Research Article

Continuous Time Formulation and Lagrangian Relaxation Algorithm for the Gate Assignment Problem

Yuan Yuan , Ping Yan, and Liqiang Zhao

College of Economics and Management, Shenyang Aerospace University, Shenyang 110136, China

Correspondence should be addressed to Yuan Yuan; yy_sau@163.com

Received 2 February 2020; Revised 20 July 2020; Accepted 27 July 2020; Published 25 August 2020

Academic Editor: Bruno G. M. Robert

Copyright © 2020 Yuan Yuan 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.

Gates are important operating facilities and resources in civil airports. It is a core task in the airport operation management to select reasonable gates for inbound and outbound flights. We present a continuous time formulation with second-order cone programming (SOCP) for the gate assignment problem which allocates flights to available gates to optimize both the transfer time of passengers and the robustness of the airport operations schedules. The problem is formulated as a mixed integer nonlinear program, and then, the quadratic objective that minimizes the walking distance of transferring passengers is linearized, and the objective that minimizes the variance of idle time at the gates is transformed to a second-order cone constraint with a linear objective function. Then, a Lagrangian relaxation algorithm is developed by exploiting the problem structure. Computational tests are carried out to illustrate the efficiency of the model and the algorithms. It is shown that the continuous time formulation is more efficient than the existing model, and the Lagrangian relaxation algorithm can obtain better solutions faster than a commercial solver.

1. Introduction

The gate at civil airports serves as the end point of the arriving aircraft and the starting point of the departing aircraft. It provides a place for aircraft parking, passengers boarding and disembarking, and baggage loading, as well as a transit point for aviation and ground transportation. With the significant increase in civil aviation business in the past decades, the management and coordination ability of airports and airlines are facing enormous challenges. Gates are important operating facilities and scarce resources which play a critical role in the airports operation system, linking the service of airside and landside, to ensure the smooth operation of the airport ground. It is not only costly but also limited by many external factors to relieve the capacity pressure of the gates by expanding the airport. Therefore, utilizing the limited available gates more efficiently becomes the main focus of the management of airports.

As one of the important issues that airport operations managers face every day, the gate assignment problem (GAP) is to allocate flights to available gates under a set of

technical and management constraints in order to maximize the satisfaction of the passengers and the operational efficiency of the airport. With the development of the air transportation industry, the demand for gates at the airports is increasing rapidly. Inherent delays of air transport operations will bring uncertainties and have a great impact on the gate assignment plan. Therefore, robustness of the gate assignment plan is important to deal with potential risks due to flight delays. We study the gate assignment problem under the deterministic conditions to provide an initial gate assignment plan with robustness, which facilitates the online adjustments. In order to balance the multiple interests of different stakeholders in the gate assignment process, this article considers the gate assignment problem with three weighted objectives, including the number of ungated flights, i.e., flights assigned to the remote apron stand, total distance that passengers need to transfer flights, and variance of idle time at the gates. A lot of studies have focused on the gate assignment problem with different characteristics. The first research studies which focus on the gate assignment appeared in the early 1970s, e.g., study by Braaksma and

Shortreed [1], utilized the critical path method to construct a simulation model to improve gate utilization at Toronto International Airport. Over the past few decades, numerous modelling and solution techniques were developed to improve the utilization of gates at airports. A comprehensive survey of the developments of the gate assignment problems was provided by Dorndorf et al. [2], and Bouras et al. [3] gave an updated review for the research progress of the problem and the various solution methods. Recently, Daş et al. [4] reviewed the formulation, objectives, and also solution approaches for variants of the gate assignment problem. The main existing work for the GAP is summarized as follows.

Mangoubi and Mathaisel [5] solved the GAP for minimizing passenger walking distance using the linear programming relaxation of an integer program and a heuristic algorithm. Yan and Chang [6] also tackled the problem for minimizing passenger walking distance but formulated the problem using the multicommodity network flow framework and designed an algorithm based on Lagrangian relaxation. Haghani and Chen [7] modelled the problem considering walking distance of transferring passengers as a quadratic assignment problem and proposed a heuristic procedure to solving the problem. Bolat [8] dealt with the GAP considering the minimization of the variance of idle time at the gates with a quadratic objective function and developed a branch-and-bound method to solve it. Bolat [9] continued to provide different models for the same problem and identified the polynomial time solvable conditions. Yan and Huo [10] studied a multiobjective gate assignment problem which minimizes the walking distance and waiting time for the passengers and developed a solution algorithm combining column generation, simplex, and branch-and-bound. Lim et al. [11] studied the GAP considering that arrival and departure time of the flights can change in a time window and developed Tabu search as well as memetic algorithms to solve the problem. Yan and Tang [12] studied the GAP with stochastic delay of the flights and developed a heuristic algorithm to handle the gate assignments. Dorndorf et al. [13] modelled the gate assignment problem as a resource constrained project scheduling problem based on which two approaches of robustness were proposed. Drexel and Nikulin [14] proposed a quadratic integer program for the GAP with multiple objectives and developed a Pareto simulated annealing algorithm. Dorndorf et al. [15] considered that a flight can be allocated to different gates when arriving at and departing from the airport. They transformed the problem to the clique partitioning problem and developed a heuristic approach based on the ejection chain to solve it. Later, this study was expanded for solving a multiple period problem in Dorndorf et al. [16]. Maharjan and Matis [17] considered the objective of minimizing aircraft fuel consumption and formulated the GAP as a binary integer multicommodity flow network model. Seker and Noyan [18] studied the GAP where the flight arrival and departure time is uncertain and developed a stochastic programming approach to measure the robustness. Diepen et al. [19] designed a column generation approach to deal with the robust gate assignment problem which is formulated as an

