Journal of Air Transport Management 115 (2024) 102529

Contents lists available at ScienceDirect

Journal of Air Transport Management

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

An airport gate reassignment problem with gate closures

Dursen Deniz Poyraz [a,b,*], Meral Azizoğlu [b]

[a] Rotterdam School of Management, Erasmus University, 3062 PA Rotterdam, Netherlands
[b] Department of Industrial Engineering, Middle East Technical University, Ankara, 06800, Turkey

ARTICLE INFO

Keywords:
Airport gate reassignment problem
Mixed integer linear programming
Nondominated objective vectors

ABSTRACT

In this study, we consider an airport gate reassignment problem where an airport has assigned gates to aircraft, but then a disruption occurs at some of the gates. After the disruption, we need to reassign the aircraft to the gates while taking into account both efficiency and stability measures. For efficiency, we want to use the gates as much as possible, considering both the number of aircraft and the number of passengers in these aircraft. For stability, we want to stick as closely as possible to the initial plan.

We suggest solution procedures for finding two extreme ends of the nondominated objective vectors, all extreme supported nondominated objective vectors, and all nondominated objective vectors with respect to our efficiency and stability measures. An optimal decomposition rule is presented to simplify the complexity of the solution. Our extensive experiments have shown that our optimization procedures can handle the instances with up to 150 aircraft and 40 gates, and approximation algorithms can handle the instances with up to 200 aircraft and 40 gates.

1. Introduction

Gates are important resources of airports whose proper allocations are crucial for effective air transport operations. In the airport gate assignment problem (AGAP), aircraft with prespecified arrival and departure times are assigned to available gates under some prespecified objective. Some noteworthy objectives studied in the literature are maximizing the number of aircraft assigned to gates, maximizing the number of passengers assigned to gates, and minimizing the walking distances of the passengers. Aircraft can also be assigned to a remote gate, the so-called apron. Apron is an area at the airport where the aircraft are parked if they cannot be assigned to any gate. The satisfactory solutions to gate assignment problems usually avoid assignments to the apron, due to its remote nature.

Daily airport operations may experience disturbances in various forms: flight earliness and delays, flight cancellations, maintenance operations, flight and gate breakdowns, adverse weather conditions, emergency flights, and even major incidents such as labor strikes of airport employees and abnormal meteorological conditions which may even result in temporary airport closures.

Additionally, airports have experienced a new type of disturbance owed to the recent corona virus disease. Existing studies in the literature are varied: its effects as potential aeropolitics issues (Macilree and Duval, 2020), understanding its implications on commercial revenue management through examining passenger spending drivers (Choi, 2021), how the aviation industry has bounced back and returned to normalcy after a shock period (Sun et al., 2023), the most influential factors on passenger satisfaction before and during the pandemic (Pereira et al., 2023). Sun et al. (2022) reviewed selected studies for a pandemic-resilient future in aviation.

As a result of such disruptions, the optimal solution to any gate assignment problem may become undesired or even infeasible to implement. Hence, a need to reassign the aircraft to the gates arises. We consider a gate reassignment problem (AGRP) where the aircraft are already assigned to the gates or to the apron, and a disruption that affects a subset of the gates occurs. After the disruption, the aircraft assigned to the disrupted gates should be shifted to the remaining available gates or the apron. This shift may also trigger some assignment changes for the aircraft of the nondisrupted gates to give room for the aircraft of the disrupted gates. Such adjustments, which are referred to as reassignment, should ensure efficiency en route to low airport operating costs and stability en route to low setup costs.

The original assignment of aircraft to the gates or to the apron before any disruption is referred as the initial plan. We assume that the initial plan was efficient, so our new plan should have its objective function as an efficiency measure. Moreover, we assume that the preparations are already made according to the initial plan, hence the new plan should stay faithful to the initial one.

In our efficiency criterion, we focus on an airport's vital need of utilizing its gate resources most efficiently. Gates are used both by

* Corresponding author at: Rotterdam School of Management, Erasmus University, 3062 PA Rotterdam, Netherlands.
E-mail addresses: poyraz@rsm.nl (D.D. Poyraz), ma@metu.edu.tr (M. Azizoğlu).

https://doi.org/10.1016/j.jairtraman.2023.102529
Received 22 May 2023; Received in revised form 22 November 2023; Accepted 13 December 2023
Available online 29 December 2023

aircraft and by passengers. We define our efficiency criterion through two objectives with hierarchy. The first objective that goes into our efficiency criterion is the maximization of the number of aircraft assigned to gates. With this objective, the number of ungated aircraft is minimized, i.e., the gate utilization is maximized, thus the most efficient assignment plan is obtained in terms of apron related costs. The second objective that goes into our efficiency criterion is the maximization of the number of passengers assigned to gates. By assigning the aircraft with a higher number of passengers to a gate, a contribution to the minimization of passenger walking distance or general customer satisfaction, since a smaller number of passengers will be routed to the remote apron, is inherently being made. Therefore, with our efficiency criteria, we maximize primarily the number of aircraft assigned to gates and secondarily the number of passengers in these aircraft. In doing so, we consider aeronautical concerns, i.e., concerns that are directly related to the passengers and aircraft; but not non-aeronautical concerns, such as airport retailing, advertising, car rentals, car parking, and land rentals. We refer the reader to the review by Kidokoro and Zhang (2023) for the detailed discussion of the non-aeronautical objectives and their importance. Furthermore, we show that the gate assignment problem that minimizes our efficiency measure is solvable in polynomial time.

In our stability criterion, as the name suggests, we focus on staying stable, i.e., the new plan should resemble the initial plan as much as possible, preserving most of the initial assignments that remain feasible. In the event of a disruption, first and foremost, we aim to reassign the already gated aircraft to a gate again. We believe that reassigning an already gated aircraft to the apron, will give a high deviation from the initial plan in terms of similarity to the initial plan and passenger discomfort. Secondly, we would like to focalize on the number of passengers in the already assigned set of aircraft. This consideration is in parallel with that of the secondary objective of the efficiency criterion. Lastly, from an opportunistic point of view, we would like to reassign the ungated aircraft to a gate whenever possible. With the decreased apron usage, we would be reaping the benefits of increased efficiency. Thus, a similar reassignment plan to the initial plan is obtained through considerations in three folds: we maximize the number of aircraft reassigned to gates that were initially assigned to gates as the primary objective, the number of passengers in these aircraft as the secondary objective, and the number of aircraft reassigned to gates that were initially assigned to the apron as the tertiary objective. We show that the gate assignment problem that minimizes our stability measure is NP-hard.

Our performance measures are studied in a multicriteria context as an increase in the efficiency value would lead to a decrease in the stability value, and vice versa, hence a very fruitful trade-off analysis could be made. Recognizing this fact, we study several trade-off problems.

Firstly, a hierarchical optimization is considered such that the efficiency (stability) value is maximized while the stability (efficiency) value is kept at its optimal. This approach returns two extreme points and is important for decision makers who have a strong preference for one objective.

Secondly, we assume that the decision maker has a linear preference function that is expressed as a weighted combination of the criteria. We produce a set of objective vectors each of which is optimal for a particular weight range. These objective vectors altogether form the extreme supported nondominated objective vectors.

Finally, we assume that the decision maker has an unknown utility function for efficiency and stability criteria. We produce the set of nondominated objective vectors, one of which is optimal for a particular nonincreasing utility function. Using this set, the decision maker can make a trade-off between a certain amount of increase in efficiency value and a certain amount of decrease in stability value, and vice versa. All trade-off problems of our concern are NP-hard since the problem that minimizes our stability measure is NP-hard.

We present a model-based optimization approach to generate the exact set of nondominated objective vectors and a heuristic approach to generate an approximate set of nondominated objective vectors. We also develop a decomposition rule where the problem is decomposed into subproblems, where each of which is dealt with independently, and then their corresponding solutions are combined by a mathematical model. We observe that if one is faced with instances for which the decomposition rule can be applied, the exact nondominated objective vectors can be found considerably easier.

Our experiments, where we chose to set the number of gates almost equal to that of the airports in the three largest cities in Turkey: İstanbul Airport in İstanbul, Esenboğa Airport in Ankara, and İzmir Adnan Menderes Airport in İzmir, namely, have shown that the exact approaches can be used to tackle real-life instances with many aircraft and many gates.

Related literature on AGAP and AGRP is reviewed in Section 2. In Section 3, our AGRP is defined, and a mathematical model is given. The solution procedures that are used to find two extreme nondominated objective vectors, all extreme supported nondominated objective vectors and all nondominated objective vectors are presented in Section 4. Section 5 defines the optimal decomposition rule that decomposes the main problem into smaller instances. In Section 6, the results of our extensive experiments are reported. Conclusions and suggestions for future research are made in Section 7.

2. Literature review

We give literature reviews on AGAP and AGRP in Sections 2.1 and 2.2, respectively.

2.1. Airport gate assignment studies

In a review by Daş et al. (2020) on AGAP, many objective functions of this assignment problem are classified under some defined categories. Karsu et al. (2021) studied an AGAP with two minimization objectives: the number of apron assignments and total passenger walking distance. They give exact and heuristic solution approaches. Their problem instances mimic the real-life airports in Turkey.

Yan and Chang (1998) formulated AGAP as a multi-commodity network flow problem to minimize total passenger walking distance. They used real-life data from an international airport in Taiwan. Another multi-commodity flow model with two objectives (robustness and taxiing times) is developed by Wang et al. (2022). They used real-life data from the Paris-Charles-de-Gaulle international airport in France. Cai et al. (2019) worked to minimize the total passenger walking distance and the total robust cost. They put an upper limit on the number of aircraft that can be assigned to the apron and defined compatibility related constraints such as gate sizes: small/large, gate and airline leasing contracts, and flight types. They made an application for the Baiyun airport in Guangzhou, China.

Yu et al. (2017b) focused on the robustness and some traditional costs: the expected conflict cost, tow cost, and passenger transfer distance. They designed an adaptive large neighborhood search with some novel multiple local search operators. Liu et al. (2023) focused on gate utilization and running time of aircraft including parking time and taxi time.

2.2. Airport gate reassignment studies

The gate reassignment problem literature mostly considers multi-objective studies. Pternea and Haghani (2019) proposed hierarchical optimization for a gate reassignment problem to minimize the costs of (i) flight assignment, (ii) successful passenger connections, and (iii) failed passenger connection. Zhang and Klabjan (2017) defined an efficient gate reassignment methodology for disruptions. They come up with two multi-commodity network flow models and two heuristic algorithms. They handle the minimization of total flight delays, the number of gate reassignment operations, total passenger transfer distance, and the number of missed passenger connections with a weighted sum approach.

Dorndorf et al. (2012) studied the multi objectives: some assignment preference scores, the number of unassigned flights during overload periods, the number of tows, some robustness measures, and the one most familiar to our work which is a deviation from a given reference schedule. Yan et al. (2011) assumed to handle the uncertainty around aircraft arrival and departure times: that some flights which are closer to the time of planning reassignments tend to be more certain. That is why they divided flights into the following two categories: deterministic flights and stochastic flights. They made an application to the Taiwan Taoyuan Airport in Taiwan.

Flight delays, whether in the form of early or tardy flight arrivals or the form of tardy flight departures, in general, constitute the disruptions worked on by Tang et al. (2010), where they emphasized the crucial need of developing a framework for the gate reassignment problem, stating that the traditional manual flight reassignment method has too many shortcomings.

In their study, Deng et al. (2017) worked with multi-objectives that take into consideration the loss of passengers, cost of airport operating, and economic loss of airlines, in one criterion and for the other criterion, constructed a measure called the most important index of disturbance value to manage the deviation from the initial plan. They integrated two metaheuristics: the genetic algorithm and the ant colony algorithm to propose a two-stage hybrid method.

Wang et al. (2013) handled flight delays in two categories: certain and uncertain delays. For the former case, they minimize the apron and gate disturbance values and for the latter case, they minimize the gate and time disturbance values. They presented an ant colony-based heuristic. Maharjan and Matis (2011) considered the passengers who are either connecting to or originating from an airport where their boarding passes were issued before the gate reassignments. They implemented their work with Continental Airlines at the George W. Bush Intercontinental Airport in Houston, Texas.

