International Journal of Industrial Engineering, 30(2), 387-401, 2023

OPTIMAL REASSIGNMENT OF FLIGHTS TO AIRPORT BAGGAGE UNLOADING CAROUSELS IN RESPONSE TO TEMPORARY MALFUNCTIONS

Shangyao Yan[1], Chung-Cheng Lu[2,*], and Yi-Ting Chen[1]

[1]Department of Civil Engineering
National Central University
Zhongli, Taiwan

[2]Department of Transportation and Logistics Management
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
[*]Corresponding author's e-mail: jasoncclu@nycu.edu.tw

Being able to efficiently reassign outbound flights to baggage unloading carousels (BUCs) following temporary malfunctions is very important for airport operators. This study proposes an optimization model with a heuristic to solve the carousel reassignment problem. The objective is to minimize the total disturbance and overlapping time caused by the reassignment of outbound flights. A heuristic is developed to efficiently solve large-sized instances. The proposed approach is then applied to solve real-world instances of the problem at a major international airport in Taiwan. The computation time is about two minutes. The objective value obtained with the heuristic is more than 15% better than that obtained by the manual approach currently used by the operator. The improvement is gained mostly from the reduction in total temporal disturbance and overlapping time. The proposed approach could assist the operator in reassigning outbound flights to BUCs in response to malfunctions.

Keywords: Baggage Unloading Carousel; Flight-to-Carousel Reassignment; Airport Disturbance Management; Mathematical Programming.

(Received on October 10, 2022; Accepted on January 10, 2023)

1. INTRODUCTION

Baggage handling systems (BHSs) are an essential component of airport infrastructure. They are comprised of a series of zones (such as piers, chutes, carousels, and conveyor belts) for handling baggage. Airport baggage can be categorized into three types: inbound, transfer, and outbound. Timely handling of outbound and transfer baggage is critical for airport operations as both directly impact departing flights and passenger boarding. This study focuses on the baggage unloading system (BUS) responsible for outbound and transfer baggage for departing flights. The BUS assigns each piece of outbound or transfer baggage to an unloading zone (or carousel) after it passes through security screening and the main sorter.

The airport operator assigns baggage unloading carousels (BUCs) to scheduled departing flights. Unexpected disruptions, such as loss of power or mechanical failure, can cause BUS malfunctions that render the predetermined BUC assignment inapplicable. Disturbances resulting from BUS malfunctions can significantly delay scheduled flight departures and cause annoyance and inconvenience to boarding passengers. For instance, on December 8th, 2019, over 20 departing flights were delayed due to a BUS malfunction at Israel's Ben Gurion Airport. Another BUS malfunction occurred on March 8th, 2019, at Taiwan's Taoyuan International Airport, causing significant delays to over 40 departing flights and affecting more than 10,000 passengers. Considering that BUS malfunctions have such a significant impact on airport performance, solutions for mitigating their impact are extremely valuable to airport operators.

Airport operators generally reassign flights affected by BUC malfunctions to other available carousels and/or postpone the unloading of baggage from some flights. However, the reassignment of affected flights to available carousels inevitably causes disturbances to the original carousel assignments and inconvenience to baggage handlers, as well as passengers. Therefore, from the perspective of airport operators, it is desirable to minimize disturbances due to carousel reassignment in order to reduce the impact of BUC malfunctions on airport operations. However, the manual approach typically adopted by airport operators in practice may not be able to effectively reassign carousels immediately after BUC malfunctions. For

DOI: 10.23055/ijietap.2023.30.2.8645 ISSN 1943-670X

instance, the operator of the airport modeled in this study reassigns affected flights to available carousels with remaining capacity based on prior experience, giving priority to large-sized flights. If there is no available carousel with remaining capacity to reassign a flight (flight A), then flight A is delayed until a carousel with remaining capacity becomes available. If the delay is too long (i.e., more than two hours), then flight A will be reassigned to a carousel that was originally assigned to a later flight (flight B). In turn, flight B will have to be reassigned, and the process is repeated until all affected flights are reassigned. As can be seen, the manual approach is a local adjustment approach and is neither efficient nor effective.

To the best of our knowledge, the BUC reassignment problem in response to BUS malfunctions has never been considered, although the planning problem for BUC assignment and its variants have been addressed in some previous studies. The aforementioned practical need and this gap in the literature have motivated us to address the problem of reassigning departure flights to baggage unloading carousels in response to temporary BUS malfunctions.

An innovative optimization model with a solution algorithm is developed to solve the carousel reassignment problem from the perspective of the airport operator, which minimizes disturbances due to the reassignment while satisfying a set of operational constraints. The carousel reassignment problem under consideration can be viewed as a resource (carousel)-constrained assignment problem, which is characterized as NP-hard (Garey and Johnson, 1979). To efficiently solve large-sized instances of the proposed model, this study develops a heuristic based on the genetic algorithm (GA). The GA is an adaptive heuristic search method inspired by population genetics (Goldberg, 1989; Holland, 1975), which has been widely adopted to solve combinatorial optimization problems (e.g., Garcia-Najera *et al*., 2011; Gen and Cheng, 2000; Lu and Yu, 2012; Yu *et al*., 2019). The differences between the proposed heuristic and typical GAs are discussed below. Typical GAs adopt a mutation operator to avoid being trapped in local optima. However, it should be noted that determining a suitable mutation rate is critical to this mechanism. If the mutation rate is too high, better chromosomes cannot be retained from generation to generation, rendering the search process highly random. On the other hand, if the mutation rate is too low, the search process cannot jump out of a local optimum. To avoid the problem of determining a suitable mutation rate, this study employs a 2-Swap local search operator to replace the mutation operator used in typical GAs. Incorporating the 2-Swap local search operator can be viewed as adding a depth-search mechanism to the GAs, which already feature breadth-search in the crossover operation to increase the probability of finding high-quality solutions. Moreover, the scheme for coding the genes in a chromosome is simple and clear, facilitating the crossover and 2-Swap operations in the heuristic. The proposed model and the heuristic are evaluated using problem instances generated from real data from a major international airport in Taiwan. The performance of the proposed heuristic is compared with that of the manual approach currently used by the airport operator. Sensitivity and scenario analyses are also conducted to examine the influence of the parameters on the solution of the model.

From a theoretical perspective, this study represents an initial but significant effort to address the problem of carousel reassignment in response to BUS malfunctions. We have developed an optimization model and a heuristic to solve this problem and demonstrated the performance of our approach using real-world problem instances. In practical terms, our proposed approach provides a systematic solution that outperforms the manual approach currently used by airport operators to reassign outbound flights to available carousels in response to temporary malfunctions.

The remainder of this paper is organized as follows: Section 2 provides a literature review, Section 3 presents the problem description and assumptions, Section 4 describes the mathematical model for optimal carousel reassignment, and Section 5 presents the solution algorithm. In Section 6, we report the computational results, and we conclude the paper with some final remarks in Section 7.

