Computers & Operations Research 151 (2023) 106073

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

A new formulation and an effective matheuristic for the airport gate assignment problem

Özlem Karsu [a], Oğuz Solyalı [b,*]

[a] Department of Industrial Engineering, Bilkent University, Ankara 06800, Turkey
[b] Business Administration Program, Middle East Technical University Northern Cyprus Campus, Kalkanli 99738, Turkey

ARTICLE INFO

Keywords:
Airport gate assignment problem
Mixed integer linear programming
Matheuristic

ABSTRACT

This study considers an airport gate assignment problem where a set of aircraft arriving to an airport are assigned to the fixed gates of the airport terminal or to the apron. The aim is to lexicographically minimize the number of aircraft assigned to the apron, and then the total walking distance by passengers. A new mixed integer linear programming formulation and a matheuristic is proposed for the problem. The proposed formulation is based on the idea of flow of passengers and has smaller size compared to the existing formulations in the literature. The proposed matheuristic, which relies on solving a restricted version of the proposed formulation of the problem, is not only easy to implement but is also very effective. A computational study performed on benchmark instances reveals that the proposed formulation and the matheuristic outperform the existing exact and heuristic algorithms in the literature.

1. Introduction

Global air traffic has increased steadily over the past decades and is likely to increase further after recovery from the current pandemic. This increase, together with the passenger expectations and the highly competitive environment, calls for efficient use of available resources in the airline industry (Daş et al., 2020). The decision makers face operational problems that are challenging due to being large scale and involving competing criteria that should be concurrently considered. In line with this, various optimization problems related to airline operations have been defined and addressed in the literature.

One of the key operational decisions that airport professionals face on a daily basis is assigning aircraft to available gates, known as the airport gate assignment problem (AGAP). This problem concerns itself with assigning aircraft to gates so as to optimize various objective functions that are relevant. Daş et al. (2020) provide a recent comprehensive review on airport gate assignment problems, in which the existing works are classified with respect to the type of formulation, objective function (passenger, airport/airline or robustness-oriented) and solution methodology (exact or heuristic). See also Dorndorf et al. (2007) for another detailed survey on flight gate assignment approaches.

As stated in Daş et al. (2020), early approaches to the AGAP consider single objective formulations and propose various exact methods such as branch and bound. Some noteworthy initial attempts are due to Babić et al. (1984) and Bihr (1990), who minimize passenger walking distance in their models. Other studies considering exact solution approaches to various variants of AGAP are due to Bolat (1999), who proposes a branch-and-bound algorithm and a heuristic for minimizing the difference between the minimum and maximum slack times and Yu et al. (2016), who consider network flow formulations to minimize a weighted aggregation of the total distance traveled by transfer passengers, the towing costs, and the expected conflict time between schedules. Jaehn (2010) optimally solves a single objective problem that considers a special case where the flight/gate preferences are maximized, through dynamic programming. Recently, Li et al. (2021) consider an AGAP with the objective of minimizing arrival delays and propose a column generation-based exact algorithm. There are also studies that propose (meta)heuristic algorithms for several variants of AGAP (see e.g. Daş, 2017; Deng et al., 2017; Ding et al., 2005; Dorndorf et al., 2008; Li et al., 2022; Yu et al., 2016) as well as studies considering robust approaches (Cai et al., 2019; Xu et al., 2017) and multiobjective extensions that simultaneously address criteria related to different perspectives of passengers, airlines and robustness (Daş et al., 2020). Owing to their relevance, most studies consider minimizing passenger walking distance and minimizing the number of flights assigned to the apron as two objectives (Dell'Orco et al., 2017; Ding et al., 2005; Drexl and Nikulin, 2008); while there are also studies focusing on various passenger and robustness-oriented objectives (Daş, 2017; Şeker and Noyan, 2012; Tang and Wang, 2013; Yan and Huo, 2001).

* Corresponding author.
E-mail addresses: ozlemkarsu@bilkent.edu.tr (Ö. Karsu), solyali@metu.edu.tr (O. Solyalı).

https://doi.org/10.1016/j.cor.2022.106073
Received 8 March 2022; Received in revised form 14 September 2022; Accepted 31 October 2022
Available online 7 November 2022

We note here that these AGAP variants differ from the problem addressed in this work with respect to the objective functions, the assumptions on the operational dynamics, and the methodology employed (see also Daş et al., 2020; Dorndorf et al., 2007 and the references therein for more information on different variants of the airport gate assignment problems).

In this work, we consider the AGAP, in which airport-oriented and passenger-oriented objectives are considered in a lexicographic manner. Specifically, we ensure that total walking distance by passengers, the most used passenger-oriented objective in the literature according to Daş et al. (2020), is minimized while keeping the number of assignments to the apron at its minimum. This problem (AGAP-WD&AA) has been recently addressed by Karsu et al. (2021), where a mixed integer linear programming (MILP) formulation, a branch and bound algorithm and two branch and bound-based heuristics (beam search and filtered beam search) are proposed. Their branch and bound algorithm utilizes lower bounds for the passenger walking distance in a branch and bound framework, in which each level corresponds to the assignment of an aircraft and each node at a level is for the assignment of the corresponding aircraft to one of the feasible gates. Beam search and filtered beam search algorithms proposed by Karsu et al. (2021) are based on the idea of keeping attractive nodes when evaluating the branch and bound tree and discarding the rest permanently. The attractive nodes are determined with respect to their lower bounds on the passenger walking distance in beam search while filtered beam search uses a two-step procedure in which an initial screening based on realized distances is followed by a second phase evaluation based on the lower bounds on passenger walking distance.

Among the studies that consider only minimizing the passenger walking distance in AGAP (AGAP-WD), we observe that the ones finding exact solutions propose linearized formulations and solve them to optimality (Haghani and Chen, 1998; Maharjan and Matis, 2012; Xu and Bailey, 2001; Karsu et al., 2021). These linearizations are based on replacing the quadratic term with new variables which results in four-indexed variables in the respective MILP formulations. There are also several heuristics proposed for the AGAP-WD. Cheng et al. (2012) propose a genetic algorithm, a simulated annealing heuristic, a tabu search heuristic and a hybrid of simulated annealing and tabu search heuristics (SATS). They assess the performance of these heuristics on the real-world instances of Incheon International Airport of Korea (ICN) and find that the SATS performs the best among considered heuristics. Cheng et al. (2017) develop a tabu search heuristic with a path relinking feature (TSPR), which performs better than the SATS on ICN instances. Dell'Orco et al. (2017) and Deng et al. (2019) propose a fuzzy bee colony optimization algorithm (REFBCO) and an improved ant colony optimization algorithm (REICMPACO), respectively. Recently, Li et al. (2022), which develop a probability learning-based feasible and infeasible tabu search (PLFITS) heuristic, run REFBCO and REICMPACO heuristics on ICN instances besides PLFITS. They also run these heuristics on instances introduced by Karsu et al. (2021) which are based on Esenboğa (ESB) and İstanbul Atatürk airports (ATA) of Turkey. Results of Li et al. (2022) reveal that PLFITS heuristic finds the best-known solution in all instances of ICN, ESB and ATA data sets. Thus, one can say that PLFITS is the state-of-the-art heuristic method for the AGAP-WD.

Our contributions can be summarized as follows:

- Unlike (Haghani and Chen, 1998; Maharjan and Matis, 2012; Xu and Bailey, 2001; Karsu et al., 2021), we propose a new MILP formulation that uses three-indexed continuous variables instead of four-indexed variables so as to tackle the quadratic term in the AGAP-WD&AA. We consider the flow of commodity (transfer passengers of aircraft) between gates and propose a flow-based formulation that is novel for the airport gate assignment problem. The proposed formulation solves larger instances to optimality with smaller solution times than the exact algorithms and MILP formulation in Karsu et al. (2021).
- We suggest a mathematical programming-based heuristic (a.k.a a matheuristic) that exploits the strength of our new formulation and a newly proposed strong upper bound that is quickly obtained. The proposed matheuristic, which relies on solving a restricted version of the proposed formulation, is not only easy to implement but is also very effective.
- For the AGAP-WD&AA, we demonstrate that the proposed matheuristic outperforms the heuristics in Karsu et al. (2021) on benchmark instances.
- We also validate our matheuristic on the benchmark instances for the AGAP-WD. We compare our matheuristic with the heuristics proposed by Cheng et al. (2017, 2012), Dell'Orco et al. (2017), Deng et al. (2019), Li et al. (2022). The computational results show that our matheuristic performs much better than the existing heuristics and yields new best-known solutions in all ICN instances, and is comparable to the best-known approach in the literature in ESB and ATA instances.

