Exact and heuristic solution approaches for the airport gate assignment problem ☆

Özlem Karsu [a,*], Meral Azizoğlu [b], Kerem Alanlı [b]

[a] *Department of Industrial Engineering, Bilkent University, Ankara 06800, Turkey*
[b] *Department of Industrial Engineering, Middle East Technical University, Ankara 06800, Turkey*

ARTICLE INFO

ABSTRACT

In this study, we consider an airport gate assignment problem that assigns a set of aircraft to a set of gates. The aircraft that cannot be assigned to any gate are directed to an apron. We aim to make aircraft-gate assignments so as to minimize the number of aircraft assigned to apron and among the apron usage minimizing solutions, we aim to minimize total walking distance travelled by all passengers. The problem is formulated as a mixed-integer nonlinear programming model and then it is linearized. A branch and bound algorithm, beam search and filtered beam search algorithms that employ powerful lower and upper bounding mechanisms are developed. The results of the computational experiment have shown the satisfactory performance of the algorithms.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

Assignment problem deals with allocating a particular set of tasks, activities or people to a set of resources so as to maximize the utility of allocation or minimize its cost. Allocation of tasks to people at workplaces, allocation of vehicles to service areas, allocation of financial resources to government agencies (capital budgeting and planning) are some notable assignment problems encountered in real-life [31].

One application area in which assignment problems are of great relevance is airline operational planning. It is a very important as much complex area, where the managers and operation planners rely on assignment problems like crew, fleet and gate assignments to find efficient and effective ways to handle airline operations [4].

We study the airport gate assignment problem (AGAP) that considers the allocation of the aircraft, with pre-specified arrival and departure times to the gates in airports.

Due to its practical importance, several variants of the AGAP have been considered in the literature. The variants differ with respect to the objectives, constraints and preferences, which are determined based on the characteristics of the specific setting.

Major airports have fixed gates, which allow passengers to board into the aircraft using a jet bridge and an apron that is used

when the fixed gates are not sufficient. The passengers reach the apron by bus or by walking longer distances and then climb to the aircraft using steep stairs. The fixed gates are usually more convenient for the passengers; hence they are preferred over the apron.

There are two types of passengers: transfer passengers and non-transfer passengers. Transfer passengers are those who arrive at the airport by a flight and depart on another flight. Non-transfer passengers are those who either arrive at the airport from the city and depart with a flight or arrive at the airport by a flight and leave the airport afterwards.

The airports can be categorized based on the type of flights that they serve. In Turkey, some airports are only available for domestic flights, whereas airports in larger cities and touristic areas are "international airports", serving international flights as well as domestic flights. In airports with both domestic and international flights, the international and domestic terminals are separated due to their different control procedures (passport/identity check). Some metropolitan airports allow towing, i.e., transportation of the aircraft from one place to another in the airport using specialized ground vehicles. In such airports, the aircraft can change from its pre-assigned gate to another when necessary.

The air transportation traffic has been roughly doubled from the early 1980s to 2006 [14] and according to International Air Transport Association [20], average yearly increase in the number of airline passengers is 5.85% between the years 2006 and 2017. As the air traffic grows and operational environment changes throughout the years, needs of airport operators and airline companies have evolved. Thus, many different objectives have been consid-

---

ered in the AGAP. The most widely-considered ones are minimizing the number of un-gated (unassigned) aircraft, maximizing preferences/utility of assigning certain aircraft to specific gates (utility can be, for example, a function of profitability of shops), minimizing total passenger walking distance, minimizing the deviation from the original/reference schedule [42] and minimizing towing costs [14].

There are also studies that consider minimizing other metrics such as gate idle time, waiting time, total connection time, baggage transport distance, total duration of ungated flights, number of conflicts, and buffer times [2].

Metropolitan airports are usually located in large areas, so it may take a significant amount of time and effort for a passenger to travel within the airport. Therefore, minimizing total passenger walking distance is the most considered objective [2]. Total passenger walking distance includes the distances between the gates including the apron (for transfer passengers), between the gates and airport entrance and airport exit (for non-transfer passengers) and from the gates to the common areas such as luggage claim area (for all passengers). Minimizing total passenger distance traveled is a passenger oriented concern that would be achieved by proper assignment of the aircraft to the gates and apron.

Our objective is to minimize the total passenger distance traveled subject to the minimum number of aircraft assigned to the apron. In that sense, we consider a lexicographic approach for a bicriteria optimization problem, where the criteria are apron usage and passenger walking distance. Minimizing the number of aircraft assigned to an apron is an important managerial concern whose reduction contributes to the proper organization of the airline operations and reduces the walking distances of the passengers. We formulate the problem as a nonlinear mathematical model and provide its linearized version. We design a branch-and-bound algorithm and beam and filtered beam search algorithms along with powerful bounding schemes. The computational runs are performed using the layouts of Esenboğa and Atatürk Airports. To the best of our knowledge we are proposing the first implicit enumeration algorithm for the hierarchical AGAP where the primary and secondary objectives are the number of aircraft assigned to apron and the total passenger walking distance, respectively.

In Section 2, we provide a review of the related literature and clarify our contribution. We give the mathematical models of the AGAP, and give linear programs that return the minimum number of aircraft assigned to apron, in Sections 3 and 4, respectively. In Section 5, we explain our Branch and Bound, Beam Search and Filtered Beam Search algorithms. In Section 6, we report the results of our computational experiment. Section 7 concludes the paper and discusses some future research directions.

2. Literature review

The airport gate assignment problem (AGAP) is a widely studied problem in the operational research literature attributed to its practical importance (see [34] for an early work). Detailed surveys are due to Dorndorf et al. [14] and Daş et al. [11].

One can classify the AGAP variants into two categories with respect to formulation: static AGAP and stochastic & robust AGAP [9,28]. In the static AGAP, a deterministic model is formulated, typically with the objectives of minimizing waiting time, ungated flights or total walking distance. Stochastic and robust AGAP are formulated considering some stochastic aspects like flight delays and disruptions. Commonly used objectives in the stochastic AGAP are minimizing the idle time, gate conflicts and flight delays.

Another classification of the AGAP studies is made with respect to the solution methodology [9]. The solution methodologies are categorized into three: expert system approaches, exact solution approaches and heuristic approaches.

Expert systems can be defined as software systems that aim to simulate the decision-making process of human experts. The rules obtained from human knowledge are given to the software for suggesting solutions. Some studies that propose expert systems for solving the AGAP are Brazile & Swigger [7], Gosling [17], Srihari & Muthukrishnan [33] and Su & Srihari [35].

Attributed to our main focus, we emphasize the AGAP studies on exact methods.

One of the earliest studies is due to Babić et al. [3] where the average walking distance covered by the arriving and departing passengers is minimized. They assumed that there are no transfer passengers and in the flight schedule one airplane could always be assigned to an unoccupied gate. They proposed a branch-and-bound algorithm along with a lower bound that underestimates the total walking distance of the upcoming passengers.

Bihr [5] assumed fixed departure gates and formulated an assignment model to assign arriving flights to gates so as to minimize the total walking distance of the passengers.

Mangoubi & Mathaisel [24] proposed a greedy heuristic and an LP relaxation for the total walking distance minimizing AGAP. They assumed that a passenger arriving at a gate would be equally likely to board his next flight at any gate; hence used expected walking distances through a uniform distribution for transfer passengers, which simplifies the problem. They used real time data from Toronto International Airport (with 138 aircraft and 20 gates) to assess the performance of their solution approaches.

Bolat [6] proposed a branch-and-bound algorithm and a heuristic called branch-and-trim for solving the robust AGAP so as to minimize the difference between the minimum and maximum slack times. Their computational studies performed using data from Riyadh International Airport showed that the branch-and-trim heuristic significantly outperforms the currently used procedure in the number of ungated aircraft and towing operations.

Yu et al. [40] focused on the robustness issue and proposed mathematical models for the robust AGAP. Their objectives were minimizing the expected conflict time between schedules (for robustness), minimizing towing costs and minimizing the total distance traveled by transfer passengers. They developed three mathematical models: a network flow model with a quadratic objective function and two mixed integer programming (MIP) models with linearized objective functions. They also proposed heuristics and compared their performance relative to the CPLEX branch-and-cut scheme. Their experimental results showed that one MIP model is far superior to the network flow model with quadratic objective. They also showed that exact quadratic expressions are clearly better than approximate ones that use the average distance assuming a uniform distribution for the experience of transfer passengers.

Attributing to the complexity of the problem, heuristic approaches are more commonly studied than exact algorithms. Some example works proposing various heuristic and meta-heuristic approaches for the variants of the AGAP are Haghani & Chen [18], Yan & Huo [38], Yan et al. [39], Xu & Bailey [37], Ding et al. [13], Cheng et al. [9], Genç et al. [16], Şeker & Noyan [32], Marinelli et al. [25], Aktel et al. [2], Kim et al. [23], Hu & Di Paolo [19], Mokhtarimousavi et al. [26], Deng et al. [12], Daş [10], Yu et al. [41] and Jiang et al. [22].

Mangoubi & Mathaisel [24] minimized the passenger walking distance and assumed that a transfer passenger arriving at a gate would be equally likely to board his next flight at any gate. Hence they took expected walking distances of the transfer passengers from uniform distribution, which simplifies the problem. We formulate the minimum total walking distance problem as a quadratic assignment model with overlap and minimum apron assignment constraints; then linearize the quadratic term and propose a mixed integer linear programming formulation (see Pentico [31] and Loiola et al. [43] for reviews on assignment problems and

quadratic assignment problems, respectively. See also Bouras et al. [44], and the references therein, for quadratic programming formulations).

Other studies that formulate the AGAP as a quadratic assignment problem are Haghani & Chen [18], Jalalian et al. [21] and Drexl & Nikulin [45]. Haghani & Chen [18] minimized total passenger walking distance and formulated a quadratic assignment problem. They then proposed an integer linear programming reformulation and developed a heuristic algorithm. Jalalian et al. [21] and Drexl & Nikulin (2008) considered multi-objective formulations and suggested simulated annealing to find an approximate Pareto set. Jalalian et al. [21] considered CO2 emissions, cruise flight time and passenger walking distances as objectives whereas Drexl & Nikulin [45] considered total passenger walking distances, gate preferences and number of ungated flights as objectives. Another multi objective gate assignment study is due to Chao et al. [8] who considered carbon emissions for airports and carbon emission costs for airlines. We study a hierarchical AGAP where the primary and secondary objectives are minimizing the number of aircraft assigned to apron and the total passenger walking distance, respectively. We propose a branch and bound algorithm for exact solutions as well as beam search and filtered beam search algorithms for heuristic solutions.

