{j,k} * (d_i - a_j) * (d_j - a_i) \leq 0, \tag{109}$$

$$x_{i,k} \in \{0, 1\}, \tag{110}$$

$$\forall 1 \leq i, j \leq n, i \neq j, \quad \forall 1 \leq k \leq c. \tag{111}$$

Constraint (107) indicates that each aircraft is assigned to at most only one gate. Constraint (108) represents a method to compute the auxiliary variable y_{ij} from x_{ik} . Constraint (109) ensures that one gate can only be assigned at most one aircraft at the same time. Some additional constraints in the real operations are ignored. Constraint (110) represents binary value of the decision variables.

As mentioned in Section 2.1.3, Bolat [31] proposed two models formulated as a mixed integer linear program which have been transformed from a mixed integer nonlinear program. The proposed mixed integer nonlinear program was as follows:

$$X_{jk} = \begin{cases} 1, & \text{if flight } j \text{ is assigned to gate } k, \\ 0, & \text{otherwise,} \end{cases}$$

$$P_{jk} = \begin{cases} 1, & \text{if the assignment of flight } j \text{ to gate } k \\ & \text{can satisfy all considerations,} \\ 0, & \text{otherwise.} \end{cases} \tag{112}$$

Model P1. Consider

$$\text{Minimize } \sum_{k=1}^M \sum_{j=1}^{N+1} S_{jk}^2 \tag{113}$$

subject to

$$\sum_{k=1}^M P_{jk} X_{jk} = 1 \quad j = 1, \dots, N, \tag{114}$$

$$E_{1k} = \text{Maximize } \{A_1 X_{1k}, B_k\} \quad k = 1, \dots, M, \tag{115a}$$

$$E_{jk} = \text{Maximize } \{A_j X_{jk}, L_{j-1,k}\} \quad j = 2, \dots, N, k = 1, \dots, M, \tag{115b}$$

$$L_{jk} = E_{jk} + G_j X_{jk} \quad j = 1, \dots, N, k = 1, \dots, M, \tag{116}$$

$$S_{1k} = E_{1k} - B_k \quad k = 1, \dots, M, \tag{117a}$$

$$S_{jk} = E_{jk} - L_{j-1,k} \quad j = 2, \dots, N, k = 1, \dots, M, \tag{117b}$$

$$S_{N+1,k} = F_k - L_{Nk} \quad k = 1, \dots, M, \tag{118}$$

$$X_{jk} = 0 \text{ or } 1 \quad j = 1, \dots, N, \quad k = 1, \dots, M, \tag{119}$$

$$E_{jk}, L_{jk}, S_{jk}, S_{N+1,k} \geq 0 \quad j = 1, \dots, N, \quad k = 1, \dots, M. \tag{120}$$

Bolat [31] also proposed two alternative formulations for homogenous and heterogeneous gates. The proposed extended formulation for the homogenous gates was as follows:

$$Z = \max_{k=1, \dots, M} \{F_k - B_k\}, \tag{121}$$

$$I_{ij} = \begin{cases} A_j - D_i, & \text{if } A_j \geq D_i, \\ Z, & \text{otherwise,} \end{cases} \tag{122}$$

$$I_{0j} = A_j \quad j = 1, \dots, N, \tag{123}$$

$$I_{i,N+1} = H - D_i \quad i = 1, \dots, N. \tag{124}$$

Model P3. Consider

$$\text{Minimize } \sum_{i=0}^N \sum_{j=i+1}^{N+1} I_{ij}^2 Y_{ij} \tag{125}$$

subject to

$$\sum_{i=0}^N \sum_{j=i+1}^{N+1} Y_{ij} = N + M, \tag{126}$$

$$\sum_{j=1}^{N+1} Y_{0j} \leq M, \tag{127}$$

$$\sum_{i=0}^N Y_{i,N+1} \leq M, \tag{128}$$

$$\sum_{i=0}^{j-1} Y_{ij} = 1 \quad j = 1, \dots, N, \tag{129}$$

$$\sum_{j=i+1}^{N+1} Y_{ij} = 1 \quad i = 1, \dots, N, \tag{130}$$

$$Y_{ij} = 0 \text{ or } 1 \quad i = 0, 1, \dots, N, \quad j = i + 1, \dots, N + 1. \tag{131}$$

In addition, the proposed extended formulation for the heterogeneous gates was as follows:

$$I_{ijk} = \begin{cases} A_j - D_i, & \text{if } A_i > B_k, \quad A_j \geq D_i, \\ P_{ik} = P_{jk} = 1, & \\ Z, & \text{otherwise,} \end{cases} \tag{132}$$

$$I_{0jk} = \begin{cases} A_j - B_k, & \text{if } A_j \geq B_k, \\ Z, & \text{otherwise,} \end{cases} \tag{133}$$

$$I_{i,N+1,k} = \begin{cases} F_k - D_i, & \text{if } D_i \leq F_k, \\ Z, & \text{otherwise.} \end{cases} \tag{134}$$

Model P5. Consider

$$\text{Minimize } \sum_{k=1}^M \sum_{i=0}^N \sum_{j=i+1}^{N+1} I_{ijk}^2 Y_{ijk} \tag{135}$$

subject to

$$\sum_{k=1}^M \sum_{i=0}^N \sum_{j=i+1}^{N+1} Y_{ijk} = N + M, \tag{136}$$

$$\sum_{j=1}^{N+1} Y_{0jk} \leq 1 \quad k = 1, \dots, M, \tag{137}$$

$$\sum_{k=1}^M \sum_{i=0}^{j-1} Y_{ijk} = 1 \quad j = 1, \dots, N, \tag{138}$$

$$\sum_{k=1}^M \sum_{j=i+1}^{N+1} Y_{ijk} = 1 \quad j = 1, \dots, N, \tag{139}$$

$$Y_{ijk} + \sum_{\substack{t=1 \\ t \neq k}}^M \sum_{v=j+1}^{N+1} Y_{jvt} \leq 1 \quad i = 0, \dots, N - 1 \tag{140}$$

$$j = i + 1, \dots, N \quad k = 1, \dots, M,$$

$$Y_{ijk} = 0 \text{ or } 1 \quad i = 0, \dots, N, \quad j = i + 1, \dots, N + 1, \tag{141}$$

$$k = 1, \dots, M$$

$$Y_{abc} + \sum_{\substack{t=1 \\ t \neq c}}^M \sum_{v=b+1}^{N+1} Y_{bvt} \leq 1. \tag{142}$$

As mentioned in Section 2.1.3, different instances have been studied according to the number of the gates: small (five gates), medium (10 gates), and large (20 gates). Instances with more than 20 gates were not considered. The results were as follows: average numbers of flight were 26.125, 52.25, and 105.417 and the average utilizations were 45.725, 66.548, and 88.871% according to the gate size, respectively.

2.1.5. Quadratic Programming

(1) *Quadratic Mixed Binary Programming.* Zheng et al. [33] formulated the GAP as a mixed binary quadratic program with minimizing the slack time overall variance as the objective function; an assumption has been stated such that the flights are sequenced with the smallest arrival time. The proposed mixed binary quadratic model was as follows:

$$\text{Minimize } f = \sum_{i=1}^{N+1} \sum_{k=1}^M (R_{ik} - \bar{R})^2 \tag{143}$$

subject to

$$\sum_{k \in M} y_{ik} = 1, \tag{144}$$

$$y_{ik} \geq \sum_{j \in N} z_{ijk}, \tag{145}$$

$$y_{jk} \geq \sum_{i \in N} z_{ijk}, \tag{146}$$

$$R_{jk} = a_j - ST_k, \quad j = \min(j), \quad j \in (j \mid y_{jk} = 1), \tag{147}$$

$$R_{N+1,k} = ET_k - d_i, \quad i = \max(i), \quad i \in (i \mid y_{jk} = 1), \tag{148}$$

$$R_{ik} = a_j - d_i, \quad \forall (i, j, k) \in \{(i, j, k) \mid z_{ijk} = 1\}, \tag{149}$$

$$a_j + (1 - z_{ijk})S \geq d_i + I, \tag{150}$$

$$v_i \leq e_k + (1 - y_{ik})S, \tag{151}$$

$$y_{ik} = \begin{cases} 1, & \text{if flight } i \text{ is assigned to gate } k, \\ 0, & \text{otherwise,} \end{cases} \tag{152}$$

$$z_{ijk} = \begin{cases} 1, & \text{if flight } i \text{ and flight } j \text{ are both} \\ & \text{assigned to gate } k, \\ 0, & \text{otherwise,} \end{cases} \tag{153}$$

where the indices i, j, k in (143)–(151) denote $i, j \in N, k \in M$. Equation (143) represents the objective function with the aim of minimizing overall variance of slack time. Constraint (144) imposes the assignment of every flight to one gate. Constraint (145) obliges every flight to have at most one immediate precedent flight. Constraint (146) enforces every flight to have at most one immediate succeeding flight. Constraints (147) and (148) define the first and last slack time of each gate, and constraint (149) defines the other slack times. Constraint (150) stipulates that the flight can be assigned to the gate when the preceding flight has departed for dwell time. Constraint (151) indicates that the different type of gate allows parking different type of flight.

Solutions were obtained using tabu search based on some initial (starting) solutions; the results were compared with those of a random algorithm developed in the literature. Using data from Beijing International Airport (10 gates and 100 of flights between 6:00 and 16:00), the initial solutions using metaheuristic and random algorithm were 9821 and 15775, respectively.

Bolat [34] formulated the AGAP as a mixed binary quadratic programming model to minimize the variance of idle times and used branch and bound algorithm and proposed two heuristics which were “single pass heuristic” (SPH) and “heuristic branch and bound” (HBB) for solving the proposed model. The proposed mixed binary quadratic model was stated as follows:

$$\text{minimize } Z = \sum_{k=1}^M \sum_{j=1}^{N+1} S_{jk}^2 \tag{154}$$

subject to

$$\sum_{k=1}^M P_{jk} X_{jk} = 1, \quad j = 1, \dots, N, \tag{155}$$

$$E_{jk} \geq A_j X_{jk}, \quad j = 1, \dots, N, \quad k = 1, \dots, M, \tag{156}$$

$$E_{jk} \geq L_{j-1,k}, \quad j = 1, \dots, N, \quad k = 1, \dots, M, \tag{157}$$

$$L_{jk} = E_{jk} + G_j X_{jk}, \quad j = 1, \dots, N, \quad k = 1, \dots, M, \tag{158}$$

$$S_{jk} = E_{jk} - L_{j-1,k}, \quad j = 1, \dots, N, \quad k = 1, \dots, M, \tag{159}$$

$$S_{N+1,k} = L_{N+1,k} - L_{Nk}, \quad k = 1, \dots, M, \tag{160}$$

$$X_{jk} = \begin{cases} 1, & \text{if flight } j \text{ is assigned to gate } k \\ 0, & \text{otherwise,} \end{cases} \tag{161}$$

$$j = 1, \dots, N, \quad k = 1, \dots, M,$$

$$E_{jk}, L_{jk}, S_{jk}, S_{N+1,k} \geq 0, \quad j = 1, \dots, N, \tag{162}$$

$$k = 1, \dots, M.$$

Real instances, from King Khalid International Airport (72 generated sets), were used. During the initial phase, the proposed heuristic methods gave an average improvement of 87.39% on the number of remote assigned flights, whereas the average improvement on the number of towed aircrafts during the real time phase was 76.19%.

Xu and Bailey [14] formulated the GAP as a mixed binary quadratic programming model (Model 1) and the objective was to minimize the passenger connection time. The proposed model (Model 1) was reformulated (linearized) into another model (Model 2) in which the objective function and the constraints have been linearized (the resultant model was a mixed binary integer model). Model 1 and Model 2 are listed below.