Further literature include Pternea and Haghani (2018) who studied passenger connections, Yan et al. (2009) who assumed major events that result in temporary airport closures, Gu and Chung (1999) who implemented a genetic algorithm for the minimization of extra delay times, Yu et al. (2017a) who also integrated taxiway scheduling and Ali et al. (2019) where they proposed a passenger-centric model that minimizes the transit time of transfer passengers.

The most closely related study to ours is Pternea and Haghani (2019) where they used hierarchical optimization approaches to handle their multi criteria problems. In addition to the hierarchical optimization approaches, we deal with simultaneous optimization and generate all extreme supported nondominated and all nondominated objective vectors with respect to our efficiency and stability criteria. To the best of our knowledge, no reported gate reassignment study considers the performance measures simultaneously.

3. Problem definition and the mathematical model

We study an AGRP with n aircraft, m gates, and an apron. Each aircraft has specified arrival and departure times and is either assigned to a gate or the apron from its arrival time to its departure time. This is compatible with the real-life application of renting gates to airlines for fixed periods, i.e., time intervals. Moreover, each aircraft has a specified number of passengers who have either entered from the entrance point or transferred from other aircraft. The apron, assumed to have infinite aircraft capacity, is so far away from the gates that using it is not favored for any reason. Thus, only the aircraft that cannot be assigned to a gate, are assigned to the apron.

We assume that there is an initial plan where each aircraft is assigned to either a gate or to the apron. A disruption at the beginning of the planning horizon affects a specified set of gates and makes them inoperable. After the disruption, a new plan is formed where the affected aircraft are assigned to one of the not-affected gates or apron. The not-affected aircraft may be reassigned to their initial gates or any one of the not-affected gates or apron. We use the following terms in the literature interchangeably: 'initial plan', 'current assignment', 'initial assignment'; 'affected gates', 'disrupted gates'; 'affected aircraft', 'disrupted aircraft'; 'new plan', 'reassignment'.

We further assume that there is no assignment restriction, i.e., all aircraft can be assigned to one of the m gates. Moreover, the arrival and departure times of aircraft, the number of passengers, and all other parameters are known with certainty and not subject to any change. That is, the system we consider is deterministic and static.

We define efficiency and stability criteria as our performance measures and assume that the initial plan is known and found by the efficiency concerns of the decision makers.

In Sections 3.1 and 3.2, our efficiency and stability criteria are discussed, respectively. In Section 3.3, the mathematical model is provided.

3.1. Efficiency criterion

From an airport and a passenger point of view, an assignment plan should meet the following requirements: i. its number of aircraft assigned to gates should be as high as possible, and ii. if there is a tie among multiple aircraft that are potentially competing against each other to be assigned to a gate, then the decision should be in favor of the one with the highest number of passengers.

We call such an assignment plan *efficient* due to it having the least apron usage both by aircraft and their corresponding passengers.

The efficiency criterion is defined through the objective function, E, with the direction of maximization.

$E1$	Number of aircraft assigned to gates, the primary objective
$E2$	Number of passengers assigned to gates, the secondary objective

$E = E1 + \varepsilon_E E2$ where ε_E is a sufficiently small number that gives priority to $E1$ and breaks the ties in favor of $E2$. That is, ε_E ensures that the maximum number of aircraft assigned to gates is preserved while the number of passengers assigned to gates is being maximized. Hence, ε_E should be set so small that the $E1$ value does not decrease even by one unit for the highest improvement of the $E2$ value, Eq. (1). Note that, in addition to establishing the hierarchy between objective functions, the parameter ε_E also performs a rescaling between the objective functions, where one objective is in units of aircraft and the other is in units of passengers.

$$E1 + \varepsilon_E E2_{min} \geq E1 - 1 + \varepsilon_E E2_{max} \quad (1)$$

where $E2_{min}$ is the smallest possible $E2$ value and $E2_{max}$ is the largest possible $E2$ value.

We define p_i as the number of passengers in aircraft i. We estimate $E2_{min}$ to be the case where all gates are busy with exactly one aircraft, so $E2_{min} = \sum_{i=1}^{m} p_{[i]}$ where $p_{[i]}$ is the ith smallest p_i value, i.e., it is the summation of the smallest m p_i values. We estimate $E2_{max}$ to be the case where all aircraft are assigned to gates, so $E2_{max} = \sum_{i=1}^{n} p_i$.

Eq. (1) reduces to $\varepsilon_E E2_{min} \geq \varepsilon_E E2_{max} - 1$ and subsequently, $\varepsilon_E \leq \frac{1}{E2_{max} - E2_{min}}$. Putting $E2_{min}$ and $E2_{max}$ into this expression, we have $\varepsilon_E \leq \frac{1}{\sum_{i=1}^{n} p_i - \sum_{i=1}^{m} p_{[i]}}$.

In our experiments, we set

$$\varepsilon_E = \frac{1}{\sum_{i=1}^{n} p_i - \sum_{i=1}^{m} p_{[i]} + 1}. \quad (2)$$

Putting Eq. (2) into E, we get $E = E1 + \frac{1}{\sum_{i=1}^{n} p_i - \sum_{i=1}^{m} p_{[i]} + 1} E2$. To have an integer value for E, we multiply it by $\sum_{i=1}^{n} p_i - \sum_{i=1}^{m} p_{[i]} + 1$ and get the following expression for our efficiency criterion, $E = \left[\sum_{i=1}^{n} p_i - \sum_{i=1}^{m} p_{[i]} + 1\right] E1 + E2$.

3.2. Stability criterion

In the AGRP, the new plan after a disruption can have some resemblance to the initial plan. With our stability criterion, the new plan stays faithful to the initial plan, and the number of gate assignments preserved, their corresponding number of passengers, and the number of gate assignments that were initially assigned to the apron are maximized in this very order.

S_1	Set of aircraft that are assigned to gates in the initial plan
S_2	Set of aircraft that are assigned to the apron in the initial plan
I	Set of all aircraft, $S_1 \cup S_2$
$ST1$	Number of aircraft in S_1 assigned to their initial gates
$ST2$	Number of passengers in aircraft in S_1 assigned to their initial gates
$ST3$	Number of aircraft in S_2 assigned to gates

Our stability aim is primarily to maximize $ST1$. Among the optimal solutions of $ST1$, we prefer the one having the maximum $ST2$. ε_{ST1} is a number that guarantees that the maximum number of aircraft is assigned to their initial gates while the number of passengers assigned to their initial gates is being maximized. Hence, we first want to maximize the partial objective function, $ST1+\varepsilon_{ST1}ST2$, where ε_{ST1} is a sufficiently small number so that the $ST1$ value does not reduce even by one unit for the highest improvement of the $ST2$ value, Eq. (3).

$$ST1 + \varepsilon_{ST1}ST2_{min} \geq ST1 - 1 + \varepsilon_{ST1}ST2_{max} \tag{3}$$

where $ST2_{min}$ and $ST2_{max}$ are the smallest and largest possible $ST2$ values, respectively. We estimate $ST2_{min}$ to be the case where all aircraft are assigned to the apron, so $ST2_{min} = 0$. We estimate $ST2_{max}$ to be the case where no aircraft is disrupted, so $ST2_{max} = \sum_{i\in S_1} p_i$.

Eq. (3) reduces to $\varepsilon_{ST1}ST2_{min} \geq \varepsilon_{ST1}ST2_{max} - 1$ and then $\varepsilon_{ST1} \leq \frac{1}{ST2_{max}} = \frac{1}{\sum_{i\in S_1} p_i}$.

In our experiments, we set

$$\varepsilon_{ST1} = \frac{1}{\sum_{i\in S_1} p_i}. \tag{4}$$

Putting Eq. (4) into the objective function, we get $ST1+\frac{1}{\sum_{i\in S_1} p_i}ST2$. We multiply it by $\sum_{i\in S_1} p_i$ to get an integer value for the objective function as $\left(\sum_{i\in S_1} p_i\right)ST1 + ST2 = ST_A$.

Among the optimal solutions to ST_A, we prefer the one having the largest $ST3$ value. ε_{ST2} is a number that ensures that the optimality of ST_A is preserved while the number of aircraft in $S2$ assigned to gates is being maximized. Hence, we maximize $ST = ST_A + \varepsilon_{ST2}ST3$, where ε_{ST2} is a sufficiently small number which is found by using the ideas of ε_E and ε_{ST1}, Eq. (5). Similarly, parameters ε_{ST1} and ε_{ST2} also perform rescaling between the objective functions as well as handling the hierarchy between them.

$$ST_A + \varepsilon_{ST2}ST3_{min} \geq ST_A - 1 + \varepsilon_{ST2}ST3_{max} \tag{5}$$

where $ST3_{min}$ is the smallest possible $ST3$ value and $ST3_{max}$ is the largest possible $ST3$ value. We take $ST3_{min}$ as 0, i.e., it is the case where all aircraft are assigned to the apron. We take $ST3_{max}$ to be the case where all apron assignments in the initial plan are shifted to gates, so $ST3_{max} = |S_2|$, i.e., it is the number of aircraft assigned to the apron in the initial plan. Eq. (5) reduces to $\varepsilon_{ST2}ST3_{min} \geq \varepsilon_{ST2}ST3_{max} - 1$ and subsequently, $\varepsilon_{ST2} \leq \frac{1}{ST3_{max}} = \frac{1}{|S_2|}$.

In our experiments, we set

$$\varepsilon_{ST2} = \frac{1}{|S_2|+1}. \tag{6}$$

Putting Eq. (6) into the objective function, the overall stability measure is expressed as $ST = ST_A + \frac{1}{|S_2|+1}ST3$. To have an integer value for ST, we multiply it by $|S_2|+1$ and get the following expression for our overall stability criterion:

$$ST = \left(|S_2|+1\right)\left(\sum_{i\in S_1} p_i\right)ST1 + \left(|S_2|+1\right)ST2 + ST3.$$

Our efficiency and stability measures altogether provide a broad perception to cover many efficiency and customer satisfaction objectives of the airlines. Recently, Yu (2023) defines three categories for airport performance: productivity and efficiency; financial performance; and service quality and passenger satisfaction. One leg of our stability measure, the number of passengers assigned to different gates in the initial and new plans, falls under the passenger satisfaction category, given that passengers would be dissatisfied with a changed gate. Moreover, passengers of ungated aircraft are also dissatisfied. This follows, one leg of our efficiency measure, the number of passengers assigned to gates, falls under the passenger satisfaction category as well. The other legs of our performance measures belong to the category of productivity and efficiency.

3.3. Mathematical model

We give the mathematical model for our AGRP. The model is an assignment-based model that is also used by Karsu et al. (2021) for their AGAP. It uses the following sets and parameters:

I	Set of aircraft
K	Set of gates (apron included)
n	Number of aircraft ($\|I\|$)
m	Number of gates ($\|K\|-1$ gates, gate $m+1$ is apron)
p_i	Number of passengers in aircraft i, $i = 1,\dots,n$
a_i	Arrival time of aircraft i, $i = 1,\dots,n$
d_i	Departure time of aircraft i, $i = 1,\dots,n$
R	Number of distinct a_i and d_i values, where $R-1$ is the number of time intervals

Set $\{ad_1, ad_2, \dots, ad_R\}$ is the distinct a_i and d_i values in chronological order. During interval $[a_i, d_i]$, aircraft i stays at the airport and during interval $[ad_r, ad_{r+1}]$ there is no arrival or departure, $r = 1,\dots,R-1$.

$$o_{ir} = \begin{cases} 1, & \text{if aircraft } i \text{ is in the airport at interval } r, \\ & \text{r} = 1,\dots,\text{R}-1 \\ 0, & \text{otherwise} \end{cases}$$

We define the stability related parameter, c_{ik}, as:

$$c_{ik} = \begin{cases} 1, & \text{if aircraft } i \text{ is assigned to gate } k \text{ in the} \\ & \text{initial plan, } i = 1,\dots,n \;\; k = 1,\dots,m+1 \\ 0, & \text{otherwise} \end{cases}$$

The assignment decision variable, x_{ik}, is defined as:

$$x_{ik} = \begin{cases} 1, & \text{if aircraft } i \text{ is assigned to gate } k \text{ in the} \\ & \text{new plan, } i = 1,\dots,n \;\; k = 1,\dots,m+1 \\ 0, & \text{otherwise} \end{cases}$$

The constraints are as given below:

$$\sum_{k=1}^{m+1} x_{ik} = 1 \qquad i = 1,\dots,n \tag{A}$$

$$\sum_{i=1}^{n} o_{ir}x_{ik} \leq 1 \qquad k = 1,\dots,m \;\; r = 1,\dots,R-1 \tag{B}$$

$$x_{ik} \in \{0,1\} \qquad i = 1,\dots,n \;\; k = 1,\dots,m+1 \tag{C}$$

Constraint set (A) ensures each aircraft i is assigned to a single gate. The overlapping of the aircraft is handled by binary parameter o_{ir}, where in constraint set (B) aircraft that are in the system at the same time interval cannot be assigned to the same gate. Lastly, Constraint set (C) states that the decision variable x_{ik} is binary.