integer linear programming model. Wang et al. [20] considered the optimal modelling and algorithm design for real-time redistribution of gates when flight time is disturbed. Kim and Feron [21] studied the robust gate assignment by introducing a queuing model, which reduces the gate conflicts due to departure metering. Guépet et al. [22] studied the problem that allocates aircraft activities to the gates. The problem was formulated as a mixed integer program with several strengthening strategies. Yu et al. [23] studied the robust gate assignment problem and transformed the quadratic model equivalently to a more efficient MIP model. Liu et al. [24] considered the GAP with operational safety constraints, presented an optimization model, and designed a genetic algorithm to solve the problem. Dorndorf et al. [25] investigated the GAP where start and end time of flight activities is stochastic and formulated it as a clique partitioning model. Dell'Orco et al. [26] proposed a fuzzy bee colony optimization method for the GAP with uncertainty. Aktel et al. [27] proposed a new Tabu search algorithm and compared the performance of the algorithm with two metaheuristic algorithms. Kaliszewski et al. [28] approached the problem by multiobjective optimization as well as evolutionary multiobjective optimization. Daş [29] investigated the GAP with a new objective of shopping revenues at the airport and developed a hybrid heuristic algorithm for the multiobjective problem. Schaijk and Visser [30] considered stochastic gate constraints instead of deterministic ones in order to mitigate the impact of stochastic flight delays and improve the robustness of solutions. Xu et al. [31] proposed a new formulation for the robust gate assignment problem and transformed it into a series of binary models which are tractable. Deng et al. [32] proposed a multiobjective model for the robust gate assignment problem and solved it using a representative optimizer.

Among the existing researches, many mathematical programming approaches have been proposed for the GAP and achieved good optimization effects such as reducing the walking distance of passengers, improving the utilization of the gates, and equilibrating the idle time of gates. But there are still some shortcomings, such as the mathematical models are not efficient enough and less consideration of multiple objectives. Therefore, it is necessary to develop new modelling and solution methods for the multiobjective GAP considered in this article.

Early modelling methods developed for the GAP were based on discrete time methods; that is, the planning period was divided into equal time periods, and all the events (such as start and end of the activities) occurred at the endpoints of the time period. This modelling method suffers from the drawback that the dimension of model increases fast with the number of time periods. Continuous time modelling methods, which include sequence-based modelling method and event-based modelling method, were first proposed by the researchers in chemical engineering to avoid the limitations of discrete time modelling method (see a review by Floudas and Lin [33]). The sequence-based continuous time modelling method which obtains a full schedule by deciding

pair-wise sequences of the tasks has applications for the gate assignment problems in recent years, such as Liu et al. [24]. In this article, a more efficient formulation for the GAP is proposed using the event-based modelling method. Time slots are defined as several time periods with arbitrary length located at the planning period of a gate, each time slot representing an event, i.e., the duration of a flight staying at the gate. Figure 1 shows the definition of time slots for the gate assignment problem. The timings of all the events can be presented by special logical constraints. Thus, modelling the gate assignment problem by event-based continuous time modelling method will substantially reduce the number of binary variables comparing the discrete time modelling method and sequence-based continuous time modelling method.

Moreover, many studies examined the GAP as a quadratic assignment problem, as a result common optimization software or exact algorithms cannot solve it efficiently, and then, most researchers focus on resolving different variants of the problem by meta-heuristics methods. In this article, we optimize the gate assignment problem by formulating it as a mixed integer SOCP model. SOCP has been widely used in various fields, such as combinatorial optimization, robust optimization, machine learning, and optimal control, because it can be effectively solved by the interior point algorithm, see a review of its applications by Alizadeh and Goldfarb [34]. Mainstream optimization solvers such as IBM ILOG CPLEX can solve SOCP with integer variables efficiently. Utilizing the characteristic of second-order cone, the gate assignment problem that minimizes the variance of idle time at the gates can be transformed into a program with linear objective and second-order cone constraints.

The remainder of the paper is organized as follows. The gate assignment problem considered in this paper is described and formulated in Section 2; model reformulation strategies and settings are also discussed. A Lagrangian relaxation algorithm is developed in Section 3 for solving large problems. Computational experiments which test the performance of the modelling and solution approach are presented in Section 4. The conclusions are presented in Section 5.

2. Formulation of the GAP

2.1. Problem Description. The gate assignment problem involves the interests of multiple parties such as airports, passengers, and airlines, which is a multiobjective combinatorial optimization problem. Previous researches on the gate assignment problem mainly focus on improving the quality of passenger services and optimizing the operating efficiency of airports and airlines. In order to balance the multiple interests of different stakeholders in the gate assignment process, the gate assignment problem considered in this article optimizes three objectives:

- (1) Minimize the number of ungated flights

Terminal gate is the linkage between airside and landside. Passengers can walk directly from one side

FIGURE 1: Definition of time slots.

to the other through the bridge equipped at the gate. However, gates are very scarce resources at the airport. If no gate is available when a flight arrives, it should be allocated to the apron. Then, passengers need to be transferred between the aircraft and the terminal building by transfer buses, which will increase the connection time and operational cost, and also decrease the satisfactory of passengers. Thus, the primary goal is to minimize the number of ungated flights.

- (2) Minimize the total distance that passengers need to transfer flights

From the passengers' points of view, walking distance is a common consideration. In practice, arriving passengers and departing passengers do not pay much attention to the walking distance in the airport, while transferring walking distance between two connecting flights is most concerned since the idle time between two flights may be limited due to flight delay or variations. Thus, the walking distance for transferring passengers can be minimized by a well-designed gate assignment.

- (3) Minimize the variance of idle time at the gates

In order to maintain the robustness of the airport operating system, the gate assignment plan should be insensitive to the disruptions such as flight delay or cancellations due to severe weather and air traffic control. A reasonable way to achieve this goal is to retain enough idle time before arrival of each flight, which is to say the idle time before arrival of the flights should be equilibrium. In mathematics, this objective can be expressed by minimization of the variance of idle time.

Moreover, the following constraints need to be met to produce a feasible flight gate assignment:

- (1) Exclusive constraint. The flight is exclusive to the gate in time and space; i.e., one gate can serve no more than one flight at any time.

- (2) Unique constraint. Each flight needs to be assigned to one gate or the apron in the whole process from its arrival to departure.
- (3) Buffer time constraint. For any two flights that are consecutively assigned to the same gate, there should be a necessary safety time interval, called the buffer time at the gate, to ensure safety of the flights, and to give sufficient ground working time. Moreover, the buffer time can also relieve the pressure from stochastic delays of flights.
- (4) Type-matching constraint. The type of aircraft should match the type of gate; that is, large aircrafts are not allowed to be assigned to small gates, but small aircraft can be assigned to large gates, which can meet the service requirements of small aircrafts. However, the latter treatment will reduce the utilization of gates and affect the assignment for subsequent large aircrafts.