Model 1. Consider

$$\text{Minimize } \sum_{i,j \in N} \sum_{k,l \in K} f_{ijkl} y_{ik} y_{jl} \tag{163}$$

subject to

$$\sum_{k \in K} y_{ik} = 1, \quad \forall i \in N, \tag{164}$$

$$y_{ik} = \sum_{j \in N} z_{ijk}, \quad \forall i \in N, \quad \forall k \in K, \tag{165}$$

$$y_{jk} = \sum_{i \in N} z_{ijk}, \quad \forall j \in N, \quad \forall k \in K, \tag{166}$$

$$t_i \geq a_i + \alpha, \quad \forall i \in N, \tag{167}$$

$$t_i \leq d_i - \theta_i \sum_{j \in N} f_{ji}, \quad \forall i \in N, \tag{168}$$

$$t_i + \theta_i \sum_{j \in N} f_{ji} \leq t_j + (1 - z_{ijk})M, \quad \forall_{i,j} \in N, \forall_k \in K, \tag{169}$$

$$a_j + (1 - z_{ijk})M \geq d_i + \beta, \quad \forall_{i,j} \in N, \forall_k \in K, \tag{170}$$

$$y_{ik} = \begin{cases} 1, & \text{if flight } i \text{ is assigned to gate } k, \\ 0, & \text{otherwise,} \end{cases} \tag{171}$$

$$z_{ijk} = \begin{cases} 1, & \text{if flight } i, j \text{ are both assigned to gate } k; \\ & \text{flight } i \text{ immediately precedes flight } j, \\ 0, & \text{otherwise,} \end{cases} \tag{172}$$

$$z_{ijk}, z_{ijk} \in \{0, 1\}, \quad \forall_{i,j} \in N, \forall_{k,l} \in K, \tag{173}$$

$$t_i \geq 0, \quad \forall_i \in N, \tag{174}$$

where objective function (163) seeks to minimize the total connection times by passengers. Constraint (164) specifies that every flight must be assigned to one gate. Constraint (165) indicates that every flight can have at most one flight immediately followed at the same gate. Constraint (166) indicates that every flight can have at most one preceding flight at the same gate. Constraints (167) and (168) stipulate that a gate must open for boarding on a flight during the time between its arrival and departure and also must allow sufficient time for handling the passenger boarding, which is assumed to be proportional to the number of passengers going on board. Constraint (169) establishes the precedence relationship for the binary variable z_{ijk} and the time variables t_i and t_j and is only effective when $z_{ijk} = 1$. It stipulates that if flight i is assigned immediately before flight j to the same gate k , the gate must open for flight i earlier than for flight j . Therefore, it ensures each gate only serves one flight at any particular time. Constraint (170) further states that the aircraft can only arrive at the gate when the previous flight has departed for certain time.

Model 2. Consider

$$x_{ijkl} = \begin{cases} 1, & \text{if and only if flight } i \text{ is assigned to gate } k, \\ & \text{and flight } j \text{ is assigned to gate } l, \\ 0, & \text{otherwise,} \end{cases} \tag{175}$$

$$\text{Minimize } \sum_{i,j \in N} \sum_{k,l \in K} f_{ij} c_{kl} x_{ijkl} \tag{176}$$

subject to

$$\sum_{k \in K} y_{ik} = 1, \quad \forall_i \in N, \tag{177}$$

$$y_{ik} = \sum_{j \in N} z_{ijk}, \quad \forall_i \in N, \forall_k \in K, \tag{178}$$

$$y_{jk} = \sum_{i \in N} z_{ijk}, \quad \forall_j \in N, \forall_k \in K, \tag{179}$$

$$t_i \geq a_i + \alpha, \quad \forall_i \in N, \tag{180}$$

$$t_i \leq d_i - \theta_i \sum_{j \in N} f_{ji}, \quad \forall_i \in N, \tag{181}$$

$$t_i + \theta_i \sum_{j \in N} f_{ji} \leq t_j + (1 - z_{ijk})M, \quad \forall_{i,j} \in N, \forall_k \in K, \tag{182}$$

$$a_j + (1 - z_{ijk})M \geq d_i + \beta, \quad \forall_{i,j} \in N, \forall_k \in K, \tag{183}$$

$$z_{ijk}, z_{ijk} \in \{0, 1\}, \quad \forall_{i,j} \in N, \forall_{k,l} \in K, \tag{184}$$

$$t_i \geq 0, \quad \forall_i \in N, \tag{185}$$

$$x_{ijkl} \leq y_{ik}, \quad \forall_{i,j} \in N, \forall_{k,l} \in K, \tag{186}$$

$$x_{ijkl} \leq y_{jl}, \quad \forall_{i,j} \in N, \forall_{k,l} \in K, \tag{187}$$

$$y_{ik} + y_{jl} - 1 \leq x_{ijkl}, \quad \forall_{i,j} \in N, \forall_{k,l} \in K, \tag{188}$$

$$x_{ijkl} \in \{0, 1\}, \quad \forall_{i,j} \in N, \forall_{k,l} \in K, \tag{189}$$

where constraints (186) and (187) state that a binary variable x_{ijkl} can be equal to one if flight i is assigned to gate k ($y_{ik} = 1$) and flight j is assigned to gate l ($y_{jl} = 1$). Constraint (188) further gives the necessary condition which is that x_{ijkl} must be equal to one if $y_{ik} = 1$ and $y_{jl} = 1$.

The B&B and tabu search algorithm were used to solve the generated instances (seven instances, up to 400 flights and 50 gates for 5 consecutive working days). The results of the analyzed instances showed an average saving of the connection time of 24.7%.

(2) *Binary Quadratic Programming.* Ding et al. [6, 35] developed a binary quadratic programming model for the overconstrained AGAP to minimize the number of ungated flights. A greedy algorithm was designed to obtain an initial solution, which has been improved using tabu search (TS). The developed model was stated as follows:

$$\text{Minimize } \sum_{m+1}^n y_{i,m+1}, \tag{190}$$

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{m+1} \sum_{l=1}^{m+1} f_{i,j} w_{k,l} y_{i,k} y_{j,l} + \sum_{i=1}^n f_{0,i} w_{0,i} \tag{191}$$

$$+ \sum_{i=1}^n f_{i,0} w_{i,0}$$

subject to

$$\sum_{k=1}^{m+1} y_{i,k} = 1 \quad (\forall i, 1 \leq i \leq n), \tag{192}$$

$$a_i < d_i \quad (\forall i, 1 \leq i \leq n), \tag{193}$$

$$y_{i,k} y_{j,k} (d_j - a_i) (d_i - a_j) \leq 0 \tag{194}$$

$$(\forall i, 1 \leq i, j \leq n, k \neq m + 1),$$

$$y_{ik} = \begin{cases} 1, & \text{if flight } i \text{ is assigned to gate } k, \\ & (0 < k \leq m + 1) \\ 0, & \text{otherwise} \end{cases} \quad (195)$$

$$\forall i, 1 \leq i \leq n, \forall k, 1 \leq k \leq m + 1, \quad (196)$$

where constraint (192) ensures that every flight must be assigned to one and only one gate or assigned to the apron. Constraint (193) specifies that the departure time of each flight is later than its arrival time. Constraint (194) says that an assigned gate cannot admit overlapping the schedule of two flights.

In 2005, Ding et al. [7] developed a binary quadratic programming model for the overconstrained AGAP to minimize the number of ungated flights. The developed model was as follows:

$$\text{Minimize } \sum_{m+1}^n y_{i,m+1}, \quad (197)$$

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^n \sum_{i=1}^{m+1} f_{i,j} w_{k,l} y_{i,k} y_{j,l} + \sum_{i=1}^n f_{0,i} w_{0,i} + \sum_{i=1}^n f_{i,0} w_{i,0} \quad (198)$$

subject to

$$\sum_{k=1}^{m+1} y_{i,k} = 1 \quad (\forall i, 1 \leq i \leq n), \quad (199)$$

$$y_{i,k} y_{j,k} (d_j - a_i) (d_i - a_j) \leq 0 \quad (200)$$

$$(\forall i, 1 \leq i, j \leq n, k \neq m + 1),$$

$$y_{ik} = \begin{cases} 1, & \text{if flight } i \text{ is assigned to gate } k, \\ & (0 < k \leq m + 1), \\ 0, & \text{otherwise,} \end{cases} \quad (201)$$

$$\forall i, 1 \leq i \leq n, \quad \forall k, 1 \leq k \leq m + 1, \quad (202)$$

where constraint (199) ensures that every flight must be assigned to one and only one gate or assigned to the apron and constraint (200) requires that flights cannot overlap if they are assigned to the same gate.

Using the same case study by Ding et al. [6, 35], a greedy algorithm was designed to obtain an initial solution, which has been improved using simulated annealing (SA) and a hybrid of simulated annealing and tabu search (SA-TS).

2.1.6. Multiple Objective AGAP Formulations. Hu and Di Paolo [36] mathematically formulated the multiobjective GAP (MOGAP) as a minimization problem and solved this problem using a new genetic algorithm with uniform crossover. The developed MOGAP model was presented as follows:

$$\text{Minimize } J_{\text{MOGAP}} \quad (203)$$

subject to

$$\sum_{g=1}^{N_G} H_g = N_{AC}, \quad (204)$$

$$E_{Q_g(j)} = \begin{cases} P_{Q_g(j)}, & j = 1 \\ \max(P_{Q_g(j)}, E_{Q_g(j-1)}, G_{Q_g(j-1)}), & j > 1, \end{cases} \quad (205)$$

$$j = 1, \dots, H_g, \quad g = 1, \dots, N_G,$$

$$W_i = E_i - P_i, \quad i = 1, \dots, N_{AC}, \quad (206)$$

$$J_{\text{TPWD}} = \sum_{g=1}^{N_G+1} \sum_{j=1}^{H_g} \sum_{i=1}^{N_{AC}+1} M_P(Q_g(j), i) M_{\text{PWD}}(g, v_i), \quad (207)$$

$$J_{\text{TBTD}} = \sum_{g=1}^{N_G+1} \sum_{j=1}^{X_g} \sum_{i=1}^{N_{AC}+1} M_P(Q_g(j), i) M_{\text{PTD}}(g, v_i), \quad (208)$$

$$J_{\text{TPWT}} = \sum_{g=1}^{N_{AC}} W_i \sum_{i=1}^{N_{AC}+1} (M_P(i, j) + M_P(j, i)), \quad (209)$$

$$J_{\text{MOGAP}} = \alpha J_{\text{TPWD}} + \beta J_{\text{TBTD}} + (1 - \alpha - \beta) \varphi J_{\text{TPWT}}, \quad (210)$$

$$\alpha + \beta \leq 1, \quad 0 \leq \alpha \leq 1, \quad 0 \leq \beta \leq 1. \quad (211)$$

Wei and Liu [16] considered the AGAP as a fuzzy model and adopted a hybrid genetic algorithm to solve the developed model. The main objectives were minimizing passengers' total walking distance and gates idle times variance. They developed the following model:

$$\text{minimize } Z_1 = \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^{M+1} \sum_{l=1}^{M+1} f_{ij} T_{kl} y_{ik} y_{il}, \quad (212)$$

$$\text{minimize } Z_2 = \sum_{i=1}^N S_i^2 + \sum_{k=1}^M S_k^2 \quad (213)$$

subject to

$$\sum_{k=1}^M y_{ik} = 1, \quad \forall i \in N, \quad (214)$$

$$(d_j - a_i + \alpha) (d_i - a_j + \alpha) y_{ik} y_{jk} \leq 0, \quad (215)$$

$$\forall i, j = 1, \dots, N, \quad \forall k = 1, \dots, M,$$

$$y_{ik} \in \{0, 1\}, \quad \forall i \in N, \quad \forall k \in K, \quad (216)$$

where objective function (212) reflects the total walking distance of passengers. y_{ik} is 0-1 variable; $y_{ik} = 1$ if flight i is assigned to gate k ; otherwise it is 0; f_{ij} describes the number of passengers transferring from flight i to j , and T_{kl} is walking distance for passenger from gate k to l . Objective function (213) is used as a surrogate for the variance of idle times. The actual number of assignments is N and the number of nondummy idle times is $N + M$. Constraint

(214) indicates that every flight must be assigned to one gate. Constraint (215) shows that flights that have overlap schedule cannot be assigned to the same gate, where α is the least safe time between continuous aircrafts assigned to the same gate. Constraint (216) denotes that y_{ik} is a binary variable.