In our efficiency problem, $E1$, the number of gated aircraft $\sum_{i=1}^{n}\sum_{k=1}^{m} x_{ik}$, and $E2$, their corresponding number of passengers $\sum_{i=1}^{n}\sum_{k=1}^{m} p_i x_{ik}$, are primarily and secondarily maximized, respectively. We give the integer-valued aggregate objective function for the efficiency criterion, $E_A = \frac{1}{\varepsilon_E}\sum_{i=1}^{n}\sum_{k=1}^{m} x_{ik} + \sum_{i=1}^{n}\sum_{k=1}^{m} p_i x_{ik}$ where $\varepsilon_E = \frac{1}{\sum_{i=1}^{n} p_i - \sum_{i=1}^{m} p_{[i]} + 1}$.

Karsu et al. (2021) showed that the maximization of the number of aircraft assigned to gates, $\left(\max \sum_{i=1}^{n}\sum_{k=1}^{m} x_{ik}\right)$, is solved in polynomial time using a network flow model. The same network structure holds when the arc costs with '1' that emanate from the node representing aircraft i are replaced by $\frac{1}{\varepsilon_E} + p_i$. This follows that our efficiency problem can also be solved in polynomial time.

In our stability problem, the objective functions $ST1$, $ST2$, and $ST3$, (the number of aircraft-gate assignments preserved $\sum_{i=1}^{n}\sum_{k=1}^{m} c_{ik} x_{ik}$, their corresponding number of passengers $\sum_{i=1}^{n}\sum_{k=1}^{m} p_i c_{ik} x_{ik}$, and the number of aircraft assigned to gates that were initially assigned to apron $\sum_{i|c_{im+1}=1}\sum_{k=1}^{m} x_{ik}$) are maximized primarily, secondarily, and tertiarily, respectively. We give the integer-valued aggregate objective function for the stability criterion as follows:

We give the integer-valued aggregate objective function for the stability criterion as follows:

$$ST_A = \frac{1}{\varepsilon_{ST1}\varepsilon_{ST2}}\sum_{i=1}^{n}\sum_{k=1}^{m} c_{ik}x_{ik} + \frac{1}{\varepsilon_{ST2}}\sum_{i=1}^{n}\sum_{k=1}^{m} p_i c_{ik} x_{ik} + \sum_{i|c_{im+1}=1}\sum_{k=1}^{m} x_{ik}$$

where $\varepsilon_{ST1} = \frac{1}{\sum_{i=1}^{n}\sum_{i=1}^{m} c_{ik}+1}$ and $\varepsilon_{ST2} = \frac{1}{\sum_{i=1}^{n} c_{im+1}+1}$.

Jaehn (2010) shows that the maximizing the total aircraft-gate score $\left(\max \sum_{i=1}^{n}\sum_{k=1}^{m} p_{ik} x_{ik}\right)$ problem is NP-hard. Our stability measure reduces the total aircraft-gate score when $p_{ik} = \frac{1}{\varepsilon_{ST1}\varepsilon_{ST2}} c_{ik} + \frac{1}{\varepsilon_{ST2}} p_i c_{ik} + 1$. This follows that our stability problem is also NP-hard. We refer to the constraint sets (A), (B), and (C) as $x \in X_A$ and state the mathematical model as:

$$\begin{aligned} \max\quad & E_A \\ \max\quad & ST_A \\ \text{subject to}\quad & x \in X_A \end{aligned}$$

A gate assignment solution r in $x \in X_A$ is called an efficient solution if there is no other solution q in $x \in X_A$ where $E_q \geq E_r$ and $ST_q \geq ST_r$ with strict inequality holding at least once ($E_q > E_r$ and $ST_q \geq ST_r$ or $E_q \geq E_r$ and $ST_q > ST_r$). The associated objective vector (E_r, ST_r) is said to be a nondominated objective vector (ndov). The solution q is dominated by solution r and the objective vector (E_q, ST_q) is dominated by the nondominated objective vector (E_r, ST_r).

A nondominated objective vector is called an extreme nondominated objective vector if it has the largest objective function value for one objective, in a maximization problem.

Our efficiency and stability concerns constitute two different perspectives on the problem. Hence, their solutions are treated as the two extreme ends of a solution spectrum.

An efficient solution is called a supported efficient solution if it is optimal for the linear combination of E and ST, i.e., $wE + (1-w)\,ST$ for any positive w. If an efficient solution does not optimize $wE + (1-w)\,ST$ for all positive w, then it is unsupported efficient.

A supported efficient solution is called an extreme supported efficient solution if it can be found by changing the value of w. A supported efficient solution is called a nonextreme supported efficient solution if it is a linear combination of two extreme supported efficient solutions.

Table 1
Summary of the objective vectors.

Objective vector	Definition
Nondominated objective vector (ndov)	Set of objective vectors that are not dominated by any other solution
Extreme nondominated objective vector	Nondominated objective vectors with the maximum values in any one of the objectives
Supported nondominated objective vector	Nondominated objective vector that is optimal for a linear combination of two objectives
Extreme Supported nondominated objective vector	Supported nondominated objective vector that is found by a linear combination of two objectives
Unsupported nondominated objective vector	Nondominated objective vector that does not optimize any linear combination of two objectives

The nondominated objective vectors corresponding to supported, unsupported, extreme supported, and nonextreme supported efficient solutions are referred to as supported, unsupported, extreme supported, and nonextreme supported nondominated objective vectors, respectively. In Table 1, we summarize the objective vectors, for the sake of completeness.

4. Finding the nondominated objective vectors

In this section, the generation of the nondominated objective vectors is discussed. In Sections 4.1 and 4.2, algorithms to generate extreme nondominated objective vectors are provided. The generation of all exact nondominated objective vectors and approximate set of nondominated objective vectors are discussed in Sections 4.3 and 4.4, respectively. The problems that are studied in Sections 4.2 and 4.3 are all NP-hard due to the NP-hardness of our stability problem.

4.1. Finding the extreme nondominated objective vectors

We discuss the generation of the extreme nondominated objective vectors with the largest E and the largest ST values in Sections 4.1.1 and 4.1.2, respectively.

4.1.1. Finding the extreme nondominated objective vector with the largest E value

Consider the following problem:

$$\begin{aligned} \max\quad & E \\ \text{subject to}\quad & x \in X_A \end{aligned}$$

Let E^* be the optimal objective function value. E^* is an upper bound on the efficiency values of all nondominated objective vectors. However, any feasible solution with an efficiency value of E^* does not necessarily comprise a nondominated objective vector since there may exist another feasible solution with a larger ST value. The solution having the maximum ST value among the solutions having an efficiency value of E^* can be found using the two-step Procedure 1.

Procedure 1	Finding an extreme nondominated objective vector with the largest E value
Input:	Arrival and departure times of the aircraft, initial plan
Step 1.	Solve $\max\ E$ subject to $x \in X_A$ Let E_1^* be the optimal objective function value.
Step 2.	Solve $\max\ ST$ subject to $E = E_1^*$ and $x \in X_A$ Let ST_1^* be the optimal objective function value.
Output:	Extreme nondominated objective vector with the largest E value, (E_1^*, ST_1^*) and the new plan

4.1.2. Finding the extreme nondominated objective vector with the largest ST value

Consider the following problem:

$$\begin{aligned} \max \quad & ST \\ \text{subject to} \quad & x \in X_A \end{aligned}$$

In the optimal solution, the initial plan should be implemented to its greatest extent. To achieve this, we fix the aircraft-gate assignments that are not affected by disruptions. In doing so, the first and second terms of the stability objective are readily maximized. Hence, a great emphasis is put on fixing the aircraft-gate assignments that are not affected by disruptions.

By keeping the initial plan for the aircraft, that were initially assigned to gates and are not affected by disruptions, we ensure that the new plan will be the most faithful one to the initial plan. To this end, we preprocess our stability problem and create a so-called reduced problem to work with in Procedure 2.

Procedure 2	Creating a reduced problem for the stability problem
Input:	Arrival and departure times of the aircraft, initial plan
Step 1.	Fix the aircraft-gate assignments that are not affected by disruptions.
Step 2.	Among the remaining unassigned aircraft, find which ones can only be assigned to the apron, and assign these aircraft to the apron.
Step 3.	Among the remaining unassigned aircraft, find which ones have a one-to-one relationship with an available time interval at a gate, i.e., an aircraft can only be assigned to a specific gate and this specific gate have no other possible unassigned aircraft for said open time interval. Make this a one-to-one assignment. The remaining unassigned aircraft and their possible gate assignments constitute the reduced problem.
Output:	A reduced problem and a partial assignment plan

We define a network for each aircraft in the reduced problem, where arcs represent possible aircraft-gate assignments. Note that each aircraft has a set of eligible gates due to the partial assignment plan.

n'	Number of aircraft in the reduced problem
$S(i)$	Set of gates that are eligible for aircraft $i, i = 1, \ldots, n'$

The objective function of the stability problem for the reduced problem is to maximize $ST + \epsilon'_{ST} E$ where $ST = \frac{1}{\epsilon_{ST1}\epsilon_{ST2}} ST1 + \frac{1}{\epsilon_{ST2}} ST2 + ST3$, $E = \frac{1}{\epsilon_E} E1 + E2$, $\epsilon'_{ST} = \frac{1}{E_{max} - E_{min} + 1}$, E_{min} is the smallest possible E value, and E_{max} is the largest possible E value considering the set of aircraft in the reduced problem, $i = 1, \ldots, n'$. We estimate E_{min} to be the case where all aircraft in the reduced problem are assigned to the apron, so $E_{min} = 0$. We estimate E_{max} to be the case where all aircraft in the reduced problem are assigned to gates, so $E_{max} = \frac{1}{\epsilon_E} n' + \sum_{i=1}^{n'} p_i$.

The constraint sets are as follows:

$$\sum_{k \in S(i)} x_{ik} = 1 \qquad i = 1, \ldots, n'$$

$$\sum_{i=1}^{n'} o_{ir} x_{ik} \le 1 \quad \forall k \in \cup_{i=1,\ldots,n'} S(i) \quad r = 1, \ldots, R-1$$

$$x_{ik} \in \{0, 1\} \qquad i = 1, \ldots, n' \quad \forall k \in \cup_{i=1,\ldots,n'} S(i)$$

Let (E', ST') be the optimal objective vector of the reduced problem. Extreme nondominated objective vector with the largest ST value, (E_2^*, ST_2^*), is found by also factoring in the partial assignment plan. $E_2{}^* = E' + \frac{1}{\epsilon_E} E1 + E2$ and $ST = \frac{1}{\epsilon_{ST1}\epsilon_{ST2}} ST1 + \frac{1}{\epsilon_{ST2}} ST2 + ST3$ where $\frac{1}{\epsilon_E} E1 + E2$ and $\frac{1}{\epsilon_{ST1}\epsilon_{ST2}} ST1 + \frac{1}{\epsilon_{ST2}} ST2 + ST3$ are the efficiency and stability of the partial assignment plan, and ϵ_E, ϵ_{ST1}, and ϵ_{ST2} are parameters of the main problem.