3. Problem definition

We study an AGAP where aircraft have to be assigned to fixed gates and an apron. The aircraft are either associated to the domestic flights or international ones. There are fixed gates in domestic and international terminals. There is only one apron in the airport that serves both terminals. The apron is far from all other gates and the airport exit.

Once the aircraft arrives at the airport at the arrival time, its assigned gate is empty and it occupies the gate till its departure time. Each aircraft has a specified set of passengers, who are either transfer passengers departing from/arriving for other aircraft or non-transfer passengers that enter to and exit from the airport using the same point. The passengers may include the crew, cargo/flight shipping and maintenance workers, catering/provisioning employees who transfer to other aircraft or enter/exit from the airport.

Our primary aim is to minimize the number of aircraft assigned to apron. Among the solutions with minimum apron usage we aim to minimize the total walking distance by all passengers, which can be expressed as the sum of distances covered by non-transfer passengers (from entrance to the gate through security check and from gate to exit through baggage hall) and transfer passengers (from gate to gate via transit counters). In small airports, passenger walking is convenient; hence the passenger walking distances are the actual walking distances. However in medium and large airports, the passengers may not be allowed to walk for long distances, hence walking may mean being transported by bus, which may be more convenient to reach fixed gates and apron. The proposed model is still valid as the walking distance used may correspond to the distance covered by other means.

We show that the minimum number of aircraft to be assigned to apron is found via maximum cost network flow models that are explained in Section 4. We discourage apron assignments by imposing minimum number of aircraft assigned to apron as a constraint.

We assume that all parameters are known and not subject to any change, i.e., the system is deterministic and static.

The sets and parameters related to the aircraft are as follows:

I_D: set of all domestic aircraft; $|I_D|$: number of domestic aircraft

I_I: set of all international aircraft; $|I_I|$: number of international aircraft

I: set of all aircraft $= I_D \cup I_I$

n: number of the aircraft (i.e., the cardinality of the set I, $|I|$)

p_{ij}: number of transfer passengers between aircraft i and aircraft j ($\forall i$, $\forall j \in I$)

e_i: number of passengers coming from the entrance for aircraft i ($\forall i \in I$)

f_i: number of passengers leaving the airport from the exit after aircraft i arrives ($\forall i \in I$)

a_i: arrival time of aircraft i ($\forall i \in I$)

d_i: departure time of aircraft i ($d_i > a_i$) ($\forall i \in I$)

Aircraft i stays in the airport in interval $[a_i, d_i]$. During that interval the boarding/ unboarding of the passengers and crew, loading/unloading of the baggages, cargo, catering materials; some preparation tasks like cleaning, fueling and maintenance are performed.

$\{ad_1, ad_2, ..., ad_R\}$: set of distinct a_i and d_i values in chronological order, i.e., $ad_1 < ad_2 < ... < ad_R$, where R is the number of distinct a_i and d_i values.

$$comp_{ir} = \begin{cases} 1, & \text{if aircraft } i \text{ is in the airport} \\ & \text{at interval } [ad_r, ad_{r+1}) \quad \forall i \in I, r \in \{1, ..., R-1\} \\ 0, & \text{otherwise} \end{cases}$$

$$g(i): \begin{cases} D, & \text{if aircraft } i \text{ is domestic} \\ I, & \text{if aircraft } i \text{ is international} \end{cases} \quad \forall i \in I$$

NA^*: minimum number of aircraft assigned to apron (Found as explained in Section 4)

The sets and parameters related to gates are as follows:

K_D: set of all gates (fixed gates+ apron) at the domestic terminal

K_I: set of all gates (fixed gates + apron) at the international terminal

K: set of all gates$= K_D \cup K_I$

m: number of all fixed gates; m: $|K|$ - 1, gate $m+1$ is the apron

d_{kl}: distance between gates k and l ($\forall k, l \in K$)

ed_k: distance between gate k and airport entrance/exit point ($\forall k \in K$)

Without loss of generality, ed_k may be defined as distance between gate k and the baggage claim area + distance between the baggage claim area and the entrance/exit point

The decision variable is as follows:

$$x_{ik} = \begin{cases} 1, & \text{if aircraft } i \text{ is assigned to gate } k \ \forall (i, k) \in (I_D, K_D) \text{ and } (I_I, K_I) \\ 0, & \text{otherwise} \end{cases}$$

The nonlinear objective function is stated below:

$$Min \sum_{i=1}^{n-1} \sum_{k \in K_{g(i)}} \sum_{j=i+1}^{n} \sum_{l \in K_{g(j)}} p_{ij} d_{kl} x_{ik} x_{jl} + \sum_{i \in I_D} \sum_{k \in K_D} (e_i + f_i) \, ed_k x_{ik}$$
$$+ \sum_{i \in I_I} \sum_{k \in K_I} (e_i + f_i) \, ed_k x_{ik} \tag{O1}$$

The first term is the total distance covered by the transfer passengers that relies on two assignment decisions, making the function nonlinear. The second term is the total distance covered by the non-transfer passengers. The constraint sets are as follows:

$$\sum_{k \in K_D} x_{ik} = 1 \quad \forall i \in I_D \tag{1}$$

$$\sum_{k \in K_I} x_{ik} = 1 \quad \forall i \in I_I \tag{2}$$

$$\sum_{i=1}^{n} comp_{ir} x_{ik} \leq 1 \quad \forall k \in (K \setminus \{(m+1)\}), r = 1, \ldots, R-1 \tag{3}$$

$$\sum_{i \in I} x_{im+1} = NA^* \tag{4}$$

$$x_{ik} \in \{0, 1\} \quad \forall (i, k) \in (I_D, \ K_D) \ and \ (I_I, \ K_I) \tag{5}$$

Constraint sets (1) and (2) assign each aircraft to exactly one gate in its respective terminal. Constraint set (3) guarantees that there are no overlapping aircraft in any fixed gate. Two aircraft are said overlapping if the time intervals they spent in the airport overlap. The minimum number of aircraft assigned to apron is defined in Constraint (4). Constraint set (5) ensures that all decision variables are binary.

We now discuss the linearization of the objective function expressed in (O1). In doing so, we introduce a new decision variable as:

$$y_{ijkl} = \begin{cases} 1, & \textit{if } x_{ik} \textit{ and } x_{jl} \textit{ are both equal to } 1 \\ 0, & \textit{otherwise } \forall (i, k) \textit{ and } (j > i, l) \in (I_D, \ K_D) \textit{ and } (I_I, \ K_I) \end{cases}$$

We introduce the following constraint sets.

$$y_{ijkl} \geq x_{ik} + x_{jl} - 1 \quad \forall (i, k) \ and \ (j > i, l) \in (I_D, \ K_D) \ and \ (I_I, \ K_I) \tag{6}$$

$$y_{ijkl} \geq 0 \quad \forall (i, k) \ and \ (j > i, l) \in (I_D, \ K_D) \ and \ (I_I, \ K_I) \tag{7}$$

Note that at an optimal solution y_{ijkl} variables take value 1 only when both x_{ik} and x_{jl} are 1, as they are penalized with positive coefficients in the following linearized objective function:

$$Min \sum_{i=1}^{n-1} \sum_{k \in K_{g(i)}} \sum_{j=i+1}^{n} \sum_{l \in K_{g(j)}} p_{ij} d_{kl} y_{ijkl} + \sum_{i \in I_D} \sum_{k \in K_D} (e_i + f_i) \ ed_k x_{ik}$$
$$+ \sum_{i \in I_I} \sum_{k \in K_I} (e_i + f_i) \ ed_k x_{ik} \tag{O2}$$

The compact forms of the models are as stated:

Nonlinear AGAP Model: Min O1 subject to constraint sets (1) through (5)
Linear AGAP Model: Min O2 subject to constraint sets (1) through (7)

4. Finding the minimum number of aircraft assigned to apron

In this section we present the models that find minimum number of aircraft assigned to apron, thereby maximum number of aircraft assigned to fixed gates. We let NA_D^* and NA_I^* be the maximum number of aircraft assigned to domestic gates and international gates, respectively. The minimum number of aircraft assigned to apron, NA^* is $n - NA_D^* - NA_I^*$. NA_D^* and NA_I^* can be found independently, as domestic and international aircraft are served in different terminals.

Consider a maximum cost network flow model with nodes representing the domestic aircraft and with arcs between non-overlapping aircraft. Fig. 1 depicts this network structure.

The network has $|I_D|+1$ arcs departing from node 0, one to each domestic aircraft and one to terminal node $|I_D|+1$. There are $|I_D|+1$ arcs arriving to node $|I_D|+1$, one from each domestic aircraft and one from the source node (node 0). There is an arc between two nodes if the corresponding domestic aircraft do not overlap. All arc weights are 1.

We send $|K_D|-1$ units of flow from node 0 to node $|I_D|+1$ so as to maximize the number of arcs visited. The maximum cost value from node 0 to node $|I_D|+1$ gives the maximum number of domestic aircraft that can be feasibly assigned to $|K_D|-1$ gates.

The associated mathematical model for the domestic aircraft is as stated below.

A_D: set of arcs representing domestic aircraft on the maximum cost network

$$z_{ij} = \begin{cases} 1, & \textit{if arc } (i, j) \textit{ is selected} \\ 0, & \textit{otherwise } (i, \ j) \in A_D \end{cases}$$

$$Max \ Z_D = \sum_{(i,j) \in A_D} z_{ij}$$

$$\sum_{(0,j) \in A_D} z_{0j} \leq |K_D| - 1$$

$$\sum_{(i,|I_D|+1) \in A_D} z_{i(|I_D|+1)} \leq |K_D| - 1$$

$$\sum_{(i,j) \in A_D} z_{ij} = \sum_{(j,i) \in A_D} z_{ji} \ i = 1, \dots, \ |I_D|$$

$$\sum_{(i,j) \in A_D} z_{ij} \leq 1 \ i = 1, \dots, \ |I_D|$$

The optimal objective function value of the model, Z_D^*, is the maximum number of domestic aircraft that can be assigned to all domestic gates and $NA_D^* = |I_D| - Z_D^*$ is the minimum number of domestic aircraft on the apron.

The model for the international aircraft is the same except that $|I_D|$ is replaced by $|I_I|$; $|K_D|$ is replaced by $|K_I|$; NA_D^* is replaced by NA_I^* and A_D is replaced by A_I (set of arcs representing international aircraft).

The total unimodularity structure of the maximum cost network flow model implies that it can be solved in polynomial time by LP methods as its optimal solution and the optimal solution to its LP relaxation are identical (see [29]). This follows NA_D^* and NA_I^*, thereby NA^*, can be found in polynomial time.