2.2. Mathematical Model. According to the objectives and constraints described in Section 2.1, a nonlinear programming model with binary variables is formulated for the gate assignment problem using the event-based continuous time modelling method.

2.2.1. Sets and Parameters

N : set of all the flights at the airport during the planning period

M : set of all the gates available at the airport during the planning period

M_i : set of feasible gates available for flight i , $i \in N$

M^0 : the remote apron stand, $M^+ = M \cup M^0$ and $M_i^+ = M_i \cup M^0$, $i \in N$

E : set of time slots of each gate; $|E|+1$ indicates the end of the planning period, the set $E^+ = E \cup \{|E|+1\}$

A_i : arrival time of flight i , $i \in N$

D_i : departure time of flight i , $i \in N$

B : buffer time for consecutive flights at the gate

$p_{kk'}$: distance from gate k to gate k' , $k, k' \in M^+$

$n_{ii'}$: number of transferring passengers from flight i to flight i' , $i, i' \in N$

T : time period considered in the problem

α : weight coefficient for the objective of assigning flights to the remote apron stand

β : weight coefficient for the objective of total walking distance of transferring passengers

γ : weight coefficient for the objective of variance of idle time at the gates

2.2.2. Decision Variable

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

$$i \in N, k \in M_i^+.$$

$$y_{ikl} = \begin{cases} 1, & \text{if flight } i \text{ is in time slot } l \text{ of gate } k, \\ 0, & \text{otherwise} \end{cases},$$

$$i \in N, k \in M_i^+, l \in E.$$

w_{kl} : idle time at gate k before time slot l , $k \in M$, $l \in E^+$

2.2.3. Mixed Integer Nonlinear Programming Model.

$$\min \alpha \sum_{i \in N} x_{iM^0} + \beta \sum_{i \in N} \sum_{k \in M_i^+} \sum_{i' \in N} \sum_{k' \in M_{i'}^+} n_{ii'} p_{kk'} x_{ik} x_{i'k'} + \gamma \sqrt{\sum_{k \in M} \sum_{l \in E^+} w_{kl}^2}, \quad (1)$$

$$\text{s.t.} \quad \sum_{k \in M_i^+} x_{ik} = 1, \quad i \in N, \quad (2)$$

$$\sum_{l \in E} y_{ikl} = x_{ik}, \quad i \in N, k \in M_i, \quad (3)$$

$$\sum_{i \in N} y_{ikl} \leq 1, \quad k \in M, l \in E, \quad (4)$$

$$y_{ikl} + y_{i'k,l-1} \leq 1, \quad i, i' \in N, A_i - D_{i'} < B, k \in M_i \cap M_{i'}, l \in E \setminus \{1\}, \quad (5)$$

$$\sum_{i \in N} y_{ik,l-1} \geq \sum_{i \in N} y_{ikl}, \quad k \in M, l \in E, \quad (6)$$

$$w_{k1} \geq A_i y_{ik1}, \quad i \in N, k \in M_i, \quad (7)$$

$$w_{kl} \geq (A_i - D_{i'}) (y_{ikl} + y_{i'k,l-1} - 1), \quad i, i' \in N, A_i - D_{i'} \geq B, k \in M_i \cap M_{i'}, l \in E \setminus \{1\}, \quad (8)$$

$$w_{k,|E|+1} \geq (T - D_i) \left(y_{ikl} - \sum_{i' \in N} y_{i'k,|E|+1} \right), \quad i \in N, k \in M_i, l \in E \setminus \{|E|\}, \quad (9)$$

$$w_{k,|E|+1} \geq (T - D_i) y_{ik|E|}, \quad i \in N, k \in M_i, \quad (10)$$

$$x_{ik} \in \{0, 1\}, \quad i \in N, k \in M_i^+, \quad (11)$$

$$y_{ikl} \in \{0, 1\}, \quad i \in N, k \in M_i, l \in E, \quad (12)$$

$$w_{kl} \geq 0, \quad k \in M, l \in E^+. \quad (13)$$

The objective (1) minimizes the weighted summation of the penalty of assigning flights to the remote apron stand, the total distance that passengers need to transfer flights, and the variance of idle time. Constraint (2) ensures that each flight have to be assigned to one gate or the apron. Constraint (3) indicates that any flight assigning to a gate must be assigned to an event of the gate. Constraint (4) indicates that each event can be assigned with at most one flight. Constraint (5) avoids flight assignments with conflict arrival and departure time. Constraint (6) indicates that an event is assigned unless its previous event is assigned. Constraints (7)–(10) indicate the relationship between the idle time before each event and the assignment variables. Constraints (11)–(13) are boundary constraints on the variables.

It is worthy to note that since the number of time slots must be greater than the number of flights assigned to the gate, there will be some empty time slots to which no flights will be assigned. For constraint (6), we hope that all of the actual events are continuously arranged without empty time slots between them, and all the empty time slots are arranged at the end. This can facilitate the calculation of w_{kl} , i.e., constraints (7)–(10), and this treatment does not affect the feasible assignments since the events are defined flexibly according to the assignment rather than defined on fixed time table.

2.3. Model Reformulation. Due to the complexity of the model presented above, it is extremely difficult to solve the model by optimization software or exact algorithms such as branch and bound. In this subsection, the objective function of minimizing the total walking distance of transferring passengers is transformed into a linear objective and a set of linear constraints, and the second part of objective function that minimizes the variance of idle time is transformed into a linear objective and a second-order cone constraint.