4.2. Finding the extreme supported nondominated objective vectors

The extreme supported nondominated objective vectors are found through the following objective function: $w\left(\frac{E - E_{min}}{E_{max} - E_{min}}\right) + (1 - w)\left(\frac{ST - ST_{min}}{ST_{max} - ST_{min}}\right)$ where (E_{max}, ST_{min}) is the 1st extreme nondominated objective vector, and (E_{min}, ST_{max}) is the 2nd extreme nondominated objective vector. Through this scaling, the same set of extreme supported vectors as without performing any scaling is found but with more dispersed weights.

For the sake of simplicity, we write our scaled objective function as $wE_{scaled} + (1 - w)ST_{scaled}$ where $E_{scaled} = \frac{E - E_{min}}{E_{max} - E_{min}}$ and similarly, $ST_{scaled} = \frac{ST - ST_{min}}{ST_{max} - ST_{min}}$.

We adapt the method used by Özlen and Azizoğlu (2009) to generate all extreme supported nondominated objective vectors in Procedure 3.

We start by finding the two extreme nondominated objective vectors either as done in Sections 4.2 and 4.3 or by setting $w = 1$ and $w = 0$, respectively.

Then, we solve the following relation to finding a range for w : $wE_{scaled}(1) + (1 - w)ST_{scaled}(1) = wE_{scaled}(2) + (1 - w)ST_{scaled}(2)$ where $E_{scaled}(i)$ and $ST_{scaled}(i)$ are E_{scaled} and ST_{scaled} values of the ith extreme nondominated objective vector, respectively. Rearranging the terms, we get $w = \frac{ST_{scaled}(2) - ST_{scaled}(1)}{ST_{scaled}(2) - ST_{scaled}(1) + E_{scaled}(1) - E_{scaled}(2)}$.

In ranges $[0, w)$ and $(w, 1]$, extreme nondominated objective vectors with the largest ST and E are favored, respectively.

We solve the following problem to get the third extreme supported nondominated objective vector:

$$\begin{aligned} \max \quad & wE_{scaled} + (1 - w)ST_{scaled} \\ \text{subject to} \quad & x \in X_A \end{aligned}$$

Let us say the optimal objective vector is $(E_{scaled}(3), ST_{scaled}(3))$. We reorder these three extreme supported objective vectors so that $E_{scaled}(3) < E_{scaled}(2) < E_{scaled}(1)$ and $ST_{scaled}(1) < ST_{scaled}(2) < ST_{scaled}(3)$.

Then, we calculate new weights, w_1 and w_2, to search for other extreme supported nondominated objective vectors.

$$w_1 = \frac{ST_{scaled}(2) - ST_{scaled}(1)}{ST_{scaled}(2) - ST_{scaled}(1) + E_{scaled}(1) - E_{scaled}(2)}$$

$$w_2 = \frac{ST_{scaled}(3) - ST_{scaled}(2)}{ST_{scaled}(3) - ST_{scaled}(2) + E_{scaled}(2) - E_{scaled}(3)}$$

The extreme supported nondominated objective vectors are reordered when a new one is found. New weights are calculated, and new ranges are searched iteratively as shown in Procedure 3.

Procedure 3 returns an extreme supported nondominated objective vector at each iteration. This extreme supported nondominated objective vector is either a new or an already known one.

If at the first iteration, one of the extreme nondominated objective vectors is returned, meaning no other extreme supported nondominated objective vector exists, then we stop. If one of the known extreme supported nondominated objective vectors is returned, then we stop searching the weight range at hand and move to a new one.

4.3. Finding all nondominated objective vectors

An optimal solution to

$$\begin{aligned} \max \quad & E + \epsilon_E ST \\ \text{subject to} \quad & ST \le k \ \text{ and } \ x \in X_A \end{aligned}$$

or equivalently,

$$\begin{aligned} \max \quad & E \\ \text{subject to} \quad & x \in X_A \end{aligned}$$

Procedure 3 Finding the extreme supported nondominated objective vectors

Input: Arrival and departure times of the aircraft, initial plan

Step 0. Find extreme nondominated objective vectors, (E_{max}, ST_{min}) and (E_{min}, ST_{max}).
Find the scalarized weight,

$$w = \frac{ST_{scaled}(2) - ST_{scaled}(1)}{ST_{scaled}(2) - ST_{scaled}(1) + E_{scaled}(1) - E_{scaled}(2)}.$$

Solve $\max \quad wE_{scaled} + (1-w)ST_{scaled}$

subject to $x \in X_A$

If the optimal objective vector is either (E_{max}, ST_{min}) or (E_{min}, ST_{max}), go to Step 3.
Let k be the number of extreme supported nondominated objective vectors excluding the two extremes.
Let $k = 1$ and let $(E_{scaled}(3), ST_{scaled}(3))$ be this optimal objective vector.

Step 1. Reorder the extreme nondominated objective vectors and update w_k and w_{k+1} as:

$$w_k = \frac{ST_{scaled}(k+1) - ST_{scaled}(k)}{ST_{scaled}(k+1) - ST_{scaled}(k) + E_{scaled}(k) - E_{scaled}(k+1)}$$

$$w_{k+1} = \frac{ST_{scaled}(k+2) - ST_{scaled}(k+1)}{ST_{scaled}(k+2) - ST_{scaled}(k+1) + E_{scaled}(k+1) - E_{scaled}(k+2)}.$$

Step 2. Solve $\max \quad w_k E_{scaled} + (1-w_k)ST_{scaled}$

subject to $x \in X_A$

If the optimal objective vector is either $(E_{scaled}(k+1), ST_{scaled}(k+1))$ or $(E_{scaled}(k), ST_{scaled}(k))$, let $k = k+1$, go to Step 2.
If the optimal objective vector is a new extreme supported nondominated objective vector, go to Step 1.
If all weight ranges are searched, go to Step 3.

Step 3. Stop, $k+1$ extreme supported solutions are generated.

Output: Set of all extreme supported nondominated objective vectors and the new plans.

produces an efficient solution (see Haimes (1971)). Let E^* be the optimal E value. Using

$$\max \quad ST$$
$$\text{subject to} \quad E = E^* \ \text{and} \ x \in X_A$$

and the fact that (E, ST) is integer, Procedure 4 finds all nondominated objective vectors.

We demonstrate Procedure 4 on a small-sized problem (50 aircraft, 10 gates, with Disruption Type 1, and arrival and departure times belonging to Set 1, as described in detail in Section 6.1). Procedure 4 starts by finding the extreme nondominated objective vector with the largest E value following Procedure 1. Subsequently, the first nondominated objective vector (E, ST) is found as $(11054, 21640)$ and labeled as $r = 1$. Then, a new constraint is introduced, where the stability value is incrementally increased by 1 and the problem is solved to find the corresponding optimal efficiency value at each iteration. Meaning, the second nondominated objective vector is found by introducing a constraint where the stability value must be greater than or equal to $21640+1$. Hence, $r = 2$ is found as $(11004, 22265)$. After 12 iterations, all nondominated objective vectors are found. We illustrate the trade-off curve of this example in Fig. 1 whose data are tabulated in Table 2.

4.4. Finding the approximate nondominated objective vectors

Our experiments have shown that the exact algorithm becomes computationally intractable as the number of aircraft and gates increases, hence a need for a heuristic procedure arises. We generate a set of approximate nondominated objective vectors by fixing some aircraft-gate assignments and solving a reduced problem.

Procedure 4 Finding all nondominated objective vectors

Input: Arrival and departure times of the aircraft, initial plan

Step 0. To find the extreme nondominated objective vector with the largest E value
solve $\max \quad E$

subject to $x \in X_A$

Let $E(1)$ be the optimal E value.
Solve $\max \quad ST$

subject to $E = E(1)$ and $x \in X_A$

Let $ST(1)$ be the optimal ST value.
Then, $(E(1), ST(1))$ is the 1^{st} nondominated objective vector.
Let r be the number of nondominated objective vectors and set $r = 1$.

Step 1. Solve $\max \quad E$

subject to $ST \geq ST(r) + 1$ and $x \in X_A$

Step 2. Let $r = r + 1$ and $E(r)$ be the optimal E value.
Solve $\max \quad ST$

subject to $E = E(r)$ and $x \in X_A$

Let $ST(r)$ be the optimal ST value.
Then, $(E(r), ST(r))$ is the r^{th} nondominated objective vector, go to Step 1.

Step 3. Stop, all r nondominated objective vectors are generated.

Output: Set of all nondominated objective vectors and the new plans.

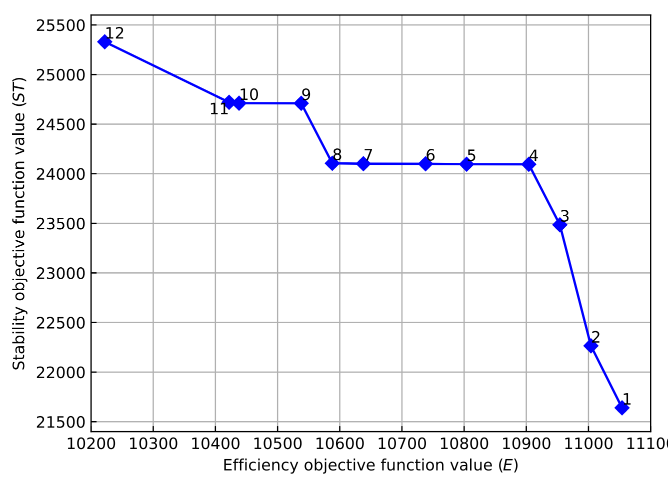

Fig. 1. Efficient frontier for the example instance.

Table 2
Nondominated objective vectors of an example instance.

r	E	ST	r	E	ST
1	11 054	21 640	7	10 638	24 101
2	11 004	22 265	8	10 588	24 105
3	10 954	23 485	9	10 538	24 710
4	10 904	24 095	10	10 438	24 711
5	10 804	24 096	11	10 422	24 720
6	10 738	24 100	12	10 222	25 330

Procedure 5 starts with finding the two extreme nondominated objective vectors, each of which is obtained in reasonable times. Then, the similarity of these new plans is detected. For a gate, if the set of aircraft assigned in the extreme nondominated objective vector with the largest ST value is a superset of the set of aircraft assigned in the extreme nondominated objective vector with the largest E value, then we fix these aircraft-gate assignments, obtain a partial assignment plan, and create a reduced problem. The idea behind this is that if a set of aircraft is assigned to the same gate in the two extreme nondominated objective vectors, i.e., the furthest nondominated objective vectors, then it is very likely that said aircraft be assigned to the same gate in the nondominated objective vectors in between also.

Procedure 5	Finding the approximate nondominated objective vectors
Input:	Arrival and departure times of the aircraft, initial plan
Step 0.	Find two extreme nondominated objective vectors, $r = 1$ and $r = 2$, with the largest E and ST values, respectively.
	Let S_{kr} be the set of aircraft assigned to gate k in extreme nondominated objective vector r.
	Let N be the set of all aircraft.
	Let M be the set of all gates.
	Let $k = 1$.
Step 1.	If S_{k2} is a superset of S_{k1}, i.e., all aircraft in S_{k1} are also in S_{k2} at gate k, update $N = N \setminus S_{k2}$ and $M = M \setminus \{k\}$.
Step 2.	If $k = m$, then go to Step 3.
	Let $k = k + 1$ and go to Step 1.
Step 3.	For the reduced problem with gates in M and aircraft in N, find all nondominated objective vectors using Procedure 4.
Output:	Set of approximate nondominated objective vectors and the new plans.

5. Optimal decomposition rule

A meaningful question in generating an assignment plan would be *"What would happen if the problem could be decomposed?"*. The optimal decomposition of an AGRP would mean that the main problem has time intervals where no aircraft occupies the gates, i.e., in such time intervals no aircraft is present in the system. After decomposing the main problem by clustering the aircraft, we get preferably equal-sized, smaller instances. Obtaining an assignment plan for each small instance and constructing a full solution for the main problem might be a good idea to explore. We propose Procedure 6 for our optimal decomposition rule which uses the following notation.