2. LITERATURE REVIEW

Many decisions in airport management involve the assignment of limited infrastructure resources to many competing activities with a given schedule, such as the assignment of flights to gates, runways or maintenance facilities (e.g., Abdulmalek and Savsar, 2022; Chen and Schonfeld, 2022; Kim *et al.*, 2017; Lieder and Stolletz, 2016; Sena Daş *et al.* 2020; Tang and Wang, 2013; Yan and Huo, 2001; Yan *et al.*, 2011). The outbound BUC assignment problem is a specific problem within this broad research area. Abdelghany *et al.* (2006) adopted an activity selection heuristic to solve the problem of assigning departing flights to outbound baggage-handling facilities (called piers) to optimize the use of available piers while satisfying operational requirements. Ascó *et al.* (2012, 2014) used evolutionary and construction heuristics to solve the outbound baggage sorting station assignment problem. Huang *et al.* (2016) presented a stochastic vector assignment model for assigning outgoing flights with scheduled departure times to baggage unloading zones (chutes) under uncertainty to minimize the total expected assignment cost in the system. Frey *et al.* (2017) developed a time-indexed mathematical programming formulation for planning the outbound baggage carousel assignment. They proposed a decomposition procedure in combination with a column generation scheme to solve practical problem instances. Huang *et al.* (2018) considered uncertain flight departure times due to flight mechanical problems or weather changes and proposed a scenario-based robust optimization model to find a robust plan for assigning flights to baggage unloading areas to minimize the expected number of unassigned flights and the number of changes between planned and actual assignments. More recently, Barth and Pisinger (2021) presented a mixed integer model for assigning baggage carousels to a set of arriving flights with the aim of achieving a balance between customer satisfaction and operational needs.

3. PROBLEM DESCRIPTION AND ASSUMPTIONS

An example is used to illustrate the carousel reassignment problem under consideration, as shown in Figure 1. There are eight departure flights (two of them are large-sized flights) that are serviced by three BUCs. The original carousel assignments are shown by the arrows connecting the flights to their assigned carousels. For instance, flight #1, which is not a large-sized flight, is assigned to carousel #1 from 1:30 to 3:00; flight #3, which is a large-sized flight, is assigned to carousel #2 from 4:00 to 5:30. However, it is assumed that carousel #1 is out of order from 3:00 to 3:30 (i.e., the malfunction event lasts for 30 minutes). As a result, flights #2 to #8 are affected in this case. The aim of the carousel reassignment problem is to optimally reassign the affected outbound flights to BUCs within the planning horizon so that the total disturbance due to the reassignment is minimized. Each flight is assigned to a BUC within the associated time window. The time window represents the time allotted for baggage unloading, which depends on the amount of baggage designated to go on that flight. Any flight can be assigned to any of the available carousels. Note that this small example was solved using both CPLEX and the proposed heuristic method, and the results are presented in Section 6.2.

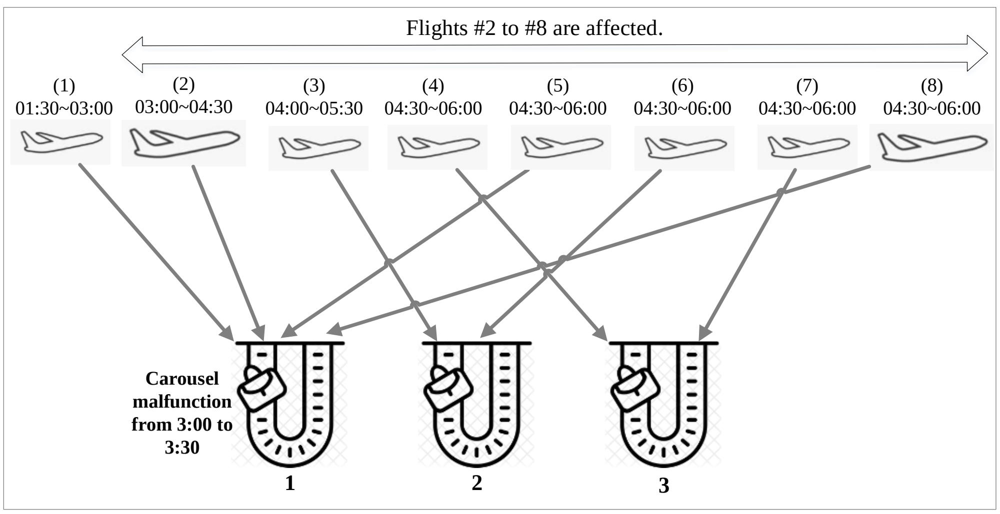

Figure 1. Illustrative Example of the Carousel Reassignment Problem

The main difference between this problem and the existing planning problem for the BUC assignment lies in the objective function. The focus in such planning problems is often on either minimizing the total assignment cost (e.g., Huang *et al*., 2016) or maximizing carousel utilization (e.g., Abdelghany *et al*., 2006). However, the aim of the carousel reassignment problem is to minimize the total disturbance due to the reassignment. Specifically, the objective function consists of two components: (i) the spatial and temporal disturbances resulting from the carousel reassignment and (ii) the time overlap for flights simultaneously using the same carousels. Reassigning flights to other available carousels typically generates spatial disturbances, while postponing the baggage unloading of some flights often results in temporal disturbances. Both spatial and temporal disturbances adversely affect airport operations, airlines, and passengers. Moreover, the reassignment of affected flights to carousels that are already being used by other flights causes an overlap in time for the flights simultaneously using the same carousels, which increases the workload of baggage handlers. The intention of the airport operator is to minimize the aforementioned disturbances and the overlapping time when reassigning flights to different carousels. Minimizing these performance measures also implies that the solution to the carousel reassignment problem does not deviate much from the original assignment.

The following assumptions are made to facilitate the modeling of the problem without loss of generality:

(a) The starting and ending times of the carousel malfunction scenario are known. The number of available carousels and their layout in the airport are given in the carousel malfunction scenario. This study does not consider the case in which all BUCs are out of order or flight cancellations.
(b) There is a buffer time (typically less than 30 minutes) for the airport operator to evaluate the impact of the BUS malfunction and to reassign the flights to the available carousels. Therefore, the starting time of the planning horizon is set to be 30 minutes after the starting time of the malfunction. The ending time of the planning horizon is determined by the severity and duration of the carousel malfunction. The (directly or indirectly) affected flights are those which were assigned to the carousels within the planning horizon.
(c) The planning horizon is discretized into a number of equal and sequentially numbered time intervals. The departure times of all the affected flights that are to be reassigned are assumed to be the integer indices for the respective time intervals. The flight number and the number of passengers for each affected flight are also known. The assumption is that each passenger checks in at least one piece of baggage, and according to the airport operator, a flight with more than 300 passengers is considered to be a large-sized flight.
(d) The original carousel assignment plan before the malfunction is given. The baggage unloading time for each flight is not affected by the malfunction.
(e) The spatial disturbance for a flight due to the reassignment is proportional to the distance (or conveyance time) between the original carousel and the reassigned carousel for that flight. Moreover, for the same distance, the spatial disturbance for a large-sized flight is larger than that for a small-sized flight because it is less preferred to reassign large-sized flights to far-away carousels.
(f) The temporal disturbance reflects the impact of postponing the assignment of a flight to a carousel at a later time interval. The temporal disturbance is computed as the number of delayed time intervals multiplied by a magnification factor (or weight). Although different weights might be applied to different flights, the airport under consideration assumes the same weights for the flights. To avoid significantly affecting connecting or downstream flights operated by the same aircraft, there is an upper bound on the maximum time (i.e., longest possible delay) for delaying the baggage unloading of an impacted flight. This largest possible delay (e.g., two hours) is determined by the airport operator.