5. Branch-and-bound, beam search and filtered beam search algorithms

The airport gate assignment problem is shown to be strongly NP-hard [30]. Hence, to find exact solutions, one should rely on mathematical models or implicit enumeration techniques like a Branch and Bound (B&B) algorithm. We propose a B&B algorithm for the exact solutions of medium sized instances that could not be solved by our mathematical model. We also propose Beam Search (BS) and Filtered Beam Search (FBS) algorithms for the heuristic solutions of the large sized instances.

BS and FBS algorithms are curtailed B&B algorithms that evaluate a subset of all partial solutions. As some nodes are not evaluated, the solutions are usually obtained quickly however with no guarantee of optimality. We refer the reader to Morton and Pentico [27] for the details of the BS and FBS algorithms and their generalizations.

We index the domestic (international) gates except apron in non-decreasing order of their distance to the airport entrance & exit point. Accordingly, gate 1 ($|K_D|$) is the closest domestic (international) gate and gate $m+1$, i.e., apron, is the farthest gate, to the airport entrance & exit point. The aircraft are indexed by their non-increasing order of arrival times to the airport.

The B&B tree has at most n levels and each level has at most $m+1$ nodes. Level i of the tree represents the assignment of aircraft i. Node k at level i is for the assignment of aircraft i to gate k. The nodes representing the infeasible assignments are not created. The B&B algorithm employs a depth-first strategy. At any level, it selects the node with the smallest lower bound on the total weighted walking distance value and proceeds to the succeeding levels. If all nodes at any level are evaluated or discarded we backtrack to previous level. We terminate whenever level 0 is reached.
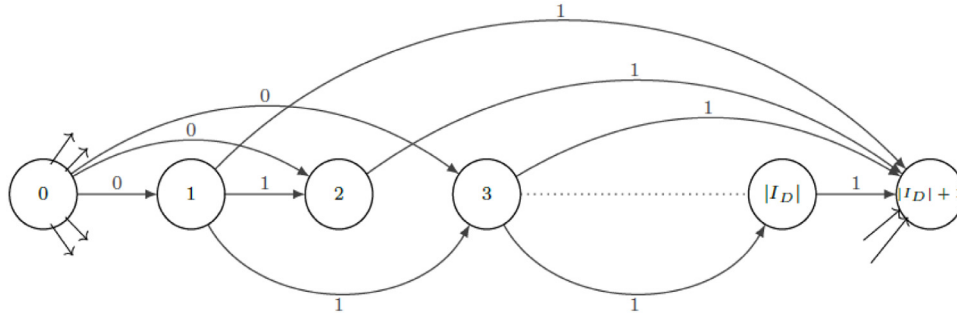
Fig. 1. Maximum cost network flow model.

The BS and FBS trees have at most n levels where level i of the tree represents the assignment of aircraft i. The BS algorithm evaluates the nodes by their lower bounds on the total distance travelled values and selects the most attractive beam width nodes among the promising ones for further branching while discarding the rest permanently. The FBS algorithm selects the most attractive nodes among the promising nodes in two steps: At first, the algorithm evaluates the nodes with respect to the realized walking distances and chooses at most filter width promising nodes for further evaluation, discarding the rest. The selected nodes are further evaluated based on the lower bounds on total distance travelled; the most attractive beam width nodes are chosen for further branching, while the rest are removed.

We call a node promising if it is not discarded by our elimination rules, if it does not make more apron assignments than the initial upper bound and if its lower bound is smaller than the initial upper bound. That is, when the initial upper bound is not apron feasible (does not make minimum number of apron assignments), we allow violation of constraint (4).

We now discuss the elimination rules, lower bounds and an initial upper bound used to evaluate the partial solutions of the B&B, BS and FBS algorithms.

5.1. Elimination rules based on properties of feasible and optimal solutions

At any partial solution, we find the set of eligible gates for each unassigned aircraft and determine the unassigned aircraft with no eligible gate other than the apron. Let:

$AA_D(AA_I)$ = the number of domestic (international) aircraft already assigned to apron + the number of yet-to-be assigned domestic (international) aircraft with no eligible gate other than the apron.

The node is fathomed by feasibility if $AA_D > NA_D^*$ or $AA_I > NA_I^*$.

We calculate a lower bound on the number of yet-to-be-assigned domestic aircraft to apron by checking the intervals defined by the chronological ordering of the arrival times.

We let

Int_r = set of not yet assigned domestic aircraft that will be at the airport in interval $[ad_r, ad_{r+1})$

m_r = number of domestic gates not yet occupied in interval $[ad_r, ad_{r+1})$

We start with $[ad_1, ad_2)$. If $|Int_1| > m_1$ then at least $|Int_1| - m_1$ aircraft should be assigned to apron. While checking $[ad_2, ad_3)$ we consider set $Int_2 \backslash Int_1$ to avoid double counting of the aircraft in Int_1 that were considered for $[ad_1, ad_2)$. If $|Int_2 \backslash Int_1| > m_2$ then at least $|Int_2 \backslash Int_1| - m_2$ more aircraft should be assigned to apron. Similarly, for $[ad_3, ad_4)$ we consider set $|Int_3 \backslash Int_2 \backslash Int_1|$

to avoid double counting of the aircraft in Int_1 and Int_2 that were considered in $[ad_1, ad_3)$. If $|Int_3 \backslash Int_2 \backslash Int_1| > m_3$ then at least $|Int_3 \backslash Int_2 \backslash Int_1| - m_3$ more aircraft should be assigned to apron. Below is the formal description of the lower bound.

$lower\ bound = 0$

for $r = 1$ to R do

If $|Int_r| > m_r$ then $lower\ bound = lower\ bound + |Int_r| - m_r$

$Int_t = Int_t \backslash Int_r$ $t = r + 1,\ r + 2,\ \ldots,\ R$

end for

The same method is applied to find a lower bound on number of yet-to-be-assigned international aircraft to apron.

We eliminate the partial solution if the domestic (international) aircraft already assigned to apron + $lower\ bound > NA_D^*$ (NA_I^*).

We eliminate some other nodes using the results of the properties of optimal solutions that are stated below.

Property 1. If domestic (international) aircraft i overlaps with less than $|K_D|$-1 domestic ($|K_I|$-1 international) aircraft then it is not assigned to apron.

Proof. Assume a solution S in which domestic (international) aircraft i is assigned to apron. Taking domestic (international) aircraft i from the apron and replacing it to one of the fixed gates, is feasible as there is at least one empty gate in interval $[a_i, d_i]$ (due to at most $|K_D|$– 1 overlaps). Such a replacement improves the objective function as the apron is the most distant gate to all other gates and exit and entrance points. Hence, a solution that assigns domestic (international) aircraft i to apron, cannot be optimal.

We do not consider a node that represents the assignment of aircraft i (that satisfies the condition of the Property 1) to apron.

Property 2. For two domestic (international) aircraft i and j, if $a_i \leq a_j, d_i \leq d_j, e_i \leq e_j, f_i \leq f_j, p_{ik} \leq p_{jk}$ for all $k \neq i,\ j$, then aircraft i is assigned to apron only when aircraft j is assigned to apron.

Proof. Assume a solution S in which domestic (international) aircraft i, but not domestic (international) aircraft j, is assigned to apron. Replacing the gates of domestic (international) aircraft i and domestic (international) aircraft j is feasible as $a_i \leq a_j$ and $d_i \leq d_j$. Such a replacement improves the objective function as $e_i \leq e_j, f_i \leq f_j,\ p_{ik} \leq p_{jk}$ for all $k \neq i,\ j$ and apron is the most distant gate to all other gates and exit and entrance points. Hence, a solution that assigns domestic (international) aircraft i, but not domestic (international) aircraft j, to apron cannot be optimal.

We use Property 2 as follows. Consider two domestic (international) aircraft i and j that satisfy the condition of Property 2, if domestic (international) aircraft i is assigned to apron then for domestic (international) aircraft j only apron should be considered.
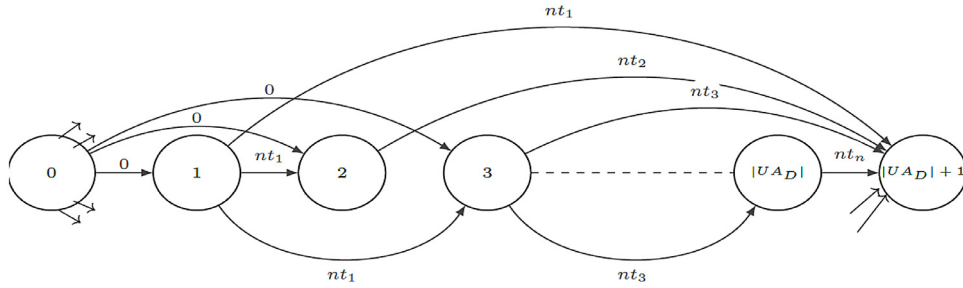
Fig. 2. Generalized network of domestic non-transfer passengers.

5.2. Lower bounds

For each partial solution, two lower bounds are calculated: one for non-transfer passengers and one for transfer passengers.

Non-Transfer Passengers Walking Distance Lower Bound: The distance to be covered by all non-transfer passengers is the number of non-transfer passengers of aircraft **i** times the distance between the gate of aircraft **i** and the airport entrance & exit point. The distance of the gate of aircraft **i** is a decision that could be underestimated by the distance of gate 1 to entrance & exit point as gate 1 is the closest gate due to our indexing. At any node s, let UA be the set of unassigned aircraft. A valid lower bound is;

$$\sum_{i \in UA} nt_i ed_1$$

where,

nt_i: number of non-transfer passengers in aircraft $i = e_i + f_i$ ($\forall i \in I$).

This lower bound is improved by the maximum number of domestic (international) non-transfer passengers that can be assigned to a domestic (international) gate. This number is found through a longest path algorithm on a network with nodes representing the aircraft and with arcs between non-overlapping aircraft.

We explain the lower bound calculations for domestic non-transfer passengers (LB_{NTD}). For international ones, we perform the same type of calculations. Let UA_D be the set of all unassigned domestic aircraft.

The arc weights in the network represent the number of non-transfer passengers where nt_i is the weight of the arc from aircraft $i \in UA_D$ to aircraft $j \in UA_D$. Fig. 2 depicts the longest path network for any domestic gate.

The above network has $|UA_D|+1$ arcs departing from node 0, one to each (unassigned) aircraft and one to terminal node $|UA_D| + 1$. There are $|UA_D|+1$ arcs arriving to node ($|UA_D| + 1$), each from one aircraft and one from the source node (node 0). The longest path between node 0 and node ($|UA_D|+1$) gives the maximum number of passengers that can be feasibly assigned (without any overlaps) to any one of the gates.