The rest of the paper is as follows. In Section 2, we describe the problem we address in detail and give the new MILP formulation for the AGAP-WD&AA and its adaptation to the AGAP-WD. We describe the matheuristic we propose in Section 3 and report the results of our computational experiments in Section 4. We conclude the discussion in Section 5, in which we also point out some future research directions that could be pursued.

2. Problem description and formulations

We study an airport gate assignment problem where arriving aircraft have to be assigned to the fixed gates of the airport terminal or to the apron of the airport. Some of the fixed gates of the airport are solely allocated for international flights whereas the rest are used only for domestic flights. Whenever an aircraft arrives at the airport, it is assigned to a fixed gate until its departure if any fixed gate is available. If there are no available fixed gates, then the aircraft is assigned to the apron, which has an unlimited capacity and serves both domestic and international flights but is far from the airport exit and fixed gates of the terminal. Each aircraft has a number of passengers who are either transfer or non-transfer passengers. Non-transfer passengers come from the entrance of the airport to the gates of their aircraft or leave the airport via its exit after the arrival of the aircraft. Transfer passengers travel from their arrival gate (including apron) to the departure gate (including apron).

The main objective is to minimize the number of aircraft assignments to the apron. Then, among the solutions having the minimum number of aircraft assignments to the apron, the aim is to find the one with the minimum total walking distance by passengers. Note that total passenger walking distance may include not only the actual distance walked by passengers but also any distance traveled via buses, trains or any means of transportation used by passengers within the airport. Karsu et al. (2021) show that the minimum number of aircraft assignments to the apron can be found by solving maximum cost network flow problems. Thus, the obtained minimum number of aircraft assignments to the apron can be imposed as a constraint and the objective is to minimize the total passenger walking distance. This problem is referred to as the AGAP-WD&AA. When the minimum number of aircraft assignments to the apron is not considered, the problem is referred to as the AGAP-WD.

We first present the MILP formulation proposed by Karsu et al. (2021) and then propose a new MILP formulation for the AGAP-WD&AA. Before presenting the formulations, the parameters and variables used in the formulations are defined in the following. Note that all parameters are assumed to be known.

Parameters:

d_{kl}: distance traveled when going from gate k to gate l.

ed_k: distance between gate k and the entrance/exit of the airport.

e_i: number of passengers coming from the entrance of the airport to aircraft i.

f_i: number of passengers leaving the airport via its exit after the arrival of the aircraft i.

$g(i) = D$ if aircraft i is domestic; and $g(i) = I$ if aircraft i is international.

I: set of all domestic and international aircraft.

I_{Dt}: set of domestic aircraft overlapping in a time interval t.

I_{It}: set of international aircraft overlapping in a time interval t.

T_D: set containing I_{Dt} for all t. $T_D = \{I_{D1}, I_{D2}, \ldots\}$

T_I: set containing I_{It} for all t. $T_I = \{I_{I1}, I_{I2}, \ldots\}$

m: number of fixed gates; $m+1$ denotes the apron.

K'_D: set of domestic fixed gates.

K'_I: set of international fixed gates.

K_D: set of domestic gates (fixed gates and apron).

K_I: set of international gates (fixed gates and apron).

$K' = K'_D \cup K'_I$.

$K = K_D \cup K_I = K' \cup \{m+1\}$.

NA^*: minimum number of aircraft that must be assigned to the apron.

n: number of aircraft.

p_{ij} : number of passengers transiting from aircraft i to aircraft j.

Decision Variables:

y_{ijkl}: fraction of passengers transiting from aircraft i to aircraft j via gates k and l.

x_{ik}: 1 if aircraft i is assigned to gate k, and 0 otherwise.

The MILP formulation (KAA) proposed by Karsu et al. (2021) is as follows:

KAA :

$$\text{Min} \quad \sum_{i\in I}\sum_{j\in I}\sum_{k\in K_{g(i)}}\sum_{l\in K_{g(j)}} p_{ij}d_{kl}y_{ijkl} + \sum_{i\in I}\sum_{k\in K_{g(i)}} (e_i+f_i)ed_kx_{ik} \tag{1}$$

$$\text{s.t. } y_{ijkl} \geq x_{ik} + x_{jl} - 1 \quad \forall i \in I, j \in I, k \in K_{g(i)}, l \in K_{g(j)} \tag{2}$$

$$\sum_{k\in K_{g(i)}} x_{ik} = 1 \quad \forall i \in I \tag{3}$$

$$\sum_{i\in I_{Dt}} x_{ik} \leq 1 \quad \forall I_{Dt} \in T_D, \forall k \in K'_D \tag{4}$$

$$\sum_{i\in I_{It}} x_{ik} \leq 1 \quad \forall I_{It} \in T_I, \forall k \in K'_I \tag{5}$$

$$\sum_{i\in I} x_{i,m+1} = NA^* \tag{6}$$

$$x_{ik} \in \{0,1\} \quad \forall i \in I, k \in K_{g(i)} \tag{7}$$

$$y_{ijkl} \geq 0 \quad \forall i \in I, j \in I, k \in K_{g(i)}, l \in K_{g(j)}. \tag{8}$$

The objective function (1) is the sum of total walking distance by transfer passengers and non-transfer passengers, respectively. Note that the first term of (1) is actually the linearization of quadratic term $p_{ij}d_{kl}x_{ik}x_{jl}$ by replacing $x_{ik}x_{jl}$ with y_{ijkl} variables and defining constraints (2) and (8). Constraints (2) ensure that $y_{ijkl} = 1$ when both $x_{ik} = 1$ and $x_{jl} = 1$. Constraints (3) stipulate that each aircraft is assigned to a fixed gate or apron. Constraints (4) and (5) ensure that at most one aircraft can be assigned to a fixed gate during a certain time interval. Constraints (4) are for domestic aircraft whereas (5) are for international aircraft. Note that the sets I_{Dt} and I_{It} are maximal cliques which mean all aircraft in a set overlap with each other in a time interval and there is no other aircraft that overlaps with all aircraft in that set. The sets I_{Dt} and I_{It} are easily found using the polynomial time algorithm in Krishnamoorthy et al. (2012). Actually, Karsu et al. (2021) consider each interval between sorted arrival and departure times to derive constraints (4) and (5), which results in higher number of constraints with some being redundant (i.e., clique constraints that are not maximal). Constraint (6) guarantees that NA^* aircraft are assigned to the apron. Note that we use NA^*, which is found a priori using the procedure in Karsu et al. (2021), as a parameter in our model. Constraints (7) and (8) are for integrality and nonnegativity of variables, respectively. Note that although y variables are defined as continuous variables, they are indeed 0–1 variables which automatically take 0–1 values due to constraints (2) and (7).