In 2001, Yan and Huo [2] formulated the AGAP as a model with two objectives: minimizing (3) the walking distance, and (4) the waiting time for the passengers. The proposed mathematical model is binary integer linear programming:

$$\text{Minimize } Z_1 = \sum_{i=1}^M \sum_{j=1}^N \sum_{k=B_i}^{L_i} (c_{ij} - x_{ijk}), \quad (217a)$$

$$\text{Minimize } Z_2 = \sum_{i=1}^M \left(P_i \left(\sum_{j=1}^N \sum_{k=B_i}^{L_i} kx_{ijk} - B_i \right) \right) \quad (217b)$$

subject to

$$\sum_{j=1}^N \sum_{k=B_i}^{L_i} x_{ijk} = 1 \quad \forall i \quad (217c)$$

$$\sum_{i \in F_j} \sum_{k \in H_{is}} x_{ijk} \leq 1 \quad \forall j, \forall S \quad (217d)$$

$$x_{ijk} = 0 \text{ or } 1, \quad \forall i, \forall j, \forall k, \quad (217e)$$

where objective (217a) represents the minimum total passenger walking distance. Objective (217b) represents the minimum total passenger waiting time. Constraint (217c) denotes that every flight must be assigned to one and only one gate. Constraint (217d) ensures that at most one aircraft is assigned to every gate in every time window.

Column generation approach, simplex method, and B&B algorithm were used to solve the proposed problem, which was a case study in Chiang Kai-Shek Airport, Taiwan. The problem consisted of 24 gates (of which two were temporary; eight out of 24 gates were only available for the wide type of aircrafts, whereas the rest were available for the other types) and 145 flights. The results showed that the obtained solution (7,300,660 s the best feasible solution found so far) was away from the optimal one by 0.077% (5595s).

Wipro Technologies [17] proposed a binary multiple objective integer quadratic programming model for the AGAP with a quadratic objective function. The proposed model was reformulated into a mixed binary integer linear programming model (linear objective functions and constraints). The proposed model has been solved using greedy heuristic, SA, and TS (MIP solvers based B&B cannot solve the proposed model within a reasonable time). The developed model was represented as follows.

Generic Model. Consider

$$\text{Minimize } \sum_{i \in N} y_{i(m+1)}, \quad (218)$$

$$\text{Minimize } \sum_{i,j \in N} \sum_{k,l \in K} f_{ij} c_{kl} y_{ik} y_{jl} \quad (219)$$

subject to

$$\sum_{k \in K} y_{ik} = 1, \quad \forall i \in N, \quad (220)$$

$$y_{ik} = 0, \quad \forall i \in N, \forall k \in K_i, \quad (221)$$

$$y_{ik} > y_{jl}, \quad \forall i \in N, \forall k \in K, \forall j \in N_i, \forall l \in K_k, \quad (222)$$

$$(y_{ik} + y_{jl}) \leq 1, \quad \forall i \in N, \forall k \in K, \forall j \in N_i, \forall l \in K_k, \quad (223)$$

$$y_{ik} = \sum_{j \in N} z_{ijk}, \quad \forall i \in N, \forall k \in K, \quad (224)$$

$$y_{ik} = \sum_{i \in N} z_{ijk}, \quad \forall j \in N, \forall k \in K, \quad (225)$$

$$t_i \geq a_i + \alpha, \quad \forall i \in N, \quad (226)$$

$$t_i \leq d_i - \theta_i * \sum_{j \in N} f_{ji}, \quad \forall i \in N, \quad (227)$$

$$\left(t_i + \theta_i * \sum_{j \in N} f_{ji} \right) \leq (t_j + (1 - z_{ijk}) * M), \quad \forall i, j \in N, \quad (228)$$

$$(a_j + (1 - z_{ijk}) * M) \geq (d_i + \beta), \quad \forall i, j \in N, \forall k \in K, \quad (229)$$

$$y_{ik}, z_{ijk} \in \{0, 1\}, \quad \forall i, j \in N, \forall k, l \in K, \quad (230)$$

$$t_i \geq 0, \quad \forall i \in N, \quad (231)$$

where objective function (218) aims at minimizing the number of flights that must be assigned to the apron, that is, those left ungated. Objective function (219) seeks to minimize the total connection times by passengers. Constraint (220) specifies that every flight must be assigned to one gate. Constraint (221) shows the equipment restriction on certain gates. Constraints (222) and (223) restrict the assignment of specific adjacent flights to adjacent gates. Constraints (224) and (225) indicate that every flight can have at most one flight immediately following and at most one flight immediately preceding, at the same gate. Constraints (226) and (227) stipulate that a gate must open for boarding on a flight during the time between its arrival and departure, and it also must allow sufficient time for handling the passenger/luggage boarding, which is assumed to be proportional to the number of passengers going on board. Constraint (228) ensures that each gate only serves one flight at any particular time (i.e., if flight i is assigned immediately before flight j to the same gate k , the gate must open for flight i earlier than for flight j). Constraint (229) further states the aircraft can only arrive at the gate when the previous flight has departed, while also including the buffer time between the

flights. Constraints (230) and (231) specify the binary and nonnegative requirements for the decision variables.

The Reformulated Model. Minimizing (218) will remain the same:

$$\text{Minimize } \sum_{i,j \in N} \sum_{k,l \in K} f_{ij} c_{cl} x_{ijkl} \quad (232)$$

subject to: Constraints... (220)–(231) from the generic model,

$$x_{ijkl} \leq y_{ik}, \quad \forall i, j \in N, \forall k, l \in K, \quad (233)$$

$$x_{ijkl} \leq y_{jl}, \quad \forall i, j \in N, \forall k, l \in K, \quad (234)$$

$$y_{ik} + y_{jl} - 1 \leq x_{ijkl}, \quad \forall i, j \in N, \forall k, l \in K, \quad (235)$$

$$x_{ijkl} \in \{0, 1\}, \quad \forall i, j \in N, \forall k, l \in K, \quad (236)$$

where constraints (233), (234), and (235) specify that x_{ijkl} can be equal to one if and only if flight i is assigned to gate k and flight j is assigned to gate l . Constraint (236) expresses the binary requirement for the decision variable x_{ijkl} .

Kaliszewski and Miroforidis [37] considered agap with the objective of assigning incoming flights to airport gates with some assumptions; those assumptions were as follows: gate assignment has no significant impact on passenger walking distance and no restrictions on the gates (all gates can take any type of airplanes) and neighboring gate operations can be carried out without any constraints. The model was stated as follows:

minimize $f_1(x)$

$$= \delta \max_j \left(\sum_{i=1}^m x_{j,i}^{a_j+1} + 2 \sum_{i=1}^m x_{j,i}^{a_j+2} + \dots + \alpha \sum_{i=1}^m x_{j,i}^\alpha \right) \quad (237)$$

$$\text{minimize } f_2(y) = \sum_{j=1}^n y_j \quad (238)$$

subject to

$$\sum_{i=1}^m \sum_{t=a_j}^\Delta x_{j,i}^t \leq 1, \quad \text{for } j = 1, \dots, n, \quad (239)$$

$$x_{j,i}^t \leq y_i^u \quad \text{for } u = t, t+1, \dots, t+g_j, \quad (240)$$

$$\sum_{j=1}^n x_{j,i}^t \leq 1, \quad t = 1, \dots, \Delta, \quad i = 1, \dots, m, \quad (241)$$

$$\sum_{i=1}^m \sum_{t=a_j}^\Delta x_{j,i}^t \leq 1 - y_j, \quad \text{for } j = 1, \dots, n. \quad (242)$$

2.1.7. Stochastic Models. Yan and Tang [10] designed a framework for a stochastic AGAP (flight delays are stochastic). The framework included three main parts: the gate assignment model, a rule for the reassignments, and two adjustment methods for penalties. The performance of the developed framework has been evaluated using simulation-based evaluation method.

The formulation of the stochastic gate assignment model (the objective was to minimize the total waiting time of the passengers) was addressed as follows:

$$\begin{aligned} \text{Minimize } ZP = & \sum_{k \in K} \sum_{ij \in A^k} c_{ij}^k x_{ij}^k \\ & + \sum_{s \in \Omega} p^s \sum_{k \in K} \sum_{ij \in QA^k} u_{ij}^{s,k} x_{ij}^k + w \sum_{s \in \Omega} p^s h^s \end{aligned} \quad (243)$$

subject to

$$\sum_{j \in N^k} x_{ij}^k - \sum_{r \in N^k} x_{ri}^k = 0 \quad \forall i \in N^k, \forall k \in K, \quad (244)$$

$$\sum_{k \in K} \sum_{ij \in F_t} x_{ij}^k = 1 \quad \forall t \in AF, \quad (245)$$

$$0 \leq x_{ij}^k \leq g^k \quad \forall (i, j) \in CA^k, \forall k \in K, \quad (246)$$

$$\sum_{k \in K} \sum_{ij \in QA^k} u_{ij}^{s,k} x_{ij}^k - h^s \leq \sum_{s \in \Omega} p^s \sum_{k \in K} \sum_{ij \in QA^k} u_{ij}^{s,k} x_{ij}^k \quad \forall s \in \Omega, \quad (247)$$

$$h^s \geq 0 \quad \forall s \in \Omega, \quad (248)$$

$$x_{ij}^k \in Z_+ \quad \forall (i, j) \in CA^k, \forall k \in K, \quad (249)$$

$$x_{ij}^k = 0, 1 \quad \forall (i, j) \in A^k - CA^k, \forall k \in K, \quad (250)$$

where function (243) denotes the minimization of the total passenger waiting time, the expected penalty value for all n scenarios, and the expected semideviation risk measure (SRM) for all n scenarios multiplied by the weighting vector w . Constraint (244) is the flow conservation constraint at every node in each network. Constraint (245) denotes that every flight is assigned to only one gate and one time window. Constraint (246) ensures that the number of gates used in each network does not exceed its available number of gates. Constraints (247) and (248) are used to calculate the SRM. Constraint (249) ensures that the cycle arc flows are integers. Constraint (250) indicates that, except for the cycle arcs, all other arc flows are either zero or one.

The value of the performance measure (the objective, minimizing the total waiting time of the passengers) for each scenario in the real time stage was calculated as follows:

$$zr^s = wt^s + it^s, \quad (251)$$

$$ZR = AWT + AIT + w \times SRMR$$

$$\begin{aligned} &= \sum_{s \in \Omega} p^s wt^s + \sum_{s \in \Omega} p^s it^s \\ &+ w \sum_{s \in \Omega} p^s \left(\max \left(0, zr^s - \sum_{s \in \Omega} p^s zr^s \right) \right) \\ &= \sum_{s \in \Omega} p^s zr^s + w \sum_{s \in \Omega} p^s \left(\max \left(0, zr^s - \sum_{s \in \Omega} p^s zr^s \right) \right). \end{aligned} \quad (252)$$

For each iteration, the penalties were calculated, using the developed two adjustment methods for penalties, as follows.