The length of the longest path L_D gives the maximum number of domestic non-transfer passengers (whose aircraft is not assigned to a gate yet) that can be served by one gate.

To find L_D we solve the longest path problem using the reaching algorithm (see [1]) proposed for acyclic networks.

After L_D is found we let $m^* : m^* L_D \leq \sum_{i \in UA_D} nt_i$ and $(m^* + 1)L_D > \sum_{i \in UA_D} nt_i$.

As each gate can serve at most L_D non-transfer passengers, there should be at least m^* gates, to serve all non-transfer passengers. To guarantee a lower bound on the total walking distance of those passengers, we take the first m^* domestic gates, hence the closest gates to the entrance & exit point. Starting from the first gate, L_D passengers are matched with each of the ($m^* - 1$) domes-

tic gates and the remaining non-transfer passengers are assigned to domestic gate m^*. This provides the following lower bound for the total walking distance of non-transfer domestic passengers.

$$LB_{NTD} = \sum_{k=1}^{m^*-1} L_D ed_k + \left(\sum_{i \in UA_D} nt_i - (m^* - 1)L_D \right) ed_{m^*}$$

Recall that $AA_D=$ the number of domestic aircraft already assigned to apron + the number of yet-to-be assigned domestic aircraft with no eligible gate other than the apron.

Max $\{0, NA_D^* - AA_D\}$ is the number of domestic aircraft that should be added to apron. This additional assignment produces a minimum walking distance of $(ed_{m+1} - ed_{m^*})$ per non-transfer passenger and there are at least $\sum_{i=1}^{NA_D^* - AA_D} nt_{[i]}$ ([i] is the index for the ith smallest nt_i value among the not-yet-assigned domestic aircraft) non-transfer domestic passengers. Hence to strengthen LB_{NTD} the following term is added, if $NA_D^* > AA_D$:

$$\sum_{i=1}^{NA_D^* - AA_D} nt_{[i]} (ed_{m+1} - ed_{m^*}).$$

LB_{NTI} (International part) is calculated and strengthened similarly.

LB_{NT} is found for domestic and international aircraft separately and their sum is taken.

The realized walking distance is $RC_{NT} = \sum_{i=1}^{s} \sum_{k \in K_{g(i)}} nt_i ed_k x_{ik}$

The overall lower bound for non-transfer passengers walking distance is $LB_{NT} + RC_{NT}$

Transfer Passengers Walking Distance Lower Bound 1 (LB_{T1}): At any level of the branch-and-bound tree, LB_{T1} is calculated, considering the assigned and not-yet-assigned aircraft. For level $s \in I$, three cases exist:

Case 1 Aircraft i, $j \leq s$ are both assigned. The realized travel distance of transfer passengers, RC_T, is;

$$\sum_{i=1}^{s-1} \sum_{k \in K_{g(i)}} \sum_{j=i+1}^{s} \sum_{l \in K_{g(j)}} p_{ij} d_{kl} x_{ik} x_{jl}$$

For other cases, we find eligible gates for each unassigned aircraft i. A gate is said to be eligible for aircraft i if none of its overlapping aircraft have yet been assigned to that gate.

We let $eg(i)$ be the set of eligible gates for aircraft i. At the root node when all the gates are empty, $eg(i)$ is the set of all gates of the same type as aircraft i. Once an overlapping aircraft with aircraft i is assigned to gate k, $eg(i)$ is updated as $\{eg(i) \setminus k\}$.

Case 2 Aircraft $i \leq s$, $j > s$, i.e., i is assigned, j is not yet assigned.

Let gt_i be the gate that aircraft i is assigned. Two sub-cases exist:

Case 2.1 Aircraft i and aircraft j are non-overlapping.

The minimum distance is zero if aircraft i and j are of the same type; i.e., both are domestic or international; otherwise it is the minimum distance between gt_i and gates of the other type. The lower bound on the realized weighted distance is as follows:

$$LB_{Tij} = \begin{cases} p_{ij} \times \min_{k \in eg(j)} \left\{ d_{gt_i, k} \right\} & \text{if } i \text{ and } j \text{ are of different type} \\ 0, & \text{otherwise} \end{cases}$$

Case 2.2 Aircraft i and aircraft j are overlapping

Overlapping aircraft i and aircraft j should be assigned to different gates. The minimum travel distance is $\min\limits_{\substack{k \neq gt_i \\ k \in eg(j)}} \{d_{gt_i, k}\}$.

The lower bound on the realized weighted distance is
$$LB_{Tij} = p_{ij} \times \min_{\substack{k \neq gt_i \\ k \in eg(j)}} \{d_{gt_i, k}\}.$$

Case 3. Aircraft $i > s$, $j > s$, i.e., i and j are both unassigned.

Note that Case 1 returns the realized walking distance. Cases 2 and 3 return lower bounds using realized and not-yet-realized assignments, respectively. Therefore, the overall lower bound $LB_{T1} = \sum\limits_{i=1}^{n-1} \sum\limits_{j=i+1}^{n} LB_{Tij}$ is the sum of Case 2 and Case 3 lower bounds.

$$LB_{Tij} = p_{ij} \times \min_{\substack{k \in eg(i) \\ l \in eg(j)}} \{d_{kl}\}$$

We further strengthen the lower bound on the transfer passengers so as to include the movements from fixed gates to apron. In doing so, we first find a lower bound on the number of transfers to be realized to apron as follows:

Let $\alpha =$ the number of aircraft at fixed gates that has transfer passengers to the aircraft at apron, i.e., number of transfers from fixed gates to apron

$LB(\alpha) =$ a lower bound on the α value

$LB(\alpha) = \sum\limits_{i=1}^{NA^*} (h_i - i + 1)$, where

h_i is the number of aircraft that aircraft i has transfer passengers with.

We index h_i values as $h_1 \leq h_2 \leq \ldots \leq h_n$ and find a lower bound on the number of transfer passengers for any $LB(\alpha)$ transfers as follows:

Let $p_{[t]} = t$th smallest p_{ij} value over all aircraft pairs (i and j). $\sum\limits_{t=1}^{LB(\alpha)} p_{[t]}$ is a lower bound on the number of transfer passengers to apron.

$P =$ number of transfer passengers who bear the apron distance so far.

If $\sum\limits_{t=1}^{LB(\alpha)} p_{[t]} > P$ then at least $(\sum\limits_{t=1}^{LB(\alpha)} p_{[t]} - P)$ more passengers should be assigned to apron and the following term is added to LB_{T1}.

$$\left(\sum_{t=1}^{LB(\alpha)} p_{[t]} - P \right) \times (D - max_{k,l \in K/\{m+1\}} \{d_{kl}\})$$

where D is the distance between the apron and its closest gate.

Transfer Passengers Walking Distance Lower Bound 2 (LB_{T2}): We define another lower bound, LB_{T2}, which can be used as an alternative to LB_{T1}.

We use the following notation.

$M_D (M_I) =$ the maximum number of domestic (international) aircraft that can be served by a domestic (international) gate in the set of not-yet-assigned aircraft

$DO_i (IO_i) =$ the set of not-yet-assigned domestic (international) aircraft overlapping with aircraft i that is kept in nonincreasing order of p_{ij} values.

$DNO_i (INO_i) =$ the set of not-yet-assigned domestic (international) aircraft nonoverlapping with aircraft i that is kept in nonincreasing order of p_{ij} values.

For each aircraft i, we calculate the contribution to LB_{T2}, denoted as LB_{Ti}. LB_{T2} is then calculated as $LB_{T2} = \sum\limits_{i \in I} LB_{Ti}$.

The following pseudo code shows the LB_{Ti} calculations for domestic aircraft i when assigned to any gate in the partial solution.

```
1: DD: the set of ordered distances of domestic gates from the assigned gate
   of aircraft i (including itself).
2: DI: the ordered distances of international gates from the assigned gate of
   aircraft i;
3:  LB_Ti = 0
4: next=2, count=0.
5: for j ∈ DO_i do
6:   LB_Ti = LB_Ti + p_ij DD_next;
7:   count++;
8:   if count = M_D and DD_next < d_{m+1} then
9:     next++;
10:  end if
11: end for
12: next=1, count=0.
13: for j ∈ DNO_i do
14:   LB_Ti = LB_Ti + p_ij DD_next;
15:   count++;
16:   if count = M_D and DD_next < d_{m+1} then
17:     next++;
18:  end if
19: end for
20: next=1, count=0.
21: for j ∈ IO_i ∪ INO_i do
22:   LB_Ti = LB_Ti + p_ij DI_next;
23:   count++;
24:   if count = M_I and DI_next < d_{m+1} then
25:     next++;
26:  end if
27: end for
```

For domestic aircraft i that has been assigned to a gate, we first consider the not-yet-assigned domestic aircraft overlapping with aircraft i (DO_i) (Line 5). The aircraft in DO_i cannot share the same gate with aircraft i, so we assign their passengers, starting from the closest domestic gate to the gate of aircraft i. To ensure a lower bound on the transfer walking distance we use a nonincreasing order of the number of the passengers. In set DD, the domestic gates are ordered by their distances to the gate of aircraft i, the first element being the gate itself. Hence for overlapping aircraft in set DO_i, we start from its second element, DD_2. This is demonstrated as setting the parameter *next* as 2 in Line 4, using DD_{next} in Line 5 and M_D while making the assignments. If the number of assignments to the closest gate (*count*) exceeds M_D, we consider the next closest domestic gate and so on (Lines 8-10). Note that if the considered distance is the apron distance (d_{m+1}), this increase is not made (See Line 8).

For the set of not-yet-assigned domestic aircraft that do not overlap with aircraft i (DNO_i), we perform the same computations (Lines 12-19), but now considering the possibility of assigning them to the gate of aircraft i, hence setting *next* as 1 at Line 12. Finally, we make the calculations for the international aircraft, this time considering the distances between the gate of aircraft i and the set of international gates (Lines 21-27).

For an unassigned domestic aircraft, the procedure is the same. The only difference is that the distance sets DD and DI are not defined for a specific gate; but correspond to the ordered sets of the distance levels between two domestic gates and between a domestic and an international gate, respectively.

For any international aircraft a similar procedure is applied. However, the aircraft and distance sets are changed accordingly.

Specifically, we replace DO_i, DNO_i with IO_i, and INO_i, and vice versa. M_D is replaced with M_I and vice versa. Finally, the distance set DD is replaced with II. The distance sets are determined based on whether the aircraft is already assigned or not, as in domestic aircraft case.

We strengthen this bound in the same way as in Transfer Passengers Walking Distance Lower Bound 1.

Now we discuss the use of our lower bounds in evaluating the partial solutions.