Defining w^i_{kl} as the number of passengers of aircraft i traveling from gate k to gate l, we propose the following flow-based MILP formulation which is referred to as the FC formulation:

FC :

$$\text{Min} \quad \sum_{i\in I}\sum_{k\in K_{g(i)}}\sum_{l\in K} d_{kl}w^i_{kl} + \sum_{i\in I}\sum_{k\in K_{g(i)}} (e_i+f_i)ed_kx_{ik} \tag{9}$$

s.t.(3)–(7),

$$\sum_{l\in K} w^i_{kl} = (\sum_{j\in I} p_{ij})x_{ik} \quad \forall i \in I, k \in K_{g(i)} \tag{10}$$

$$\sum_{l\in K_{g(i)}} w^i_{lk} = \sum_{j\in I} p_{ij}x_{jk} \quad \forall i \in I, k \in K \tag{11}$$

$$w^i_{kl} \geq 0 \quad \forall i \in I, k \in K_{g(i)}, l \in K. \tag{12}$$

The objective function (9) is equivalent to (1). Constraints (10) and (11) are the multicommodity flow balance equations for passengers. Constraints (10) ensure that if an aircraft i is assigned to gate k, then all passengers originating from aircraft i must travel from gate k to other gates. Constraints (11) ensure that all passengers between aircraft i and other aircraft assigned to gate k must travel to gate k from the gate aircraft i is assigned to. Note that multicommodity flow-based constraints similar to (10) and (11) are proposed by Erdoğan and Tansel (2007) for the quadratic assignment problem (QAP) but Erdoğan and Tansel (2007) did not present any computational results obtained with the formulation involving these constraints for the QAP. By using these multicommodity flow-based constraints in AGAP-WD&AA, it enables us to formulate the problem with three-index variables instead of four-index variables in KAA formulation. Note that FC formulation has $O(nm^2)$ continuous variables, $O(nm)$ binary variables, and $O(nm)$ constraints whereas the KAA formulation has $O(n^2m^2)$ continuous variables, $O(nm)$ binary variables, and $O(n^2m^2)$ constraints. This makes the FC formulation computationally advantageous compared to the KAA formulation as can be seen in Section 4.

Different multicommodity flow-based formulations are proposed for the AGAP variants in the literature (see e.g., Maharjan and Matis, 2012; Wang et al., 2022; Yu et al., 2016). These multicommodity flow-based formulations use binary z_{ijk} variables meaning assignment of aircraft j just after aircraft i to fixed gate k to prevent assignment of more than one aircraft to the same gate during a certain time interval. Recently, Wang et al. (2022) proposed a new multicommodity flow-based formulation using the same z variables to optimize robustness of aircraft assignments to gates. However, these multicommodity flow-based formulations require $O(n^2\ m)$ binary variables and they do not deal with the quadratic term in the objective function associated with the passenger walking distance, unlike our multicommodity flow-based formulation (FC), where we linearize the quadratic term due to the passenger walking distance using $O(nm^2)$ continuous variables (i.e., w variables).

In order to strengthen the proposed FC formulation, we propose the following valid inequalities which are based on the set of domestic and international aircraft overlapping in a time interval t:

$$\sum_{i\in I_{Dt}} x_{i,m+1} \geq |I_{Dt}| - |K'_D| \qquad \forall I_t \in T : |I_{Dt}| > |K'_D| \tag{13}$$

$$\sum_{i\in I_{It}} x_{i,m+1} \geq |I_{It}| - |K'_I| \qquad \forall I_t \in T : |I_{It}| > |K'_I| \tag{14}$$

Constraints (13) stipulate that among the domestic aircraft overlapping in a time interval at least a certain number of them (i.e., the number of domestic aircraft overlapping in a time interval less the number of fixed gates for domestic flights) must be assigned to the apron. Constraints (14) ensure the same for international aircraft. In the rest of the paper, whenever the FC formulation is referred to, it includes constraints (13) and (14) as well.

3. The proposed matheuristic

In this section, we propose a matheuristic which relies on constructing and solving a small-size FC formulation. In the matheuristic, we first obtain an initial feasible solution (i.e., a feasible assignment of aircraft to fixed gates and apron) by solving the following approximate formulation:

IFS :

$$\text{Min} \sum_{i\in I}\sum_{k\in K_{g(i)}}((d_k^{aveto}\sum_{j\in I}p_{ji} + d_k^{avefrom}\sum_{j\in I}p_{ij}) + ed_k(e_i+f_i))x_{ik} \tag{15}$$

s.t. (3)–(7).

where $d_k^{aveto} = \sum_{l\in K} d_{lk}/|K|$ and $d_k^{avefrom} = \sum_{l\in K} d_{kl}/|K|$.

The objective function (15) of the IFS formulation approximates the walking distance of transfer passengers by using the average distances between a gate and other gates (i.e., d_k^{aveto} and $d_k^{avefrom}$) and the number of passengers associated with aircraft assigned to that gate. If the distance matrix is symmetric, d_k^{aveto} and $d_k^{avefrom}$ will be equal. Note that this approximation enables us to eliminate w variables and constraints (10)–(12) of the FC formulation.

Using the values of x variables in the optimal solution of above formulation, we can also easily compute the true objective function value which can be used as an initial upper bound. As will be observed from Section 4, the IFS formulation provides a quick and good quality initial upper bound for the AGAP-WD&AA and AGAP-WD.

The main idea of the proposed heuristic is to solve a restricted FC formulation by replacing parameter $K_{g(i)}$ with $\bar{K}_{g(i)}$, where $\bar{K}_{g(i)}$ defines only a subset of the available gates to which aircraft i can be assigned. This new parameter $\bar{K}_{g(i)}$ reduces the number of x and w variables in the original formulation which helps solving the formulation to optimality. On the other hand, not defining all x and w variables in the FC formulation may cause infeasibility when the formulation is solved because assignment of all aircraft to a restricted set of gates (without exceeding the number of aircraft that must be assigned to the apron) may not be possible in some time intervals. Thus, the feasible solution obtained by solving the IFS formulation is critical to ensure feasibility of the restricted FC formulation as will be discussed below. Note that the restricted FC formulation can simply be constructed by setting $x_{ik} = 0$ if $k \notin \bar{K}_{g(i)} \forall i \in I$.

In the proposed matheuristic, which is referred to as the restricted formulation-based heuristic, we determine $\bar{K}_{g(i)}$ as follows: For each aircraft i, we sort all fixed gates in nondecreasing order of $d_k^{aveto}\sum_{j\in I} p_{ji} + d_k^{avefrom}\sum_{j\in I} p_{ij} + ed_k(e_i+f_i)$ and take the first $\beta - 1$ of them where $\beta < |K_{g(i)}|$. Note that we take $\beta - 1$ of fixed gates because the apron is always included in $\bar{K}_{g(i)}$ as the β^{th} gate. If this set does not include the regular gate that the aircraft is assigned to in the initial feasible solution found by solving the IFS formulation, we replace the $(\beta-1)^{th}$ gate with the one from the initial feasible solution. With this definition of $\bar{K}_{g(i)}$, we guarantee obtaining a feasible solution when solving the restricted formulation.

We set β to an initial value and increase it until the stopping condition is satisfied. The algorithm stops when until either no improvement regarding the objective function value is obtained or a time limit is reached.

Procedure Pseudocode of the restricted formulation-based heuristic

```
Solve the IFS formulation to obtain an initial solution x.
Initialize β. Set time = 0, improvement = 1. Determine K̄_g(i).
while improvement > 0 and time < 3600 do
    Solve the restricted formulation with K̄_g(i) with time limit 3600-time.
    Update improvement and time.
    Set β = β + 1 and update K̄_g(i) accordingly.
Return the best solution.
```

4. Computational results

We conducted computational experiments on benchmark instances introduced by Karsu et al. (2021) and Cheng et al. (2012) for the problem with the objectives of minimizing the passenger walking distance and minimizing the assignment of aircraft to apron (i.e., AGAP-WD&AA) and for the problem with the objective of minimizing the passenger walking distance (i.e., AGAP-WD).

Karsu et al. (2021) used the layouts of Ankara Esenboğa airport with 18 fixed gates (ESB instances) and İstanbul Atatürk airport with 38 fixed gates (ATA instances) to generate instances with 50, 100, 150, and 200 aircraft. Two sets of instances, set 1 and set 2, are generated for each problem size (i.e., number of fixed gates (m) and number of aircraft (n)) combination. There is low (resp. high) apron usage in Set 1 (resp. Set 2). For each problem size and set combination, 10 instances are generated. Thus, there are 80 ESB instances and 80 ATA instances in total. In addition to ESB and ATA instances, Karsu et al. (2021) randomly generated small size instances containing 12 fixed gates and up to 30 aircraft in order to test the exact approaches (see Karsu et al., 2021 for details).