Procedure 6	Finding all nondominated objective vectors with decomposition rule
Input:	Arrival and departure times of the aircraft, initial plan
Step 0.	Find set $A(v)$ for all $v = 1, \ldots, V$ by using Procedure 4.
	Find ST_{LB} and ST_{UB}.
	Let $t = ST_{LB}$ and $r = 1$.
Step 1.	Solve P_E first and then solve P_{ST}.
	Let the optimal objective function values of P_E and P_{ST} be (EFF^*, STA^*) which is the r^{th} nondominated objective vector.
Step 2.	If $STA^* < ST_{UB}$, then let $t = STA^* + 1$, $r = r + 1$ and go to Step 1, else stop, all r nondominated objective vectors are generated.
Output:	Set of all nondominated objective vectors and the new plans.

Let $A(v)$ be the set of nondominated objective vectors of the decomposed problem, $v = 1, \ldots, V$ and EFF_u and STA_u be the corresponding E, and STA values of the solution u in A, where $A = \bigcup_{v=1}^{V} A(v)$. We construct a full solution by using two mathematical models: P_E and P_{ST}. The decision variable of these models is:

$$z_u = \begin{cases} 1, & \text{if solution } u \text{ is selected} \\ 0, & \text{otherwise} \end{cases} \qquad u \in A$$

$$(P_E) \quad \max \quad \sum_{v=1}^{V} \sum_{u \in A(v)} EFF_u z_u$$

subject to

$$\sum_{v=1}^{V} \sum_{u \in A(v)} STA_u z_u \geq t \tag{D1}$$

$$\sum_{u \in A(v)} z_u = 1 \qquad v = 1, \ldots, V \tag{E}$$

$$z_u \in \{0, 1\} \qquad u \in A \tag{F}$$

In P_E, we maximize the efficiency value of the full solution while the summation of the stability value counterparts of each smaller solution that make it up are forced to be at least some lower bound t (described further below and in Procedure 6) through constraint (D1). Constraint set (E) makes sure that exactly one solution is selected for each nondominated objective vector. Constraint set (F) is the binary constraint for all decision variables. Let EFF^* be the optimal objective function value.

$$(P_{ST}) \quad \max \quad \sum_{v=1}^{V} \sum_{u \in A(v)} STA_u z_u$$

subject to

$$\sum_{v=1}^{V} \sum_{u \in A(v)} EFF_u z_u = EFF^* \tag{D2}$$

$$\sum_{u \in A(v)} z_u = 1 \qquad v = 1, \ldots, V \tag{E}$$

$$z_u \in \{0, 1\} \qquad u \in A \tag{F}$$

In P_{ST}, we maximize the stability value of the full solution while the summation of the efficiency value counterparts of each smaller solution that make it up are forced to be exactly the optimal efficiency value of the full solution found by the previous model through constraint (D2). Constraint sets (E) and (F) are utilized as before. Let STA^* be the optimal objective function value. (EFF^*, STA^*) is a nondominated objective vector when t is between $ST_{LB} = \sum_{v=1}^{V} \min_{u \in A} STA_u$ and $ST_{UB} = \sum_{v=1}^{V} \max_{u \in A} STA_u$, the lower and upper bounds for the ST values of all efficient solutions, respectively.

6. Computational experiments

Performances of the algorithms are tested. In Section 6.1, the data generation scheme is discussed. The performance measures are stated in Section 6.2. In Section 6.3 the results of our computational experiment are analyzed.

6.1. Data generation scheme

Airports in the largest three cities in Turkey, İstanbul Airport in İstanbul, Esenboğa Airport in Ankara, and İzmir Adnan Menderes Airport in İzmir, have about 40, 20, and 10 gates, respectively (World Airport Guides, 2023; TAV Airports, 2023; Airport Technology, 2023). In our experiments, we select the number of gates, m, compatible with these real-life instances.

The number of aircraft, n, starts at 50 and changes in increments of 25, for each m scenario.

Arrival and departure times, i.e., a_i and d_i are generated as stated in Karsu et al. (2021). According to this scheme, low and high waiting instances, or having low and high chances of apron assignments, namely Set I and Set II are defined, respectively.

Set I $\quad a_i \sim U[0, 300]$ and $d_i \sim U[0, 30] + 30 + a_i$

Set II $\quad a_i \sim U[0, 150]$ and $d_i \sim U[0, 60] + 60 + a_i$

The time unit for the arrival and departure times is taken as minutes. For example, when an aircraft has an arrival time of 65, it means that this aircraft will be in the airport in the 65th minute of the planning horizon, which is compatible with the time intervals used in real-life.

The number of passengers, p_i, is generated as follows: $p_i \sim T(50, 100, 300)$ where T is the triangular distribution, 50 is the minimum, 100 is the mode, and 300 is the maximum value.

We use an initial plan that is optimal for the efficiency measure. Then, we assume that disruptions occur at time zero and the affected gates do not become available thereafter. We define three types of disruption scenarios for our experiments, where affected gates are selected randomly. Disruption Type I depicts a small disruption where only a single gate is closed. Disruption Type II shows a more serious case where one fifth of the gates is affected. For the sake of completeness, Disruption Type III depicts more severe incidents where half the gates become inoperable. These disruption scenarios depict cases with emergency flights where a gate is preempted, gate breakdowns, planned or unplanned maintenance operations, adverse weather conditions that render some gates inoperable, labor strikes of airport employees that result in the closure of some gates, and so on.

For each n, m, arrival and departure time set, and disruption type, 10 problem instances are generated. We set a termination limit of 2 h for each mathematical model.

All mathematical models and algorithms are developed using ILOG CPLEX Optimization Studio 20.1.0, and solved by CPLEX Optimizer 20.1.0. Furthermore, a computer with quad-core Intel(R) Core(TM) i7-10510U CPU @1.80 GHz–2.30 GHz, 16 GB RAM, and Windows 11 is used. Reported CPU times are expressed in seconds.

6.2. Performance measures

We report the average and maximum (worst case) CPU times for all procedures. We also include the statistics for the number of nondominated objective vectors.

To evaluate the performance of the heuristic algorithm that generates the approximate set of nondominated objective vectors, we first use P, the percentage of exact nondominated objective vectors found by the heuristic, $P = \frac{|ES \cap HS|}{|HS|} \times 100$ where ES and HS are exact and approximate sets of nondominated objective vectors, respectively, and $|ES \cap HS|$ is the number of exact nondominated objective vectors found by the heuristic.

To evaluate the closeness of the approximate solutions to their exact counterparts, we use two statistics, $D1$ and $D2$, as in Czyżak and Jaszkiewicz (1998). They define $D1$ and $D2$ as the average and maximum distance between the exact and heuristic nondominated objective vectors, respectively. We assume (E^r, ST^r) is in ES and (E^q, ST^q) is in HS, and calculate the ranges of E values, $R(E)$, and ST values, $R(ST)$, as follows:

$$R(E) = \max_{(E^r,ST^r)\in ES} E^r - \min_{(E^r,ST^r)\in ES} E^r$$

$$R(ST) = \max_{(E^r,ST^r)\in ES} ST^r - \min_{(E^r,ST^r)\in ES} ST^r$$

$$f\left((E^q, ST^q), (E^r, ST^r)\right) = \max\left\{0, \frac{1}{R(E)}(E^q, E^r), \frac{1}{R(ST)}(ST^q, ST^r)\right\}$$

Then, measures $D1$ and $D2$ are calculated as follows:

$$D1 = \frac{1}{|ES|} \sum_{\substack{(E^r,ST^r)\\ \in ES}} \min_{\substack{(E^q,ST^q)\\ \in HS}} \left\{ f\left((E^q, ST^q), (E^r, ST^r)\right) \right\}$$

$$D2 = \max_{\substack{(E^r,ST^r)\\ \in ES}} \left\{ \min_{\substack{(E^q,ST^q)\\ \in HS}} \left\{ f\left((E^q, ST^q), (E^r, ST^r)\right) \right\} \right\}$$

The higher percentage of exact nondominated objective vectors found by the heuristic, P, is preferred.

Furthermore, lower $D1$ and $D2$ values, average and maximum distances between the exact and heuristic nondominated objective vectors, respectively, is preferred.

6.3. Analysis of the results

In this section, we discuss the computational results for the solution procedures.

6.3.1. Extreme and extreme supported nondominated objective vectors

We first discuss the performance of the extreme and extreme supported algorithms (Procedures 1, 2 and 3). Tables 3 and 4 report average and maximum CPU times for extreme nondominated objective vector with the largest E value, extreme nondominated objective vector with the largest ST value, extreme supported nondominated objective vectors, and average and maximum number of extreme supported nondominated objective vectors, for Set 1 and Set 2, respectively.

As expected, both tables demonstrate higher CPU times for the larger number of aircraft. Under both low and high apron usage scenarios, we observe that CPU times for finding the extreme nondominated objective vector with the largest ST value are smaller than those of the extreme nondominated objective vector with the largest E value.

With this finding, the importance of using a reduced problem in Procedure 2, is emphasized. Especially, in instances with a larger number of aircraft, the difference between CPU times of the two extreme nondominated objective vectors is more dramatic, in some cases up to 40 times.

We also observe that CPU times to find the extreme supported nondominated objective vectors are higher for Set 1 and it increases with the number of aircraft.

The disruption types also affect the complexity of the solutions. Disruption Type 1 leads to a less complex problem than its counterparts, Disruption Types 2 and 3 where more gates are closed.

Lastly, we observe that the average and maximum number of extreme supported nondominated objective vectors is higher in Set 1 as opposed to Set 2 which is compatible with higher CPU times.

As expected, under the low apron usage scenario, there exist more solutions compared to the more restrictive high apron usage scenario. To illustrate, when n is 150 and m is 20, the average number of extreme supported nondominated objective vectors is 14.6 (Disruption Type 2) and 17.9 (Disruption Type 3) for Set 1, whereas, for Set 1, it is 5.1 (Disruption Type 2) and 6.8 (Disruption Type 3).

6.3.2. All nondominated objective vectors

We discuss the performance of the exact algorithm in Procedure 4 that generates all nondominated objective vectors. Tables 5 and 6 report the average and maximum number of nondominated objective vectors, average and maximum CPU times, and average CPU time per nondominated objective vector for Set 1 and Set 2, respectively. From Tables 5 and 6, we observe that the number of nondominated objective vectors is higher in Set 1 than that of Set 2. This is due to the high apron usage nature of Set 2, there are not as many possible aircraft-gate assignments, hence the solution space is narrower. Furthermore, we observe that CPU times increase as n increases for both sets. This can be attributed to the increase in the complexity of models due to the increase in the problem size. Set 1 instances are harder to solve than Set 2 instances. This is consistent with the higher number of nondominated objective vectors of Set 1. For example, under Disruption Type 2, when n is 125 and m is 20, the average (maximum) CPU times are 1330.44 (5985.94) s for Set 1, whereas it is 15.01 (18.23) s for Set 2. Similarly, under disruption type 3, when n is 125 and m is 20, the average (maximum) CPU times are 623.93 (2850.83) s for Set 1, whereas it is 22.25 (32.13) s for Set 2. For both Set 1 and Set 2, we also observe that Disruption Type 2 takes longer to solve than its Disruption Type 3 counterpart. Moreover, the average CPU time per nondominated objective vector is higher for Set 1. Its largest values occur with the large number of aircraft and a relatively large number of gates, for example when n is 125 and m is 20. As the problem size increases, so do the solution times for both sets. However, due to its narrower solution space, there are not as many solutions for Set 2. This results in a higher average CPU time per nondominated objective vector for Set 1.

Table 3
Extreme solutions (ES), extreme supported solutions (ESS) for set 1.