Any partial solution cannot lead to an optimal solution if

$$LB_{T1} + RC_T + LB_{NT} + RC_{NT} > U$$

and cannot lead to a unique optimal solution if

$$LB_{T1} + RC_T + LB_{NT} + RC_{NT} \geq UB$$

where UB: the best known objective function value

Note that since LB_{T1} and LB_{T2} are alternative lower bounds for walking distance of transit passengers, once LB_{T2} is used, LB_{T1} is replaced with LB_{T2} in the above equations.

We apply the lower bounds in a sequel. We first find the realized travel distances and fathom the partial solution if the following condition holds:

$$RC_T + RC_{NT} \geq UB$$

If not, we calculate LB_{NT} and check whether the following condition holds:

$$RC_T + RC_{NT} + LB_{NT} \geq UB$$

If not, we compute the first lower bound for the transfer passengers walking distance (LB_{T1}) and check whether the following holds:

$$LB_{T1} + RC_T + LB_{NT} + RC_{NT} \geq UB$$

if the lower bound with LB_{T1} does not eliminate the node, then LB_{T2} is calculated and used by checking the following:

$$LB_{T2} + RC_T + LB_{NT} + RC_{NT} \geq UB$$

Note that we are proceeding from easier to compute bound to a more difficult one to increase the speed of elimination.

If all nodes at a level are eliminated by UB, we backtrack to the previous level.

The initial upper bound is found by a heuristic procedure discussed next and it is updated whenever a complete assignment with a better objective function value is reached. The upper bound solution at termination is the optimal solution.

Initial Upper Bound (UB) Procedure:

We make assignments to one gate at a time starting with gate 1, using a longest path algorithm. The longest path structure is the same as the one defined for lower bound for non-transfer passengers' walking distance (see Fig. 2); but the arc weights are different. The arc weight between aircraft i and j is $p_{ij} + nt_i$, i.e. the number of transfer passengers between these two aircraft plus the number of non-transfer passengers arriving at and departing by aircraft i. Arc weight between the root node and aircraft i is zero and arc weight between aircraft i and the dummy end node is the number of non-transfer passengers arriving at and departing by aircraft i (nt_i).

On this network, the length of the longest path from node 0 to the terminal node gives the maximum number of transfer passengers whose arrival and departure aircraft are assigned to the same gate once the non-transfer passengers are ignored. We also count the number of non-transfer passengers as their travel distance should also be shortened.

The first problem considers all aircraft and solves a longest path problem for gate 1. The aircraft appearing on the longest path are assigned to gate 1. After these aircraft are assigned to gate 1, we

update the aircraft set and solve a longest path problem for gate 2, this time only considering the remaining aircraft. Recall that gate 2 is the second closest gate to entrance & exit point and should receive higher priority for assignment due to non-transfer passengers. We continue for gate 3 and so on, till all gates are scheduled, or all aircraft are assigned; i.e., the assignment schedule is complete.

The upper bounds for the domestic and international terminal gate assignments are found separately. The sum of these upper bounds is used to start the branch-and-bound algorithm. Below is the stepwise description of the initial upper bound heuristic for the domestic aircraft. The modifications for the international aircraft are direct.

Step 0. Set $N = I_D$ (set of domestic aircraft) and $k = 1$.
Step 1. Solve the longest path problem for gate k with set N with arc weights
  $nt_i + p_{ij}$ between aircraft i and j,
  0 between node 0 and aircraft i,
  nt_i between aircraft i and node $|N|+1$.
Step 2. Let S_k be the set of aircraft appearing on the longest path.
Set $N = N \setminus S_k$.
If $N = \emptyset$ or $k = |K_D| - 1$ then go to Step 3.
Set $k = k + 1$ and go to Step 1.
Step 3. If $k = |K_D| - 1$ and $N \neq \emptyset$ then $k = k + 1$ and $S_k = N$
Upper bound schedule is formed by $\cup_{j=1,\ldots,k} S_j$
UB is the total weighted distance value of the upper bound schedule.

In our algorithms, we start with an initial upper bound if it satisfies the minimum number of apron assignments constraint. Otherwise the initial upper bound is set to a very large number.

At termination, the B&B algorithm returns an optimal solution. On the other hand, the BS and FBS algorithms return at most beam width number of complete solutions some of which may violate minimum number of apron assignments constraint.

The generalization of our model may include additional aircraft requirements some of which are stated below:

- Gate eligibility: some aircraft cannot be assigned to some gates
- Aircraft compatibility: some aircraft may not be assigned to neighbor gates due to dimension concerns
- Airline balancing: the number of aircraft assigned to apron from each airline should not exceed a specified value

Once any constraint representing a special aircraft requirement is included into our minimum apron usage network flow model, its total unimodularity nature would dispel. The resulting network flow model becomes an integer program whose optimal solution would require an exponential effort. Hence, for the cases having special aircraft requirements, in place of using the network flow model to find the minimum apron usage value and then find the minimum total weighted distance, our mathematical model and solution procedures could be modified to deliver the minimum apron usage value and the total weighted distance solution among the ones having the minimum apron usage value, simultaneously. The modified solution procedures would require developing lower bounds on the apron usage values as their optimal values would be no more available.

6. Computational results

In this section, we discuss our data generation scheme and the results of our computational experiments. We first investigate the effect of the proposed lower bound on the performance of the mathematical model and demonstrate the performance of the B&B algorithm relative to solving the mathematical model by the CPLEX MIP solver, on smaller-sized-instances. We then provide further results on the performance of our heuristic procedures, the upper bound and the beam search (BS) and filtered beam search (FBS) algorithms, on problems of larger size.
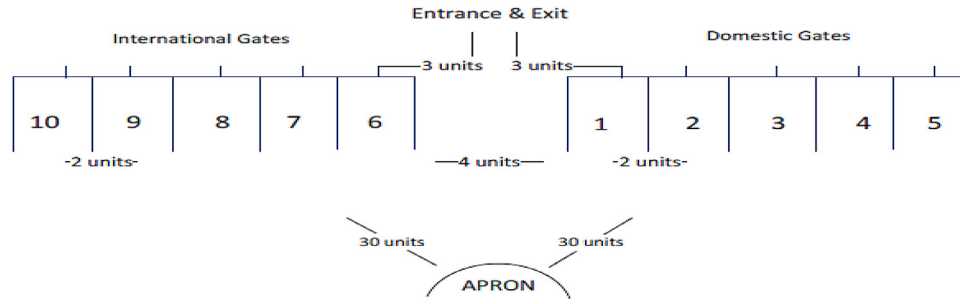
Fig. 3. The layout structure for 10 gates.

Table 1
Results of the exact approaches.

				Model without LB			Model with LB			B&B Algorithm		
				CPU Time			CPU Time			CPU Time		
m	*n*	TAD	TAI	Avg	Max	NotOpt	Avg	Max	NotOpt	Avg	Max	NotOpt
Set 1	8	15	0	0	1.04	2.19	0	1.36	2.57	0	0.03	0.13	0
		20	0	0	22.81	83.63	0	72.29	558.12	0	2.14	6.66	0
		25	1	0	185.37	418.95	0	166.83	371.18	0	392.25	3600	1
	10	15	0	0	2.78	4.69	0	4.10	12.96	0	0.12	0.51	0
		20	0	0	116.44	660.20	0	86.32	279.55	0	7.86	36.68	0
		25	1	1	1269.24	3600	2	1115.41	2389.82	0	608.40	3600	1
	12	15	0	0	2.13	3.28	0	2.55	4.03	0	0.26	1.42	0
		20	0	0	94.30	354.51	0	96.53	192.72	0	9.56	58.06	0
		25	0	0	2076.46	3600	4	3328.15	3600	8	890.13	3600	2
Set 2	8	15	16	13	4.33	8.14	0	3.97	14.61	0	0.79	1.42	0
		20	29	34	87.51	169.75	0	21.93	32.10	0	33.82	116.11	0
		25	57	48	245.42	819.61	0	71.07	162.39	0	294.09	608.24	0
	10	15	9	7	133.26	1065.68	0	18.27	32.92	0	4.72	19.58	0
		20	26	21	750.42	1484.73	0	200.81	333.13	0	653.84	3377.47	0
	12	15	2	2	88.22	321.68	0	410.83	3600	1	18.55	90.14	0
		20	10	12	2869.19	3600	5	1777.66	3309.39	0	1360.86	3600	1

Fig. 3 shows the layout structure used in the first experiment set, for an instance with 10 gates, whose distance data are provided in Table 1. These distances are based on the discussion in Ersan [15] that uses Atatürk Airport layout as its reference. The apron is taken too far from all fixed gates; hence a larger number is assigned to each associated distance. We use Manhattan distance to set the distances of the gates (except the apron) to the entrance/exit point and take $\{ed_k\} = \{3, 5, 7, 9, 11, 5, 7, 9, 11\}$ for 10-gate case.

We generate additional instances with 8 and 12 gates. For $m=8$, the layout in Fig. 3 is used by removing gates 5 and 10 and for $m=12$, adding one domestic and one international gate next to gates 5 and 10 in Fig. 3. The distance data are generated accordingly.

In the first part of the experiments on smaller-sized instances, we consider three values for the number of fixed gates, m: 8, 10, 12. For each m value, we generate instances by varying the number of aircraft, n, starting from 15 in increments of 5. We generate 10 problem instances for each (m,n) combination.

Other parameters are generated from discrete uniform distributions (DU) as:

- Number of transfer passengers between aircraft i and j (p_{ij}) ~ $DU(0, 200/n)$.
- Number of non-transfer passengers in aircraft i (nt_i) ~ $DU(0, 100)$.

The maximum number of passengers in an aircraft is taken as 300 based on information reported on the website of a commercial airline company [36]. We assume that the total number of transfer passengers does not exceed 200 and that a transfer passenger has an equal connection probability to all other aircraft. We assume

that maximum 50 passengers go from an aircraft to airport exit and maximum 50 passengers arrive at airport from outside for the same aircraft. The total number of non-transfer passengers in an aircraft is generated from $DU(0, 100)$.

We generate the following two sets of the arrival times in minutes (a_i) and durations of stay of the aircraft at the airport in minutes (dur_i); representing low and high apron requirement instances.

Set 1: $a_i \sim DU(0, 300)$ $dur_i \sim 30 + DU(0, 30)$.
Set 2: $a_i \sim DU(0, 150)$. $dur_i \sim 60 + DU(0, 60)$.

The departure time of an aircraft i in minutes (d_i) is set to $a_i+ dur_i$.

The B&B, BS and FBS algorithms are coded in C++ and solved by a dual-core (Intel Core i7 2.70 GHz) computer with 8 GB RAM. All models are solved by CPLEX 12.7.0. The solution times are expressed in Central Processing Unit (CPU) seconds.