Method 1. Consider

$$(u_{ij}^{s,k})^{m+1} = (u_{ij}^{s,k})^m + (dw_{ij}^{s,k} + dt_{ij}^{s,k})^m. \quad (253)$$

Method 2. Consider

$$(dv_{ij}^{s,k})^m = \begin{cases} 1, & \text{if } dw_{ij}^{s,k} > 0 \text{ or } dt_{ij}^{s,k} > 0 \\ 0, & \text{if } dw_{ij}^{s,k} = 0, dt_{ij}^{s,k} = 0 \end{cases} \quad (254)$$

$$\forall (i, j) \in QA^k, \quad \forall k \in K, \quad \forall s \in \Omega,$$

$$b^m = \max \left\{ 0, \frac{-\sum (dv_{ij}^{s,k})^m (d_{ij}^{s,k})^{m-1}}{\|(d_{ij}^{s,k})^{m-1}\|^2} \right\}, \quad (255)$$

$$(d_{ij}^{s,k})^m = (dv_{ij}^{s,k})^m + b^m (d_{ij}^{s,k})^{m-1} \quad \forall (i, j) \in QA^k, \quad (256)$$

$$\forall k \in K, \forall s \in \Omega$$

$$t^m = \frac{\lambda |ZR^m - ZP^m|}{\|dv_{ij}^{s,k}\|^2}, \quad 0 < \lambda \leq 2, \quad (257)$$

$$(u_{ij}^{s,k})^{m1} = (u_{ij}^{s,k})^m + t^m (d_{ij}^{s,k})^m \quad \forall (i, j) \in QA^k, \quad \forall k \in K, \quad \forall s \in \Omega. \quad (258)$$

The data was taken from the Chiang Kai-Shek (CKS) airport (172 flights, 2 gate types, and 14 aircraft types); the distributions for the flight delays were obtained from the actual data taken from the CKS airport. The obtained results were 197 minutes which was the longest solution time of the framework, which was efficient in the planning stage, but after 40 scenarios, the solution times increased significantly but the solution results were more stable.

Genç et al. [38] developed a stochastic model for AGAP with the objective of minimizing the gate duration, gate duration defined as the total time of the allocated gates (for all flights in a day):

$$F_{\text{fitness}} = \sum_{k=1}^{Ng} \sum_{l=1}^{N_l} \text{any}(M_c(k, l)) \quad (259)$$

subject to

$$\text{any}(M_c(k, l)) = \begin{cases} 1, & \text{if } M_c(k, l) \neq 0 \\ 0, & \text{otherwise.} \end{cases} \quad (260)$$

Şeker and Noyan [9] also developed a stochastic model considering the minimization of the number of conflicts and the expected variance of the idle times as a performance measure; the proposed performance measure was a part of the mixed integer programming model presented in Section 2.1.3:

$$\text{Minimize } \sum_{s \in S} V_s p_s + \Lambda \sum_{i \in N^b} \sum_{k \in M} \sum_{s \in S} (c_{i,k,s}^p + c_{i,k,s}^f) p_s \quad (261)$$

subject to

$$\sum_{k \in M} x_{i,k} = 1, \quad i \in N, \quad (262)$$

$$x_{0,k} = 1, \quad k \in M, \quad (263)$$

$$x_{n+k,k} = 1, \quad k \in M, \quad (264)$$

$$\sum_{j \in L_{i,s}} x_{j,k} + x_{i,k} \leq 1, \quad i \in N^b, \quad k \in M, \quad s = 0, \quad (265)$$

$$c_{i,k,s}^p \geq \sum_{j \in L_{i,s}^p} x_{j,k} + (x_{i,k} - 1)(n + m), \quad (266)$$

$$i \in N^b, \quad k \in M, \quad s = S,$$

$$c_{i,k,s}^f \geq \sum_{j \in L_{i,s}^f} x_{j,k} + (x_{i,k} - 1)(n + m), \quad (267)$$

$$i \in N^b, \quad k \in M, \quad s = S,$$

$$A_{i,s} \leq (2 - x_{i,k} - x_{j,k})Z + a_{j,s}, \quad (268)$$

$$i \in N^b \setminus \{0\}, \quad j \in L_{i,s}^f, \quad k \in M, \quad s = S,$$

$$A_{i,s} \leq (2 - x_{i,k} - x_{j,k})Z + a_{j,s}, \quad (269)$$

$$i \in N^b \setminus \{0\}, \quad j \in L_{i,s}^p, \quad k \in M, \quad s = S,$$

$$A_{i,s} \leq (2 - x_{i,k} - x_{j,k})Z + a_{j,s}, \quad (270)$$

$$i \in N^b \setminus \{0\}, \quad j \in L_{i,s}^f, \quad k \in M, \quad s = S,$$

$$\sum_{i \in N^b \setminus \{0\}} A_{i,s} = \sum_{i \in N} a_{i,s} + a_{0,s}m, \quad s = S, \quad (271)$$

$$I_{i,s} = A_{i,s} - d_{i,s} + s_{i,s}^p + s_{i,s}^f, \quad i \in N^b \setminus \{0\}, \quad s = S, \quad (272)$$

$$I_{i,s} \leq (2 - x_{i,k} - x_{j,k})Z, \quad (273)$$

$$i \in N^b \setminus \{0\}, \quad j \in L_{i,s}^p, \quad k \in M, \quad s = S,$$

$$I_{i,s} \leq (2 - x_{i,k} - x_{j,k})Z, \quad i \in N^b \setminus \{0\}, \quad (274)$$

$$j \in L_{i,s}^f, \quad k \in M, \quad s = S,$$

$$s_{i,s}^p \leq Z \sum_{k \in M} c_{i,k,s}^p, \quad i \in N^b, \quad s = S, \quad (275)$$

$$s_{i,s}^f \leq Z \sum_{k \in M} c_{i,k,s}^f, \quad i \in N^b, \quad s = S, \quad (276)$$

$$x_{i,k} \in \{0, 1\}, \quad i \in N^b, \quad k \in M, \quad (277)$$

$$\text{All remaining variables} \geq 0. \quad (278)$$

2.2. AGAP Related Problems. In some of the publications on the GAP the researchers have formulated the AGAP as

well-known related problems such as quadratic assignment problem (QAP), clique partitioning problem (CPP), and scheduling problem or even as a network representation. However, some of the researchers formulated the AGAP as a robust optimization model. In this section, we will present the work that has been done on the GAP as a well-known related problem.

2.2.1. *Quadratic Assignment Problem (QAP)*. Drexl and Nikulin [3] modeled the multicriteria airport gate assignment as quadratic assignment problem (QAP) and solved the problem using Pareto simulated annealing. The performance measures were as follows: minimizing connection times or total passenger walking distances, maximizing the preferences of total gate assignment, and minimizing the number of ungated flights:

$$\min z_1 = \sum_{i=1}^n \pi_{i,m+1}, \tag{279a}$$

$$\begin{aligned} \min z_2 = & \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^{m+1} \sum_{l=1}^{m+1} f_{i,j} \omega_{k,l} \pi_{i,k} \pi_{j,l} + \sum_{i=1}^n \sum_{k=1}^{m+1} f_{0,i} \omega_{0,k} \pi_{i,k} \\ & + \sum_{i=1}^n \sum_{k=1}^{m+1} f_{i,0} \omega_{k,0} \pi_{i,k}, \end{aligned} \tag{279b}$$

$$\min z_3 = \sum_{i=1}^n \sum_{k=1}^{m+1} v_i u_{i,k} \pi_{i,k} \tag{279c}$$

subject to

$$\sum_{k=1}^{m+1} \pi_{i,k} = 1, \quad 1 \leq i \leq n, \tag{280}$$

$$\pi_{i,k} \pi_{j,k} (d_j - a_i) (d_i - a_j) \leq 0, \quad 1 \leq i, j \leq n, k \neq m + 1, \tag{281}$$

$$\pi_{i,k} \in \{0, 1\}, \quad 1 \leq i \leq n, 1 \leq k \leq m + 1, \tag{282}$$

where objective (279a) addresses the number of flights that are not assigned to any terminal gate (i.e., to the apron). Objective (279b) represents the total passenger walking distance. It consists of three terms: the walking distance of transfer passengers, originating departure passengers, and disembarking arrival passengers. Objective (279c) represents the total value for flight gate assignment preference. Constraint (280) ensures that every flight must be assigned to exactly one gate including the apron. Constraint (281) prohibits schedule overlapping of two flights if they are assigned to the same terminal gate. Constraint (282) defines the variables to be Boolean.

Haghani and Chen [13] modeled the AGAP as QAP with minimizing the total passenger walking distances (transfer

passengers and local passenger) as a performance measure. The QAP model was expressed as follows:

$$\begin{aligned} \text{Minimize } & \sum_i \sum_{i'} \sum_j \sum_{j'} P_{ii'} D_{jj'} X_{ijt} X_{i'j't'} \\ & + \sum_i \sum_{i'} (P_{i0} D_{j0} + P_{0i} D_{0j}) X_{ijt}, \end{aligned} \tag{283}$$

$$P_{ii'} D_{jj'} \leftarrow P_{i0} D_{j0} + P_{0i} D_{0j} \quad \forall i, j, \tag{284}$$

$$\text{Minimize } \sum_i \sum_{i'} \sum_j \sum_{j'} P_{ii'} D_{jj'} X_{ijt} X_{i'j't'}. \tag{285}$$

According to the simplicity of solving linear models, the previous model was transformed into a linear model as follows:

$$Y_{ijj't'} = X_{ijt} X_{i'j't'}, \tag{286}$$

$$Y_{ijj't'} = \begin{cases} 1, & \text{if flight } i \text{ is assigned to gate } j \\ & \text{and flight } i' \text{ is assigned to gate } j', \\ 0, & \text{otherwise,} \end{cases} \tag{287}$$

$$\text{Minimize } \sum_i \sum_{i'} \sum_j \sum_{j'} P_{ii'} D_{jj'} Y_{ijj't'} \tag{288}$$

subject to

$$\sum_j X_{ijt} = 1 \quad \forall i, t_i^a \leq t \leq t_i^d, \tag{289}$$

$$\sum_i X_{ijt} \leq 1 \quad \forall j, t, \tag{290}$$

$$X_{ijt} \leq X_{ij(t+1)} \quad \forall t, t_i^a \leq t \leq t_i^d - 1, \tag{291}$$

$$X_{ijt_i}^a + X_{ijt_j}^a - 2Y_{ijj't'}^a \geq 0 \quad \forall i < i', j, j', \tag{292}$$

$$\sum_j \sum_{j'} Y_{ijj't'} = 1 \quad \forall i \neq i'. \tag{293}$$

The results and conclusions were as follows: the proposed approach was efficient which provided results close to the optimal solution according to the percent of improvement from the starting solution (initial solution), in the case of 10 flights and 10 gates, 20 flights and 5 gates, and 30 flights and 7 gates.

2.2.2. *Scheduling Problems*. In 2010, Li [39] formulated the GAP as a parallel machines scheduling problem and used the dynamic scheduling and the direct graph model to solve the proposed model; B&B was used in solving the small size problems while the large size problems have been solved using dynamic scheduling.

2.2.3. *Clique Partitioning Problem (CPP)*. Dorndorf et al. [8] developed an optimization model for the GAP and transformed that model into a CPP model; the two models are written below. A heuristic approach that was developed

by Dorndorf and Pesch (1994), based on the ejection chain algorithm, has been used to solve the transformed model (CPP model).