4. MATHEMATICAL MODEL

Sets and indices:

I the set of affected flights, indexed by *i* and *r*
T the set of time intervals, indexed by *s* and *t*
J_t/J_s the set of available carousels at time interval *t* or *s*, indexed by *j*
IF the set of large-sized flights

Parameters:

α_{ij} the spatial disturbance for reassigning flight *i* from its original carousel to carousel *j* (unit: min)
β_{it} the temporal disturbance for reassigning flight *i* from its original time interval to time interval *t* (unit: min)
w_i the earliest time interval to start unloading baggage to flight *i*
l_i the latest time interval to start unloading baggage to flight *i*
d_i the baggage processing (unloading) time for flight *i* (number of time intervals used to process the baggage of flight *i*)
q_{ir} the (weighted) overlapping time for flights *i* and *r* using the same carousel in a time interval.
h the maximum number of flights that can be simultaneously assigned to a carousel in a time interval (i.e., carousel capacity)

Variables:

x_{ijt} $=1$ if flight *i* is reassigned to carousel *j* starting at time interval *t* (i.e., baggage unloading of flight *i* on carousel *j* starts at time interval *t*); otherwise, $x_{ijt} = 0$
y_{irs} the overlapping time for flights *i* and *r* using the same carousel in time interval *s*

The following sets are defined for presenting the model:

T_i the set of possible time intervals for starting the baggage unloading for flight i

M_s the set of pairs (i, t) for time interval s, where t is a possible starting time interval for unloading baggage for flight i; $M_s = \{(i,t)|w_i \le t \le l_i, s-d_i \le t \le s, i \in I\}$

T_{is} the set of time intervals for starting the baggage unloading for flight i before time interval s; that is $T_{is} = \{t|w_i \le t \le l_i, s-d_i \le t \le s\}$

F_s the subset of M_s for large-sized flights; $F_s = \{(i,t)|w_i \le t \le l_i, s-d_i \le t \le s, i \in IF\}$

O_s the set of flight pairs (i, r) with overlapping time for unloading baggage at time interval s; $O_s = \{(i,r)|w_i \le s \le l_i + d_i, w_r \le s \le l_r + d_r\}$

The optimization model is formulated as follows:

$$Min = \sum_{s\in T}\sum_{(i,r)\in O_s} y_{irs} + \sum_{i\in I}\sum_{t\in T_i}\sum_{j\in J_t}(\alpha_{ij} + \beta_{it}) \cdot x_{ijt} \quad (1)$$

Subject to

$$\sum_{w_i \le t \le l_i}\sum_{j\in J_t} x_{ijt} = 1 \qquad \forall i \in I \quad (2)$$

$$\sum_{(i,t)\in M_s} x_{ijt} \le h \qquad \forall j \in J_s, s \in T \quad (3)$$

$$\sum_{(i,t)\in F_s} x_{ijt} \le 1 \qquad \forall j \in J_s, s \in T \quad (4)$$

$$q_{ir} \cdot \left(\sum_{t\in T_{is}} x_{ijt} + \sum_{t\in T_{rs}} x_{rjt} - 1\right) - y_{irs} \le 0 \qquad \forall j \in J_s, (i,r) \in O_s, s \in T \quad (5)$$

$$x_{ijt} = 0 \text{ or } 1 \qquad \forall j \in J_t, t \in T, i \in I \quad (6)$$

$$y_{irs} \ge 0 \qquad \forall (i,r) \in O_s, s \in T \quad (7)$$

The objective function (Eq. (1)) is to minimize the sum of (i) the overlapping time of the flights simultaneously using the same carousels and (ii) the spatial and temporal disturbances resulting from reassigning the flights to the available carousels. Eq. (2) ensures that each affected flight is reassigned to one carousel. The constraints in Eq. (3) are the capacity constraints for the available carousels: the number of flights assigned to a carousel within a time interval cannot exceed the preset threshold, h. Eq. (4) prescribes that at most one large-sized flight can be assigned to a carousel in a time interval. Eq. (5) computes the overlapping time between each pair of flights (i, r) for each time interval s. In Eq. (5), q_{ir} denotes the (weighted) overlapping time for flights i and r using the same carousel in a time interval, and the phrase in parentheses $\left(\sum_{t\in T_{is}} x_{ijt} + \sum_{t\in T_{rs}} x_{rjt} - 1\right)$ computes the number of time intervals in which two flights i and r use the same carousel simultaneously. The product of these two values determines the overlapping time (y_{irs}) between a pair of flights i and r. The objective is to minimize the total overlapping time obtained by these constraints. Note that the overlapping time (q_{ir}) between a large-sized flight and a small-size flight for a time interval is a multiple of that between two small-size flights. In the numerical example, the multiple is equal to 2, as suggested by the airport operator. Eq. (6) indicates that decision variable x_{ijt} is binary, while Eq. (7) states that variable y_{irs} is non-negative.

5. SOLUTION ALGORITHM

5.1 Overview

The proposed model is formulated as a mixed integer program with binary integer variables (x_{ijt}) and without very complicated constraints, so the GA is naturally useful for coding and searching for effective solutions. A GA-based heuristic is developed to efficiently solve large-scale problem instances that are too large for the commercially available CPLEX software to solve in one day. The proposed heuristic adopts a 2-Swap operator to enhance the depth search instead of using the mutation operator in conventional GAs. The heuristic is called GA+2-Swap, hereafter in the paper.

In the heuristic, each gene in a chromosome is coded using a 4-digit integer and represents the assignment of a flight to a carousel in a time interval. The number of genes in a chromosome is equal to the number of (affected) flights that need to be reassigned to the available carousels. An illustrative example of a chromosome is shown in Figure 2. There are five genes in this chromosome, each of which corresponds to a flight. For instance, the first gene corresponds to flight A. The code of the first gene (i.e., 1522) indicates that this flight is assigned to carousel 15 at time interval 22. The code of the second gene (i.e., 0217) states that flight B starts unloading luggage at carousel 02 at time interval 17.

Gene #	1	2	3	4	5
Code	1522	0217	1004	1126	0713
Corresponding flight	A	B	C	D	E

Figure 2. Illustrative Example of the Coding for a Chromosome

A flowchart of the proposed heuristic is depicted in Figure 3. The algorithm starts by generating an initial population of feasible chromosomes. The population size for each generation is K. Fitness is defined as the inverse of the objective value because the model aims to minimize the objective function (Eq. (1)). The population evolves by creating new generations of offspring through an iterative process until a maximum number of generations is reached. The creation of a new generation of individuals involves three major operators: selection, crossover (recombination), and 2-Swap. The elitism strategy, which passes on the better chromosomes in a generation to the next one, is also applied to evolving the population. The heuristic is described in detail in the following subsections. The decoding of the final chromosome to arrive at a solution is straightforward. For each flight, the code of the corresponding gene is retrieved and divided into two parts. The first part denotes the carousel, and the second part is the time interval for the baggage of that flight to be unloaded.