6.1. Results for the exact approaches

We now discuss the performance of the mathematical model (when solved by CPLEX MIP solver) and B&B algorithm; and summarize the results in Table 1. To see the extent that the proposed bounds can speed the MIP solver implementation, we calculate our proposed lower bound at the root node (LB) and solve the Linear AGAP Model by the CPLEX MIP solver with and without constraint O2≥LB.

For each set and (m, n) combination, we report the total numbers of domestic and international assignments to apron over all the ten instances, in TAD and TAI columns. We also report the average and maximum solution times for all methods: the mathe-

Table 2
Results of the BS and FBS algorithms – Set 1 (all instances are found apron feasible).

		BS		FBS		
m	n	Avg CPU Time	Avg WD Dev. (All)	Avg CPU Time	Avg WD Dev. (All)	OptKnown
8	15	0.13	0	0.11	0.82	10
	20	0.18	0.11	0.10	1.55	10
	25	0.26	0.06	0.10	1.29	10
10	15	0.14	0	0.10	0.75	10
	20	0.25	0	0.10	0.45	10
	25	0.47	0.27	0.11	1.11	10
12	15	0.17	0.07	0.11	0.54	10
	20	0.39	0	0.12	0.81	10
	25	0.79	0.01	0.12	1.71	8

matical model (without LB); the variant with the additional constraint (with LB) and B&B algorithm. We set a time limit of 1 hour for both mathematical model and B&B algorithm. The tables also include the number of instances that could not be solved optimally within 1 hour (in NotOpt columns).

For Set 1 instances, where the apron usage is low, the B&B algorithm has notably lower solution times than the mathematical model without LB for most (m,n) combinations and solves more of the larger-sized instances to optimality. For relatively small-sized and medium-sized instances $((m,n) = (8,15), (8, 20), (10, 15), (12, 15))$ both approaches use negligible time, while the B&B algorithm always being faster. For medium-sized instances where $(m,n) = (10, 20)$ and $(12, 20)$ the average solution time of the B&B algorithm is less than one tenth of the solution time of the mathematical model without LB. As expected, for fixed m, the solution times increase as n increases and both approaches fail to return solutions in one hour.

For Set 2 instances, where the apron usage is significantly high, the B&B algorithm still outperforms the mathematical model without LB in most instances. It is also seen that the mathematical model with LB has notably better performance than the model without LB in most combinations, demonstrating the value of the proposed lower bound. In Set 1 $(m,n) = (10, 25)$ instances the model with LB returns the optimal solutions for all instances, while the model without LB fails to do so in 2 of the 10 instances. A similar effect is observed in Set 2 $(m,n) = (12, 20)$ instances. Note that there are also instances where the LB increases the solution time of the mathematical model. For example, the mathematical model with LB fails to terminate in one hour for almost all $(m,n) = (12, 25)$ instances of Set 1. For this set, B&B algorithm is the best performer, returning the optimal solution in 8 instances. Overall, the results show that the B&B algorithm outperforms the mathematical model without LB in most problem instances in terms of the solution time and has comparable performance to the mathematical model with LB.

6.2. Results for the heuristic approaches

In this section we report on the performances of the BS and FBS algorithms and the upper bound (UB). For initial observations on the performance of the algorithms, we use Set 1 and Set 2, since the optimal solutions are known for most instances of these sets. Tables 2 and 3 summarize the performance results on Set 1.

For each (m,n) combination, in the BS algorithm we set beam width to $(m \times n)$. In the FBS algorithm we set filter width to $(m \times n)/20$ and beam width to filter width/4. In Table 2, we report the average solution times. We also report the deviation of total walking distance from its optimal value as a percentage of the optimal, over the instances with known optimal solutions (found using the B&B algorithm or the CPLEX MIP solver within 1 hour). In OptKnown column we report the number of instances with known optimal solutions. We observe that for all Set 1 instances, the BS

and FBS algorithms both return solutions with minimum number of aircraft assigned to apron in negligible time and these solutions slightly deviate from the optimal walking distance. We also observe that the BS algorithm provides shorter total walking distances.

Our upper bound used at the root node of the B&B algorithm delivers an implementable feasible solution. In Table 3, we report comparative results for the BS, FBS and UB solutions. For each (m,n) combination, we report the average solution times (calculated over all instances), and the deviation of the total walking distance from its optimal value. Note that the solutions may not guarantee the minimum number of apron assignments. To make a fair comparison, we consider the walking distance deviations only for the instances with known optimal solutions and in which the UB, BS and FBS solutions have the same number of apron assignments. The number of such instances is given in the column named ApronSame and OptKnown. Note from the table that all average UB deviations are very low (below 3%). In most instances the BS and FBS algorithms improve over the upper bound solutions, resulting in even better solution quality.

We also investigate the quality of the BS, FBS and UB solutions with respect to the level of apron usage. In Set 1, the BS and FBS algorithms achieve minimum apron usage over all instances while the UB returns one apron infeasible solution with $(m,n) = (8, 25)$. Nevertheless, the infeasible solution assigns only one more aircraft to the apron.

Tables 4, 5 and 6 summarize the results on Set 2 instances, i.e., the instances with high apron requirements. In Table 4, we report the average solution times and deviations from the optimal walking distance, calculated over all instances[1] and over apron feasible instances. The results show that both algorithms provide very high quality solutions in negligible time.

Table 5 compares the BS, FBS and UB solutions with respect to the solution time and total walking distance, reporting the average walking distance deviation from the optimal, over the instances having the same number of apron assignments. Similar to the results for Set 1, it is seen that all algorithms return solutions with very low walking distance deviation while the BS and FBS algorithms improve the UB solution in negligible time. The solution times are very low for all small-sized instances.

Table 6 compares the BS, FBS and UB solutions for the level of apron usage. We report the number of instances for which minimum apron usage is achieved (#Apron Feasible) and average difference from the minimum number of apron assignments over all instances (Apron Dif. (All)), and over the apron infeasible instances, for the BS, FBS and UB solutions. The results show that the BS and FBS algorithms find more solutions with minimum apron usage compared to UB.

---

[1] In this set, the optimal solutions of all instances are known.

Table 3
Results of the BS and FBS algorithms and UB – Set 1.

		BS		FBS		UB		
m	n	Avg CPU Time	Avg WD Dev.	Avg CPU Time	Avg WD Dev.	Avg CPU Time	Avg WD Dev.	Apron Same and OptKnown
8	15	0.13	0	0.11	0.82	0.11	0.86	10
	20	0.18	0.11	0.10	1.55	0.10	2.70	10
	25	0.26	0.05	0.10	1.01	0.10	1.83	9
10	15	0.14	0	0.10	0.75	0.10	1.17	10
	20	0.25	0	0.10	0.45	0.10	1.69	10
	25	0.47	0.27	0.11	1.11	0.10	2.04	10
12	15	0.17	0.07	0.11	0.54	0.11	1.05	10
	20	0.39	0	0.12	0.81	0.12	2.16	10
	25	0.79	0.01	0.12	1.71	0.11	2.67	8

Table 4
Results of the BS and FBS algorithms – Set 2 (See Table 6 for Apron feasible instances).

		BS			FBS			
m	n	Avg CPU Time	Avg WD Dev. (All)	Avg WD Dev. (Apron Feas.)	Avg CPU Time	Avg WD Dev. (All)	Avg WD Dev. (Apron Feas.)	OptKnown
8	15	0.12	3.17	3.17	0.10	4.27	4.27	10
	20	0.16	2.48	1.48	0.10	2.92	2.11	10
	25	0.23	2.42	1.58	0.10	2.95	0.53	10
10	15	0.14	1.25	1.25	0.10	2.41	2.41	10
	20	0.23	2.75	2.73	0.10	6.10	6.32	10
12	15	0.18	1.18	1.18	0.11	1.69	1.69	10
	20	0.35	2.23	2.23	0.11	3.13	3.13	10

Table 5
Comparison of the BS and FBS algorithms and UB – Set 2.

		BS		FBS		UB		
m	n	Avg CPU Time	Avg WD Dev	Avg CPU Time	Avg WD Dev	Avg CPU Time	Avg WD Dev	Apron Same and Opt Known
8	15	0.12	1.32	0.10	1.70	0.10	2.32	8
	20	0.16	2.73	0.10	2.86	0.10	3.14	8
	25	0.23	1.31	0.10	1.31	0.10	1.31	6
10	15	0.14	0.87	0.10	1.20	0.10	1.67	9
	20	0.23	2.15	0.10	2.17	0.10	2.17	5
12	15	0.18	1.18	0.11	1.69	0.11	1.69	10
	20	0.35	0.96	0.11	1.21	0.11	1.21	8

Table 6
Results of the BS and FBS algorithms and UB – Set 2.

		BS			FBS			UB		
m	n	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)
8	15	10	0		10	0		8	0.2	1
	20	7	0.3	1	7	0.3	1	5	0.5	1
	25	5	0.5	1	3	0.8	1.1	3	1	1.4
10	15	10	0		10	0		9	0.1	1
	20	8	0.2	1	9	0.1	1	4	0.9	1.5
12	15	10	0		10	0		10	0	
	20	10	0		10	0		8	0.2	1

Overall, the satisfactory performance of the upper bound contributes to the success of our BS and FBS algorithms and through these algorithms we achieve even better performance. For the apron infeasible instances, the differences from the minimum apron levels are quite low (respective average differences are at most 1, 1.1. and 1.5 for BS, FBS and UB).

To evaluate the performance of the heuristic solutions on larger-sized instances, we perform an additional experiment using Ankara Esenboğa airport and İstanbul Atatürk airport layouts, which have 18 gates (9 domestic, 9 international) and 38 gates (12 domestic, 26 international), respectively. The layouts are provided in Appendix A.

We use the same generation scheme for the arrival and departure times, and number of transfer passengers. For the number non-transfer passengers, we use $DU(0, 100)$ and $DU(0, 75)$ for the domestic and international aircraft, respectively, assuming that the ratio of the transfer passengers is higher for the international aircraft. In line with our previous notation, Set 1 and Set 2 correspond to the arrival time and duration sets with low and high apron requirements, respectively.

For Esenboğa Airport Layout, we generate moderate-sized instances with 50, 100 aircraft as well as large-sized instances with 150, 200 aircraft. Tables 7 - 9 summarize the results for the Ankara Esenboğa Airport layout. For moderate-sized instances, we perform

Table 7

Results of the BS and FBS algorithms and UB – Esenboğa Airport.

		BS			FBS			UB
Set	n	Avg CPU Time	Avg WD Imp. (%)	Same Apron	Avg CPU Time	Avg WD Imp. (%)	Same Apron	Avg CPU Time
1	50	54.64	0.54	10	0.54	0.01	10	0.112
	100	663.47	0	5	7.20	0	3	0.113
2	50	36.58	0	3	0.36	0	1	0.105
	100	458.75	0	10	0.69	0	1	0.106

Table 8

Results of the BS and FBS algorithms and UB – Esenboğa Airport.