Disruption type	Aircraft, n	Gates, m	ES- Efficiency		ES- Stability		ESS			
			CPU time				Number of ESSs		CPU time	
			Avg	Max	Avg	Max	Avg	Max	Avg	Max
1	50	10	0.31	0.41	0.20	0.83	3.6	5	1.05	1.52
		20	0.39	0.47	0.39	0.84	1	1	0.78	1.23
	75	10	0.50	0.66	0.11	0.45	4.6	7	1.62	2.72
		20	0.79	1.41	0.35	0.66	1.4	3	1.43	2.52
	100	10	1.22	1.58	0.03	0.11	4.7	7	3.93	6.11
		20	2.37	3.58	0.31	0.89	4.3	6	5.92	9.36
		40	2.37	3.78	0.78	1.77	1	1	3.15	5.38
	125	10	2.10	2.55	0.04	0.11	4.5	7	5.51	8.05
		20	5.10	8.55	0.19	0.45	4.5	6	10.36	13.56
		40	3.13	3.58	1.26	3.41	1	1	4.39	6.66
	150	10	3.80	4.84	0.07	0.27	4.2	6	8.06	11.36
		20	10.62	16.83	0.15	0.39	6.3	8	23.99	38.67
		40	3.89	4.28	1.12	2.00	1	1	5.01	5.92
	175	10	5.02	7.05	0.12	0.38	4.9	7	10.74	17.53
		20	21.38	30.14	0.10	0.33	5.6	7	36.70	46.56
		40	10.34	15.22	1.90	3.19	2.1	3	16.17	23.42
	200	10	4.07	4.98	0.10	0.47	4.3	7	10.05	17.36
		20	22.87	30.61	0.21	0.67	5.6	8	45.57	55.48
		40	15.93	37.44	1.39	2.33	3.2	4	28.52	55.53
2	50	10	0.29	0.52	0.21	0.47	5.9	8	1.80	2.50
		20	0.39	0.45	0.56	1.28	1	1	0.95	1.66
	75	10	0.55	0.81	0.18	0.53	6.9	12	4.58	10.38
		20	0.88	1.06	0.61	1.22	4.2	7	4.16	5.50
	100	10	0.74	1.00	0.05	0.11	7	8	5.26	7.17
		20	2.32	3.02	0.60	0.88	11.3	16	18.68	29.36
		40	2.33	2.55	2.77	5.31	1	1	5.10	7.69
	125	10	1.52	1.92	0.06	0.19	6.8	9	7.35	11.80
		20	5.10	7.28	0.35	0.80	11.5	19	29.49	49.91
		40	3.36	3.77	3.80	7.08	1	1	7.16	10.23
	150	10	2.21	2.67	0.07	0.22	6.5	10	12.37	21.03
		20	8.10	11.91	0.13	0.28	14.6	18	52.71	68.92
		40	6.06	8.25	3.57	5.30	2.4	6	15.63	35.08
	175	10	3.83	4.50	0.08	0.28	7.6	11	13.81	19.77
		20	14.65	21.55	0.36	0.77	13	17	60.23	71.94
		40	24.97	48.19	3.08	5.13	10.4	18	95.67	217.11
	200	10	3.59	4.08	0.09	0.30	6.7	10	16.88	28.34
		20	18.13	20.97	0.33	1.28	12.7	17	81.50	111.67
		40	45.41	69.88	2.33	3.41	19.3	24	316.24	701.03
3	50	10	0.33	0.45	0.23	0.77	8.8	14	3.23	6.44
		20	0.52	0.66	0.84	1.37	3.3	5	2.33	3.22
	75	10	0.58	0.73	0.14	0.44	8.6	11	5.47	7.61
		20	0.93	1.13	0.79	1.61	10.7	16	10.58	16.91
	100	10	0.78	0.88	0.10	0.42	9.7	14	8.56	14.20
		20	1.76	2.17	0.45	0.83	15.6	20	25.06	34.81
		40	3.14	4.03	3.60	6.03	3.2	7	11.97	18.58
	125	10	1.10	1.47	0.14	0.63	9.1	13	12.68	17.42
		20	2.80	3.19	0.35	0.63	17.8	21	51.22	71.86
		40	7.18	10.70	4.82	6.69	9.8	15	58.28	98.39
	150	10	1.24	1.38	0.12	0.34	9.4	11	10.28	11.66
		20	3.73	4.14	0.28	0.84	17.9	20	50.75	56.83
		40	11.43	15.95	5.16	7.50	16	19	127.84	244.45
	175	10	2.43	3.00	0.08	0.38	9.7	15	17.36	27.84
		20	7.15	8.44	0.25	0.56	16.1	19	63.57	80.53
		40	35.40	45.08	4.96	8.09	24.2	28	344.99	608.61
	200	10	2.15	2.36	0.07	0.25	8.9	11	15.31	19.58
		20	5.60	6.17	0.16	0.41	17	21	71.47	108.80
		40	33.00	54.36	3.08	5.47	26.1	34	329.08	490.97

Table 4
Extreme solutions (ES), extreme supported solutions (ESS) for set 2.

Disruption type	Aircraft, n	Gates, m	ES- Efficiency		ES- Stability		ESS			
			CPU time				Number of ESSs		CPU time	
			Avg	Max	Avg	Max	Avg	Max	Avg	Max
1	50	10	0.26	0.30	0.05	0.14	2.4	3	0.84	1.45
		20	0.53	0.61	0.07	0.17	2	3	1.30	2.05
	75	10	0.47	0.66	0.05	0.17	2.2	3	1.19	2.03
		20	1.29	1.83	0.15	0.66	2.3	3	2.38	3.31
	100	10	0.76	0.92	0.03	0.09	1.7	3	1.91	3.13
		20	1.75	2.23	0.05	0.19	2.3	4	4.32	6.69
		40	4.28	5.02	0.25	0.97	2.1	3	5.27	8.47
	125	10	1.05	1.16	0.07	0.20	2.2	3	3.04	3.73
		20	2.20	2.53	0.09	0.42	2.1	3	6.64	11.14
		40	6.04	7.69	0.17	0.42	2.3	3	7.90	11.25
	150	10	1.37	1.53	0.08	0.34	1.7	2	4.70	6.16
		20	2.89	3.08	0.10	0.25	2.5	3	12.83	19.83
		40	9.16	11.42	0.23	0.61	2.8	3	11.90	13.64
	175	10	2.02	2.36	0.06	0.27	2	3	5.00	6.23
		20	3.82	4.41	0.10	0.39	2.1	3	25.05	32.38
		40	13.42	15.33	0.27	1.08	2.4	4	26.00	48.58
2	50	10	0.27	0.34	0.04	0.14	3.2	5	1.09	2.00
		20	0.50	0.64	0.19	0.81	4.7	9	2.93	4.81
	75	10	0.48	0.67	0.07	0.25	2.8	4	1.61	3.09
		20	1.06	1.34	0.13	0.56	4.9	7	4.97	7.05
	100	10	0.69	0.78	0.06	0.25	2.9	4	2.13	3.42
		20	1.73	2.02	0.11	0.31	4.8	6	6.54	7.89
		40	4.09	4.97	0.39	1.08	7.6	12	19.00	28.39
	125	10	1.07	1.19	0.07	0.33	2.9	5	3.20	4.83
		20	2.04	2.17	0.06	0.20	5.1	7	11.05	14.81
		40	5.09	5.86	0.15	0.50	7.8	10	28.27	33.23
	150	10	1.58	2.66	0.08	0.38	2.5	4	3.82	4.72
		20	2.75	3.11	0.12	0.39	5.1	7	16.28	22.95
		40	7.02	7.77	0.34	1.11	8.6	12	42.32	62.45
	175	10	1.69	2.02	0.04	0.19	2.4	4	5.75	8.86
		20	3.25	3.91	0.14	0.38	4.3	7	27.29	33.42
		40	12.33	14.94	0.27	0.59	7.5	11	95.33	147.88
3	50	10	0.25	0.28	0.08	0.30	4.1	5	1.37	1.81
		20	0.46	0.58	0.19	0.73	6.3	9	4.42	6.91
	75	10	0.47	0.56	0.04	0.11	3.7	6	2.14	3.61
		20	0.82	1.03	0.13	0.31	7.9	10	6.90	8.83
	100	10	0.61	0.67	0.04	0.14	4.2	6	3.37	4.31
		20	1.33	1.44	0.09	0.19	7.2	9	8.82	12.88
		40	3.03	3.97	0.42	0.94	11.6	14	32.36	49.47
	125	10	0.96	1.11	0.07	0.27	4.1	5	4.13	4.78
		20	1.91	2.25	0.08	0.23	6.9	9	13.24	18.20
		40	4.00	4.50	0.11	0.41	11.4	15	47.86	60.45
	150	10	1.23	1.42	0.05	0.23	4	7	4.71	8.16
		20	2.35	2.58	0.07	0.20	6.8	9	18.39	27.39
		40	5.38	6.14	0.13	0.44	11.4	16	65.88	91.42
	175	10	1.60	2.19	0.07	0.23	3.4	4	5.10	6.16
		20	2.90	3.17	0.07	0.25	6.8	11	21.97	36.78
		40	8.91	10.17	0.21	0.45	10.5	14	85.72	114.05

Table 5
All exact nondominated objective vectors for set 1.

Disruption type	Aircraft, n	Gates, m	Number of ndovs		CPU time		Average CPU time per ndov
			Avg	Max	Avg	Max	
1	50	10	6.5	12	2.71	7.23	0.42
		20	1	1	0.51	0.83	0.51
	75	10	10.4	21	10.90	39.91	1.05
		20	1.6	3	1.34	2.45	0.84
	100	10	9.5	15	13.36	45.27	1.41
		20	7.1	14	14.89	30.66	2.10
		40	1	1	2.86	4.58	2.86
	125	10	9.2	18	18.84	61.80	2.05
		20	8.3	13	212.02	1671.61	25.54
		40	1	1	5.13	7.83	5.13
2	50	10	11.5	19	7.36	14.00	0.64
		20	1	1	0.75	1.08	0.75
	75	10	18.2	33	26.83	118.42	1.47
		20	6.5	11	8.99	22.69	1.38
	100	10	18	25	23.75	54.30	1.32
		20	32.8	69	452.11	1338.13	13.78
		40	1	1	4.14	4.81	4.14
	125	10	16.1	24	28.23	52.33	1.75
		20	45.2	72	1330.44	5985.94	29.43
		40	1	1	5.17	6.08	5.17
3	50	10	19.2	24	10.78	13.42	0.56
		20	4.2	9	3.33	7.23	0.79
	75	10	32.6	45	30.85	58.70	0.95
		20	26	57	59.60	143.33	2.29
	100	10	32.3	54	39.32	82.72	1.22
		20	52.4	74	589.38	2840.03	11.25
		40	4.3	10	16.84	40.27	3.92
	125	10	30.4	56	39.07	71.34	1.29
		20	76.3	103	792.51	1309.72	10.39
		40	20.8	33	623.93	2850.83	30.00

Table 6
All exact nondominated objective vectors for set 2.

Disruption type	Aircraft, n	Gates, m	Number of ndovs		CPU time		Average CPU time per ndov
			Avg	Max	Avg	Max	
1	50	10	2.8	4	0.85	1.28	0.31
		20	2.8	5	1.75	2.91	0.63
	75	10	2.4	4	1.49	2.88	0.62
		20	2.5	4	2.98	4.48	1.19
	100	10	1.9	5	1.94	5.05	1.02
		20	2.7	7	5.23	11.36	1.94
		40	3.1	5	11.72	17.91	3.78
	125	10	2.9	6	4.38	11.38	1.51
		20	2.3	3	5.47	7.30	2.38
		40	2.7	4	13.98	22.56	5.18
2	50	10	4.2	6	1.78	2.88	0.42
		20	10	19	7.64	13.28	0.76
	75	10	3.6	6	2.18	4.13	0.61
		20	8	12	9.51	14.83	1.19
	100	10	3.4	6	3.12	6.08	0.92
		20	6.1	9	10.55	15.41	1.73
		40	28.4	48	118.32	258.56	4.17
	125	10	4.4	10	5.82	15.36	1.32
		20	7.6	10	15.01	18.23	1.98
		40	13.4	22	57.63	96.55	4.30

(*continued on next page*)

Table 6 (*continued*).