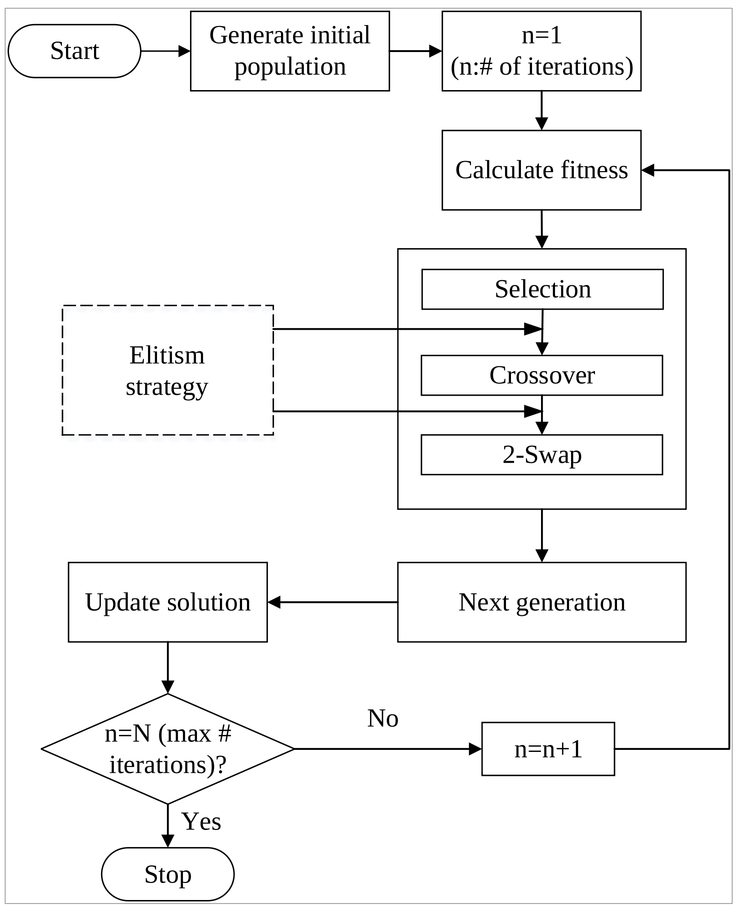

Figure 3. Flowchart of the Proposed Heuristic

5.2 Selection

This study integrates the Tournament method with the Roulette Wheel method for the selection of better chromosomes (e.g., Lu and Yu, 2012). This method involves running several tournaments between two chromosomes chosen at random from the population. The winner of each tournament (i.e., the one with a larger fitness value) is selected for reproduction. In the Roulette Wheel method, the number of replications (e_c) of the winner chromosome c that enters the mating pool is proportional to its fitness value and is calculated using the following formula:

$$e_c = C \times \frac{f_{max} - f_c}{\sum_{k=1}^{K} f_k}, \tag{8}$$

where C is the number of chromosomes in the population, f_c is the fitness value of chromosome c, and f_{max} is the largest fitness value in the population. The tournament is repeated until the mating pool is filled, i.e., the population size K is reached).

5.3 Crossover

The commonly used two-point crossover method is adopted in this study (e.g., Gen and Cheng, 2000; Lu and Yu, 2012). This crossover operator begins with the random selection of two distinct chromosomes from the mating pool. Two crossover points are then randomly selected for each of the two chromosomes. The substrings defined by the crossover points are exchanged to produce proto-children. An illustration of the two-point crossover operator is presented in Figure 4. Two chromosomes, A1 and B1, are randomly selected from the mating pool. The two crossover points are (0217 and 0713) and (0716 and 0809) in A1 and B1, respectively. The two substrings defined by the crossover points are then exchanged to produce offspring A2 and B2. Infeasible offspring (i.e., those that violate constraints (3) and/or (4)) must be repaired before they enter the next generation. The process is repeated until the preset size for the next generation is reached.

A1	1522	0217	1004	1126	0713
B1	2210	0716	0520	1430	0809

A2	1522	0716	0520	1430	0809
B2	2210	0217	1004	1126	0713

Figure 4. Illustration of the Two-Point Crossover Operator

5.4 2-Swap local search

The proposed heuristic adopts a 2-Swap local search after the crossover operator instead of a mutation operator typically used in GAs. The 2-Swap local search diversifies the search direction and improves solution quality. For each chromosome in the current population, two genes are randomly selected, and their positions are exchanged to produce a child chromosome. Figure 5 illustrates an example of the 2-Swap local search. The two genes, 0217 and 1126, are randomly selected from the parent chromosome C1. These two are then swapped to obtain the child chromosome C2. Infeasible child chromosomes are repaired to satisfy constraints (3) and (4). If the fitness value of the child chromosome is better than the parent chromosome, then the child replaces the parent in the next generation; otherwise, the parent is retained. The process is applied to all of the chromosomes in the population.

C1	1522	0217	1004	1126	0713
C2	1522	1126	1004	0217	0713

Figure 5. Illustration of the 2-Swap Local Search

5.5 Repair of infeasible chromosomes

Infeasible chromosomes resulting from the application of the crossover operator or the 2-Swap local search operator must be repaired, and only feasible offspring are allowed to enter the next generation population. Essentially, the repair ensures that (i) no flights are assigned to dysfunctional carousels, and (ii) the capacity constraint (Eq. (3)) and the large-sized flight constraint (Eq. (4)) for the available carousels are satisfied for a chromosome.

The repair procedure is depicted in Figure 6. After inputting the data, the procedure first checks whether or not all the carousel-time interval combinations satisfy the capacity constraint (Eq. (3)). If there are some carousels in some time intervals that violate this capacity constraint, the procedure will try to fix the violations by relocating extra flights one-by-one to other carousels with sufficient capacity in the same time intervals. If there is no available carousel to relocate a flight in the same time interval, then the start of the unloading time interval of that flight has to be delayed until a carousel with enough capacity is found, resulting in a temporal disturbance. Note that a change in the gene corresponding to that flight may be involved due to the delay of its unloading start time. If the unloading start time of more than one flight has to be postponed, the flight with the smallest temporal disturbance is selected to be delayed. This process is repeated until the capacity constraint is satisfied for all carousels and time intervals. If all the capacity violations can be fixed, then the procedure moves forward to check the violations for the large-sized flight constraint (Eq. (4)); otherwise, this chromosome will be abandoned, and the procedure will be stopped (i.e., there is at least one carousel-time interval combination that violates the capacity constraint and cannot be fixed). If there are some carousels in some time intervals that violate the large-sized flight constraint in Eq. (4), the repair procedure will try to fix the violations by relocating the extra large-sized flights to other carousels with sufficient capacity and delaying the unloading start time of extra large-sized flights. The process is similar to that for repairing the violation of the capacity constraint and is therefore not described again.

The flight relocation and postponement operations required for the aforementioned repair procedure can be efficiently operated in a one-dimensional data structure. Moreover, very few infeasible chromosomes are encountered in the heuristic. Thus, the repair procedure does not affect the computational efficiency of the heuristic. It should also be noted that in the crossover step, very few infeasible chromosomes that cannot be repaired are abandoned, and another pair of chromosomes is randomly selected for the next iteration of the crossover. In the 2-Swap local search step, the original feasible chromosome can be retained if an infeasible chromosome cannot be repaired. Therefore, there is always a feasible chromosome obtained in this step.