		BS			FBS			UB		
Set	n	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)
1	50	10	0		10	0		10	0	
	100	2	2	2.5	2	1.3	1.6	0	2.9	2.9
2	50	0	1.4	1.4	4	0.7	1.2	0	2.6	2.6
	100	5	0.5	1	5	0.5	1	5	0.5	1

Table 9

Results of the FBS algorithm and UB – Esenboğa Airport.

		FBS						UB			
Set	n	Avg CPU Time	Avg WD Imp. (%)	Same Apron	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)	Avg CPU Time	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)
1	150	30.84	0.00	1	0	2.8	2.8	0.13	0	7.7	7.7
	200	91.32	0.00	1	0	6.6	6.6	0.16	0	9.1	9.1
2	150	21.10	0.00	10	5	0.6	1.2	0.11	5	0.6	1.2
	200	67.63	0.00	10	5	0.8	1.6	0.12	5	0.8	1.6

a comparative analysis of all procedures in Tables 7 and 8. Table 9 summarizes the results of FBS and UB for larger instances.

In Table 7 we report the solution times of the BS, FBS and UB for instances with n=50 and 100. Since the exact approaches could not solve these instances in reasonable time, we cannot provide information on the walking distance deviations. Instead, we report on the percentage walking distance improvement of the BS and FBS solutions over the UB solution for the instances having the same number of apron assignments as UB. In Table 8, we report on the performance for the number of apron assignments.

Overall, we observe that for the medium-sized instances the BS algorithm still performs well, returning solutions in reasonable time. In Set 1 instances with 50 aircraft, all algorithms return solutions with minimum apron assignment. The numbers of apron assignments are slightly higher than their minimum levels in larger instances; however the average differences are very low (below 2, 1.3 and 2.9 for BS, FBS, and UB, resp.). In Set 2 instances, the differences from the minimum number of apron assignments are even lower (all below 0.5).

We observe that the BS and FBS algorithms improve the UB solution with respect to both the number of apron assignments and total walking distances. Note that in most instances the walking distance improvements are 0 since these are only calculated when the BS, FBS algorithms and UB return the same number of apron assignments. Since the BS and FBS solutions improve the UB solution with respect to the number of apron assignments, the walking distance improvement does not truly reflect their improvements. We also observe the computational advantage of the FBS algorithm over the BS algorithm, in returning fewer apron assignments in less time.

In Table 9, we report the walking distance and apron difference results for instances with n=150 and 200. It is seen that the FBS

returns good solutions in negligible time; outperforming UB with respect to number of apron assignments in Set 1 instances. The difference from the minimum number of apron assignments is even lower in Set 2 instances.

We generate even larger-sized instances with 50, 100, 150, 200 aircraft and 38 gates, based on İstanbul Atatürk Airport Layout. Tables 10 and 11 present the results of the BS, FBS algorithms and UB for n=50. For larger n, we report the performances of the FBS algorithm and UB in Table 12. We observe that in most instances the BS and FBS solutions are better than the UB solutions with respect to the number of apron assignments. When they make the same number of apron assignments, the BS and FBS solutions improve the walking distances.

We also observe that the algorithms return high-quality solutions with minimum apron assignments for most Set 1 instances and with very low differences from the minimum apron assignments for all Set 2 instances.

For the larger-sized instances shown in Table 12, UB provides satisfactory results, with low difference from the minimum number of apron assignments. FBS is observed to provide even better results in most instances, either with smaller walking distances or with less apron assignments.

The results demonstrate that the BS and FBS algorithms are both viable options for obtaining high quality solutions with near minimum apron assignments for medium-sized instances. Moreover, the experiments performed on the smaller-sized instances with known optimal walking distances indicate that the walking distance deviations of these solutions are very low. For large-sized instances the FBS algorithm provides good quality solutions reasonable time, improving the UB solutions, which are also of acceptable quality. The good performance of the BS and FBS algo-

Table 10

Results of the BS and FBS algorithms and UB –Atatürk Airport.

		BS		FBS		UB	
Set	*n*	Avg CPU Time	Avg WD Imp. (%)	Avg CPU Time	Avg WD Imp. (%)	Avg CPU Time	Same Apron
1	50	1602.65	11.26	4.30	10.47	0.112	10
2	50	1352.73	4.66	4.24	3.51	0.107	8

Table 11

Results of the BS and FBS algorithms and UB –Atatürk Airport.

		BS			FBS			UB		
Set	*n*	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)
1	50	10	0	0	10	0		10	0	0
2	50	9	0.1	1	10	0		8	0.2	1

Table 12

Results of the BS and FBS algorithms and UB –Atatürk Airport.

		FBS						UB			
Set	*n*	Avg CPU Time	Avg WD Imp. (%)	Same Apron	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)	Avg CPU Time	# Apron Feasible	Avg Apron Dif. (All)	Avg Apron Dif. (Apron Inf.)
1	100	71.6	12.56	10	10	0		0.126	10	0	
	150	361.2	6.06	10	10	0		0.154	10	0	
	200	1095.0	0	1	4	0.9	1.5	0.184	0	4.7	4.7
2	100	60.9		0	1	2.2	2.4	0.115	0	5.1	5.1
	150	281.1	0	2	0	3.1	3.1	0.130	0	4.2	4.2
	200	804.3	0	9	2	1.5	1.9	0.130	2	1.6	2

rithms can be attributed to the algorithms' prioritizing rule using the powerful walking distance lower bounds.

7. Conclusions

In this study we consider an airport gate assignment problem that aims to minimize the total passenger distance traveled subject to the minimum number of aircraft assigned to apron. We assume that the fixed gates can handle only one aircraft at a time and the apron has an unlimited capacity. We find the minimum number aircraft assigned to apron using maximum cost network flow models.

We formulate the problem as a mixed integer nonlinear programming model and then linearize it. We then develop a branch-and-bound algorithm and beam search and filtered beam search algorithms along with powerful lower and upper bounding mechanisms.

The results of our computational experiments based on the layout data of Atatürk Airport have revealed the superiority of the proposed branch-and-bound algorithm over the mathematical model for most problem combinations.

To the best of our knowledge we propose the first optimization algorithm for the problem of minimizing the total passenger distance traveled subject to the minimum number of aircraft assigned to apron. The algorithm could return optimal solutions to the problem instances with up to 25 aircraft when there are 8 gates and up to 25 and 20 aircraft there are 10 and 12 gates, respectively, in our

termination limit of one hour. For the problem instances of larger sizes, we propose beam search and filtered beam search algorithms that use the same mechanisms with our branch and bound algorithm. The beam search algorithm could return good quality solutions to the real life problem instances with up to 100 aircraft and 18 gates and 50 aircraft and 38 gates, while filtered beam search returns solutions for problems of even larger sizes, with up to 200 aircraft and 38 gates.

We assume that the arrival times and departure times are known with certainty and not subject to any change. Future research may consider algorithms that incorporate stochastic arrival and/or departure times. Moreover reassignment studies that would take our initial assignments as base-plan and minimize the deviations between the base-plan and new plan are worth-studying.

One may extend our models and solution procedures to include some practical aircraft assignment requirements like gate eligibility (some aircraft cannot be assigned to some gates) and aircraft compatibility (some aircraft may not be assigned to neighbor gates due to dimension concerns).

Future research may also consider the extension of our approach to the problems with additional objectives such as balancing the airport congestion and minimizing baggage movement.
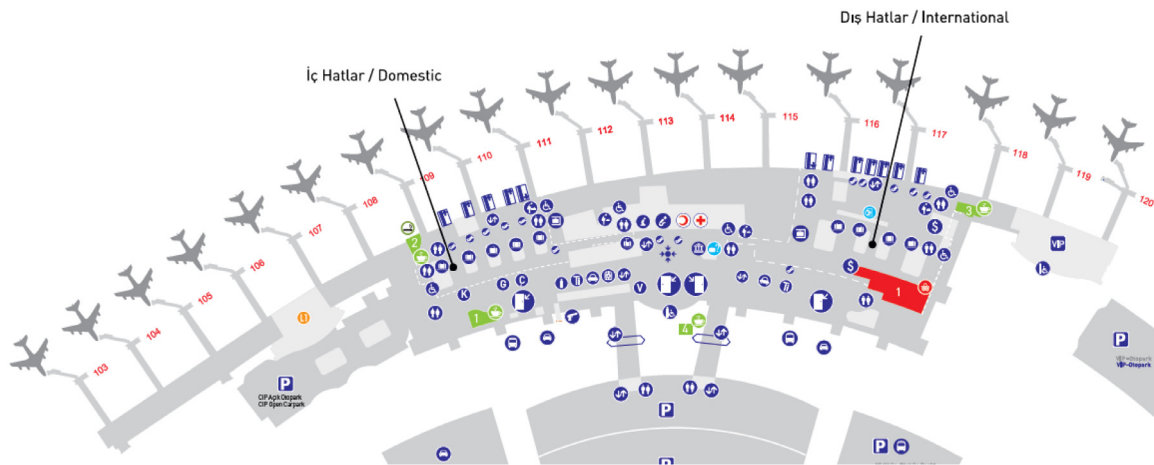
Fig. 4. Layout of Esenboğa Airport (source: https://webcmsesb.tav.aero/files/1551878584_ESB%20Gelis_2019.pdf). (Accessed on September 14, 2020).
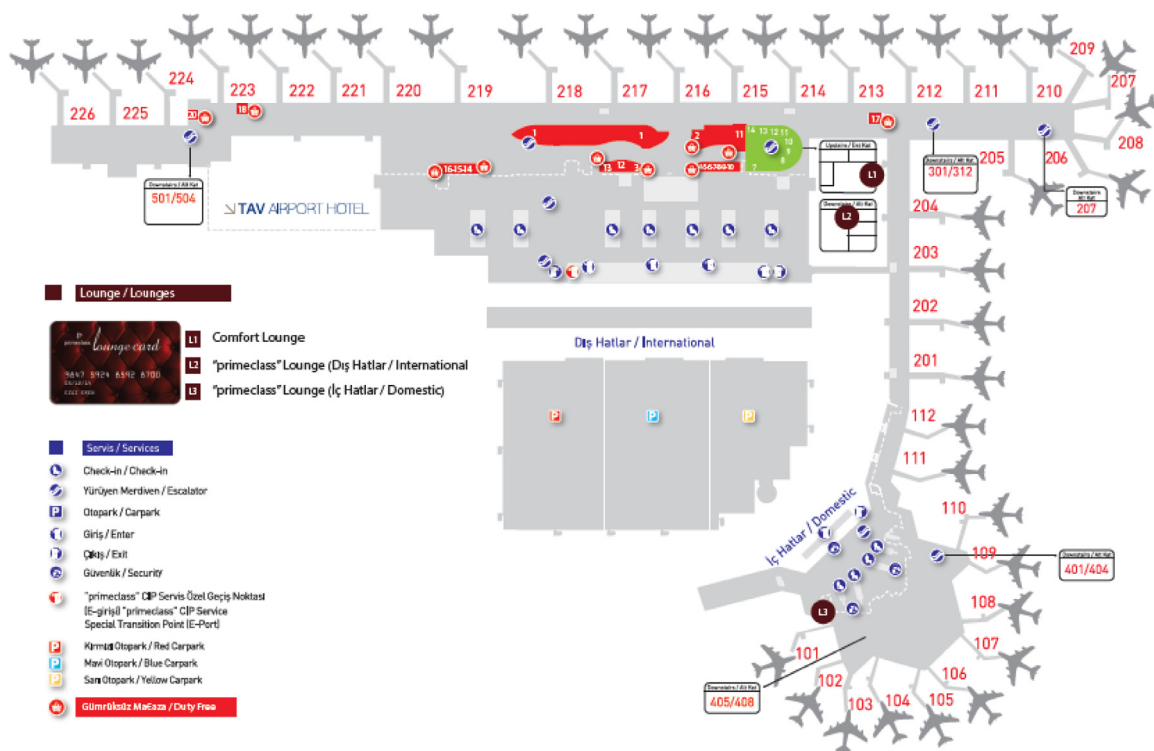