The Optimization Model for the AGAP. Consider

$$\text{Minimize}_{\{f|f:N \rightarrow M\}} \alpha_1 z_1(f) + \alpha_2 z_2(f) + \alpha_3 z_3(f) \quad (294)$$

subject to

$$\begin{aligned} f(i) &\in M(i) \quad \forall i \in N, \\ f(i) &\neq f(j) \quad \forall t_{ij} < 0, \end{aligned} \quad (295)$$

$$\begin{aligned} f(i) &\neq n + m, \quad \forall i, j \in N, \\ f(i) &\neq k \vee f(i) \neq l \quad \forall (i, k, j, l) \in S, \end{aligned}$$

$$\begin{aligned} z_1 &= -\sum_{i=1}^n p_{if(i)}^*, \\ z_2 &= |\{i \in N \mid U(i) \neq 0 \wedge f(i) \neq f(U(i))\}|, \end{aligned} \quad (296)$$

$$z_3 = \sum_{\{(i,j)|i < j, f(i) = f(j) \neq n+m\}} \max\{t_{\max} - t_{ij}, 0\}.$$

The CPP Transformation of the Problem. Consider

$$\text{maximize} \sum_{1 \leq i < j \leq a} w_{ij} x_{ij} \quad (297)$$

subject to

$$\begin{aligned} x_{ij} + x_{jk} - x_{ik} &\leq 1 \quad \text{for } 1 \leq i < j < k \leq a, \\ x_{ij} - x_{jk} + x_{ik} &\leq 1 \quad \text{for } 1 \leq i < j < k \leq a, \end{aligned} \quad (298)$$

$$\begin{aligned} -x_{ij} + x_{jk} + x_{ik} &\leq 1 \quad \text{for } 1 \leq i < j < k \leq a, \\ x_{ij} &\in \{0, 1\} \quad \text{for } 1 \leq i < j \leq a, \\ V &:= \{1, 2, \dots, n + m - 1\}, \end{aligned} \quad (299)$$

$$w_{ij} := \begin{cases} -\infty & \text{if } t_{ij} < 0 \\ \alpha_2 & \text{if } t_{ij} \geq 0 \\ \wedge (U(i) = j \vee U(j) = i) \\ -\alpha_3 \cdot \max\{t_{\max} - t_{ij}, 0\} & \text{if } t_{ij} \geq 0 \wedge U(i) \neq j \\ \wedge U(j) \neq i \forall i, j < n, \end{cases} \quad (300)$$

$$w_{ij} := \begin{cases} -\infty & \text{if } j \notin M(i) \\ \alpha_1 \cdot p_{ij}^* & \text{if } j \in M(i) \end{cases} \quad (301)$$

$$\begin{aligned} \forall i \leq n, \quad j > n \\ w_{ij} &:= -\infty \quad \forall i, j > n. \end{aligned} \quad (302)$$

2.2.4. Network Representation. Maharjan and Matis [40] formulated the GAP as a binary integer multicommodity network flow model with minimizing the passengers comfort and aircraft fuel burn as a performance measure. For passengers comfort and with arguments of distance and time for connection a penalty function in three dimensions was specified. For large size problem and based on a methodology of zoning a decomposition approach was provided and compared with the assignments made by the airline, and the results showed that the developed methodology was shown to be computationally efficient.

The Mathematical Formulation. Consider

- X_{Si}^k : binary variable representing initial assignment of gate $k \in \mathbf{K}$ to aircraft $i \in \mathbf{F}$;
- X_{ji}^k : binary variable representing assignment of gate $k \in \mathbf{K}$ to aircraft $j \in \mathbf{D}$ followed by $i \in \mathbf{A}$;
- X_{jT}^k : binary variable representing last assignment of gate $k \in \mathbf{K}$ to aircraft $j \in \mathbf{D}$;
- X_{ST}^k : binary variable representing no assignment of gate $k \in \mathbf{K}$ to any aircraft;

$$\begin{aligned} \text{Minimize } Z &= \sum_{i \in \mathbf{A}} \sum_{k \in \mathbf{K}} \left(C_k + \frac{2f_i f_c}{f_s \{d_{k1} + d_{k2}\}} \right) \left(X_{S,i}^k + \sum_{j \in \mathbf{D}} X_{j,i}^k \right) \\ &+ \sum_{(i \in \mathbf{F}, i=j, i'=j', i \neq i', j \neq j')} \sum_{(i' \in \mathbf{F}, i=j, i'=j', i \neq i', j \neq j')} \sum_{k \in \mathbf{K}} \sum_{k' \in \mathbf{K}} N_{i'i'} C_{kk'}^{i'i'} X_{i,j}^k X_{i',j'}^{k'} \end{aligned} \quad (303)$$

subject to

$$\sum_i X_{s,i}^k + X_{S,T}^k = 1 \quad \forall k \in K, i \in F, \tag{304}$$

$$X_{s,i}^k + \sum_i \ln_{ij} X_{j,i}^k = X_{m,n}^k \quad \forall i, m \in A, j, n \in D, \tag{305}$$

$$i \neq j, m = n,$$

$$\sum_i \ln_{ij} X_{j,i}^k + X_{j,T}^k = X_{m,n}^k \quad \forall i, m \in A, j, n \in D, \tag{306}$$

$$i \neq j, m = n,$$

$$\sum_i X_{jT}^k + \sum_i X_{ST}^k = 1 \quad i \in F, k \in K, \tag{307}$$

$$\sum_i X_{ij}^k = 1 \quad \forall i \in A, j \in D, i = j, \tag{308}$$

$$X_{s,i}^k, X_{j,i}^k, X_{m,n}^k, X_{S,T}^k, X_{jT}^k = \{0, 1\}, \tag{309}$$

$$\ln_{ij} = \begin{cases} 1, & \text{if } (A_i - D_j) \geq p \quad \forall i \neq j \\ 0, & \text{otherwise,} \end{cases} \tag{310}$$

where (303) represents the expected taxi in and out fuel burn cost of assigning a plane to a particular gate based on the expected runway distance corresponding to arrival and departure cities for the flight. Equation (304) is referred to as a flow-in constraint because it deals with the gate flow from the source node to the arrival flight node. Equation (305) is referred to as conservation of flow at the arrival node. Equation (306) is conservation of flow at the departure node. Equation (307) is the flow-out constraint that forces all the flow to leave the departure node to the terminal node. Equation (308) is referred to as a unit flow serving arc constraint as it allows only one unit gate $k \in K$ to flow through serving arc. Equation (309) is the binary constraints.

The above model was with a quadratic objective function and a linearization for that objective was made as follows:

$$\text{Minimize } Z_2 = \sum_{(i \in F, i=j, i'=j', i \neq i', j \neq j')} \sum_{(i' \in F, i=j, i'=j', i \neq i', j \neq j')} \sum_{k \in K} \sum_{k' \in K} N_{ii'} C_{kk'}^{ii'} X_{i,j}^k X_{i',j'}^{k'}. \tag{311}$$

The linearization has been made by replacing the quadratic term $(X_{i,j}^k X_{i',j'}^{k'})$ by a new variable $Y_{ii'}^{kk'}$ defined as follows:

$$Y_{ii'}^{kk'} = \begin{cases} 1, & \text{if } (X_{ij}^k = 1, X_{i'j'}^{k'} = 1 \quad \forall i = j, \\ & i' = j', i \neq i', j \neq j', k \neq k') \\ 0, & \text{otherwise} \end{cases} \tag{312}$$

subject to

$$Y_{ii'}^{kk'} - X_{ij}^k \leq 0 \quad \forall i = j, i' = j', i \neq j', k \neq k', \tag{313}$$

$$Y_{ii'}^{kk'} - X_{i'j'}^{k'} \leq 0 \quad \forall i = j, i' = j', i \neq j', k \neq k', \tag{314}$$

$$X_{ij}^k + X_{i'j'}^{k'} - Y_{ii'}^{kk'} \leq 1 \quad \forall i = j, i' = j', i \neq j', k \neq k', \tag{315}$$

$$X_{ij}^k + X_{i'j'}^{k'} - Y_{ii'}^{kk'} \geq 0 \quad \forall i = j, i' = j', i \neq j', k \neq k', \tag{316}$$

where inequalities (313) and (314) indicate that variable $Y_{ii'}^{kk'}$ is equal to 1 if and only if binary variables X_{ij}^k and $X_{i'j'}^{k'}$ are equal to 1. Equation (315) specifies that $Y_{ii'}^{kk'}$ cannot be greater than 1, and (316) further specifies that $Y_{ii'}^{kk'}$ cannot be less than zero. Due to the binary nature of X_{ij}^k and $X_{i'j'}^{k'}$ with the above constraints, $Y_{ii'}^{kk'}$ is forced to be a binary variable.

According to the passengers comfort, a cost function $(C_{ii'}^{kk'})$ was represented as follows:

$$C_{ii'}^{kk'} = \begin{cases} s \sqrt{d_{kk'}} (2 - \Delta t_{ii'})^2, & \forall 0 < \Delta t_{ii'} \leq t_{\max} \\ 0, & \text{otherwise,} \end{cases} \tag{317}$$

where (317) is a surface plot of the cost function.

Mathematical Formulation Using the Methodology of Zoning. Consider

$$\begin{aligned} \text{Minimize } Z = & \sum_{i \in F} \sum_{k \in K_z} \left(\frac{2f_i f_c}{f_s \{d_{k1} + d_{k2}\}} \right) \left(X_{s,i}^k + \sum_{j \in D} X_{j,i}^k \right) \\ & + \sum_{(i \in F, i=j, i'=j', i \neq i', j \neq j')} \sum_{(i' \in F, i=j, i'=j', i \neq i', j \neq j')} \sum_{k \in K_z} \sum_{k' \in K} N_{ii'} C_{kk'}^{ii'} X_{ij}^k X_{i'j'}^{k'} \end{aligned} \tag{318}$$

subject to the following.

The constraints of (304) and (308) are modified and replaced with

$$\sum_j X_{si}^k + X_{ST}^k = |K_z| \quad \forall k \in K_z, i \in F, \tag{319}$$

$$\sum_j X_{jT}^k + \sum_k X_{ST}^k = |K_z| \quad \forall i \in F, k \in K_z. \tag{320}$$

In order to solve the developed model a code for the developed model was written using an AMPL/CPLEX 11.2 package, and as mentioned before the results showed that the developed methodology was shown to be computationally efficient.

2.3. Robust Optimization. Diepen et al. [41] formulated a completely new integer linear programming formulation for the GAP with a robust objective function that is based on the so-called gate plans. The objective was to maximize the robustness of a solution, which can be expressed as an allocation of a maximum possible idle time between each pair of consecutive flights to guarantee that each flight can afford to land with some slight earliness or tardiness without the need for re-planning the schedule:

$$x_j = \begin{cases} 1, & \text{if gate plan } j \text{ is selected} \\ 0, & \text{otherwise,} \end{cases} \tag{321}$$

$$\text{Minimize } \sum_{i=1}^N c_i x_i$$

subject to

$$\sum_{i=1}^N g_{vi} x_i = 1 \quad \text{for } v = 1, \dots, V, \tag{322}$$

$$\sum_{i=1}^N e_{ia} x_i = S_a \quad \text{for } a = 1, \dots, A, \tag{323}$$

$$x_j \in \{0, 1\} \quad \text{for } j = 1, \dots, n, \tag{324}$$

where

$$g_{vi} = \begin{cases} 1, & \text{if flight } v \text{ is in gate plan } i, \\ 0, & \text{otherwise,} \end{cases} \tag{325}$$

$$e_{ia} = \begin{cases} 1, & \text{if flight } i \text{ is of type } a, \\ 0, & \text{otherwise.} \end{cases}$$

In addition, to add the preferences to the ILP model the following constraints have been added to the model:

$$\sum_{i=1}^N \sum_{v=1}^V \sum_{a=1}^A p_{vak} e_{ia} g_{vi} x_i \geq P_k \quad \text{for } k = 1, \dots, K, \tag{326}$$

where

- (i) $p_{vak} = \begin{cases} 1, & \text{if flight } v \text{ has preference for gate type } a \text{ in preference } k \\ 0, & \text{otherwise;} \end{cases}$
- (ii) P_k denotes the minimum number of flights that have to be assigned to a given gate type;
- (iii) according to preference k ;
- (iv) K denotes the total number of preferences:

$$\sum_{i=1}^N g_{vi} x_i + UAF_v = 1 \quad \text{for } v = 1, \dots, V, \tag{327}$$

where $UAF_v \geq 0$ for $v = 1, \dots, v$. and UAF_v is a penalty variable:

$$\sum_{i=1}^N (g_{vi} + g_{vA,i}) x_i + UAF_{vA} = 1,$$

$$\sum_{i=1}^N (g_{vi} + g_{vA,i}) x_i + UAF_{vA} = 1,$$

P_{ihr}

$$= \begin{cases} 1, & \text{if flight } v \text{ has preference on gate type } a \\ & \text{in preference } k, \\ 0.5, & \text{if the split version of flight } v \text{ has preference} \\ & \text{on gate type } a \text{ in preference } k, \\ 0, & \text{otherwise.} \end{cases} \tag{328}$$

The integrality of the developed model has been relaxed for the integrality and the resulting relaxed LP was exploited to obtain solutions of ILP by using column generation (CG). Table 1 summarizes all the above mathematical formulations used recently for the AGAP.