- (1) Reformulate the objective function of walking distance

Introduce decision variable $z_{ii'}$ to determine the distance that passengers need to transfer from flight i to flight i' for $i, i' \in N$:

$$z_{ii'} = \sum_{k \in M_i^+} \sum_{k' \in M_{i'}^+} p_{kk'} x_{ik} x_{i'k'}, \quad i, i' \in N. \quad (14)$$

Considering the minimum feature of the objective function, constraint (14) can be equivalently transformed into the following constraints:

$$z_{ii'} \geq p_{kk'} (x_{ik} + x_{i'k'} - 1), \quad i, i' \in N, k \in M_i^+, k' \in M_{i'}^+. \quad (15)$$

Although the transformation from (14) to (15) increased the number of constraints, the model with minimizing the total distance for passengers to transfer flights is transformed into a linear program, which is conducive to use optimization software such as CPLEX to get solutions:

$$\min \beta \sum_{i \in N} \sum_{i' \in N} n_{ii'} z_{ii'}. \quad (16)$$

- (2) Reformulate the objective function of idle time variance

In this part, we show how to reformulate the objective function of idle time variance. By introducing an auxiliary variable W to replace the total variance of idle time and considering the minimization objective function, a second-order cone constraint can be obtained:

$$\sqrt{\sum_{k \in M} \sum_{l \in E^+} w_{kl}^2} \leq W. \quad (17)$$

The objective function that minimizes the total variance of idle time is transformed into a linear one:

$$\min \gamma W. \quad (18)$$

In summary, the new formulation can be stated as follows:

$$\begin{aligned} (P) \min \quad & \alpha \sum_{i \in N} x_{iM^0} + \beta \sum_{i \in N} \sum_{i' \in N} n_{ii'} z_{ii'} + \gamma W \\ \text{s.t.} \quad & (2) - (13), (15), (17) \end{aligned} \quad (19)$$

The new formulation could be optimally solved by mainstream optimization solvers such as IBM ILOG CPLEX for small-sized instances.

2.4. Determine the Number of Time Slots. The number of time slots defined in the model directly affects the number of decision variables existing in the model. The well-designed number of time slots is very beneficial to improve the model's solving efficiency. In this article, the number of time slots is determined using the following heuristic algorithm (Algorithm 1).

3. Lagrangian Relaxation Algorithm

Since the GAP is NP-hard, only small instances can be optimally solved by applying optimization solvers. Moreover, integer variables and continuous variables are closely coupled with each other in some constraints of the model, which makes the model technically intractable. In this section, we design a Lagrangian relaxation algorithm to generate approximate solutions, the quality of which can be evaluated by the lower bounds provided by the algorithm. The framework of the Lagrangian relaxation algorithm is shown in Figure 2.

3.1. Copying Variables and Decomposition. Lagrangian relaxation algorithm is to construct a relaxed problem by assigning suitable Lagrangian multipliers to a set of constraints, which will be relaxed and added to the objective

function. The Lagrangian multipliers play a role in adding penalty to the solutions that violate the corresponding constraints. A lower bound for the objective value of the primary problem can be obtained by solving the relaxed problem optimally, and the dual of the relaxed problem provides optimal Lagrangian multipliers. Since the conflict among the three objectives of the gate assignment problem leads to the difficulties in solving the model, a decomposition scheme is designed by copying binary variables and relaxing the coupling constraints.

The decomposition scheme is started by introducing a set of auxiliary variables x'_{ik} ($i \in N, k \in M_i^+$) and y'_{ikl} ($i \in N, k \in M_i, l \in E$) and adding the following constraints to the model (P):

$$x_{ik} = x'_{ik}, \quad i \in N, k \in M_i^+, \quad (20)$$

$$y_{ikl} = y'_{ikl}, \quad i \in N, k \in M_i, l \in E, \quad (21)$$

$$x'_{ik} \in \{0, 1\}, \quad i \in N, k \in M_i^+, \quad (22)$$

$$y'_{ikl} \in \{0, 1\}, \quad i \in N, k \in M_i, l \in E. \quad (23)$$

Rewrite constraints (7)–(10) and (15) as follows:

$$w_{k1} \geq A_i y'_{ikl}, \quad i \in N, k \in M_i, \quad (24)$$

$$w_{kl} \geq (A_i - D_{i'}) (y'_{ikl} + y'_{i'k,l-1} - 1), \quad i, i' \in N, A_i - D_{i'} \geq B, k \in M_i \cap M_{i'}, l \in E \setminus \{1\}, \quad (25)$$

$$w_{k,|E|+1} \geq (T - D_i) \left(y'_{ikl} - \sum_{i' \in N} y'_{i'k,l+1} \right), \quad i \in N, k \in M_i, l \in E \setminus \{|E|\}, \quad (26)$$

$$w_{k,|E|+1} \geq (T - D_i) y'_{ik|E|}, \quad i \in N, k \in M_i, \quad (27)$$

$$z_{ii'} \geq p_{kk'} (x'_{ik} + x'_{i'k'} - 1), \quad i, i' \in N, k \in M_i^+, k' \in M_{i'}^+. \quad (28)$$

An equivalent problem is created with the objective function (19) and subject to constraints (2)–(6), (11)–(13), (17), and (20)–(28). The auxiliary variables copy the binary variables and decompose the model into three parts. The first one includes variables x_{ik} and y_{ikl} , the second includes variables x'_{ik} and $z_{ii'}$, and the last one includes y'_{ikl} , w_{kl} , and W . The three parts are coupled by the new constraints (20) and (21).

The Lagrangian relaxation algorithm is designed as follows. Relax the coupling constraints (20) and (21). Define Lagrangian multipliers λ_{ik} ($i \in N, k \in M_i^+$) as the dual variables of constraint (20). Define Lagrangian multipliers μ_{ikl} ($i \in N, k \in M_i, l \in E$) as the dual variables of constraint (21). The Lagrangian relaxation model (LR) is presented as follows:

$$\begin{aligned} \text{(LR)min} \quad & \alpha \sum_{i \in N} x_{iM^0} + \beta \sum_{i \in N} \sum_{i' \in N} n_{ii'} z_{ii'} + \gamma W + \sum_{i \in N} \sum_{k \in M_i^+} \lambda_{ik} (x_{ik} - x'_{ik}) + \sum_{i \in N} \sum_{k \in M_i} \sum_{l \in E} \mu_{ikl} (y_{ikl} - y'_{ikl}) \\ \text{s.t.} \quad & (2) - (6), (11) - (13), (17), (22) - (28) \end{aligned} \quad (29)$$

The model (LR) can be decomposed into three separated subproblems. We refer to the subproblems as LR1, LR2, and LR3, respectively, and denote their optimal objective

function as $z(\text{LR1}(\lambda, \mu))$, $z(\text{LR2}(\lambda))$, and $z(\text{LR3}(\mu))$, respectively, for given vectors λ and μ of Lagrangian multipliers:

Step 1. Sequence all the flights in decreasing order of their aircraft size and then in nondecreasing order of their arrival time.
 Step 2. Assign each flight in turn to a gate which is not occupied by another flight; if there is no available gate, assign the flight to the remote apron stand.
 Step 3. Count the number of flights already assigned to each gate, and set the maximum number among the gates to be the number of time slots for each gate in the model.