Fig. 5. Layout of Atatürk Airport (Gates 101-112 and 201-226 are considered).
(source: http://www.primeclass.com.tr/tr/Documents/istanbul%20gidi%C5%9F%20kat%C4%B1%20kroki.pdf. Accessed on September 15, 2020).

CRediT authorship contribution statement

Özlem Karsu: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft, Supervision. **Meral Azizoğlu:** Conceptualization, Methodology, Formal analysis, Writing - original draft, Supervision. **Kerem Alanlı:** Conceptualization.

Acknowledgments

Appendix A. Layouts

Figure 4, Figure 5

References

[1] Ahuja RK, Magnanti TL, Orlin JB. Network flows: theory, algorithms and applications. New Jersey, USA: Prentice Hall; 1993.
[2] Aktel A, Yagmahan B, Özcan T, Yenisey MM, Sansarcı E. The comparison of the metaheuristic algorithms performances on airport gate assignment problem. Transp Res Proc 2017;22:469–78. doi:10.1016/j.trpro.2017.03.061.
[3] Babić O, Teodorović D, Tošić V. Aircraft stand assignment to minimize walking. J Transp Eng 1984;110:55–66. doi:10.1061/(asce)0733-947x(1984)110:1(55).
[4] Barnhart C, Belobaba P, Odoni AR. Applications of operations research in the air transport industry. Transp Sci 2003;37:368–91.
[5] Bihr RA. A conceptual solution to the aircraft gate assignment problem using 0, 1 linear programming. Comput Ind Eng 1990;19:280–4. doi:10.1016/0360-8352(90)90122-3.
[6] Bolat A. Assigning arriving flights at an airport to the available gates. J Oper Res Soc 1999;50:23–34. doi:10.1057/palgrave.jors.2600655.
[7] Brazile R, Swigger K. GATES: an airline gate assignment and tracking expert system. IEEE Expert 1988;3(2):33–9. doi:10.1109/64.2103.

[8] Chao CC, Tang CH, Hsiao YH. Planned gater and runway assignments considering carbon emissions and costs. Int J Sustain Transp 2020;14:643–55. doi:10.1080/15568318.2019.1597225.

[9] Cheng C, Ho SC, Kwan C. The use of meta-heuristics for airport gate assignment. Expert Syst Appl 2012;39:12430–7. doi:10.1016/j.eswa.2012.04.071.

[10] Daş GS. New multi objective models for the gate assignment problem. Comput Ind Eng 2017;109:347–56. doi:10.1016/j.cie.2017.04.042.

[11] Daş GS, Gzara F, Stützle T. A review on airport gate assignment problems: single versus multi objective approaches. Omega 2020:102–46.

[12] Deng W, Sun M, Zhao H, Li B, Wang C. Study on an airport gate assignment method based on improved ACO algorithm. Kybernetes 2018;47:20–43. doi:10.1108/k-08-2017-0279.

[13] Ding H, Lim A, Rodrigues B, Zhu Y. The over-constrained airport gate assignment problem. Comput Oper Res 2005;32:1867–80. doi:10.1016/j.cor.2003.12.003.

[14] Dorndorf U, Drexl A, Nikulin Y, Pesch E. Flight gate scheduling: state-of-the-art and recent developments. Omega 2007;35:326–34. doi:10.1016/j.omega.2005.07.001.

[15] Ersan C. Implementation of Airport Gate Assignment Problem MS Thesis. Dept of Ind Eng, Marmara University; 2010.

[16] Genç HM, Erol OK, Eksin İ, Berber MF, Güleryüz BO. A stochastic neighborhood search approach for airport gate assignment problem. Expert Syst Appl 2012;39:316–27. doi:10.1016/j.eswa.2011.07.021.

[17] Gosling GD. Design of an expert system for aircraft gate assignment. Transp Res Part A: Gen 1990;24:59–69. doi:10.1016/0191-2607(90)90071-d.

[18] Haghani A, Chen M. Optimizing gate assignments at airport terminals. Transp Res Part A: Policy Pract 1998;32:437–54. doi:10.1016/s0965-8564(98)00005-6.

[19] Hu X, Di Paolo E. An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem. Multi-Object Memetic Algorithms 2009;71-89. doi:10.1007/978-3-540-88051-6_4.

[20] International Air Transport Association [IATA]. (2019, June 2). Annual growth in global air traffic passenger demand from 2006 to 2019 [chart]. Retrieved from https://www.statista.com/statistics/193533/growth-of-global-air-traffic-passenger-demand.

[21] Jalalian M, Gholami S, Ramezanian R. Analyzing the trade-off between CO2 emissions and passenger service level in the airline industry: mathematical modelling and constructive heuristic. J Clean Prod 2019;206:251–66 *2019*. doi:10.1016/j.jcle.2018.09.139.

[22] Jiang Y, Zeng L, Luo Y. Multiobjective gate assignment based on passenger walking distance and fairness. Math Prob Eng 2013;2013:1–7. doi:10.1155/2013/361031.

[23] Kim SH, Feron E, Clarke J. Gate assignment to minimize passenger transit time and aircraft taxi time. J. Guidance, Control, Dynam 2013;36:467–75. doi:10.2514/1.57022.

[24] Mangoubi RS, Mathaisel DF. Optimizing gate assignments at airport terminals. Transp Sci 1985;19:173–88. doi:10.1287/trsc.19.2.173.

[25] Marinelli M, Dell'Orco M, Sassanelli D. A metaheuristic approach to solve the flight gate assignment problem. Transp Res Proc 2015;5:211–20. doi:10.1016/j.trpro.2015.01.013.

[26] Mokhtarimousavi S, Talebi D, Asgari H. A Non-dominated sorting genetic algorithm approach for optimization of multi-objective airport gate assignment problem. Transp Res Record: J Transp Res Board 2018;2672:59–70.

[27] Morton TE, Pentico DW. Heuristic scheduling systems. New York: Wiley; 1993.

[28] Narciso ME, Piera MA. Robust gate assignment procedures from an airport management perspective. Omega 2015;50:82–95.

[29] Nemhauser GL, Wolsey LA. Integer and combinatorial optimization. New York, USA: John Wiley&Sons; 1988.

[30] Obata T. *Quadratic assignment problem: evaluation of exact and heuristic algorithms*, doctoral dissertation. NY, USA: Rensselaer Polytechnic Institute Troy; 1979.

[31] Pentico DW. Assignment problems: a golden anniversary survey. Eur J Oper Res 2007;176:774–93. doi:10.1016/j.ejor.2005.09.014.

[32] Şeker M, Noyan N. Stochastic optimization models for the airport gate assignment problem. Transp Res Part E: Log Transp Rev 2012;48:438–59. doi:10.1016/j.tre.2011.10.008.

[33] Srihari K, Muthukrishnan R. An expert system methodology for aircraft-gate assignment. Comput Ind Eng 1991;21:101–5. doi:10.1016/0360-8352(91)90071-d.

[34] Steuart GN. Gate position requirements at metropolitan airports. Transp Sci 1974;8:169–89. doi:10.1287/trsc.8.2.169.

[35] Su Y, Srihari K. A knowledge based aircraft-gate assignment advisor. Comput Ind Eng 1993;25:123–6. doi:10.1016/0360-8352(93)90236-q.

[36] Turkish Airlines. Retrieved September 1, 2019, from https://www.turkishairlines.com/.

[37] Xu J, Bailey G. The airport gate assignment problem: mathematical model and a tabu search algorithm. In: Proceedings of the 34th annual Hawaii international conference on system sciences; 2001. p. 3032. doi:10.1109/hicss.2001.926327.

[38] Yan S, Huo C. Optimization of multiple objective gate assignments. Transp Res Part A: Policy Pract 2001;35:413–32. doi:10.1016/s0965-8564(99)00065-8.

[39] Yan S, Shieh C, Chen M. A simulation framework for evaluating airport gate assignments. Transp Res Part A: Policy Pract 2002;36(10):885–98. doi:10.1016/s0965-8564(01)00045-3.

[40] Yu C, Zhang D, Lau H. MIP-based heuristics for solving robust gate assignment problems. Comput Ind Eng 2016;93:171–91. doi:10.1016/j.cie.2015.12.013.

[41] Yu C, Zhang D, Lau HY. An adaptive large neighborhood search heuristic for solving a robust gate assignment problem. Expert Syst Appl 2017;84:143–54. doi:10.1016/j.eswa.2017.04.050.

[42] Zhang D, Klabjan D. Optimization for gate re-assignment. Transp Res Part B: Methodol 2017;95:260–84.

[43] Loiola EM, de Abreu NMM, Boaventura-Netto PO, Hahn P, Querido T. A survey for the quadratic assignment problem. ur J Oper Res 2007;176(2):657–90.

[44] Bouras A, Ghaleb MA, Suryahatmaja US, Salem AM. The airport gate assignment problem: a survey. The scientific world journal, 2014.. Sci World J 2014.

[45] Drexl A, Nikulin Y. Multicriteria airport gate assignment and Pareto simulated annealing. IIE Trans 2008;40(4):385–97.