International Journal of Production Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tprs20

The airport shuttle bus scheduling problem

Nihat Öner , Hakan Gultekin & Çağrı Koç

To cite this article: Nihat Öner , Hakan Gultekin & Çağrı Koç (2020): The airport shuttle bus scheduling problem, International Journal of Production Research, DOI: 10.1080/00207543.2020.1841317

To link to this article: https://doi.org/10.1080/00207543.2020.1841317

Published online: 18 Nov 2020.

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
https://doi.org/10.1080/00207543.2020.1841317

Taylor & Francis
Taylor & Francis Group

Check for updates

The airport shuttle bus scheduling problem

Nihat Öner [a], Hakan Gultekin [a,b] and Çağrı Koç [c]

[a]Department of Industrial Engineering, TOBB University of Economics and Technology, Ankara, Turkey; [b]Department of Mechanical & Industrial Engineering, Sultan Qaboos University, Muscat, Oman; [c]Department of Business Administration, Social Sciences University of Ankara, Ankara, Turkey

ABSTRACT
This paper introduces the airport shuttle bus scheduling problem (ASBSP) as a new practical scheduling variant. In this problem, a number of identical vehicles that have a specific number of available seats provides transfer service between the airport and the city centre. After making a transfer in one direction, the vehicle can either make a new transfer in the opposite direction depending on the availability and the schedule of the passengers or make an empty return to make a new transfer in the same direction. The vehicles can wait in either location until their next transfer. The passengers have certain time windows for the transfer in relation to their flight times and operational rules to satisfy customer satisfaction. This is a profit-seeking service where transfer requests can also be rejected. The ASBSP aims to prepare a daily schedule for the available vehicles and to assign passengers to these vehicles with the objective of maximising the total profit. This paper presents two alternative mixed integer programming formulations and proposes two valid inequalities to get better bounds. Furthermore, it develops a hybrid metaheuristic that integrates multi-start, simulated annealing and large neighbourhood search for its solution. Extensive computational experiments on real-life benchmark instances have been made to test the performances of the formulations and the hybrid metaheuristic. Furthermore, the impacts of several problem parameters including the number of vehicles, vehicle capacity, transfer fee, transportation time and allowable passenger waiting times on the problem complexity and results have been investigated.

ARTICLE HISTORY
Received 28 January 2020
Accepted 15 October 2020

KEYWORDS
Airport shuttle bus service; scheduling; hybrid metaheuristic

1. Introduction

The aviation industry is one of the fastest growing contributors to global economic growth and helped the rise of globalisation by bringing people together. The industry has doubled in size every 15 years which increased the demand for more flights, more destinations as well as more aircrafts. The current size of worldwide commercial fleet is about 25,000 aircrafts and it is expected to be 35,000 in the next decade (KPMG 2018). To increase their competitiveness, many airlines provide auxiliary services to their passengers such as shuttle transfers. Some airline companies provide this service with an additional price to their customers (Turkish Airlines 2020). There are also private transfer companies that serve independent of the airlines (Antalya Shuttle 2020). A number of shuttle vehicle is used to serve in both directions between the airport and the city centre to satisfy the requests collected from the passengers at least 1 day ahead. This is a profit-seeking operation, and hence not all transfer requests need to be satisfied. Planning the shuttles for small number of passengers and vehicles with relatively longer time intervals between the transfers is quite easy. However, the problem complexity increases drastically with the number of passengers, the number of vehicles and the frequency of transfers. We called this real-life problem as the airport shuttle bus scheduling problem (ASBSP) which deals with the profit, service quality and customer satisfaction.

The ASBSP has some similarities with the parallel batch processing machine scheduling problem (PBPMS). Batch processing machines, such as ovens, allow simultaneous processing of several jobs. The passengers in the ASBSP can be considered as jobs of the PBPMS that have equal sizes. The vehicles in the ASBSP can be considered as identical batch processors in the PBPMS that have bounded capacities. The transfer times correspond to the processing times of the jobs. Furthermore, the current problem also includes job release times, deadlines and sequence dependent setup times. The batch processor scheduling problem is introduced by Ikura and Gimple (1986), for which many variations of the problem have been considered thereafter. These studies can be reviewed from Potts and Kovalyov (2000); Mathirajan and Sivakumar (2006); Xiao and Shao (2013). Among the

CONTACT Çağrı Koç cagri.koc@asbu.edu.tr

ones that considered parallel batching machines, several of them considered job release times and/or setup times (see, e.g. Chen, Du, and George 2010; Cakici et al. 2013; Shahvari and Logendran 2017; Yu, Huang, and Lee 2017). Trindade et al. (2018) provides mixed integer linear programming formulations for parallel batch scheduling problems with makespan objective and with and without release times. They propose several symmetry breaking constraints to reduce the size of the feasible set.