ALGORITHM 1

FIGURE 2: Framework of the Lagrangian relaxation algorithm.

$$\begin{aligned}
 (\text{LR1}) \min \quad & \alpha \sum_{i \in N} x_{iM^0} + \sum_{i \in N} \sum_{k \in M_i^+} \lambda_{ik} x_{ik} + \sum_{i \in N} \sum_{k \in M_i} \sum_{l \in E} \mu_{ikl} y_{ikl}, \\
 \text{s.t.} \quad & (2)-(6), (11)-(12) \\
 (\text{LR2}) \min \quad & \beta \sum_{i \in N} \sum_{i' \in N} n_{ii'} z_{ii'} - \sum_{i \in N} \sum_{k \in M_i^+} \lambda_{ik} x'_{ik}, \\
 \text{s.t.} \quad & (22), (28) \\
 (\text{LR3}) \min \quad & \gamma W - \sum_{i \in N} \sum_{k \in M_i} \sum_{l \in E} \mu_{ikl} y'_{ikl}, \\
 \text{s.t.} \quad & (13), (17), (23)-(27)
 \end{aligned} \tag{30}$$

For given Lagrangian multipliers λ and μ , LR1, LR2, and LR3 can be solved by mainstream optimization solver such as CPLEX. A lower bound of the problem can be obtained from the optimal objective function value of the relaxed

model. In the iterative process of the Lagrangian relaxation algorithm, the constraints of each subproblem are not changed at all, and changes only happen on the objective function of each subproblem due to the updated Lagrangian multipliers. Therefore, a substantial advantage of solving the subproblems by CPLEX is that the optimizer will retain the constraint feasible domain of the subproblems and continue to solve the model with different objectives in each iteration. Thus, all the cuts discovered in previous iterations will be inherited by subsequent iterations, which will reduce the computing time significantly.

3.2. Constructing Feasible Solution. The solutions obtained from the relaxed problem are usually infeasible since the relaxed constraints may not be satisfied. An easy way to get a feasible solution is to directly use the solution from the

subproblem LR1, as the value of variables x_{ik} and y_{ikl} can constitute a feasible assignment for the flights. Then, fix the value of binary variables in constraints (7)–(10) and (15) and calculate the value of $z_{ii'}$, w_{kl} , and W to minimize the objective function.

Obviously, the solution obtained by solving the subproblem LR1 has little contribution to the subproblems LR2 and LR3 as it did not utilize the information provided by subproblems LR2 and LR3. Therefore, the solution may not be a good solution for the original problem. Then, the following heuristic algorithm is developed to search for a better solution (Algorithm 2).

3.3. Updating Lagrangian Multipliers. Solving the Lagrangian dual problem (31) could obtain the optimal Lagrangian multipliers, which lead to a best lower bound for the model (P):

$$\min_{\lambda, \mu} z(\text{LR1}(\lambda, \mu)) + z(\text{LR2}(\lambda)) + z(\text{LR3}(\mu)). \quad (31)$$

The subgradient algorithm has been shown to be effective to solve Lagrangian dual problems. In the algorithm, all the subproblems should be optimally solved. By applying the subgradient algorithm to solve the Lagrangian dual problem (31), the multipliers are updated according to

$$\begin{aligned} \lambda_{ik}^{n+1} &= \lambda_{ik}^n + \sigma^n f(\lambda_{ik}^n), \\ \mu_{ikl}^{n+1} &= \mu_{ikl}^n + \tau^n g(\mu_{ikl}^n), \end{aligned} \quad (32)$$

where $f(\lambda_{ik}^n)$ and $g(\mu_{ikl}^n)$ are subgradients of the relaxed problem at λ_{ik}^n and μ_{ikl}^n , and the calculation of $f(\lambda_{ik}^n)$ and $g(\mu_{ikl}^n)$ is given by

$$\begin{aligned} f(\lambda_{ik}^n) &= x_{ik}^n - x'_{ik}, \\ g(\mu_{ikl}^n) &= y_{ikl}^n - y'_{ikl}, \end{aligned} \quad (33)$$

where x_{ik}^n , x'_{ik} , y_{ikl}^n , and y'_{ikl} are the solutions of x_{ik} , x'_{ik} , y_{ikl} , and y'_{ikl} for the (LR) at the n th iteration.

σ^n and τ^n are the step size at the n th iteration. The calculation of σ^n and τ^n is given by

$$\begin{aligned} \sigma^n &= \theta_1 \frac{L_{UB} - L_{LB}^n}{\|f(\lambda_{ik}^n)\|^2 + \|g(\mu_{ikl}^n)\|^2}, \quad 0 < \theta_1 < 1, \\ \tau^n &= \theta_2 \frac{L_{UB} - L_{LB}^n}{\|f(\lambda_{ik}^n)\|^2 + \|g(\mu_{ikl}^n)\|^2}, \quad 0 < \theta_2 < 1, \end{aligned} \quad (34)$$

where L_{UB} is the objective value of the best feasible solution found for the model (P) and L_{LB}^n is the optimal objective value of the relaxed problem (LR) in the n th iteration.

4. Computational Experiments

In order to test the performance of the formulation and the Lagrangian relaxation algorithm, we generate several random instances to simulate airport operations circumstances and carry out different computational experiments. The experiments are implemented on a PC with Intel(R) i7-

5600U 2.6 GHz CPU and 8G RAM. The algorithm is programmed by C++ language, and IBM ILOG CPLEX v12.6.1 is used to solve the mathematical models (regardless of the original problem or the subproblems of the Lagrangian relaxation algorithm).