The second set of benchmark instances introduced by Cheng et al. (2012) are based on real data from ICN, the largest airport in South Korea. As in the other studies (Cheng et al., 2012; Li et al., 2022) we use instances generated based on the gates and exits of Terminal 1 with 74 gates and 14 exits. In these instances, the daily flight data is obtained from FlightStats, where the number of aircraft/flights varies from 279 to 304. For each day of the week, the percentage of transfer passengers (π) are set to 0.1, 0.3, and 0.5, which results in 21 ICN instances. However as the flight data of Thursday when $\pi = 0.1$ is lost, we use 20 ICN instances as in Li et al. (2022).

The proposed matheuristic (RFH) is coded in C++ and all formulations (KAA, FC, and IFS) are solved by CPLEX 12.7 with its default settings. All experiments with RFH, FBS, KAA, FC, and IFS were performed on an Intel Core i7 2.70 GHz dual-core computer with 8 GB RAM, which is the same computational platform as in Karsu et al. (2021). All solution times are reported in CPU seconds.

4.1. Results for AGAP-WD&AA: Comparison of exact approaches

We first investigate the performance of the proposed MILP formulation (FC) compared to the three exact approaches suggested in Karsu et al. (2021): the linearized formulation with and without a lower bound constraint (KAA with LB and KAA, respectively) and the branch and bound algorithm (BB) on the same set of instances used in Karsu et al. (2021) for AGAP-WD&AA. In order to observe the computational efficiency gained by the valid inequalities (13) and (14), we also include the results of FC formulation excluding these inequalities.

We summarize the results of our experiments in Table 1. For each approach, we provide the average ($Time_{avg}$) and maximum ($Time_{max}$) solution times and the number of instances that could not be solved to optimality within a time limit of one hour (NO). For any method, we stopped solving larger instances when at least one instance could not be solved within the time limit. The results clearly demonstrate the superiority of our formulation. The FC formulation has considerably lower solution times and could solve larger instances to optimality within the time limit compared to KAA formulations and the BB. Moreover, the results show the computational advantage of using valid inequalities, especially in larger problem instances. The results also reveal that set 2 instances where apron usage is high are more difficult to solve for all exact approaches.

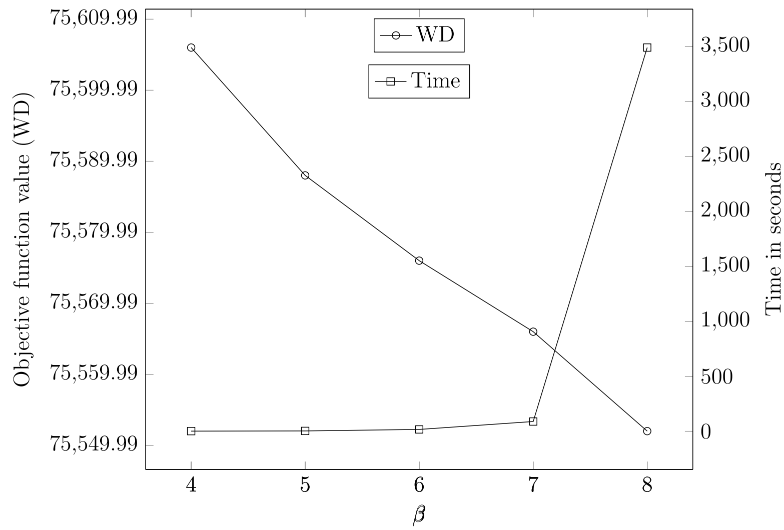

Fig. 1. Effect of β (Set 2, instance 6).

Table 1
Comparison of exact approaches.

Set	m	n	KAA			KAA with LB			BB			FC			FC without (13) and (14)		
			$Time_{avg}$	$Time_{max}$	NO	$Time_{avg}$	$Time_{max}$	NO	$Time_{avg}$	$Time_{max}$	NO	$Time_{avg}$	$Time_{max}$	NO	$Time_{avg}$	$Time_{max}$	NO
1	12	15	2.13	3.28	0	2.55	4.03	0	0.257	1.42	0	0.15	0.21	0	0.16	0.26	0
		20	94.30	354.51	0	96.53	192.72	0	9.557	58.06	0	0.46	1.59	0	0.43	1.24	0
		25	2076.46	3600.00	4	3176.85	3600.00	5	890.13	3600.00	2	6.77	54.99	0	7.19	59.01	0
		30	–	–	–	–	–	–	–	–	–	17.34	116.74	0	17.53	118.85	0
2	12	15	88.22	321.68	0	410.83	3600.00	1	18.55	90.14	0	1.08	3.31	0	1.88	6.39	0
		20	2869.19	3600.00	5	–	–	–	1360.86	3600.00	1	25.28	99.68	0	43.03	131.32	0
		25	–	–	–	–	–	–	–	–	–	368.45	1748.08	0	608.46	2217.25	0
		30	–	–	–	–	–	–	–	–	–	445.59	1538.98	0	603.49	1585.24	0

4.2. Results for AGAP-WD&AA: Comparison of heuristics

In this section, we compare our matheuristic (RFH) with other heuristics on ESB and ATA instances for AGAP-WD&AA. Specifically, we compare RFH with FBS and an initial upper bound procedure (i.e., KAA-UB) in Karsu et al. (2021), the best upper bound obtained by CPLEX with the FC formulation in one hour (FC-1hr), and the initial upper bound obtained by solving the IFS formulation (IFS).

Note that, in our experiments we set the filter width and beam width parameters in the FBS algorithm to higher values compared to Karsu et al. (2021) so as to make this algorithm more effective (also consumes more time). In particular, considering the memory requirements, we set filter width to 23000 and beam width to 11500.

Fig. 1 provides some insight on how parameter β affects the results of RFH. In this figure, we provide the results of an ESB instance with 50 aircraft, for which RFH started with $\beta = 4$ and stopped after five iterations. The figure typically illustrates the progress of RFH (computation times and objective function) while β is increasing. We see that the larger β is, the better the quality of the solution is, but this comes at the expense of higher solution times.

Based on a preliminary analysis on the ESB instances with 50 aircraft, we set the initial value of β to six in RFH. The results of this preliminary analysis are summarized in Table 2, which shows the objective function values with the best results being highlighted in boldface for each instance. All approaches are implemented with a time limit of one hour.

Table 2
Preliminary results to set β in RFH for AGAP-WD&AA.

Instance	Set 1			Set 2		
	$\beta = 4$	$\beta = 5$	$\beta = 6$	$\beta = 4$	$\beta = 5$	$\beta = 6$
1	**23 934**	**23 934**	**23 934**	73 181	73 181	**73 173**
2	**26 782**	**26 782**	**26 782**	69 446	69 446	**69 444**
3	23 404	**23 354**	**23 354**	**78 880**	**78 880**	**78 880**
4	26 459	**26 455**	**26 455**	81 156	**81 140**	**81 140**
5	**24 122**	**24 122**	**24 122**	79 689	**79 665**	**79 665**
6	**23 358**	**23 358**	23 382	**75 552**	**75 552**	**75 552**
7	**25 129**	**25 129**	**25 129**	88 365	**88 103**	**88 103**
8	**24 868**	**24 868**	**24 868**	**80 888**	**80 888**	**80 888**
9	**24 615**	**24 615**	**24 615**	**78 766**	**78 766**	**78 766**
10	27 365	**27 354**	**27 354**	**69 084**	**69 084**	**69 084**

We implement all algorithms on ESB instances with 50, 100 and 200 aircraft and present the results in Table 3. For these instances we report the average solution times (Time) as well as average % gaps with respect to the best known lower and upper bounds. We find % gap of the solutions from the best upper bound using %GapUB = $100 \times (S - BestUB)/BestUB$ where S is the solution of interest and $BestUB$ is the best upper bound found by all algorithms. Similarly, we also calculate the % gap of the solutions (S) from the lower bound ($CPXLB$) obtained by solving the FC formulation using CPLEX with a time limit of one hour. This % gap is calculated as follows: %GapLB $= 100 \times (S - CPXLB)/CPXLB$. Note that as the initial upper bound procedure suggested by Karsu et al. (2021) (i.e., KAA-UB) makes assignments of aircraft iteratively to gates by considering one gate at a time until no aircraft is left to be assigned or until the last fixed gate is considered, it does not guarantee that the apron assignments are kept at minimum. Similarly, FBS does not guarantee returning an apron feasible solution as it is based on using lower bounds on the total passenger walking distance in the nodes of the branch and bound tree. In case of a failure in finding a feasible solution to some instances, the number of feasible solutions found by that algorithm is reported in column #F. Note that %GapLB and %GapUB are calculated only when the corresponding algorithm finds an apron feasible solution, and

Table 3
Comparison of heuristics on ESB instances for AGAP-WD&AA.