3. Resolution Methods

As mentioned before in Section 2, most of the solution techniques presented in Sections 3.2 and 3.3 have been used concurrently with complex mathematical formulations that

TABLE 1: Formulations of AGAP and related problems.

Formulation	References	Criterion (comments)	Problem type
Integer linear programming (IP)	Lim et al. [24]	(i) Minimizing the sum of the delay penalties (ii) Minimizing the total walking distance	Theoretical
	Diepen et al. [25]	(i) Minimizing the deviation of arrival and departure time (ii) Minimizing replanning the schedule	Real case (Amsterdam Airport Schiphol)
	Diepen et al. [26]	Minimizing the deviations from the expected arrival and departure times	Real case (Amsterdam Airport Schiphol)
Binary integer programming	Mangoubi and Mathaisel [11]; Yan et al. [29]	Minimizing passenger walking distances	Real case (Toronto International Airport); Real case (Chiang Kai-Shek Airport)
	Vanderstraeten and Bergeron [28]	Minimizing the number off-gate event	Theoretical
	Bihl [12]	Minimizing of the total passenger distance	Theoretical
	Tang et al. [27]	Developing a gate reassignment framework and a systematic computerized tool	Real case (Taiwan International Airport)
	Prem Kumar and Bierlaire [18]	(i) Maximizing the gate rest time between two turns (ii) Minimizing the cost of towing an aircraft with a long turn (iii) Minimizing overall costs that include penalization for not assigning preferred gates to certain turns	Theoretical
Mixed integer linear programming (MILP)	Bolat [30]	Minimizing the range of slack times	Real case (King Khaled International Airport)
	Bolat [31]	Minimizing the variance or the range of gate idle time	Real case (King Khaled International Airport)
Mixed integer nonlinear programming	Li [5, 32]	Minimizing the number of gate conflicts of any two adjacent aircrafts assigned to the same gate	Real case (Continental Airlines, Houston Gorge Bush Intercontinental Airport)
	Bolat [31]	Minimizing the variance or the range of gate idle time	Real case (King Khaled International Airport)
Multiple objective GAP formulations	Hu and Di Paolo [36]	Minimize passenger walking distance, baggage transport distance, and aircraft waiting time on the apron	Theoretical
	Wei and Liu [16]	(i) Minimizing the total walking distance for passengers (ii) Minimizing the variance of gates idle times	Theoretical
	B.A.C.o.E.B. Team and A.I.C.o.E. Team [17]	(i) Minimizing walking distance (ii) Maximizing the number of gated flights (iii) Minimizing flight delays	Theoretical
	Yan and Huo [2]	(i) Minimizing passenger walking distances (ii) Minimizing the passenger waiting time	Real case (Chiang Kai-Shek Airport)
	Kaliszewski and Miroforidis [37]	Finding gate assignment efficiency which represents rational compromises between waiting time for gate and apron operations	Theoretical
Stochastic model	Yan and Tang [10]	Minimizing the total passenger waiting time	Real case (Taiwan International Airport)
	Genç et al. [38]	Maximizing gate duration, which is total time of the gates allocated	Theoretical and real case (Ataturk Airport of Istanbul, Turkey)
	Şeker and Noyan [9]	Minimizing the expected variance of the idle time	Theoretical
Quadratic assignment problem (QAP)	Drexel and Nikulin [3]	(i) Minimizing the number of ungated flights (ii) Minimizing the total passenger walking distances or connection times (iii) Maximizing the total gate assignment preferences	Theoretical
	Haghani and Chen [13]	Minimizing the total passenger walking distances	Theoretical

TABLE I: Continued.

Formulation	References	Criterion (comments)	Problem type
Scheduling problems	Li [39]	(i) Maximizing the sum of the all products of the flight eigenvalue (ii) Maximizing the gate eigenvalue that the flight assigned	Theoretical
Quadratic mixed binary programming	Bolat [34]	Minimizing the variance of idle times	Real case (King Khaled International Airport)
	Zheng et al. [33]	Minimizing the overall variance of slack time	Real case (Beijing International Airport, China)
	Xu and Bailey [14]	Minimizing the passenger connection time	Theoretical
Binary quadratic programming	Ding et al. [6, 7, 35]	Minimize the number of ungated flights and the total walking distances or connection times	Theoretical
Clique partitioning problem (CPP)	Dorndorf et al. [8]	(i) Maximizing the total assignment preference score (ii) Minimizing the number of unassigned flights (iii) Minimizing the number of tows (iv) Maximizing the robustness of the resulting schedule	Theoretical
Network representation	Maharjan and Matis [40]	Minimizing both fuel burn of aircraft and the comfort of connecting passengers	Real case (Continental Airlines at George W. Bush Intercontinental Airport in Houston (IAH))
Robust optimization	Diepen et al. [41]	Maximizing the robustness of a solution to the gate assignment problem	Real case (Amsterdam Airport Schiphol)

led to very high computing time. Section 3.1 addressed the exact solution techniques and the optimization programming language used to solve the proposed models to their optimality. Sections 3.1, 3.2, and 3.3 include the research work that has been done on the exact, heuristic, and metaheuristic approaches for solving the AGAP.

3.1. Exact Algorithms. Exact algorithms are those that yield an optimal solution. According to the literature, different exact solution techniques have been used to solve the GAP. As an example branch and bound was used as well as column generation and other methods, and in some research, the authors used some optimization programming languages like CPLEX and AMPL. In this section, only the research work that has been done on the exact solution techniques for solving the AGAP is presented.

Li [5, 32] solved the proposed hybrid mathematical model using CPLEX software. Mangoubi and Mathaisel [11] relaxed the integrality of the developed ILP model and solved the relaxed ILP model using CG; an optimal solution has been obtained for minimizing the total walking distance. Bihr [12] proposed a primal-dual simplex algorithm to find the solution and found the optimal solution. Yan and Huo [2] used simplex algorithm with column generation and weighting method to solve the provided model. Bolat [30, 34], Li [39], and Yan and Huo [2] used branch and bound algorithm to solve the models they have developed. Reference [14] used branch and bound algorithm and compared the result with tabu search algorithm.

3.2. Heuristic Algorithms. Basically the GAP is a QAP and it is an NP-hard problem as shown in Obata [21]. Since the AGAP is NP-hard, researchers have suggested various heuristic and metaheuristics approaches for solving the GAP. This section is for the heuristic algorithms; with heuristic algorithms, theoretically there is a chance to find an optimal solution. That chance can be remote because heuristics often reach a local optimal solution and get stuck at that point, so it was necessary to have modern heuristics called metaheuristic. This approach will be presented in the following part in this section; the research work that has been done on the heuristic approaches for solving the AGAP is presented.

Yan and Tang [10] developed a framework designed to deal with the GAP which has stochastic flight delays; the developed framework was with a heuristic approach embedded in it. Genç [42] used several heuristics, which are the “Ground Time Maximization Heuristic,” “Idle Time Minimization algorithm,” and “Prime Time Heuristic,” to solve the GAP with minimizing the idle gate time (or maximizing the number of assigned flights) as a performance measure. Ding et al. [6, 35] designed a greedy algorithm for solving the GAP with the objective of minimizing the number of ungated flights. Lim et al. [24] used several solution approaches, which are the “Insert Move Algorithm,” the “Interval Exchange Move Algorithm,” and a “Greedy Algorithm,” to solve the developed model for the GAP. Yan et al. [29] proposed a simulation framework and developed an optimization model (Section 2.1.2) and then solved the model using two greedy heuristics: the first was related to the number of passengers and the second was related to the

TABLE 2: Resolution methods.

Method	References	Approach/results	Problem type
Exact algorithms	Mangoubi and Mathaisel [11]	Linear programming relaxation	Real case (Toronto International Airport)
	Bihr [12]	Primal-dual simplex	Theoretical
	Yan and Huo [2]	Simplex; branch and bound	Real case (Chiang Kai-Shek Airport)
	Bolat [30, 34]; Xu and Bailey [14]; Li [39]	Branch and bound	Real case (King Khaled International Airport, KSA); theoretical
Heuristic algorithms	Thengvall et al. [43]	Bundle algorithm approach	Theoretical
	Yan and Tang [10]	Heuristic approach embedded in a framework designed	Real case (Taiwan International Airport)
	Ding et al. [6, 35]	Greedy algorithm	Theoretical
	Lim et al. [24]	The Insert Move Algorithm, the Interval Exchange Move Algorithm, and a Greedy Algorithm	Theoretical
	Diepen et al. [25]	Column generation	Real case (Amsterdam Airport Schiphol)
	Dorndorf et al. [8]	Heuristic based on the ejection chain algorithm	Theoretical
	Mangoubi and Mathaisel [11]	Heuristic approach	Real case (Toronto International Airport)
	Vanderstraeten and Bergeron [28]	ADAP	Theoretical
	Yan et al. [29]	Greedy heuristics	Real case (Chiang Kai-Shek Airport)
	Bolat [30]	Heuristic branch and trim	Real case (King Khaled International Airport, KSA)
	Bolat [34]	Heuristic branch and bound, SPH heuristic	Real case (King Khaled International Airport, KSA)
	Haghani and Chen [13]	Heuristic approach	Theoretical
	Genç [42]	Ground time maximization heuristic, and idle time minimization heuristic	Theoretical and real case (Ataturk Airport of Istanbul, Turkey)
	B.A.C.o.E.B. Team and A.I.C.o.E. Team [17]	A hybrid heuristics algorithm guided by simulated annealing and greedy heuristic	Theoretical
Bouras et al. [45]	Heuristic approach	Theoretical	
Metaheuristic algorithms	Ding et al. [6, 35]	Tabu search	Theoretical
	Ding et al. [7]	Simulated annealing, hybrid of simulated annealing and tabu search	Theoretical
	Lim et al. [24]	TS algorithm and a memetic algorithm	Theoretical
	Hu and Di Paolo [36]	New genetic algorithm with uniform crossover	Theoretical
	Drexl and Nikulin [3]	Pareto simulated annealing	Theoretical
	Xu and Bailey [14]	Tabu search	Theoretical
	Bolat [31]	Genetic algorithm	Real case (King Khaled International Airport, KSA)
	Şeker and Noyan [9]	Tabu search algorithms	Theoretical
	Zheng et al. [33]	A tabu search algorithm and metaheuristic method	Real case (Beijing International Airport, China)
	Wei and Liu [16]	A hybrid genetic algorithm	Theoretical
	Gu and Chung [44]	Genetic algorithms approach	Theoretical
	Cheng et al. [23]	Genetic algorithm (GA), tabu search (TS), simulated annealing (SA), and a hybrid approach based on SA and TS	Real case (Incheon International Airport, South Korea)
Bouras et al. [45]	Genetic algorithm (GA), tabu search (TS), and simulated annealing (SA)	Theoretical	

TABLE 2: Continued.

Method	References	Approach/results	Problem type
OPL	Li [5, 32]	Optimization programming language (CPLEX)	Real case (Continental Airlines, Houston Gorge Bush Intercontinental Airport)
	Tang et al. [27]	Using CPLEX 10.0 solver concert with C language	Real case (Taiwan International Airport)
	Prem Kumar and Bierlaire [18]	Optimization programming language (OPL)	Theoretical
	Maharjan and Matis [40]	AMPL/CPLEX 11.2	Real case (Continental Airlines at George W. Bush Intercontinental Airport in Houston (IAH))

arrival time of the flight at the gate. The distance result showed that the optimization model outperformed the heuristic but the heuristic simulation time outperformed the optimization model. Diepen et al. [25] solved the resulting LP-relaxation and the original ILP model using column generation.