Depending on the size of the airport, six sets of small instances and six sets of large instances are generated, respectively. Each set contains ten randomly generated instances with the same number of flights and gates, and the flight arrival time is uniformly distributed during the planning period. The planning period is set as 360 minutes (6 hours) for the small instances and 720 minutes (12 hours) for the large instances. The buffer time for consecutive flights at the gate is set as 15 minutes.

The performances of the model and algorithm are reported from two aspects:

- (1) The efficiency of the model to be solved optimally by optimization software.
- (2) The quality of solutions that the algorithm can generate for practical instances in a reasonable computation time.

4.1. Model Efficiency. This article presents various modelling strategies to improve the modelling efficiency of the gate assignment problem, and their performance is reported from the following three experiments. In the experiments, all the models are solved using CPLEX, and the efficiency of the proposed modelling strategies is evaluated by the speed (or CPU time need) for the solver to get the optimal solution compared with other models under the same solver configuration:

- (1) Linearized model versus quadratic model for minimizing the walking distance

In this experiment, the objective of minimizing variance of idle time is ignored, i.e., $\gamma = 0$. Thus, the linearized model is reduced to contain the objective function

$$\min \alpha \sum_{i \in N} x_{iM^0} + \beta \sum_{i \in N} \sum_{i' \in N} n_{ii'} z_{ii'} \quad (35)$$

and constraints (2)–(6), (11), (12), and (15). The quadratic model contains the objective function

$$\min \alpha \sum_{i \in N} x_{iM^0} + \beta \sum_{i \in N} \sum_{k \in M_i^+} \sum_{i' \in N} \sum_{k' \in M_{i'}^+} n_{ii'} p_{kk'} x_{ik} x_{i'k'}, \quad (36)$$

and constraints (2)–(6) and (11) and (12). Table 1 shows the average CPU time (avg. CPU time) and relative deviation returned from CPLEX (MIP gap) for solving each set of problems.

The results in Table 1 show that the linearized model can be optimally solved in a short time for most instances, and only 3 instances not optimally solved.

Step 1. Exchange between the apron and gates. Sequence the flights assigned to the apron by nondecreasing order with the arrival time. For each flight in the sequence, check if it can be exchanged with any flight that is assigned to the gates. If so, calculate the objective value obtained after exchanging the flights, and execute the exchange that leads to the minimum objective function; if not, continue to check the next flight assigned to the apron until all the flights have been checked.

Step 2. Exchange among the gates. Sequence the flights assigned to the gates by nondecreasing order with the arrival time. For each flight in the sequence, check if it can be exchanged with any flight that is assigned to other gates. If so, calculate the objective value obtained after exchanging the flights and execute the exchange that leads to the minimum objective function; if not, continue to check the next flight assigned to the gates until all the flights have been checked.

ALGORITHM 2

The average CPU time for CPLEX to solve the model is not more than 153.73 s. The quadratic model can only be optimally solved for a few instances within the time limit of 600 s. Most instances formulated by the quadratic model cannot obtain optimal solutions in 600 s, and the relative MIP gap is up to 36.93%.

(2) SOCP versus sum of squares model for minimizing variance of idle time

In this experiment, the objective of minimizing walking distance is ignored, i.e., $\beta=0$. Thus, the model (P) is reduced to contain the objective function

$$\min \alpha \sum_{i \in N} x_{iM^0} + \gamma W, \quad (37)$$

and constraints (2)–(13) and (17). The sum of squares model contains the objective function

$$\min \alpha \sum_{i \in N} x_{iM^0} + \gamma \sum_{k \in M} \sum_{l \in E^+} w_{kl}^2, \quad (38)$$

and constraints (2)–(13). Table 2 shows the average CPU time (avg. CPU time) and relative deviation returned from CPLEX (MIP gap) for solving each set of problems.

As shown in Table 2, solution time of each model increases quickly as the problem dimension increases, from less than 1 second to hundreds of seconds. But the SOCP model is solved faster than the sum of squares model, and the relative MIP gap for the SOCP model obtained by the CPLEX solver is within 0.42%. Most instances cannot obtain optimal solutions in the time limit using the sum of squares model.

(3) Event-based model versus sequence-based model

In order to test the efficiency of event-based continuous time modelling method, a model formulated by the sequence-based modelling method is employed to compare with the model presented in Section 2.3. Table 3 shows the average CPU time (avg. CPU time) and relative deviation returned from CPLEX (MIP gap) for solving each set of problems.

TABLE 1: Comparison between linearized model and quadratic model.

Problem	Avg. CPU time (s)			MIP gap (%)	
	$ N $	$ M $	T	Linearized	Quadratic
10	3	360	0.28	469.87 ^[8]	0.00
12	5	360	3.87	600.00 ^[10]	0.01
14	5	360	27.37	600.00 ^[10]	0.01
16	5	360	125.65 ^[11]	600.00 ^[10]	0.11
18	5	360	153.73 ^[11]	600.00 ^[10]	0.43
20	5	360	83.39 ^[11]	542.85 ^[9]	0.01

^[*]The number of instances not optimally solved in 600 s, and CPU time is set to 600 s.

The results in Table 3 show that the event-based model can be solved faster than the sequence-based model, and 8 instances are not optimally solved, much less than the 27 instances of the sequence-based model. The MIP gap of event-based model is much smaller than the sequence-based model.

In summary, the comparisons from the three experiments illustrate that the proposed modelling method is more efficient than the existing methods.

4.2. Algorithm Effectiveness. The effectiveness of the Lagrangian relaxation algorithm is evaluated by the accuracy of solutions obtained. For small instances, the objective value obtained by the Lagrangian relaxation algorithm is compared with model (P), and the CPU time is also compared. For large instances, 600 seconds time limit is set for CPLEX to solve the model, and the objective values obtained by the Lagrangian relaxation algorithm and model (P) are compared with the LR lower bound. Tables 4 and 5 show the comparison results for small and large instances, respectively.