Set	m	n	RFH			FC-1hr			IFS			FBS				KAA-UB			
			Time	%GapLB	%GapUB	#F	%GapLB	%GapUB	Time	%GapLB	%GapUB	Time	#F	%GapLB	%GapUB	Time	#F	%GapLB	%GapUB
1	18	50	3600.00	2.59	0.01	10	2.64	0.06	0.04	2.79	0.21	262.72	10	3.75	1.14	0.11	10	4.59	1.96
		100	3600.38	7.22	0.00	10	7.92	0.65	0.06	7.27	0.04	2321.50	7	13.23	4.95	0.11	0	–	–
		200	3600.55	5.57	0.00	7	7.34	1.67	0.38	5.58	0.00	3600.00	0	–	–	0.16	0	–	–
2	18	50	2326.95	2.78	0.04	10	2.94	0.18	0.03	2.91	0.16	264.74	6	8.96	5.37	0.11	0	–	–
		100	3295.51	2.75	0.01	10	2.90	0.16	0.11	2.81	0.07	2289.32	5	3.29	0.27	0.11	5	3.29	0.27
		200	3578.74	1.56	0.04	10	1.79	0.27	0.14	1.57	0.05	3600.00	5	1.65	0.03	0.12	5	1.65	0.03

Table 4
Comparison of heuristics on ATA instances for AGAP-WD&AA.

Set	m	n	RFH			FC-1hr			IFS			FBS				KAA-UB			
			Time	%GapLB	%GapUB	#F	%GapLB	%GapUB	Time	%GapLB	%GapUB	Time	#F	%GapLB	%GapUB	Time	#F	%GapLB	%GapUB
1	38	50	3600.52	19.57	0.00	10	21.87	1.93	0.06	20.37	0.68	1081.32	10	20.58	0.85	0.11	10	36.45	14.09
		100	3600.67	25.63	0.00	10	30.04	3.51	0.10	25.87	0.19	3732.59	10	46.69	16.77	0.13	10	46.69	16.77
		200	3448.01	29.83	0.00	4	48.58	14.47	1.28	29.88	0.03	3907.46	0	–	–	0.18	0	–	–
2	38	50	1173.92	26.45	0.00	10	28.98	2.00	0.05	26.69	0.19	1164.99	10	34.00	5.95	0.11	8	39.90	10.60
		100	2248.62	11.14	0.00	8	13.09	1.76	0.15	11.19	0.05	3776.77	0	–	–	0.12	0	–	–
		200	3313.12	–	0.00	0	–	–	0.56	–	0.02	4136.99	2	–	0.37	0.13	2	–	0.37

Table 5
Number of best solutions achieved by each method for AGAP-WD&AA.

Set	n	ESB (m=18)			ATA (m=38)		
		RFH	FC-1hr	FBS	RFH	FC-1hr	FBS
1	50	8	5	1	10	0	0
	100	10	0	0	10	0	0
	200	10	0	0	10	0	0
2	50	7	3	0	10	0	0
	100	8	1	1	10	0	0
	200	8	1	3	10	0	0

Table 6
Performance of RFH compared to FC with IFS for AGAP-WD&AA.

	Instance	RFH	FC with IFS		Instance	RFH	FC with IFS
ESB							
Set 1	1	**23 934**	23 947	Set 2	1	**73 173**	73 203
	2	**26 782**	26 823		2	**69 444**	69 459
	3	23 354	**23 342**		3	78 880	**78 790**
	4	**26 455**	26 459		4	**81 140**	81 156
	5	**24 122**	**24 122**		5	**79 665**	79 676
	6	**23 382**	23 399		6	**75 552**	75 630
	7	**25 129**	25 184		7	**88 103**	88 365
	8	**24 868**	24 879		8	80 888	**80 843**
	9	**24 615**	24 617		9	78 766	**78 726**
	10	27 354	**27 348**		10	**69 084**	69 113
ATA							
Set 1	1	**34 938**	34 976	Set 2	1	**48 092**	48 120
	2	**34 102**	34 129		2	**48 790**	48 889
	3	**34 883**	35 081		3	**55 434**	55 836
	4	**36 663**	36 709		4	**62 413**	62 451
	5	**35 280**	35 574		5	**51 653**	51 672
	6	**36 056**	36 103		6	**47 743**	47 781
	7	**35 476**	35 789		7	**65 460**	65 533
	8	**34 411**	34 564		8	**57 601**	57 738
	9	**34 434**	34 744		9	**58 094**	58 139
	10	**38 092**	38 128		10	**49 606**	49 742

the averages are calculated over instances for which an apron feasible solution is obtained.

Results in Table 3 reveal that the proposed matheuristic outperforms other approaches with respect to the solution quality. It is observed that the RFH mainly stops since no improvement is observed. IFS also performs very well, returning high quality solutions in negligible time. This is not the case for KAA-UB as it cannot find feasible solutions in many instances (i.e., the number of apron assignments exceeds the minimum) and in the instances that it can, the quality of the solution is worse than that of IFS. FBS could not find any feasible solution in more than half of the instances. Even in instances where FBS found a feasible solution, the quality of those solutions are inferior to those of RFH, FC, and IFS. In addition, FBS could not improve the solution provided by KAA-UB in any instances of Set 2 as shown by the average %GapLB and %GapUB values. While FC managed to find a feasible solution for most ESB instances, the quality of its solutions are worse than those of RFH and IFS.

We provide the computational results on ATA instances in Table 4 which has the same format as Table 3. Similar to the ESB instances, the proposed matheuristic yields results superior to other approaches on ATA instances. Average % gaps with respect to the best lower bounds are quite large in ATA instances compared to those in ESB instances. This can be explained by the fact that the FC formulation could not return good quality lower bounds due to the large size of ATA instances which have 38 fixed gates. Note that FC even failed to find a feasible solution to 18 ATA instances out of 60 with the majority of those instances having 200 aircraft. FBS and KAA-UB also failed to find feasible solutions in 28 and 30 ATA instances out of 60, respectively. Our simple and quick heuristic IFS not only found a feasible solution to all ATA instances but also outperformed both FBS and KAA-UB with respect to the solution quality and time. Note that the average % gaps for FC, FBS and KAA-UB are calculated over the instances in which they could obtain a feasible solution.

In Table 5, we provide the number of best solutions returned by each method on ESB and ATA instances. Note that more than one method may yield the best solution in some instances. According to Table 5, RFH found the best solutions in all ATA instances whereas it obtained the best solutions in 51 ESB instances out of 60. It seems that as the number of fixed gates increases, the performance of RFH improves compared to other heuristics.

Note that IFS produces very good solutions quickly and RFH improves the solution found by IFS in most instances in a very short time. We performed further experiments on the ESB instances with 50 aircraft to see the value of using the restricted formulation (i.e., RFH) compared to solving the full formulation (i.e., FC) when the solution found by the IFS is supplied to the solver as an initial solution. The results which are summarized in Table 6 demonstrate that it is better to use the restricted formulation instead of the full one with the initial solution by IFS (i.e., FC with IFS) as the former mostly yields better solutions than the latter does within the given time limit.

Table 7
Preliminary experiments for ICN instances.