Thengvall et al. [43] represented a heuristic approach for the problem of schedules recovery in airports during hub closures; the proposed approach was a bundle algorithm approach. Dorndorf et al. [8] used a heuristic approach that was developed by Dorndorf and Pesch (1994), which was based on the ejection chain algorithm, to solve the transformed model (CPP model). Mangoubi and Mathaisel [11] also used heuristic approach to solve the GAP with the objective of minimizing walking distance for the passengers and the obtained solutions were compared with the optimal solution using LP to obtain the deviation in the results. Haghani and Chen [13] used a heuristic approach to solve the GAP with the objective of minimizing walking distance for the passengers. Vanderstraeten and Bergeron [28] developed a direct assignment of flights to gates algorithm, named ADAP; the developed algorithm was an implicit enumeration which has been faster by carefully applying some variables selection criteria, which are the concept of “main chain,” the assigned weights to variables, and the single assignment constraints. Bolat [30] used branch and trim heuristic to solve the GAP with the objective of minimizing the slack times range. Bolat [34] used the HBB and SPH heuristics to solve the models that he has developed for the GAP. The first approach (HBB) is a B&B approach that has been utilized by some restrictions on the number of the nodes that has to be branched in a search tree while the second approach (SPH) developed a heuristic that after N iterations builds only one solution. The used heuristics assigned flights one at a time by considering all available gates and for determining the most permissible gate a priority function was utilized.

3.3. Metaheuristic Algorithms. As mentioned before in Section 3.2, heuristics often get stuck in a local optimal solution, but metaheuristics or “modern heuristics” introduce systematic rules to deal with this problem. The systematic rules avoid local optima or give the ability of moving out of local optima. The common characteristic of these metaheuristics is the use of some mechanisms to avoid local optima. Metaheuristics succeed in leaving the local optimum by temporarily accepting moves that cause worsening of the

objective function value. In this section, the research work that has been done on the metaheuristic approaches for solving the AGAP is presented.

Gu and Chung [44] introduced a genetic algorithm model to solve the AGAP. The developed model has been implemented in a high-level programming language; the effectiveness of the developed model has been validated by testing different scenarios and the results showed that the performance of the developed model was efficient. Şeker and Noyan [9] developed stochastic programming models; the developed model has been formulated as a mixed integer programming but in large scale. The developed models were solved using tabu search (TS) algorithm, and the obtained results were with high quality.

Cheng et al. [23] studied the performance of several metaheuristics in solving the GAP. The metaheuristics were genetic algorithm (GA), tabu search (TS), simulated annealing (SA), and a hybrid of SA and TS. Tabu search (TS) outperforms SA and GA but the hybrid approach outperforms TS in terms of solution quality. Ding et al. [6, 35] used a tabu search algorithm to solve the GAP; the starting (initial) solution was obtained using a designed greedy algorithm. Xu and Bailey [14] developed a tabu search algorithm to solve the GAP and compared the result of the developed algorithm with a branch and bound algorithm. The results showed that the two approaches provide optimal solutions for the studied problems but TS obtained was better in the CPU time.

Zheng et al. [33] developed a model for solving the GAP (see Section 2) and used a TS algorithm to obtain solutions for the developed model. Ding et al. [7] used a simulated annealing and a hybrid of SA and TS to solve the GAP model that they have developed; the starting (initial) solution was obtained using a designed greedy algorithm. Lim et al. [24] proposed TS and memetic algorithms to solve the GAP. Drexler and Nikulin [3] solved the multicriteria airport gate assignment using Pareto simulated annealing. Hu and Di Paolo [36] solved the multiobjective gate assignment problem (MOGAP) using a new genetic algorithm with uniform crossover. Bolat [31] used genetic algorithm (GA) to minimize the variance or the range of gate idle time. Wei and Liu [16] modified a hybrid genetic algorithm to solve the fuzzy AGAP model. Table 2 summarizes all the above solution techniques used recently for the AGAP.

Recently, Bouras et al. [45] approached the AGAP as a parallel machine-scheduling problem with some priority and

TABLE 3: Number of publications per year.

Year	Number of publications	References
1974	1	Steuart [1]
1985	1	Mangoubi and Mathaisel [11]
1988	1	Vanderstraeten and Bergeron [28]
1990	1	Bihl [12]
1998	1	Haghani and Chen [13]
1999	2	Bolat [30], Gu and Chung [44]
2000	1	Bolat [34]
2001	3	Bolat [31], Xu and Bailey [14], Yan and Huo [2]
2002	2	Lam et al. [15], Yan et al. [29]
2003	1	Thengvall et al. [43]
2004	2	Ding et al. [6, 35]
2005	3	Ding et al. [7], Lim et al. [24], Al-Khalifah [19]
2007	4	Diepen et al. [25], Dorndorf et al. [20, 22], Yan and Tang [10]
2008	4	Diepen et al. [26], Drexl and Nikulin [3], Li [5, 32]
2009	4	Wei and Liu [16], Hu and Di Paolo [36], Tang et al. [27], B.A.C.o.E.B. Team and A.I.C.o.E. Team [17]
2010	4	Dorndorf et al. [8], Genç [42], Zheng et al. [33], Li [39]
2011	2	Prem Kumar and Bierlaire [18], Genç et al. [38]
2012	5	Li [39], Şeker and Noyan [9], Diepen et al. [41], Maharjan and Matis [40], Kaliszewski and Miroforidis [37], Cheng et al. [23]
2014	1	Bouras et al. [45]

eligibility. They solved the problem with the aim of minimizing the following objectives: total cost, total tardiness, and maximum tardiness. They developed three heuristics and used three metaheuristics (simulated annealing, tabu search, and genetic algorithms). The evaluation was conducted over 238 generated instances but only 50 instances were presented in the report. The results showed that simulated annealing was the most efficient metaheuristic to solve the problem.

4. Conclusion and Research Trends

In this survey, we have presented the very recent publications about the airport gate assignment problem. The collected literature has the aim of identifying the contributions and the trends in the research using exact or approximate methods.

An abundant literature is listed to describe mathematical formulation on AGAP or other related problems. They have been grouped in such a way that the user is guided to identify each problem specification. For single objective, integer/binary models are described along with mixed integer ones. Nonlinear formulations are also described for mixed/integer models. Rare are the authors who really came out with exact solutions using existing commercial optimization software or their own exact methods (branch and bound...). Heuristics were suggested to build feasible solutions and improve the latter solutions using metaheuristics.

For multiobjective optimization, several models have been formulated as nonlinear objectives with little success in solving such problems with exact methods in a reasonable time.

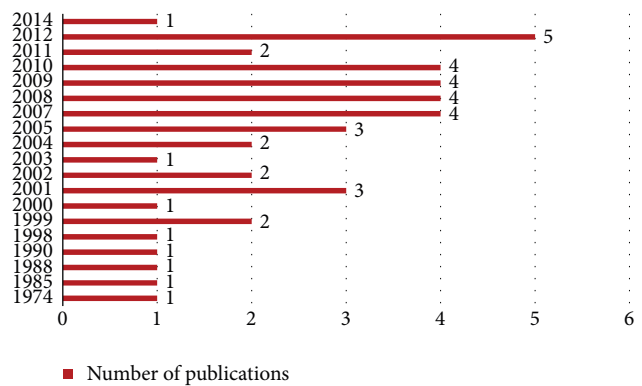


FIGURE 1: Number of publications per year.

Related problems to AGAP have also been introduced in the survey. Some cases of AGAP have been formulated as some well-known combinatorial optimization problems such as QAP.

Since 2005 (Table 3, Figures 1-2), most of the people started to consider using heuristics/metaheuristics as tools to solve AGAP since the problem is NP-hard, and the issue of time solving was still unresolved by the existing tools.

In practice, major airlines may have more than 1000 daily flights to handle at more than 50 gates, which results in billions of binary variables in formulation. B&B based MIP solvers (i.e., CPLEX) will not be able to handle such huge size problems within a reasonable time bound.

TABLE 4: Number of publications per research area.

Area	Number of publications	References
Integer linear programming (IP)	3	Lim et al. [24], Diepen et al. [25], Diepen et al. [26]
Binary integer programming	6	Mangoubi and Mathaisel [11], Yan et al. [29], Vanderstraeten and Bergeron [28], Bihr [12], Tang et al. [27], Prem Kumar and Bierlaire [18]
Mixed integer linear programming (MILP)	3	Bolat [30], Bolat [31], Kaliszewski and Miroforidis [37]
Mixed integer nonlinear programming	3	Li [5, 32], Bolat [31]
Quadratic programming (QP)	6	Bolat [34], Zheng et al. [33], Xu and Bailey [14], Ding et al. [6, 7, 35]
Multiple objective GAP formulations	5	Hu and Di Paolo [36], Wei and Liu [16], B.A.C.o.E.B. Team and A.I.C.o.E. Team [17], Yan and Huo [2], Kaliszewski and Miroforidis [37]
Stochastic models	3	Yan and Tang [10], Genç et al. [38], Şeker and Noyan [9]
Quadratic assignment problem (QAP)	2	Drexel and Nikulin [3], Haghani and Chen [13]
Scheduling problems	1	Li [39]
Clique partitioning problem (CPP)	1	Dorndorf et al. [8]
Network representation	1	Diepen et al. [41]
Robust optimization	1	Maharjan and Matis [40]
Exact algorithms	7	Mangoubi and Mathaisel [11], Bihr [12], Yan and Huo [2], Bolat [30, 34]; Xu and Bailey [14]; Li [39]
Heuristic algorithms	16	Thengvall et al. [43], Yan and Tang [10], Ding et al. [6, 35], Lim et al. [24], Diepen et al. [25], Dorndorf et al. [8], Mangoubi and Mathaisel [11], Vanderstraeten and Bergeron [28], Yan et al. [29], Bolat [30, 34], Haghani and Chen [13], Genç [42], B.A.C.o.E.B. Team and A.I.C.o.E. Team [17], Bouras et al. [45]
Metaheuristic algorithms	14	Ding et al. [6, 35], Ding et al. [7], Lim et al. [24], Hu and Di Paolo [36], Drexel and Nikulin [3], Xu and Bailey [14], Bolat [31], Şeker and Noyan [9], Zheng et al. [33], Wei and Liu [16], Gu and Chung [44], Cheng et al. [23], Bouras et al. [45]
OPL	4	Li [5, 32], Tang et al. [27], Prem Kumar and Bierlaire [18], Maharjan and Matis [40]

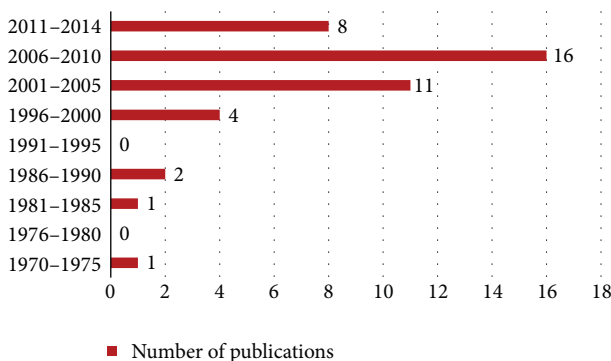


FIGURE 2: Number of publications per 5 years.

A growing interest in metaheuristics (Table 4, Figure 3) has been observed in the recent papers on AGAP. TS, SA, and GA are the most used improvement methods. Some hybridized methods combining these methods have also been suggested.

It is quite understandable that researches move towards the use of such methods. Airport managers usually face changes on their plans and need to change their plans due to uncertainties.

With a lack of studies on robust methods or stochastic methods, where only few papers appeared on these subjects, researchers have at their disposal a battery of methods such as evolutionary methods, parallel metaheuristics, and self-tuning metaheuristics to apply on this interesting problem.