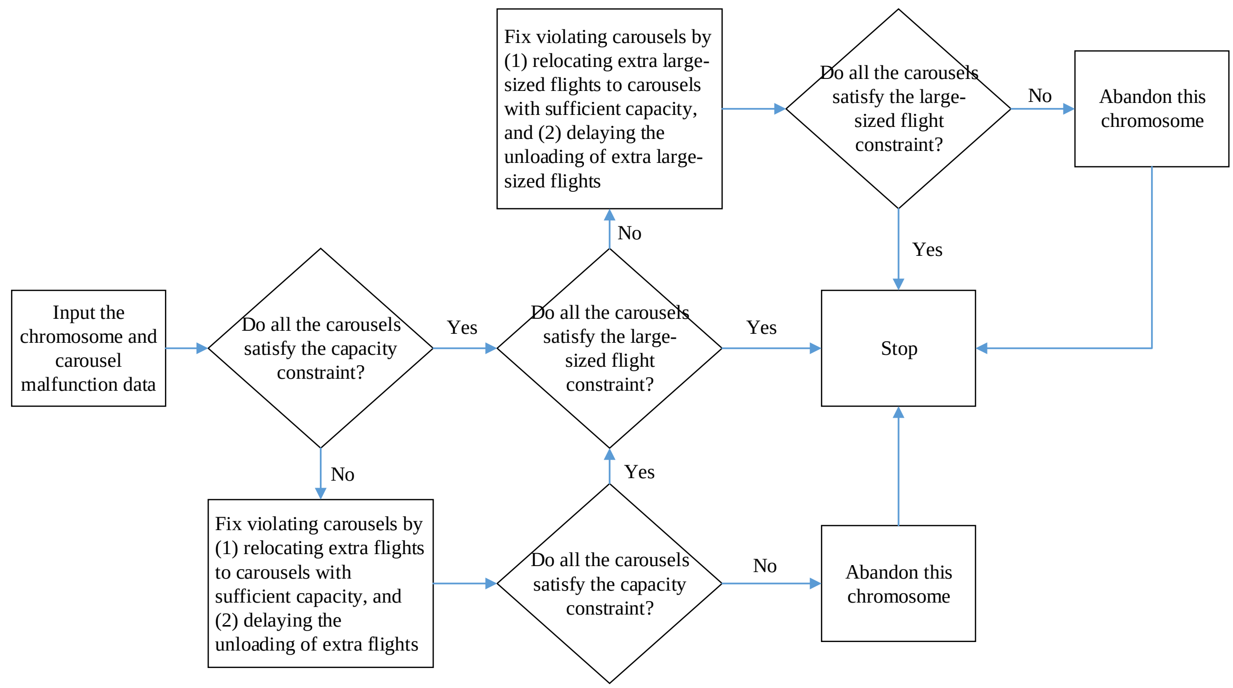

Figure 6. Repair of Infeasible Chromosomes

6. NUMERICAL EXPERIMENTS AND RESULTS

This study conducted numerical experiments to evaluate the proposed model and solution algorithm's performance in practical applications. Test instances were generated using real data from Taoyuan International Airport, the largest international airport in Taiwan. Specifically, we addressed the carousel reassignment problem for outbound flights in Terminal 2, which primarily serves long-distance flights to Europe and America that typically involve larger amounts of baggage. On average, the number of outbound flights per day is around 160, nearly equal to the number of daily inbound flights. For detailed test data and results, interested readers may contact the authors. We implemented the GA+2-Swap algorithm using the C++ programming language, and all numerical experiments were conducted on a personal computer with an Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz and 8.00 GB of RAM.

6.1 Input data

The planning horizon begins 30 minutes after the initial occurrence of the carousel malfunction scenario and extends until the end of the day. The input data required for the model comprises the layout of the baggage unloading carousels, information related to the departing flights, the original plan for carousel assignment, and the details of the carousel malfunction scenario.

(1) The layout of baggage unloading carousels is shown in Figure 7. There are 23 carousels located in three areas: six carousels in the north sector (number 1 to 6), six carousels in the south sector (number 7 to 12), and eleven carousels in the north waiting hall (number 13 to 23).

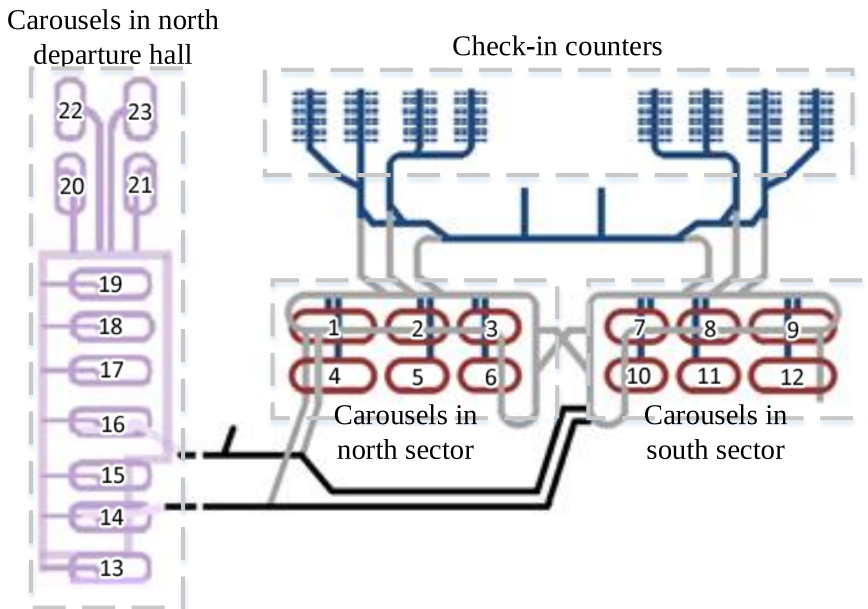

Figure 7. Layout of Baggage Unloading Carousels

(2) The data for the departure flights include the flight ID, departure time, destination, number of boarding passengers, and check-in counter for each departure flight. There is a total of 160 flights, of which 24 are large-sized flights.
(3) The original carousel assignment before the malfunction is provided by the operator. The data include the flight number, assigned BUC, and the starting and ending times for using the assigned carousel for each flight. The baggage unloading time for a flight is assumed to be 90 minutes.
(4) This study considers a malfunction scenario in which the six carousels (number 1 to 6) in the north sector break down due to a power failure. This type of carousel malfunction scenario happens occasionally in periods of high power consumption, such as on summer afternoons when demand for electricity may exceed supply. The malfunction lasts from 11:00 to 14:00, during which the six carousels in the north sector are unusable, while the other 17 carousels function normally. To allow for buffer time to generate a reassignment plan, the reassignment period starts at 11:30, with each time interval lasting for 30 minutes.