β	$\pi = 0.5$, Friday ($n = 294$)				$\pi = 0.3$, Tuesday ($n = 290$)			
	RFH with IFS′		RFH with IFS		RFH with IFS′		RFH with IFS	
	Time	WD	Time	WD	Time	WD	Time	WD
5	42.08	27,227,920	31.01	28,294,000	23.70	24,881,075	24.52	25,669,080
6	68.24	27,212,150	56.67	28,273,405	37.13	24,874,255	34.38	25,660,965
7	182.02	27,146,795	180.34	28,255,450	71.45	24,864,740	73.93	25,660,965
8	200.00	27,107,200	200.00	28,270,650	172.39	24,838,515	200.00	25,657,930
9	200.00	27,163,225	200.00	28,331,660	200.00	24,843,395	200.00	25,665,180
10	200.00	27,248,700	200.00	28,343,035	200.00	24,836,010	200.00	25,675,480

4.3. Results for AGAP-WD: Comparison of heuristics

In this section, we compare our matheuristic with the metaheuristics existing in the literature on ICN, ESB and ATA instances for AGAP-WD. Specifically, we compare the following heuristics:

- SATS: Hybrid simulated annealing and tabu search heuristic in Cheng et al. (2012)
- TSPR: Tabu search heuristic with a path relinking feature in Cheng et al. (2017)
- REFBCO: Fuzzy bee colony optimization algorithm in Dell'Orco et al. (2017)
- REICMPACO: Improved ant colony optimization algorithm in Deng et al. (2019)
- FBS: Filtered beam search heuristic in Karsu et al. (2021)
- PLFITS: Probability learning-based feasible and infeasible tabu search heuristic in Li et al. (2022)
- IFS: IFS formulation-based heuristic proposed in this study
- RFH: Restricted formulation-based heuristic proposed in this study

The results of TSPR and SATS, which are only available for ICN instances, are obtained from Cheng et al. (2017) and Cheng et al. (2012), respectively, whereas those of REFBCO, REICMPACO and PLFITS are taken from Li et al. (2022), and those of FBS are obtained from Karsu et al. (2021). Note that Li et al. (2022) implemented REFBCO, REICMPACO and PLFITS with a time limit of 200 s.

As all metaheuristics except SATS, TSPR and FBS were run with a time limit of 200 s on the benchmark instances for AGAP-WD, we implemented RFH with a slight modification in order to have a fair comparison. In particular, we solve the restricted formulation only once with a fixed β (the number of gates that can be considered per aircraft) and set the time limit to 200 s as in Li et al. (2022). For ESB and ATA instances, we set β to six as in our previous experiments whereas we set β to eight for ICN instances based on a preliminary experiment performed on two randomly selected instances with a different percentage of transfer passengers (π) and a different day implying different number of aircraft (i.e., n) (see Table 7). We observe that the number of transfer passengers is very small compared to the number of non-transfer passengers in ICN instances. Therefore, in addition to trying different β values, we tried minimizing $\sum_{i \in I} \sum_{k \in K_{g(i)}} ed_k(e_i + f_i)x_{ik}$ instead of the original objective function (15) in IFS. This modified IFS is referred to as IFS′ in the sequel.

The results of the preliminary experiments are given in Table 7 where the obtained objective function values and solution times are shown in columns WD and Time, respectively. The results in Table 7 indicate that using RFH with IFS′ and setting β to eight yield better results. Hence we perform our main experiments using IFS′ and $\beta = 8$ in ICN instances. Note that we also tried using RFH without IFS or IFS′ on these two instances and observed that RFH could not return any feasible solution for β values ranging from 5 to 10, which reveals the importance of IFS or IFS′ for RFH.

The results for ESB and ATA instances are summarized in Tables 8 and 9. In these tables we report the objective function (i.e., total passenger walking distance (WD)) achieved by FBS, REICMPACO, REFBCO, PLFITS, IFS and RFH, and the solution times (Time) of these approaches. Note that REICMPACO, REFBCO, and PLFITS were run 10 times and the reported passenger walking distances in Tables 8 and 9 are the best ones among 10 runs. The solution times given for REICMPACO, REFBCO, and PLFITS are the average solution times over 10 runs. In Tables 8 and 9 we also present the average percent improvement (Imp%) on the total passenger walking distance achieved by the RFH compared to the PLFITS ($Imp\% = 100 \times (PLFITS - RFH)/PLFITS$). Note that PLFITS has achieved the best results on all ESB and ATA instances in the literature. It is seen that RFH and PLFITS outperform FBS, REICMPACO and REFBCO on ESB instances. RFH yields superior results to PLFITS with regard to average solution quality in set 1 instances while the former obtains slightly worse results than the latter in set 2 instances where apron usage is high. IFS quickly finds slightly worse but similar results as RFH. On ATA instances which have fixed gates more than twice of the fixed gates available in ESB instances, RFH largely outperforms other heuristics in that it achieves the best average results in all ATA instances except those in set 1 with 50 aircraft and those in set 2 with 150 aircraft. Similar to the results for AGAP-WD&AA, when the number of fixed gates increases, RFH improves its performance compared to other heuristics. Like ESB instances, IFS obtains slightly worse results than RFH on ATA instances but with negligible times.

The results for ICN instances are summarized in Table 10, where we report the objective function values (WD) achieved by SATS, TSPR, REICMPACO, REFBCO, PLFITS, IFS′ and RFH, and the solution times of these approaches. We also report the percent improvement (Imp%) achieved by RFH over PLFITS in terms of WD as PLFITS obtained the best-known solutions for all ICN instances until now. It is seen that RFH yields the best results in all 20 ICN instances. Hence, our results improve the best-known results in the literature for these instances. RFH improves the best results by 3.28% on average, with average percent improvement decreasing as the percentage of transfer passengers (π) increases. Note that our simple and fast IFS′ heuristic also performs better than the existing metaheuristics with very small solution times compared to others.

Although computational platforms used by the heuristics are different, the solution times needed by the heuristics except IFS/IFS′ are comparable to each other on all data sets due to the time limit of 200 s. Note that the computational platform used for REICMPACO, REFBCO and PLFITS is slightly slower than the one used for RFH, as stated by Li et al. (2022). However, considering that the best objective values of 10 runs of REICMPACO, REFBCO and PLFITS are reported with their average solution times over 10 runs, the obtained results are in favor of RFH.

Finally, as in Section 4.2, we performed additional experiments to see how RFH compares against the full formulation where the MILP solver is fed with the solution found by IFS (i.e., FC with IFS). The results of these experiments are given in Table 11 for ESB and ATA instances with 50 aircraft. Similar to the case in AGAP-WD&AA, RFH has clear advantage over FC with IFS in these instances as the former provides better solutions in 38 out of 40 instances within the given time limit. Moreover, on ICN instances, when we solve FC providing IFS′ solution as an initial solution to the solver, we do not get any improvement over IFS′ results in any of the ICN instances whereas RFH improves IFS′ results in all ICN instances. Thus, these results clearly show the value of using the restricted formulation over the full formulation with an initial solution.

5. Conclusion

We consider the airport gate assignment problem recently addressed in Karsu et al. (2021), which lexicographically minimizes the number of aircraft assignments to apron and the total passenger walking distance. We propose a novel flow-based mixed integer linear programming

Table 8
Comparison of heuristics on ESB instances for AGAP-WD.