In Table 4, the columns under “avg. rel. dev. (%)” are calculated as

$$\text{OBJ_Dev} = \frac{\text{LR}^{\text{UB}} - \text{OBJ}^{\text{CPLEX}}}{\text{OBJ}^{\text{CPLEX}}} \times 100\%, \quad (39)$$

$$\text{LR_Gap} = \frac{\text{LR}^{\text{UB}} - \text{LR}^{\text{LB}}}{\text{LR}^{\text{LB}}} \times 100\%,$$

where $\text{OBJ}^{\text{CPLEX}}$ is the optimal objective value obtained by CPLEX through solving the model (P) and LR^{UB} and LR^{LB}

TABLE 2: Comparison between SOCP and sum of squares model.

Problem			Avg. CPU time (s)		MIP gap (%)	
$ N $	$ M $	T	SOCP	Sum of squares	SOCP	Sum of squares
10	3	360	0.43	0.57	0.00	0.00
12	5	360	8.06	292.27 ^[4]	0.00	7.42
14	5	360	45.38	394.75 ^[5]	0.01	13.38
16	5	360	94.82	387.25 ^[6]	0.01	35.25
18	5	360	213.83 ^[2]	493.04 ^[8]	0.42	45.24
20	5	360	108.23 ^[1]	442.84 ^[7]	0.05	27.21

^[*]The number of instances not optimally solved in 600 s, and CPU time is set to 600 s.

TABLE 3: Comparison between event-based model and sequence-based model.

Problem			Avg. CPU time (s)		MIP gap (%)	
$ N $	$ M $	T	Event	Sequence	Event	Sequence
10	3	360	0.43	2 .48	0.00	0.00
12	5	360	38.26	163.28 ^[2]	0.00	2.95
14	5	360	176.38	365.84 ^[5]	0.01	5.23
16	5	360	270.94 ^[4]	303.93 ^[5]	4.32	3.96
18	5	360	286.27 ^[4]	534.04 ^[8]	2.18	3.87
20	5	360	162.48 ^[2]	448.95 ^[7]	0.28	2.04

^[*]The number of instances not optimally solved in 600 s, and CPU time is set to 600 s.

TABLE 4: Comparison between CPLEX and Lagrangian relaxation for small instances.

Problem			Avg. rel. dev. (%)		Avg. CPU time (s)	
$ N $	$ M $	T	OBJ_Dev	LR_Gap	CPLEX	LR
10	3	360	0.13	2.59	0.45	4.33
12	5	360	0.32	3.37	38.48	16.04
14	5	360	1.31	3.96	176.43	45.18
16	5	360	1.38	4.42	271.38 ^[4]	81.85
18	5	360	1.76	5.24	288.37 ^[4]	90.28
20	5	360	2.01	5.43	158.28 ^[2]	72.62

^[*]The number of instances not optimally solved in 600 s, and CPU time is set to 600 s.

are the upper bound and lower bound obtained from the Lagrangian relaxation algorithm.

The results in Table 4 show that the objective value obtained by the Lagrangian relaxation algorithm is close to that obtained by CPLEX since the average relative deviation between CPLEX and Lagrangian relaxation (OBJ_Dev) is up to 2.01%. Lagrangian relaxation algorithm also provides high-quality lower bounds, as the Lagrangian duality gap (LR_Gap) is up to 5.43%. Thus, we take the lower bound as an effective criterion to evaluate the performance of the algorithm for large instances. As the size of instance increases, the time for solving by CPLEX increases quickly, while Lagrangian relaxation still works in a short time.

In Table 5, the column “MIP gap” indicates the relative deviation returned from CPLEX when the model is not optimally solved.

TABLE 5: Comparison between CPLEX and Lagrangian relaxation for large instances.

Problem			Avg. rel. dev. (%)		Avg. CPU time (s)	
$ N $	$ M $	T	MIP gap	LR_Gap	CPLEX	LR
30	6	720	62.84 ^[4]	6.08	600.00	139.89
40	8	720	80.87 ^[2]	7.63	600.00	188.73
50	10	720	86.33 ^[8]	9.28	600.00	231.34
60	12	720	—	12.84	600.00	237.30
70	14	720	—	14.37	600.00	363.32
80	16	720	—	16.89	600.00	416.30

^[*]The number of instances with no feasible solution in 600 s, and relative deviation is set to 100%. —, all the instances have no feasible solution in 600 s.

The results in Table 5 show that all the large instances cannot be optimally solved by CPLEX in 600 s, most of which even have no feasible solutions. Lagrangian relaxation can still get reasonable solutions. The dual gap of the Lagrangian relaxation algorithm keeps within 17%, and solution time does not exceed the time limit.

The computational results in Tables 4 and 5 illustrated that the Lagrangian relaxation algorithm can get high-quality solutions for small instances, and it is more acceptable than CPLEX for large instances.

5. Conclusion

In this article, the GAP with multiple objectives is studied. The problem is formulated and transformed into a linear program with second-order cone constraints which can be efficiently solved by an optimization solver. An algorithm based on Lagrangian relaxation is developed to deal with large instances of the problem. Computational results shown that the proposed model can be solved more efficiently than the models formulated by other methods in literature. For small instances, optimal solutions can be obtained from the model by a commercial solver, and the Lagrangian relaxation algorithm can provide high-quality approximate solutions faster; for large instances, CPLEX does not work in the time limit, while the Lagrangian relaxation algorithm can still get reasonable solutions. Future research will be focused on online decision-making of the GAP which is more adapted to the actual needs of current air transport industry.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the grant from the National Natural Science Foundation of China (71802141) and

Humanities and Social Science Foundation of the Ministry of Education in China (18YJC630219 and 17YJA630139).