With the actual trend of heuristics use, it would be interesting to work on a combination of these algorithms

In the framework of a hyperheuristic approach, many hard combinatorial optimization problems have already been tackled using heuristics (Burke et al. [46]), which motivates our recommendation of this approach. Strengthening weaknesses is the essence of hyperheuristics since they smartly work with search spaces of heuristics. The idea is, during a process of exploring new solutions, to choose the adequate metaheuristic where the currently used one is failing to improve or to generate new heuristics by using the components of existing ones (Soubeiga [47]; Burke et al. [48]).

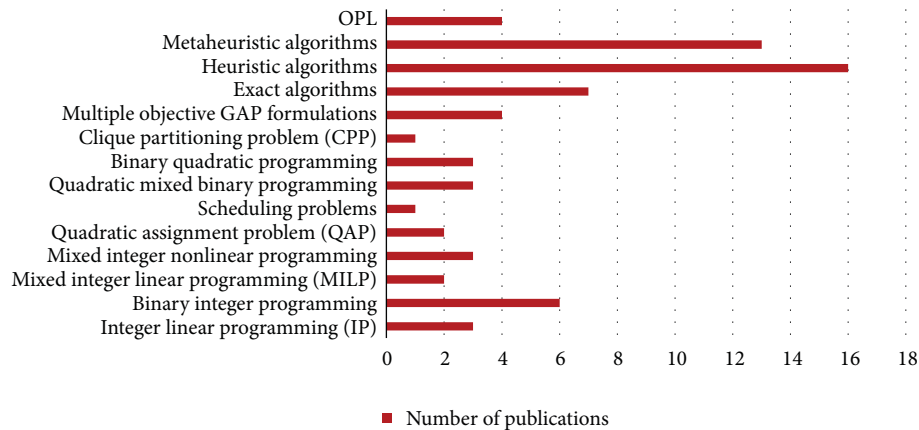


FIGURE 3: Number of publications per research area.

There is an alternative to these approximate methods, which could be explored: the use of efficient exact methods (branch and bound...) with the introduction of tight lower bounds. In our review, we found that only Tang et al. [27] developed a lower bound and used a classical branch and bound algorithm, while other authors combined special heuristics with their method (Bolat [30, 34], Li [39], and Yan and Huo [2]).

We have also noticed the absence of a data set for AGAP. It could be interesting to have a set of instances of different sizes that can be shared by researchers, with benchmarks (optimal and best-known values) and CPU times to help comparing methods as it is the case for known problems: quadratic assignment problem and travelling salesman problem.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors extend their appreciation to the Deanship of Scientific Research Center of the College of Engineering at King Saud University for supporting this work.

References

- [1] G. N. Steuart, "Gate position requirements at metropolitan airports," *Transportation Science*, vol. 8, no. 2, pp. 169–189, 1974.
- [2] S. Yan and C.-M. Huo, "Optimization of multiple objective gate assignments," *Transportation Research A: Policy and Practice*, vol. 35, no. 5, pp. 413–432, 2001.
- [3] A. Drexl and Y. Nikulin, "Multicriteria airport gate assignment and Pareto simulated annealing," *IIE Transactions*, vol. 40, no. 4, pp. 385–397, 2008.
- [4] W. Li and X. Xu, "Optimized assignment of airport gate configuration based on immune genetic algorithm," in *Measuring Technology and Mechatronics Automation in Electrical Engineering*, pp. 347–355, Springer, 2012.
- [5] C. Li, "Airport gate assignment A hybrid model and implementation," <http://arxiv.org/abs/0903.2528>.
- [6] H. Ding, A. Lim, B. Rodrigues, and Y. Zhu, "New heuristics for over-constrained flight to gate assignments," *Journal of the Operational Research Society*, vol. 55, no. 7, pp. 760–768, 2004.
- [7] H. Ding, A. Lim, B. Rodrigues, and Y. Zhu, "The over-constrained airport gate assignment problem," *Computers & Operations Research*, vol. 32, no. 7, pp. 1867–1880, 2005.
- [8] U. Dorndorf, F. Jaehn, and E. Pesch, "Flight gate scheduling with respect to a reference schedule," *Annals of Operations Research*, vol. 194, pp. 177–187, 2012.
- [9] M. Şeker and N. Noyan, "Stochastic optimization models for the airport gate assignment problem," *Transportation Research E: Logistics and Transportation Review*, vol. 48, no. 2, pp. 438–459, 2012.
- [10] S. Yan and C.-H. Tang, "A heuristic approach for airport gate assignments for stochastic flight delays," *European Journal of Operational Research*, vol. 180, no. 2, pp. 547–567, 2007.
- [11] R. S. Mangoubi and D. F. X. Mathaisel, "Optimizing gate assignments at airport terminals," *Transportation Science*, vol. 19, no. 2, pp. 173–188, 1985.
- [12] R. A. Bihl, "A conceptual solution to the aircraft gate assignment problem using 0, 1 linear programming," *Computers and Industrial Engineering*, vol. 19, no. 1–4, pp. 280–284, 1990.
- [13] A. Haghani and M.-C. Chen, "Optimizing gate assignments at airport terminals," *Transportation Research A: Policy and Practice*, vol. 32, no. 6, pp. 437–454, 1998.
- [14] J. Xu and G. Bailey, "The airport gate assignment problem: mathematical model and a tabu search algorithm," in *Proceedings of the IEEE 34th Annual Hawaii International Conference*, Delta Technology Inc., Atlanta, Ga, USA, 2001.
- [15] S. Lam, J. Cao, and H. Fan, "Development of an intelligent agent for airport gate assignment," *Journal of Airport Transportation*, vol. 7, no. 2, pp. 103–114, 2002.
- [16] D.-X. Wei and C.-Y. Liu, "Fuzzy model and optimization for airport gate assignment problem," in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS '09)*, pp. 828–832, Shanghai, China, November 2009.
- [17] B.A.C.o.E.B. Team and A.I.C.o.E. Team, *Gate Assignment Solution (GAM) Using Hybrid Heuristics Algorithm*, Wipro Technologies (WIT), 2009.

- [18] V. Prem Kumar and M. Bierlaire, "Multi-objective airport gate assignment problem in planning and operations," *Journal of Advanced Transportation*, vol. 48, no. 7, pp. 902–926, 2014.
- [19] H. S. Al-Khalifah, *Dynamic Gate Assignment at an Airport*, Department of Industrial Engineering, King Saud University, Riyadh, Saudi Arabia, 2005.
- [20] U. Dorndorf, A. Drexler, Y. Nikulin, and E. Pesch, "Flight gate scheduling: state-of-the-art and recent developments," *Omega*, vol. 35, no. 3, pp. 326–334, 2007.
- [21] T. Obata, *Quadratic Assignment Problem: Evaluation of Exact and Heuristic Algorithms*, Rensselaer Polytechnic Institute, Troy, NY, USA, 1979.
- [22] U. Dorndorf, F. Jaehn, C. Lin, H. Ma, and E. Pesch, "Disruption management in flight gate scheduling," *Statistica Neerlandica. Journal of the Netherlands Society for Statistics and Operations Research*, vol. 61, no. 1, pp. 92–114, 2007.
- [23] C.-H. Cheng, S. C. Ho, and C.-L. Kwan, "The use of meta-heuristics for airport gate assignment," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12430–12437, 2012.
- [24] A. Lim, B. Rodrigues, and Y. Zhu, "Airport gate scheduling with time windows," *Artificial Intelligence Review*, vol. 24, no. 1, pp. 5–31, 2005.
- [25] G. Diepen, J. M. van den Akker, J. A. Hoogeveen, and J. W. Smeltink, *Using Column Generation for Gate Planning at Amsterdam Airport Schiphol*, Department of Information and Computing Sciences, Utrecht University, 2007.
- [26] G. Diepen, J. M. V. D. Akker, and J. A. Hoogeveen, *Integration of Gate Assignment and Platform Bus Planning*, Department of Information and Computing Sciences, Utrecht University, 2008.
- [27] C. Tang, S. Yan, and Y. Hou, "A gate reassignment framework for real time flight delays," *4OR: A Quarterly Journal of Operations Research*, vol. 8, no. 3, pp. 299–318, 2010.
- [28] G. Vanderstraeten and M. Bergeron, "Automatic assignment of aircraft to gates at a terminal," *Computers and Industrial Engineering*, vol. 14, no. 1, pp. 15–25, 1988.
- [29] S. Yan, C.-Y. Shieh, and M. Chen, "A simulation framework for evaluating airport gate assignments," *Transportation Research A: Policy and Practice*, vol. 36, no. 10, pp. 885–898, 2002.
- [30] 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.
- [31] 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.
- [32] C. Li, "Airport gate assignment: new model and implementation," <http://arxiv.org/abs/0811.1618>.
- [33] P. Zheng, S. Hu, and C. Zhang, "Airport gate assignments model and algorithm," in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT '10)*, vol. 2, pp. 457–461, IEEE, Chengdu, China, July 2010.
- [34] A. Bolat, "Procedures for providing robust gate assignments for arriving aircrafts," *European Journal of Operational Research*, vol. 120, no. 1, pp. 63–80, 2000.
- [35] H. Ding, A. Lim, B. Rodrigues, and Y. Zhu, "Aircraft and gate scheduling optimization at airports," in *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pp. 1185–1192, January 2004.
- [36] X. B. Hu and E. Di Paolo, "An efficient Genetic Algorithm with uniform crossover for the multi-objective airport gate assignment problem," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 55–62, September 2007.
- [37] I. Kaliszewski and J. Miroforidis, "On interfacing multiobjective optimisation models the case of the airport gate assignment problem," in *Proceedings of the International Conference on Application and Theory of Automation in Command and Control Systems*, London, UK, 2012.
- [38] H. M. Genç, O. K. Erol, İ. Eksin, M. F. Berber, and B. O. Gülerüyz, "A stochastic neighborhood search approach for airport gate assignment problem," *Expert Systems with Applications*, vol. 39, no. 1, pp. 316–327, 2012.
- [39] W. Li, "Optimized assignment of civil airport gate," in *Proceedings of the International Conference on Intelligent System Design and Engineering Application (ISDEA '10)*, pp. 33–38, Changsha, China, October 2010.
- [40] B. Maharjan and T. I. Matis, "Multi-commodity flow network model of the flight gate assignment problem," *Computers and Industrial Engineering*, vol. 63, no. 4, pp. 1135–1144, 2012.
- [41] G. Diepen, J. M. van den Akker, and J. A. a. Hoogeveen, "Finding a robust assignment of flights to gates at Amsterdam AIRport Schiphol," *Journal of Scheduling*, vol. 15, no. 6, pp. 703–715, 2012.
- [42] H. M. Genç, "A new solution approach for the airport gate assignment problem for handling of uneven gate demands," in *Proceedings of the World Conference of Transport Research (WCTR '10)*, Lisbon, Portugal, 2010.
- [43] B. G. Thengvall, J. F. Bard, and G. Yu, "A bundle algorithm approach for the aircraft schedule recovery problem during hub closures," *Transportation Science*, vol. 37, no. 4, pp. 392–407, 2003.
- [44] Y. Gu and C. A. Chung, "Genetic algorithm approach to aircraft gate reassignment problem," *Journal of Transportation Engineering*, vol. 125, no. 5, pp. 384–389, 1999.
- [45] A. Bouras, M. A. Ghaleb, U. S. Suryahatmaja, and A. M. Salem, "The Airport Gate Assignment Problem (AGAP) as parallel machine scheduling with eligibility: metaheuristic approaches," *Industrial Engineering Technical Report KSUENG-IED-TR-012014-01*, Department of Industrial Engineering, King Saud University, Riyadh, Saudi Arabia, 2014.
- [46] E. K. Burke, M. Gendreau, M. Hyde et al., "Hyper-heuristics: a survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.
- [47] E. Soubeiga, *Development and Application of Hyperheuristics to Personnel Scheduling*, University of Nottingham, 2003.
- [48] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and R. Qu, "A survey of hyper-heuristics," *Tech. Rep. NOTTCS-TR-SUB-0906241418-2747*, School of Computer Science and Information Technology, University of Nottingham, 2009.



Hindawi

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