Set	m	n	FBS		REICMPACO		REFBCO		PLFITS		IFS		RFH		
			WD	Time	WD	Time	WD	Time	WD	Time	WD	Time	WD	Time	Imp%
1	18	50	25,482.2	0.54	27,896.3	154.13	26,879.3	96.64	25,123.6	72.62	25,047.3	0.04	25,009.5	200.00	0.45
		100	91,556.9	7.20	93,256.8	114.65	91,641.3	112.87	87,640.2	85.21	85,353.1	0.05	85,340.0	200.00	2.62
		150	174,984.3	30.84	181,234.5	135.63	184,631.2	97.32	173,854.3	76.32	166,777.4	0.07	166,774.5	200.00	4.07
		200	344,250.2	91.32	365,953.1	126.74	354,213.6	73.21	342,005.3	115.65	336,439.2	0.10	336,439.2	200.00	1.63
2	18	50	81,056.4	0.36	85,214.6	102.52	83,542.4	85.62	78,345.6	76.27	77,569.7	0.03	77,499.3	52.40	1.08
		100	210,763.8	4.71	232,564.5	95.65	225,634.3	114.21	209,135.5	85.32	209,908.5	0.05	209,883.1	129.37	−0.36
		150	297,333.2	21.10	325,745.1	89.72	315,263.4	164.96	295,213.3	99.14	297,009.1	0.10	296,925.9	106.46	−0.58
		200	494,978.9	67.63	548,564.3	73.25	536,254.7	189.56	494,635.9	112.62	495,354.4	0.16	495,305.5	200.00	−0.14

Table 9
Comparison of heuristics on ATA instances for AGAP-WD.

Set	m	n	FBS		REICMPACO		REFBCO		PLFITS		IFS		RFH		
			WD	Time	WD	Time	WD	Time	WD	Time	WD	Time	WD	Time	Imp%
1	38	50	36,178.8	4.30	38,645.8	185.21	37,925.3	54.65	35,420.2	118.32	35,670.9	0.05	35,464.2	200.00	−0.12
		100	84,792.7	71.58	86,452.3	146.45	88,126.5	113.21	83,820.9	105.21	83,213.4	0.10	83,099.3	200.00	0.86
		150	124,072.7	361.24	135,483.1	139.69	128,362.2	184.62	120,532.1	65.85	118,367.3	0.40	118,334.8	200.00	1.82
		200	249,034.6	1094.92	276,532.9	95.54	254,563.2	129.63	236,523.2	95.32	229,177.5	1.02	229,121.1	200.00	3.13
2	38	50	58,165.0	4.24	61,235.6	182.32	57,698.5	132.52	56,127.5	78.58	54,577.3	0.05	54,525.5	8.45	2.85
		100	278,693.7	60.91	294,652.1	132.52	280,321.3	179.32	268,432.1	87.12	262,686.6	0.11	262,558.9	21.82	2.19
		150	443,563.6	281.09	463,252.3	112.87	456,325.6	185.62	435,216.3	76.53	435,538.9	0.18	435,449.6	48.44	−0.05
		200	798,524.5	804.25	812,369.5	142.37	823,641.2	156.27	796,231.2	143.95	793,971.7	0.30	793,691.9	154.34	0.32

Table 10
Comparison of heuristics on ICN instances for AGAP-WD.

π	Date	m	n	SATS		TSPR		REICMPACO		REFBCO		PLFITS		IFS′		RFH		
				WD	Time	WD	Time	WD	Time	WD	Time	WD	Time	WD	Time	WD	Time	Imp%
0.1	Friday	74	294	27,091,415	170	26,945,560	185	26,958,710	176.53	26,883,430	185.65	25,926,530	114.82	25,070,495	8.70	25,006,045	29.69	3.55
	Saturday	74	290	27,001,350	174	26,800,315	185	26,925,450	159.43	26,879,255	174.65	25,966,585	95.25	25,031,660	2.96	24,975,980	53.27	3.81
	Sunday	74	304	30,016,505	193	29,764,555	201	29,672,665	169.63	29,784,010	165.32	28,910,295	104.92	27,974,065	4.68	27,925,355	24.97	3.41
	Monday	74	297	27,554,290	185	27,668,210	207	27,370,585	158.52	27,054,520	186.65	26,699,195	122	25,880,105	2.13	25,829,915	33.78	3.26
	Tuesday	74	290	26,055,045	180	25,780,535	196	35,740,330	189.98	25,682,365	179.53	24,906,875	122.11	24,025,135	4.69	23,981,610	33.85	3.71
	Wednesday	74	279	25,092,430	151	24,875,240	173	24,552,775	198.92	24,883,320	200.01	24,227,945	127.04	23,416,075	5.63	23,382,520	21.76	3.49
0.3	Friday	74	294	29,325,270	184	29,274,210	196	28,452,540	185.45	29,330,105	175.65	27,158,465	143.94	26,288,910	8.78	26,226,690	85.41	3.43
	Saturday	74	290	29,378,545	194	29,272,310	224	29,020,050	167.14	28,940,075	165.41	26,977,360	117.11	26,201,750	3.21	26,060,650	200.00	3.40
	Sunday	74	304	31,690,910	199	31,642,165	226	31,745,625	178.63	31,578,310	184.63	29,673,555	136.52	28,794,790	2.66	28,665,125	78.65	3.40
	Monday	74	297	29,798,700	219	30,025,230	228	29,837,770	163.21	29,413,050	145.32	27,739,185	155.72	26,927,165	2.36	26,841,520	200.00	3.24
	Tuesday	74	290	28,050,095	189	27,898,320	211	27,894,450	167.58	27,596,960	167.34	25,712,685	131.25	24,967,940	3.26	24,838,515	170.85	3.40
	Wednesday	74	279	27,816,840	156	27,558,215	174	27,557,455	192.36	27,605,650	187.65	25,456,930	145.8	24,865,820	2.19	24,780,635	200.00	2.66
	Thursday	74	289	29,472,105	186	29,402,315	205	28,509,725	132.02	28,347,840	196.54	27,056,815	145.57	26,141,685	5.13	26,040,010	200.00	3.76
0.5	Friday	74	294	31,679,360	199	31,304,880	213	31,816,275	185.65	31,519,750	156.57	27,879,920	121.33	27,361,940	3.87	27,100,040	200.00	2.80
	Saturday	74	290	31,436,665	195	31,337,070	213	30,738,010	134.61	30,868,875	168.32	27,971,855	155.17	27,239,615	2.28	26,987,935	200.00	3.52
	Sunday	74	304	35,011,240	217	35,050,585	261	35,403,830	147.32	34,260,520	175.63	30,861,150	166.91	30,024,830	2.73	29,897,910	200.00	3.12
	Monday	74	297	31,989,310	213	31,900,725	234	30,875,445	169.31	29,872,170	197.47	28,559,395	165.01	27,883,800	2.65	27,750,140	200.00	2.83
	Tuesday	74	290	30,112,340	198	30,069,210	223	29,224,860	184.23	28,912,285	179.32	26,664,255	120.78	26,022,455	2.41	25,899,600	200.00	2.87
	Wednesday	74	279	29,751,435	175	29,667,010	201	27,259,495	173.65	28,135,610	185.65	26,457,175	134.41	25,920,735	2.35	25,729,615	200.00	2.75
	Thursday	74	289	31,896,375	194	31,755,335	228	29,323,555	197.61	29,274,900	163.54	28,087,180	155.12	27,322,105	3.13	27,186,485	200.00	3.21

formulation for the problem and a matheuristic, which is based on the idea of iteratively solving a small-size (restricted) formulation of the problem.

We first test the suggested formulation and matheuristic on problem instances used in Karsu et al. (2021) for the AGAP that lexicographically minimizes apron usage and passenger walking distance. The results reveal that the suggested formulations outperform the existing ones as well as the customized branch and bound procedure suggested in Karsu et al. (2021). We also demonstrate that for larger problem instances, which are generated based on the layouts of Ankara Esenboğa and İstanbul Atatürk airports, the proposed matheuristic finds solutions that are of better quality than the Filtered Beam Search proposed in Karsu et al. (2021) as well as solving the proposed formulation with a time limit.

Finally, we compare our matheuristic to the metaheuristic approaches for the single objective AGAP, where the only objective is minimizing total passenger walking distance. We compared our matheuristic with REFBCO in Dell'Orco et al. (2017), REICMPACO in Deng et al. (2019), FBS in Karsu et al. (2021), and PLFITS in Li et al. (2022) on benchmark instances (ESB and ATA) of Karsu et al. (2021) and also with SATS in Cheng et al. (2012), TSPR in Cheng et al. (2017), REFBCO in Dell'Orco et al. (2017), REICMPACO in Deng et al. (2019), and PLFITS in Li et al. (2022) on ICN instances of Cheng et al. (2012). PLFITS by Li et al. (2022) is reported to provide the current best-known results for these benchmark instances. The results of our computational experiments demonstrated that our matheuristic outperforms PLFITS and provides the new best-known solutions for all ICN instances and is comparable to PLFITS for ESB and ATA instances.