In addition to the above input data, the model and algorithmic parameters are set as follows:

(1) The maximum number of flights that can be simultaneously assigned to a carousel in a time interval (h) is three.
(2) The spatial disturbance for reassigning flight i from its original carousel to carousel j (unit: min), denoted as α_{ij}, is proportional to the conveyance time between the original carousel and carousel j. The spatial disturbances between the three areas of the BUS are presented in Table 1. In each cell, the numbers on the left and the right of the slash represent the disturbances for small-size and large-sized flights, respectively. For instance, if the original carousel and the reassigned carousel are in the same zone, the disturbance is 30 minutes for a large-sized flight and 10 minutes for a small-size flight. If one carousel is in the north sector, but the other is in the south sector, the disturbance is 20 minutes for a large-sized flight and 40 min for a small-size flight.

Table 1. Spatial Disturbances in the Three Areas (min)

	North sector	South sector	North waiting hall
North sector	10 / 30	20 / 40	60 / 80
South sector	20 / 40	10 / 30	80 / 100
North waiting hall	60 / 80	80 / 100	10 / 30

(3) The temporal disturbance for reassigning flight *i* from its original time interval to time interval *t* (unit: min), denoted as β_{it}, reflects the impact of postponing the assignment of flight *i* to a carousel at a later time interval *t*. As recommended by the airport operator, this study sets the maximum allowable delay for serving a flight to be two hours. There are four intervals (the length of a time interval is 30 minutes) for each flight reassignment. The disturbance β_{it} is computed as the number of delayed time intervals multiplied by a magnification factor. For instance, if flight *i* is to be served at time interval $t+2$, then β_{it} will be 100 (2×50), where the magnification factor is equal to 50. The value of this factor is larger than the spatial disturbance (i.e., 30) incurred when reassigning flights to different carousels. This is because postponing flights to carousels has a greater impact than reassigning flights to different carousels. Sensitivity analysis showed that the magnitude of this factor does not significantly affect the solution.

(4) The values of the major parameters of the GA algorithm (population size and crossover rate) were determined by a set of preliminary experiments. Specifically, the following values of the two parameters were examined.
Population size (K): 100, 120, 150, 180;
Crossover rate: 0.5, 0.8, 1.0.

According to the results of preliminary experiments, it has been determined that a population size (K) of 150 for each generation and a crossover rate of 0.5 yield the best objective value. Furthermore, it was observed that the objective value remains relatively stable after 300 generations. Therefore, the maximum number of generations has been set to 300 to achieve a balance between solution effectiveness and efficiency.

The optimization model presented in Section 3 was set up using the input data mentioned above. The model involves a large number of variables (185,223) and constraints (14,248,409). One of the main challenges in solving this problem is the large number of constraints required to calculate the overlapping time between each pair of flights (*i*, *r*) for each time interval *s* (as shown in Eq. (5)). The large numbers of variables and constraints in the model impede the use of off-the-shelf optimization software, such as CPLEX, to obtain an exact solution. Generating a smaller instance that can be solved by CPLEX for a preliminary evaluation of the performance of the heuristic is an acceptable compromise. As mentioned earlier, the main objective of this study is to formulate a model for solving the BUC reassignment problem. The proposed heuristic represents an initial effort to solve the problem; further elaboration and examination of the solution approach are worthy of investigation in subsequent studies.

6.2 Evaluation of the algorithm performance using a small-scale instance

A small-scale instance was generated based on the data described in Section 6.1 to preliminarily evaluate the performance of the GA+2-Swap. We solved this small instance using GA+2-Swap and compared the solution obtained by solving the optimization model using CPLEX. The instance includes eight departing flights, two of which are large-sized flights, with an original carousel assignment shown in Table 2. The baggage unloading system has one carousel in each of the north sector, south sector, and north waiting hall. In this instance, a carousel in the north sector (#1) is out of service for 30 minutes, from 3:00 to 3:30, affecting flights #2 to #8. The planning horizon is from 3:30 to 6:30. The instance has 185,223 variables and 14,248,409 constraints.

Table 2. Original Carousel Assignment of the Flights in the Small Instance

Flight ID	1	2	3	4	5	6	7	8
Large-sized flight	No	No	Yes	No	No	No	No	Yes
Assigned carousel	1	1	2	3	1	2	3	1
Starting time	01:30	03:00	04:00	04:30	04:30	04:30	04:30	04:30
Ending time	03:00	04:30	05:30	06:00	06:00	06:00	06:00	06:00

The proposed GA+2-Swap heuristic was used to solve the small instance, and the results were compared with those obtained by CPLEX. The results are presented in Table 3. The GA+2-Swap was able to obtain the optimal solution, demonstrating its potential effectiveness in solving the problem. Moreover, GA+2-Swap was computationally more efficient than CPLEX for solving this small instance. Notably, the algorithm converged at the 7th iteration, well below the maximum number of iterations set at 300, resulting in a computational time of less than 0.9 seconds.

Table 3. Solution Results of GA+2-Swap and CPLEX for this Small Instance

Solution method	GA+2-Swap	CPLEX
Objective value (min)	520	520
Total spatial disturbance (min)	20	20
Total temporal disturbance (min)	50	50
Total overlapping time (min)	450	450
Computational time (sec)	0.9	3

6.3 Numerical example results

The results obtained by solving the numerical example using the GA+2-Swap algorithm and the manual approach currently used by the airport operator are shown in Table 4. In addition to presenting the objective value and computational time, this table provides the total temporal disturbance, total spatial disturbance, and total overlapping time of flights that are simultaneously using the same carousels. The proposed solution algorithm achieves a significantly better objective value (3,570 min) compared to the manual approach (4,120 min), with the majority of the improvement (550 min or 15.4%) coming from the reduction in total temporal disturbance and total overlapping time. The reduction in total temporal disturbance is particularly advantageous for the airline since it allows for the reassignment of flights to carousels without significant changes to the flight departure schedule. Furthermore, the workload of the baggage handling system can be reduced by decreasing the total overlapping time. Lastly, the computational time required for the GA+2-Swap algorithm to solve this instance is about two minutes (138.84 sec), which is significantly more efficient than the manual approach.

The reassignment results show that most affected flights were relocated from the carousels in the north sector (numbers 1 to 6) to the carousels in the north waiting hall (numbers 13 to 23) following a malfunction that occurred in the north sector between 11:00 and 14:00. This caused a larger spatial disturbance of 1,460 minutes. However, the temporal disturbance was relatively minor, with only 50 minutes, as most affected flights could be reassigned to available carousels in almost the same time intervals as their initial assignments.

Table 4. Solution Results Obtained for this Numerical Example

Items	GA+2-Swap	Manual
Objective value (min)	3,570	4,120
Total temporal disturbance (min)	50	680
Total spatial disturbance (min)	1,460	350
Total overlapping time (min)	2,060	3,090
Computational time (min)	2.31	> 10

6.4 Comparison of GA+2-Swap and GA+2PM

To demonstrate the superiority of the proposed GA+2-Swap over the conventional GA with a mutation operator, this study implements a classical GA with a two-point mutation operator, referred to as GA+2PM, in which two randomly selected genes are exchanged to form a new chromosome. The performance of GA+2-Swap and GA+2PM for solving the numerical example described in Section 6.1 is compared.