Disruption type	Aircraft, n	Gates, m	Number of ndovs		CPU time		Average CPU time per ndov
			Avg	Max	Avg	Max	
3	50	10	5.8	10	2.23	4.59	0.38
		20	12.3	21	8.93	16.06	0.73
	75	10	6	9	3.86	6.97	0.64
		20	13.4	22	12.57	24.61	0.94
	100	10	6.5	13	5.77	12.77	0.89
		20	11.2	16	16.81	26.98	1.50
		40	27.4	37	80.59	123.69	2.94
	125	10	6.3	8	6.91	9.42	1.10
		20	11.8	17	22.25	32.13	1.89
		40	25.3	30	91.52	107.95	3.62

Table 7
Heuristic procedure for set 1.

Disruption type	Aircraft, n	Gates, m	CPU time		P		$D1$		$D2$	
			Avg	Max	Avg	Min	Avg	Max	Avg	Max
1	50	10	3.28	8.67	82.85	45.45	0.04	0.25	0.13	0.70
		20	0.78	1.23	100.00	100.00	0.00	0.00	0.00	0.00
	75	10	8.79	26.02	70.36	28.57	0.05	0.22	0.18	0.67
		20	1.72	3.95	100.00	100.00	0.00	0.00	0.00	0.00
	100	10	11.70	30.33	92.33	66.67	0.01	0.10	0.06	0.37
		20	9.03	16.64	88.29	66.67	0.02	0.06	0.07	0.25
		40	3.15	5.38	100.00	100.00	0.00	0.00	0.00	0.00
	125	10	13.18	25.34	85.62	36.36	0.02	0.10	0.08	0.23
		20	19.00	40.22	77.54	30.77	0.03	0.12	0.10	0.25
		40	4.39	6.66	100.00	100.00	0.00	0.00	0.00	0.00
	150	10	16.58	28.72						
		20	101.95	572.41						
		40	5.01	5.92						
2	50	10	8.25	13.86	90.86	75.00	0.01	0.05	0.06	0.25
		20	0.95	1.66	100.00	100.00	0.00	0.00	0.00	0.00
	75	10	17.13	51.70	76.63	21.21	0.02	0.05	0.09	0.25
		20	8.34	20.47	94.11	77.78	0.01	0.07	0.08	0.36
	100	10	29.38	98.44	92.17	50.00	0.00	0.01	0.02	0.07
		20	232.37	1034.02	92.04	33.33	0.00	0.01	0.01	0.07
		40	5.10	7.69	100.00	100.00	0.00	0.00	0.00	0.00
	125	10	19.09	35.22	89.63	50.00	0.01	0.03	0.04	0.17
		20	706.73	4499.83	83.50	52.00	0.00	0.02	0.05	0.11
		40	7.16	10.23	100.00	100.00	0.00	0.00	0.00	0.00
	150	10	32.60	55.31						
		20	1699.25	6493.59						
		40	18.27	46.08						
3	50	10	6.27	21.09	55.86	33.33	0.03	0.06	0.11	0.17
		20	3.50	5.56	92.44	44.44	0.02	0.10	0.05	0.27
	75	10	10.55	35.22	38.19	17.07	0.03	0.05	0.13	0.24
		20	52.58	238.25	76.49	33.33	0.01	0.09	0.05	0.25
	100	10	17.53	75.53	44.97	15.91	0.03	0.05	0.15	0.32
		20	43.99	214.48	36.94	21.67	0.02	0.02	0.07	0.10
		40	18.18	38.55	100.00	100.00	0.00	0.00	0.00	0.00
	125	10	19.22	49.86	44.54	14.29	0.04	0.06	0.14	0.26
		20	286.75	1030.66	62.78	20.41	0.01	0.03	0.05	0.13
		40	761.56	4008.38	95.63	82.14	0.00	0.01	0.01	0.07
	150	10	13.37	40.30						
		20	435.45	2137.91						
		40	7396.45	59 410.89						

6.3.3. Approximate nondominated objective vectors

Procedure 5, where we take advantage of the closeness of the two extreme nondominated objective vectors to create a reduced problem and unify its solutions with the extreme supported nondominated objective vectors, is implemented on our test instances. The results are given in Tables 7 and 8, where average and maximum CPU times are reported up to 150 aircraft to show the reduced problem complexity, and average and minimum P, and average and maximum $D1$ and $D2$ values are reported for the instances used in Procedure 4.

As expected, as the problem size increases, the CPU times increase. However, it takes a much shorter time to solve the same set of problems with the heuristic procedure compared to the exact algorithm. This becomes more vivid in the following example: when n is 125 and m is 20 with Disruption Type 2 for Set 1, the average (maximum) CPU time to generate all nondominated objective vectors is 1330.44 (5985.94) s as seen in Table 5. The same instances are solved using the heuristic procedure with the average (maximum) CPU time of 706.73 (4499.83) s as shown in Table 7.

Table 8
Heuristic procedure for set 2.

Disruption type	Aircraft, n	Gates, m	CPU time Avg	CPU time Max	P Avg	P Min	$D1$ Avg	$D1$ Max	$D2$ Avg	$D2$ Max
1	50	10	1.05	1.89	94.17	66.67	0.03	0.17	0.08	0.50
		20	1.45	2.14	79.00	40.00	0.11	0.35	0.29	0.77
	75	10	1.38	2.19	97.50	75.00	0.01	0.08	0.03	0.33
		20	2.62	3.69	95.00	50.00	0.03	0.25	0.06	0.63
	100	10	2.08	3.69	96.00	60.00	0.01	0.14	0.05	0.49
		20	4.70	7.47	98.57	85.71	0.00	0.00	0.00	0.00
		40	5.49	9.16	75.67	50.00	0.12	0.23	0.37	0.69
	125	10	3.49	4.66	90.00	50.00	0.04	0.17	0.13	0.51
		20	6.95	11.41	93.33	66.67	0.03	0.17	0.10	0.50
		40	8.19	11.64	90.83	66.67	0.03	0.17	0.10	0.50
	150	10	4.98	6.58						
		20	13.19	20.53						
		40	12.90	15.62						
2	50	10	1.85	3.33	96.00	60.00	0.01	0.08	0.03	0.25
		20	6.27	10.77	90.58	68.75	0.00	0.02	0.04	0.24
	75	10	2.27	4.72	97.50	75.00	0.01	0.08	0.03	0.33
		20	8.27	13.58	94.44	66.67	0.01	0.04	0.06	0.25
	100	10	2.77	4.95	94.17	66.67	0.02	0.12	0.08	0.50
		20	9.68	11.78	98.89	88.89	0.00	0.02	0.01	0.14
		40	48.82	108.78	90.94	68.00	0.00	0.01	0.03	0.11
	125	10	5.31	10.41	94.50	70.00	0.01	0.08	0.06	0.33
		20	16.94	22.63	99.00	90.00	0.00	0.01	0.01	0.07
		40	39.35	54.95	99.55	95.45	0.00	0.00	0.00	0.04
	150	10	4.84	7.94						
		20	20.68	27.91						
		40	55.37	87.94						
3	50	10	2.73	5.22	95.71	57.14	0.01	0.09	0.03	0.29
		20	10.44	21.03	99.38	93.75	0.00	0.00	0.00	0.04
	75	10	3.87	7.11	98.00	80.00	0.01	0.06	0.03	0.32
		20	13.77	23.75	93.26	40.91	0.00	0.03	0.01	0.09
	100	10	6.95	12.72	94.53	75.00	0.01	0.03	0.05	0.19
		20	18.42	24.89	98.13	87.50	0.00	0.00	0.00	0.01
		40	66.34	108.67	98.00	80.00	0.00	0.01	0.00	0.03
	125	10	8.23	11.91	96.25	75.00	0.00	0.03	0.03	0.24
		20	25.04	36.39	97.39	88.24	0.00	0.01	0.01	0.11
		40	92.02	110.06	99.55	95.45	0.00	0.00	0.01	0.07
	150	10	8.41	13.84						
		20	29.31	41.13						
		40	101.61	137.23						

We observe a significant time reduction when the heuristic procedure is used for this sizeable problem set. In this reduced time, the average (minimum) of the reported P value is 83.50% (52%), which shows many of the nondominated objective vectors can be generated in a much shorter time. Considering the high average (maximum) number of nondominated objective vectors 45.2 (72), we would be also generating a high number nondominated objective vectors for the decision maker to choose from.

From Table 7, for Set 1, we see high P values for Disruption Types 1 and 2, and satisfactory P values for Disruption Type 3 when supported with considerable CPU time reductions compared to the exact algorithm and considering the high number of nondominated objective vectors in Set 1. From Table 8, for Set 2, we observe high P values for all disruption types which is consistent with the lower number of nondominated objective vectors in Set 2 compared to Set 1 as shown in Tables 5 and 6.

To illustrate, for Set 2, when n is 125 and m is 40 with Disruption Type 2, on average 99.55% of all nondominated objective vectors are generated using the heuristic procedure. The corresponding average CPU time is 39.35 s. On the other hand, for the same instances, the exact algorithm generates 13.4 nondominated objective vectors on average with an average CPU time of 57.63 s as shown in Table 6. We observe that a good percentage of all nondominated objective vectors can be found within a reduced time by our heuristic procedure.

6.3.4. Optimal decomposition rule

For the optimal decomposition rule proposed in Procedure 6, we generated new instances where the problem can be decomposed into two or three small problems where no aircraft pair between said small problems is in the system at the same time interval. In Tables 9–12, we report the average and maximum number of nondominated objective vectors, and CPU times with and without the optimal decomposition rule.

Table 9 is prepared for Set 1, with Disruption Type 2 and the case where the problem is decomposed into two small problems. Similarly, we observe increased problem complexity with the increased problem size. We also observe as the number of nondominated objective vectors increases, so do the CPU times. From Table 9, we observe the reduction in CPU times that decomposing provides.

Table 10 is for Set 1, with Disruption Type 2 and the case where the problem is decomposed into three small problems and Table 11 is prepared for Set 1, with Disruption Type 3 and the case where the problem is decomposed into three small problems. From Tables 10 and 11, the effect of disruption, i.e., the number of gates closed, can be inferred. As the number of closed gates increases, the average CPU time also increases in Set 1. As a striking example, when n is 100 and m is 20 with Disruption Type 3 for Set 1, the average (maximum) CPU time is reported as 162.39 (520.70) s without using the decomposition rule.

Table 9
Decomposition algorithm, set 1, disruption type 2, $r = 2$.

Aircraft, n	Gates, m	Number of ndovs		CPU time			
				Without decomposition		With decomposition	
		Avg	Max	Avg	Max	Avg	Max
75	10	11.8	19	8.45	13.50	6.09	14.52
	20	12.6	19	21.96	73.92	9.37	20.03
	40	1	1	2.93	3.45	2.14	2.88
100	10	15	24	14.45	23.45	10.70	25.17
	20	23.8	33	76.09	372.23	18.57	29.17
	40	1	1	4.13	5.59	3.28	3.86
125	10	16.4	31	20.07	35.00	11.65	19.91
	20	27.4	37	72.48	103.42	34.93	52.19
	40	1.5	4	7.97	21.00	6.16	14.92

Table 10
Decomposition algorithm, set 1, disruption type 2, $r = 3$.

Aircraft, n	Gates, m	Number of ndovs		CPU time			
				Without decomposition		With decomposition	
		Avg	Max	Avg	Max	Avg	Max
75	10	13.4	26	9.00	22.19	4.31	8.84
	20	11.4	16	10.16	18.94	4.85	6.27
	40	1	1	1.64	1.81	1.84	2.31
100	10	14.3	23	11.36	20.14	6.64	12.47
	20	16.3	23	23.38	34.91	8.21	11.69
	40	1	1	3.61	4.14	2.49	3.00
125	10	15.4	20	15.43	21.03	5.51	8.34
	20	26.8	48	63.31	138.28	14.84	26.86
	40	4.9	9	17.25	30.17	7.73	12.33

Table 11
Decomposition algorithm, set 1, disruption type 3, $r = 3$.