References

- [1] J. P. Braaksmas and J. H. Shortreed, "Improving airport gate usage with critical path," *Transportation Engineering Journal of ASCE*, vol. 97, no. 2, pp. 187–203, 1971.
- [2] 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.
- [3] A. Bouras, M. A. Ghaleb, U. S. Suryahatmaja, and A. M. Salem, "The airport gate assignment problem: a survey," *Scientific World Journal*, vol. 2014, Article ID 923859, 27 pages, 2014.
- [4] G. S. Daş, F. Gzara, and T. Stützle, "A review on airport gate assignment problems: single versus multi objective approaches," *Omega*, vol. 92, Article ID 102146, p. 12, 2020.
- [5] R. S. Mangoubi and D. F. X. Mathaisel, "Optimizing gate assignments at airport terminals," *Transportation Science*, vol. 19, no. 2, pp. 173–188, 1985.
- [6] S. Yan and C.-M. Chang, "A network model for gate assignment," *Journal of Advanced Transportation*, vol. 32, no. 2, pp. 176–189, 1998.
- [7] A. Haghani and M.-C. Chen, "Optimizing gate assignments at airport terminals," *Transportation Research Part A: Policy and Practice*, vol. 32, no. 6, pp. 437–454, 1998.
- [8] A. Bolat, "Procedures for providing robust gate assignments for arriving aircrafts," *European Journal of Operational Research*, vol. 120, no. 1, pp. 63–80, 2000.
- [9] 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.
- [10] S. Yan and C.-M. Huo, "Optimization of multiple objective gate assignments," *Transportation Research Part A: Policy and Practice*, vol. 35, no. 5, pp. 413–432, 2001.
- [11] A. Lim, B. Rodrigues, and Y. Zhu, "Airport gate scheduling with time windows," *Artificial Intelligence Review*, vol. 24, no. 1, pp. 5–31, 2005.
- [12] 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.
- [13] U. Dorndorf, F. Jaehn, C. Lin, H. Ma, and E. Pesch, "Disruption management in flight gate scheduling," *Statistica Neerlandica*, vol. 61, no. 1, pp. 92–114, 2007.
- [14] A. Drexler and Y. Nikulin, "Multicriteria airport gate assignment and Pareto simulated annealing," *IIE Transactions*, vol. 40, no. 4, pp. 385–397, 2008.
- [15] U. Dorndorf, F. Jaehn, and E. Pesch, "Modelling robust flight-gate scheduling as a clique partitioning problem," *Transportation Science*, vol. 42, no. 3, pp. 292–301, 2008.
- [16] U. Dorndorf, F. Jaehn, and E. Pesch, "Flight gate scheduling with respect to a reference schedule," *Annals of Operations Research*, vol. 194, no. 1, pp. 177–187, 2012.
- [17] B. Maharjan and T. I. Matis, "Multi-commodity flow network model of the flight gate assignment problem," *Computers & Industrial Engineering*, vol. 63, no. 4, pp. 1135–1144, 2012.
- [18] M. Seker and N. Noyan, "Stochastic optimization models for the airport gate assignment problem," *Transportation Research Part E-Logistics and Transportation Review*, vol. 48, no. 2, pp. 438–459, 2012.
- [19] G. Diepen, J. M. van den Akker, J. A. Hoogeveen, and J. W. Smeltink, "Finding a robust assignment of flights to gates at Amsterdam Airport Schiphol," *Journal of Scheduling*, vol. 15, no. 6, pp. 703–715, 2012.
- [20] H. Wang, Y. Luo, and Z. Shi, "Real-time gate reassignment based on flight delay feature in hub airport," *Mathematical Problems in Engineering*, vol. 2013, Article ID 646241, 10 pages, 2013.
- [21] S. H. Kim and E. Feron, "Impact of gate assignment on departure metering," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 699–709, 2014.
- [22] J. Guépet, R. Acuna-Agost, O. Briant, and J. P. Gayon, "Exact and heuristic approaches to the airport stand allocation problem," *European Journal of Operational Research*, vol. 246, no. 2, pp. 597–608, 2015.
- [23] C. Yu, D. Zhang, and H. Y. K. Lau, "MIP-based heuristics for solving robust gate assignment problems," *Computers & Industrial Engineering*, vol. 93, pp. 171–191, 2016.
- [24] S. Liu, W.-H. Chen, and J. Liu, "Robust assignment of airport gates with operational safety constraints," *International Journal of Automation and Computing*, vol. 13, no. 1, pp. 31–41, 2016.
- [25] U. Dorndorf, F. Jaehn, and E. Pesch, "Flight gate assignment and recovery strategies with stochastic arrival and departure times," *OR Spectrum*, vol. 39, no. 1, pp. 65–93, 2017.
- [26] M. Dell'Orco, M. Marinelli, and M. G. Altieri, "Solving the gate assignment problem through the fuzzy bee colony optimization," *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 424–438, 2017.
- [27] A. Aktel, B. Yagmahan, T. Özcan, M. M. Yenisey, and E. Sansarç, "The comparison of the metaheuristic algorithms performances on airport gate assignment problem," *Transportation Research Procedia*, vol. 22, pp. 469–478, 2017.
- [28] I. Kaliszewski, J. Miroforidis, and J. Stańczak, "The airport gate assignment problem - multi-objective optimization versus evolutionary multi-objective optimization," *Computer Science*, vol. 18, no. 1, pp. 41–52, 2017.
- [29] G. S. Daş, "New multi objective models for the gate assignment problem," *Computers & Industrial Engineering*, vol. 109, pp. 347–356, 2017.
- [30] O. R. P. V. Schaijk and H. G. Visser, "Robust flight-to-gate assignment using flight presence probabilities," *Transportation Planning & Technology*, vol. 40, no. 8, pp. 928–945, 2017.
- [31] L. Xu, C. Zhang, F. Xiao, and F. Wang, "A robust approach to airport gate assignment with a solution-dependent uncertainty budget," *Transportation Research Part B: Methodological*, vol. 105, pp. 458–478, 2017.
- [32] W. Deng, H. Zhao, X. Yang, D. Li, Y. Li, and J. Liu, "Research on a robust multi-objective optimization model of gate assignment for hub airport," *Transportation Letters*, vol. 10, no. 4, pp. 229–241, 2018.
- [33] C. A. Floudas and X. Lin, "Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review," *Computers & Chemical Engineering*, vol. 28, no. 11, pp. 2109–2129, 2004.
- [34] F. Alizadeh and D. Goldfarb, "Second-order cone programming," *Mathematical Programming*, vol. 95, no. 1, pp. 3–51, 2003.