The crossover rate is 0.5, and the maximum number of generations is 300. The mutation rates (MR) examined range from low to high MRs, including 0.1, 0.3, 0.5, and 0.8. Ten evaluations of each algorithm and parameter combination are obtained to compute the average objective value. The results are shown in Figure 8, which indicates that for the three population sizes (100, 120, and 150) and four different MRs (0.1, 0.3, 0.5, and 0.8), the average objective value of GA+2-Swap is lower than that of GA+2PM. This result demonstrates the superiority of GA+2-Swap over GA+2PM in terms of solution effectiveness. The computational times of GA+2-Swap and GA+2PM are similar (around 130 sec). Therefore, GA+2-Swap provides better solution quality without sacrificing much computational efficiency.

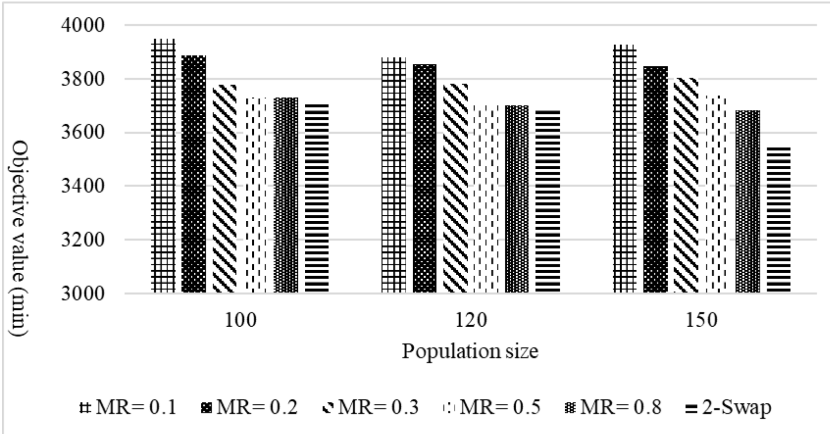

Figure 8. Comparison of GA+2-Swap and GA+2PM

6.5 Scenario and sensitivity analyses

A set of scenario and sensitivity analyses was conducted to examine the influence of the input parameters on the model solution. Although all major parameters were examined, only those with a significant influence on the solution are reported in this subsection.

6.5.1 Carousel capacity

The base case, as described in Section 5.1, allows a maximum of three flights to be assigned to a carousel simultaneously within a time interval (h). This parameter determines the carousel capacity, which affects the model through Constraint (3). One strategy the airport operator can use to deal with carousel malfunction events is to increase the carousel capacity by allowing a larger value for parameter h. To examine the influence of increasing carousel capacity on the model, the parameter is increased by one (h = 4) in addition to the base case (h = 3), as suggested by the airport operator. The computational results are shown in Table 5, which indicate a 1.13% decrease in objective value when h is increased to 4. Despite having the same total temporal disturbance (50 min) in both scenarios, the preferable solution would be to reassign flights to different carousels rather than postpone flights to later time intervals. Therefore, the temporal disturbances are the same and much lower than the spatial disturbances in both scenarios. The total overlapping time in the case of larger carousel capacity (h = 4) is smaller than that in the base case (h = 3), while the total spatial disturbance and the number of changes in carousel assignment are larger.

Increasing the carousel capacity may lead to an increase in the total overlapping time for flights, as more flights simultaneously use the same carousels. This implies a heavier workload on the staff manning the baggage handling system, which is less preferred by the airport operator. However, increasing the carousel capacity could also result in more possible combinations for carousel assignment and more changes in carousel assignment, as shown in the table. Consequently, the total spatial disturbance could also increase with the carousel capacity. This scenario analysis shows that the effect of increasing carousel capacity on the total overlapping time may actually be counteracted by the effect on the total spatial disturbance. The computational times are similar in both scenarios, indicating that the solution algorithm is efficient and not very sensitive to this parameter (h).

Table 5. Results of Scenario Analysis of the Carousel Capacity

Carousel capacity	h = 3 (base)	h = 4
Objective value (min)	3570	3530 (-1.13%)
Total spatial disturbance (min)	1460	1700
Total temporal disturbance (min)	50	50
Total overlapping time (min)	2060	1780
Number of changes in carousel assignment	30	37
Computational time (sec)	138.39	132.9

6.5.2 Duration of the carousel malfunction event

The influence of carousel malfunction duration on the solution of the model was examined, with the base case having a three-hour event window (11:00-14:00). The scenario analysis includes four additional settings, and the results are presented in Table 6. The objective value increased with the event duration. As the event duration increased, more flights were affected, resulting in increased changes to carousel assignments, total spatial disturbance, and overlapping time. However, temporal disturbances remained low across all five scenarios due to the efficient reassignment of affected flights to available carousels. The computational times demonstrated the algorithm's efficiency, with computational efficiency only minimally impacted by event duration.

Table 6. Sensitivity Analysis Result of the Event Duration

Event duration (hr)	1	2	3 (base)	4	5
Objective value (min)	3030 (-15.13%)	3130 (-12.32%)	3570 (±0%)	3810 (6.72%)	4110 (15.13%)
Total spatial disturbance (min)	790	1090	1460	1560	1790
Total temporal disturbance (min)	50	50	50	50	50
Total overlapping time (min)	1190	1990	2060	2200	2270
Number of changes in carousel assignment	22	25	30	37	40
Computational time (sec)	132.84	136.4	138.39	138.23	137.56

6.6 Discussion and Managerial Implications

This study is the first to address the carousel reassignment problem, which aims to reassign a number of outbound flights to a set of available BUCs in response to temporary malfunctions. Most airport operators around the world currently use manual approaches based on past experience, which may lead to suboptimal results. The results of the above numerical experiments show that the proposed method outperforms the manual method currently used by the airport operator in terms of both solution quality and computational efficiency. Therefore, the proposed optimization method for carousel reassignment, which aims to minimize the spatial and temporal disturbances due to the reassignment, provides a systematic approach for the airport to efficiently reallocate affected flights to available carousel-time interval combinations.

Typically, airport operations management involves making decisions about the allocation and scheduling of limited resources (e.g., check-in counters, runways, gates, and baggage carousels) for competing activities (e.g., flights) in a way that maximizes resource utilization. Optimization models and algorithms can be employed to facilitate these decisions. Although the proposed optimization model and heuristic were designed to address the carousel reassignment problem, the proposed approach and its basic concepts can be adapted to reassign or reschedule other types of airport resources for inbound or outbound flights in response to temporary malfunctions. Note that the primary objective of applying optimization approaches to the reassignment or rescheduling problems is to minimize the total disturbance due to the reassignment or rescheduling, with the total cost generally being of less concern to airport operators. Moreover, while the case study detailed in this paper uses data from an airport in Taiwan, the proposed approach can be adopted to solve the carousel reassignment problem in airports in other countries with some adjustments to the parameter settings in the model objective, constraints, and heuristic.

The proposed approach can be applied to the production re-scheduling problem in industrial engineering. For instance, in a hybrid manufacturing system or its variants (e.g., Yilmaz and Durmuşoğlu, 2018a, 2018b, 2019), the machines in the system are considered as carousels, and the tasks (or products) are treated as flights (or their baggage). The approach can be utilized to resolve the multi-machine re-scheduling problem due to temporary machine malfunctions. In this case, the decision maker needs to reassign or re-schedule the production tasks to available machines such that some objectives (e.g., disturbances, costs, or completion time) are optimized. The optimization model for the machine rescheduling problem may need to minimize the changes in the assignment of the (affected) tasks to the available machines, in addition to the original objective or multiple objectives (e.g., average flow time, number of workers, and completion time). Furthermore, the idea of integrating a local search operator, such as 2-Swap, into the GA can be extended to existing evolutionary algorithms (e.g., NSGA-II) to solve multi-objective machine rescheduling problems.