Aircraft, n	Gates, m	Number of ndovs		CPU time			
				Without decomposition		With decomposition	
		Avg	Max	Avg	Max	Avg	Max
75	10	29.4	44	50.11	107.84	5.00	8.56
	20	20.3	25	31.73	46.86	5.52	8.31
	40	1.6	3	4.92	8.13	1.52	2.08
100	10	30.9	65	46.10	110.86	7.31	14.69
	20	43.5	77	162.39	520.70	12.86	21.98
	40	11.1	15	42.48	60.69	7.25	10.38
125	10	25.2	46	41.11	90.72	5.61	7.38
	20	71.8	105	470.60	897.23	25.56	35.83
	40	21.3	31	133.52	187.25	16.11	22.58

Table 12
Decomposition algorithm, set 2, disruption type 3, $r = 2$.

Aircraft, n	Gates, m	Number of ndovs		CPU time			
				Without decomposition		With decomposition	
		Avg	Max	Avg	Max	Avg	Max
75	10	5.4	7	2.19	3.05	1.09	1.94
	20	9.6	13	6.94	9.41	3.28	4.31
	40	15.9	18	18.00	20.17	8.64	10.17
100	10	4.5	7	2.51	3.67	1.26	1.70
	20	8.7	12	9.53	16.31	4.59	7.59
	40	18.4	21	33.11	44.98	19.78	24.03
125	10	3.7	5	2.79	3.80	1.84	2.31
	20	8.6	10	10.86	13.13	6.74	8.78
	40	17.5	22	46.22	65.77	22.31	28.22

However, with the decomposition rule, the average (maximum) CPU time significantly reduces to 12.86 (21.98) s.

From Tables 9 and 10, the effect of r can be observed. Under the same apron usage scenario with the same disruption type, decomposing the problem into either two or three small problems result in further decreased CPU times in general. Table 12 is for Set 2, with Disruption Type 3 and the case where the problem is decomposed into two small problems. We again observe the improved CPU times with the decomposition rule, this time for the high apron usage scenario (Set 2).

From Tables 9–12, we deduce that both the average and maximum CPU times are considerably reduced by using the decomposition rule.

7. Conclusions and further research directions

In this study, we consider an AGRP where due to gate disruptions the initial aircraft-gate assignment plan becomes obsolete, and a new plan is required. We define efficiency and stability criteria for different concerns of the decision makers. In our efficiency criterion, maximization of gate utilization in terms of the number of aircraft assigned and their corresponding number of passengers is sought. On the other hand, in our stability criterion, maximization of the number of preserved assignments, the corresponding number of passengers, and the number of apron assignments that are shifted to gates are aimed. Both the efficiency and stability criteria are made up of multi-objectives: two objective functions are defined for the efficiency criterion and three for the stability criterion. These objective functions altogether cover various efficiency and passenger satisfaction concerns of the airlines.

We apply hierarchical optimization, i.e., maximizing the efficiency (stability) measure while keeping the stability (efficiency) value at its maximum level in handling said objective functions. We use assignment model-based approaches to generate all extreme supported and all nondominated objective vectors with respect to our efficiency and stability criteria.

To generate all nondominated objective vectors, we follow two approaches: optimization and approximation. Our optimization algorithm solves instances with up to 150 aircraft and 40 gates, in less than two hours. With the approximation algorithm, we handle instances with up to 200 aircraft and 40 gates and report excellent performance results in terms of solution times and the power of representing the exact nondominated objective vectors.

We develop an optimal decomposition rule that decomposes the main problem into smaller instances at time intervals that reside with no aircraft in the system. We find that with the optimal decomposition rule, problems could be solved in considerably small times.

All our objective functions reflect aeronautical concerns of the airline managers. Future research may consider multi-criteria problems that trade-off between aeronautical objectives and non-aeronautical objectives such as airport retailing, advertising, car rentals, and so on. Our solution procedures can be modified to make nice trade-offs between aeronautical and non-aeronautical objectives. Moreover, we anticipate that, in real-life instances, there may be few cases where our optimal decomposition rule can be used directly. As further research, some heuristic approaches that may take our rule as a basis can be developed. Hence, we propose to create a subproblem that is optimally decomposable, which is achieved by taking out a subset of aircraft and then applying our optimal decomposition rule to obtain a new plan, and lastly to reconsider the aircraft that were taken out through some insertion or exchange heuristics. As another further research direction, we propose some aircraft-gate eligibility constraints, where some gates are reserved for certain airlines. Another proposition would be to consider some side-by-side compatibility constraints, where the sizes of the aircraft factor into the decision-making process, i.e., two large aircraft cannot be assigned to juxtaposed gates. We believe implicit enumeration techniques, such as a branch and bound algorithm, can be designed to generate all nondominated objective vectors simultaneously in place of our sequential generation methods. Furthermore, optimization algorithms for a known, yet complex utility function can be developed, and different efficiency and stability measures can be tried out.

CRediT authorship contribution statement

Dursen Deniz Poyraz: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Meral Azizoğlu:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Airport Technology, 2023. Izmir Adnan Menderes International Airport. Retrieved May 17, 2023, from https://www.airport-technology.com/projects/izmir-adnan-menderes-international-airport/.

Ali, H., Guleria, Y., Alam, S., Schultz, M., 2019. A passenger-centric model for reducing missed connections at low cost airports with gates reassignment. IEEE Access 7, 179429–179444. http://dx.doi.org/10.1109/ACCESS.2019.2953769.

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. http://dx.doi.org/10.1109/TCYB.2019.2956974.

Choi, J.H., 2021. Changes in airport operating procedures and implications for airport strategies post-COVID-19. J. Air Transp. Manage. 94, 102065. http://dx.doi.org/10.1016/j.jairtraman.2021.102065.

Czyzżak, P., Jaszkiewicz, A., 1998. Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. J. Multi-Criteria Decis. Anal. 7 (1), 34–47. http://dx.doi.org/10.1002/(SICI)1099-1360(199801)7:1<34::AID-MCDA161>3.0.CO;2-6.

Daş, G.S., Gzara, F., Stützle, T., 2020. A review on airport gate assignment problems: Single versus multi objective approaches. Omega 92, 102146. http://dx.doi.org/10.1016/j.omega.2019.102146.

Deng, W., Li, B., Zhao, H., 2017. Study on an airport gate reassignment method and its application. Symmetry 9 (11), 258. http://dx.doi.org/10.3390/sym9110258.

Dorndorf, U., Jaehn, F., Pesch, E., 2012. Flight gate scheduling with respect to a reference schedule. Ann. Oper. Res. 194 (1), http://dx.doi.org/10.1007/s10479-010-0809-8.

Gu, Y., Chung, C.A., 1999. Genetic algorithm approach to aircraft gate reassignment problem. J. Transp. Eng. 125 (5), 384–389. http://dx.doi.org/10.1061/(ASCE)0733-947X(1999)125:5(384).

Haimes, Y., 1971. On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Trans. Syst. Man Cybern. (3), 296–297. http://dx.doi.org/10.1109/TSMC.1971.4308298.

Jaehn, F., 2010. Solving the flight gate assignment problem using dynamic programming. Z. Betriebswirtschaft 80, 1027–1039. http://dx.doi.org/10.1007/s11573-010-0396-9.

Karsu, Ö., Azizoğlu, M., Alanlı, K., 2021. Exact and heuristic solution approaches for the airport gate assignment problem. Omega 103, 102422. http://dx.doi.org/10.1016/j.omega.2021.102422.

Kidokoro, Y., Zhang, A., 2023. Effects of non-aeronautical service on airports: A selected review and research agenda. J. Air Transp. Res. Soc. 1 (1), 40–53. http://dx.doi.org/10.59521/5372A44CC8A85136.

Liu, J., Guo, Z., Yu, B., 2023. Optimising gate assignment and taxiway path in a discrete time–space network: integrated model and state analysis. Transp. B: Transp. Dyn. 11 (1), 1–23. http://dx.doi.org/10.1080/21680566.2022.2036650.

Macilree, J., Duval, D.T., 2020. Aeropolitics in a post-COVID-19 world. J. Air Transp. Manage. 88, 101864. http://dx.doi.org/10.1016/j.jairtraman.2020.101864.

Maharjan, B., Matis, T.I., 2011. An optimization model for gate reassignment in response to flight delays. J. Air Transp. Manage. 17 (4), 256–261. http://dx.doi.org/10.1016/j.jairtraman.2011.02.012.

Özlen, M., Azizoğlu, M., 2009. Generating all efficient solutions of a rescheduling problem on unrelated parallel machines. Int. J. Prod. Res. 47 (19), 5245–5270. http://dx.doi.org/10.1080/00207540802043998.

Pereira, F., Costa, J.M., Ramos, R., Raimundo, A., 2023. The impact of the COVID-19 pandemic on airlines' passenger satisfaction. J. Air Transp. Manage. 112, 102441. http://dx.doi.org/10.1016/j.jairtraman.2023.102441.

Pternea, M., Haghani, A., 2018. Mathematical models for flight-to-gate reassignment with passenger flows: State-of-the-art comparative analysis, formulation improvement, and a new multidimensional assignment model. Comput. Ind. Eng. 123, 103–118. http://dx.doi.org/10.1016/j.cie.2018.05.038.

Pternea, M., Haghani, A., 2019. An aircraft-to-gate reassignment framework for dealing with schedule disruptions. J. Air Transp. Manage. 78, 116–132. http://dx.doi.org/10.1016/j.jairtraman.2019.01.005.

Sun, X., Wandelt, S., Zhang, A., 2022. COVID-19 pandemic and air transportation: Summary of recent research, policy consideration and future research directions. Transp. Res. Interdiscip. Pers. 16, 100718. http://dx.doi.org/10.1016/j.trip.2022.100718.

Sun, X., Wandelt, S., Zhang, A., 2023. Aviation under the COVID-19 pandemic: A synopsis from normalcy to chaos and back. J. Air Transp. Res. Soc. 1 (1), 136–151. http://dx.doi.org/10.59521/8F3EEB54038C692E.

Tang, C.H., Yan, S., Hou, Y.Z., 2010. A gate reassignment framework for real time flight delays. 4OR 8, 299–318. http://dx.doi.org/10.1007/s10288-009-0112-1.

TAV Airports, 2023. Ankara Esenboga Airport arrival level. Retrieved May 17, 2023, from https://webcmsesb.tav.aero/files/1551878584_ESB%20Gelis_2019.pdf.

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. http://dx.doi.org/10.1016/j.trc.2021.103491.

Wang, H., Luo, Y., Shi, Z., 2013. Real-time gate reassignment based on flight delay feature in hub airport. Math. Probl. Eng. 2013, http://dx.doi.org/10.1155/2013/646241.

World Airport Guides, 2023. Istanbul International Airport terminal information. Retrieved May 17, 2023, from http://www.worldairportguides.com/istanbul-ist/terminals.php.

Yan, S., Chang, C.M., 1998. A network model for gate assignment. J. Adv. Transp. 32 (2), 176–189. http://dx.doi.org/10.1002/atr.5670320204.

Yan, S., Chen, C.Y., Tang, C.H., 2009. Airport gate reassignment following temporary airport closures. Transportmetrica 5 (1), 25–41. http://dx.doi.org/10.1080/18128600802591814.

Yan, S., Tang, C.H., Hou, Y.Z., 2011. Airport gate reassignments considering deterministic and stochastic flight departure/arrival times. J. Adv. Transp. 45 (4), 304–320. http://dx.doi.org/10.1002/atr.141.

Yu, C., 2023. Airport performance – a multifarious review of literature. J. Air Transp. Res. Soc. 1 (1), 22–39. http://dx.doi.org/10.59521/E7E8098D7A835864.

Yu, C., Zhang, D., Lau, H.H., 2017a. A heuristic approach for solving an integrated gate reassignment and taxi scheduling problem. J. Air Transp. Manage. 62, 189–196. http://dx.doi.org/10.1016/j.jairtraman.2017.04.006.

Yu, C., Zhang, D., Lau, H.Y., 2017b. An adaptive large neighborhood search heuristic for solving a robust gate assignment problem. Expert Syst. Appl. 84, 143–154. http://dx.doi.org/10.1016/j.eswa.2017.04.050.

Zhang, D., Klabjan, D., 2017. Optimization for gate re-assignment. Transp. Res. B 95, 260–284. http://dx.doi.org/10.1016/j.trb.2016.11.006.