Future research can be conducted to address other variants of the problem with different objective functions and operational dynamics. We believe that our flow based formulation and matheuristic will also prove valuable in addressing airport gate assignment problem in a multiobjective setting that involves the passenger walking distance criterion.

Moreover, the formulation and the algorithm that we propose can be used for problems including factors such as resource availability and

Table 11
Performance of RFH compared to FC with IFS for AGAP-WD.

	Instance	RFH	FC with IFS		Instance	RFH	FC with IFS
			ESB				
Set 1	1	**23 947**	23 956	Set 2	1	**73 181**	73 203
	2	**26 803**	26 825		2	**69 446**	69 459
	3	23 379	**23 346**		3	**78 880**	78 925
	4	**26 455**	26 459		4	**81 148**	81 152
	5	**24 156**	24 202		5	**79 676**	79 689
	6	**23 358**	23 399		6	**75 570**	75 630
	7	**25 139**	25 184		7	**88 354**	88 365
	8	**24 879**	24 929		8	**80 888**	80 891
	9	**24 615**	24 617		9	**78 766**	78 799
	10	**27 364**	27 369		10	**69 084**	69 113
			ATA				
	Instance	RFH	FC with IFS		Instance	RFH	FC with IFS
Set 1	1	**34 917**	35 226	Set 2	1	**48 092**	48 120
	2	**34 051**	34 339		2	**48 852**	48 889
	3	**34 827**	35 081		3	**55 530**	55 836
	4	**36 685**	36 740		4	**62 451**	**62 451**
	5	**35 424**	35 574		5	**51 659**	51 676
	6	**36 010**	36 243		6	**47 743**	47 781
	7	**35 558**	35 789		7	**65 507**	65 533
	8	**34 475**	34 564		8	**57 612**	57 738
	9	**34 570**	34 686		9	**58 128**	58 139
	10	**38 125**	38 128		10	**49 681**	49 742

allocation, in addition to layout and size considerations. Specifically, newer technologies such as Internet of Things or Video Analytics can be employed to determine the resource use at the gates, which would then be used to incorporate these concerns into the formulations and make effective decisions. We believe that modifying our models so as to consider such resource allocation aspects is a promising future direction.

CRediT authorship contribution statement

Özlem Karsu: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Oğuz Solyalı:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

We would like to thank Mingjie Li and Chun-Hung Cheng for sharing ICN instance data.

References

Babić, O., Teodorović, D., Tošić, V., 1984. Aircraft stand assignment to minimize walking. J. Transp. Eng. 110 (1), 55–66.
Bihr, R.A., 1990. A conceptual solution to the aircraft gate assignment problem using 0, 1 linear programming. Comput. Ind. Eng. 19 (1–4), 280–284.
Bolat, A., 1999. Assigning arriving flights at an airport to the available gates. J. Oper. Res. Soc. 50 (1), 23–34.
Cai, X., Sun, W., Misir, M., Tan, K.C., Li, X., Xu, T., Fan, Z., 2019. A bi-objective constrained robust gate assignment problem: Formulation, instances and algorithm. IEEE Trans. Cybern. 51 (9), 4488–4500.
Cheng, C.-H., Gunasekaran, A., Ho, S.C., Kwan, C.-L., Ng, T.D., 2017. Hybrid tabu searches for effective airport gate management. Int. J. Oper. Res. 30 (4), 484–522.
Cheng, C.-H., Ho, S.C., Kwan, C.-L., 2012. The use of meta-heuristics for airport gate assignment. Expert Syst. Appl. 39 (16), 12430–12437.
Daş, G.S., 2017. New multi objective models for the gate assignment problem. Comput. Ind. Eng. 109, 347–356.
Daş, G.S., Gzara, F., Stützle, T., 2020. A review on airport gate assignment problems: Single versus multi objective approaches. Omega 92, 102146.
Dell'Orco, M., Marinelli, M., Altieri, M.G., 2017. Solving the gate assignment problem through the fuzzy bee colony optimization. Transp. Res. C 80, 424–438.
Deng, W., Xu, J., Zhao, H., 2019. An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. IEEE Access 7, 20281–20292.
Deng, W., Zhao, H., Yang, X., Xiong, J., Sun, M., Li, B., 2017. Study on an improved adaptive PSO algorithm for solving multi-objective gate assignment. Appl. Soft Comput. 59, 288–302.
Ding, H., Lim, A., Rodrigues, B., Zhu, Y., 2005. The over-constrained airport gate assignment problem. Comput. Oper. Res. 32 (7), 1867–1880.
Dorndorf, U., Drexl, A., Nikulin, Y., Pesch, E., 2007. Flight gate scheduling: State-of-the-art and recent developments. Omega 35 (3), 326–334.
Dorndorf, U., Jaehn, F., Pesch, E., 2008. Modelling robust flight-gate scheduling as a clique partitioning problem. Transp. Sci. 42 (3), 292–301.
Drexl, A., Nikulin, Y., 2008. Multicriteria airport gate assignment and pareto simulated annealing. IIE Trans. 40 (4), 385–397.
Erdoğan, G., Tansel, B., 2007. A branch-and-cut algorithm for quadratic assignment problems based on linearizations. Comput. Oper. Res. 34 (4), 1085–1106.
Haghani, A., Chen, M.-C., 1998. Optimizing gate assignments at airport terminals. Transp. Res. A. 32 (6), 437–454.
Jaehn, F., 2010. Solving the flight gate assignment problem using dynamic programming. Z. Betriebswirtschaft 80 (10), 1027–1039.
Karsu, Ö., Azizoğlu, M., Alanlı, K., 2021. Exact and heuristic solution approaches for the airport gate assignment problem. Omega 103, 102422.
Krishnamoorthy, M., Ernst, A.T., Baatar, D., 2012. Algorithms for large scale shift minimisation personnel task scheduling problems. European J. Oper. Res. 219 (1), 34–48.
Li, Y., Clarke, J.-P., Dey, S.S., 2021. Using submodularity within column generation to solve the flight-to-gate assignment problem. Transp. Res. C 129, 103217.
Li, M., Hao, J.-K., Wu, Q., 2022. Learning-driven feasible and infeasible tabu search for airport gate assignment. European J. Oper. Res. 302 (1), 172–186.
Maharjan, B., Matis, T.I., 2012. Multi-commodity flow network model of the flight gate assignment problem. Comput. Ind. Eng. 63 (4), 1135–1144.
Şeker, M., Noyan, N., 2012. Stochastic optimization models for the airport gate assignment problem. Transp. Res. E 48 (2), 438–459.
Tang, C.-H., Wang, W.-C., 2013. Airport gate assignments for airline-specific gates. J. Air Transp. Manag. 30, 10–16.
Wang, R., Allignol, C., Barnier, N., Gondran, A., Gotteland, J.-B., Mancel, C., 2022. A new multi-commodity flow model to optimize the robustness of the gate allocation problem. Transp. Res. C 136, 103491.
Xu, J., Bailey, G., 2001. The airport gate assignment problem: mathematical model and a tabu search algorithm. In: Proceedings of the 34th Annual Hawaii International Conference on System Sciences. p. 10.
Xu, L., Zhang, C., Xiao, F., Wang, F., 2017. A robust approach to airport gate assignment with a solution-dependent uncertainty budget. Transp. Res. B 105, 458–478.
Yan, S., Huo, C.-M., 2001. Optimization of multiple objective gate assignments. Transp. Res. A 35 (5), 413–432.
Yu, C., Zhang, D., Lau, H.Y.K., 2016. MIP-based heuristics for solving robust gate assignment problems. Comput. Ind. Eng. 93, 171–191.