7. CONCLUDING REMARKS

In this paper, we addressed the BUC reassignment problem by formulating it as a resource-constrained assignment problem. The objective is to minimize spatial and temporal disturbances as well as total overlapping time resulting from reassignment while ensuring that all affected flights are assigned. The model takes into account practical requirements important to the airport operator. To solve practical-sized instances, this study developed a heuristic approach called GA+2-Swap. The proposed model and heuristic were evaluated using problem instances generated from real data from a major international airport in Taiwan. The results demonstrated that the proposed GA+2-Swap heuristic outperforms the manual method currently used by the airport operator and produces better solutions in a shorter amount of time. Our approach can assist the airport operator in making decisions on carousel reassignment in response to BUC malfunctions.

There are several directions for future research, as discussed below. First, although the airport operator does not prioritize cost-related items, they may be considered in the objective function, particularly for other airport applications. The current model does not consider cost-related items with baggage handling or transportation in the objective function. This could lead to the underestimation of the impact of carousel reassignment on airport operations. Secondly, a more generalized definition of spatial and temporal disturbances is necessary for the model to be applicable to other airports. The current computation method of the spatial and temporal disturbances is primarily designed for the airport modeled in this study and might not be suitable for other airports. Thirdly, appropriate methods need to be developed to obtain good quality lower bounds to evaluate the effectiveness of the heuristic. The performance of the heuristic was evaluated and found to be superior to the manual approach currently used by the airport operator. However, note that the focus of this study is on developing a systematic approach to address the carousel reassignment problem and conducting preliminary tests. The development of other new algorithms and comparing their performance in solving the problem are non-trivial tasks that are beyond the scope of this paper but could be investigated in future studies.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Science and Technology, Taiwan under grant numbers MOST107-2221-E-008-018-MY3 and MOST-110-2221-E-008-025-MY3.

REFERENCES

Abdelghany, A., Abdelghany, K. and Narasimhan, R. (2006). Scheduling Baggage-handling Facilities in Congested Airports. *Journal of Air Transport Management*, 12(2): 76–81.

Abdulmalek, F. and Savsar, M. (2022). Modeling and simulation analysis of aircraft maintenance operations. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 29(4).

Ascó, A, Atkin, J. A. and Burke, E. K. (2012). An Evolutionary Algorithm for the Over-constrained Airport Baggage Sorting Station Assignment Problem. *Simulated Evolution and Learning*, in *Lecture Notes in Computer Science*, 7673: 32–41.

Ascó, A, Atkin, J. A. and Burke, E. K. (2014). An Analysis of Constructive Algorithms for the Airport Baggage Sorting Station Assignment Problem. *Journal of Scheduling*, 17(6): 601–619.

Barth, T. C. and Pisinger, D. (2021). Baggage Carousel Assignment at Airports: Model and Case Study. *SN Operations Research Forum*, 2(1): 1-27.

Chen, C.-Y. and Schonfeld, P. (2022). Unit Load Device Reassignment Following Temporary Airport Closures. *International Journal of Shipping and Transport Logistics*, Online First Articles.

Frey, M., Kolisch, R. and Artigues, C. (2017). Column Generation for Outbound Baggage Handling at Airports. *Transportation Science,* 51(4): 1226–1241.

Garcia-Najera, A. and Bullinaria, J. A. (2011). An Improved Multi-Objective Evolutionary Algorithm for the Vehicle Routing Problem with Time Windows. *Computers & Operations Research,* 38(1): 287-300.

Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP Completeness*. San Francisco: Freeman.

Gen, M. and Cheng, R. (2000). *Genetic Algorithms and Engineering Optimization.* New York: John Wiley & Sons Inc.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Reading, MA: Addison-Wesley.

Holland, J. *H*. (1975). *Adaptation in Natural and Artificial Systems.* Ann Arbor, MI: University of Michigan Press.

Huang, E., Mital, P., Goetschalckx, M. and Wu, K. (2016). Optimal Assignment of Airport Baggage Unloading Zones to Outgoing Flights. *Transportation Research Part E: Logistics and Transportation Review*, 94: 110–122.

Huang, E., Liu, I., and Lin, J. T. (2018). Robust Model for the Assignment of Outgoing Flights on Airport Baggage Unloading Areas. *Transportation Research Part E: Logistics and Transportation Review*, 115: 110-125.

Kim, S. *H*., Feron, E. and Clarke, J. P., Marzuoli, A. and Delahaye, D. (2017). Airport Gate Scheduling for Passengers, Aircraft, and Operation. *Journal of Air Transportation*, 25(4): 109-114.

Lieder, A. and Stolletz, R. (2016). Scheduling Aircraft Take-offs and Landings on Interdependent and Heterogeneous Runways. *Transportation Research Part E: Logistics and Transportation Review*, 88: 167-188.

Lu, C.-C. and Yu, V. F. (2012). Data Envelopment Analysis for Evaluating the Efficiency of Genetic Algorithms on Solving the Vehicle Routing Problem with Soft Time Windows. *Computers & Industrial Engineering*, 63(2): 520-529.

Sena Daş, G., Gzara, F. and Stützle, T. (2020). A Review on Airport Gate Assignment Problems: Single Versus Multi Objective Approaches. *OMEGA*, 92: 102146.

Tang, C. *H*. and Wang, W.C. (2013). Airport Gate Assignments for Airline-Specific Gates. *Journal of Air Transport Management*, 30: 10-16.

Yan, S, and Huo, C. M. (2001). Optimization of Multiple Objective Gate Assignments. *Transportation Research Part A: Policy and Practice*, 35(5): 413-432.

Yan, S., Hsiao, F.-Y., Guo, J. and Chen, Y.-C. (2011). Effective aircraft maintenance schedule adjustment following incidents. *Transportation Planning and Technology,* 34(8): 727-745.

Yilmaz, Ö. F., & Durmuşoğlu, M. B. (2018a). Evolutionary algorithms for multi-objective scheduling in a hybrid manufacturing system. In *Handbook of research on applied optimization methodologies in manufacturing systems.* IGI Global.

Yilmaz, Ö. F. and Durmuşoğlu, M. B. (2018b). "A performance comparison and evaluation of metaheuristics for a batch scheduling problem in a multi-hybrid cell manufacturing system with skilled workforce assignment." *Journal of Industrial & Management Optimization* 14(3), 1219-1249.

Yilmaz, Ö. F. and Durmuşoğlu, M. B. (2019). Multi-objective scheduling problem for hybrid manufacturing systems with walking workers. *The International Journal of Industrial Engineering: Theory, Applications and Practice,* 26(5):625-650.

Yu, X., Zhou, Y. and Liu, X. F. (2019). A Novel Hybrid Genetic Algorithm for the Location Routing Problem with Tight Capacity Constraints. *Applied Soft Computing,* 85: Article No. 105760.