Certain characteristics of the ASBSP differentiate it from the PBPMS. First, the PBPMS includes classical time related scheduling objectives such as minimisation of the makespan, tardiness and total completion time (see, e.g. Thomasson et al. 2018; Perez-Gonzalez et al. 2019; Zhang, Yao, and Li 2020). On the other hand, different than the PBPMS literature, the objective of the ASBSP is to maximise the total profit. Additionally, in the PBPMS, all jobs have to be processed. However, the transfer requests can be rejected in the ASBSP for sake of profit maximisation. The deadlines of jobs in the ASBSP are also different than the due dates that are considered in some studies on the PBPMS. Basically, due dates are soft constraints whereas deadlines are hard constraints. Because of these differences, problem-specific solution methods developed for the PBPMS should be adapted before they can be used for the ASBSP. In Section 2.1, we show how the mathematical formulations that are developed for the PBPMS can be adapted for the ASBSP. On the other hand, in the ASBSP the processing times, i.e. transfer times are equal. Also, in the ASBSP there are multiple jobs with common ready dates and deadlines. These are some of the unique features of the ASBSP. Using these features, an alternative formulation is developed in Section 2.2. Our tests showed that this new formulation is computationally more efficient than the PBPMS based formulation.

From another perspective, scheduling of airport shuttle services can be considered as a variant of the vehicle routing problem (VRP). The objective of the classical VRP is to find a routing for a number of identical vehicles. The route of each vehicle must start and end at the depot and each customer is visited once by only one of the vehicles. Many types of solution methods have been developed for the VRP and its variations (see Toth and Vigo 2014; Koç and Laporte 2018; Koc, Laporte, and Tukenmez 2020; Diri Kenger, Koç, and Özceylan 2020). Several papers studied optimisation of airport shuttle services in different contexts. Song, Dinwoodie, and Roe (2007) proposed a Markov model to decide the size of the fleet, dispatching of the loaded vehicle and repositioning of the empty vehicles. The vehicles provide shuttle service between two terminals at the airport to satisfy random customer demands. Dong et al. (2011) considered the problem of transportation of passengers to the airport as a service provided by flight ticket sales companies. The authors formulated the problem as an extension of the VRP with time windows and aimed to minimise fixed start-up costs and variable transportation costs. Tang, Yu, and Li (2015) considered the transfer of customers to the airport by the flight ticket sales companies in China. Different than the current study, this paper assumes only a one-way transfer of the customers (to the airport) with the objective of minimising the total operating cost. Another difference is the assumption that the number of available vehicles is satisfactory and all requests of the customers must be satisfied. However, in the current study, the number of vehicles is limited and customer requests can be rejected in order to maximise the profit. Assuming two-way transfers complicates the problem since the vehicle capacity is replenished at multiple locations. Furthermore, the vehicles can either wait idle at some location for the next transfer in that direction or travel empty to the other location to make another transfer in the opposite direction.

Another model for transporting people is the ridesharing systems (see Agatz et al. 2012; Furuhata et al. 2013; Savelsbergh and Van Woensel 2016). These systems bring together travellers with similar time schedules and itineraries on short-notice. Ridesharing has significant potential to reduce the number of used cars for transporting people. Although there are some similarities between the ridesharing systems and the ASBSP, the business models are different from each other. In the ASBSP, all vehicles are owned by the company and are available for passenger transfer all the time. However, in ridesharing models, the vehicles are owned by individuals who optionally take part in the transfers. In ridesharing models, the planning problem mostly consists of matching the demand with the most suitable vehicle. However, in the ASBSP, the planning problem consists of grouping the passengers and routing the vehicles to maximise the profit.

To our knowledge, there is no other similar problem that includes all of the features of the ASBSP in the literature, and the comprehensive structure of airport shuttle services has not yet been investigated in spite of its practical importance. We believe that solving this problem contributes to the scientific literature and real practice.

The main contributions of this study can be listed as follows. First, we define a new practical scheduling problem named ASBSP. We develop a mixed integer linear programming formulation based on the PBPMS. Using unique features of the ASBSP, we propose a more efficient formulation that is based on the VRP. We propose two valid inequalities to improve its performance. Second, we develop a hybrid metaheuristic for the ASBSP which combines multi-start initialisation, passenger grouping procedures, simulated annealing and large neighbourhood search. Our third contribution is to perform extensive computational analyses on real-life benchmark instances. We provide several managerial and policy insights for the ASBSP. We investigate the effects of various problem parameters and components of the algorithm on the results and the solutions. These parameters include the number of vehicles, vehicle capacity, transportation cost per passenger, transportation time, and the allowable waiting time for the passengers at the airport and at the city centre.

The remainder of this paper is organised as follows. Section 2 describes the problem in detail and provides the mathematical formulations. Section 3 presents the metaheuristic solution method. Experimental study is reported in Section 4, followed by concluding remarks in Section 5.

2. Problem description and mathematical formulations

In the ASBSP (illustrated in Figure 1), passengers are transferred between the city centre and the airport using a fleet of homogeneous vehicles owned by the shuttle service company. It is assumed that there is a single pick-up/drop-off point both at the airport and in the city centre. Passengers can request a transfer from the airport to the city centre (A–C) or from the city centre to the airport (C–A). For each flight performed at each day, the historical data of transfer requests are available. The number of transfer requests for each flight is estimated by analysing this data. Using this information, the service planning should be made and the transfer start times should be determined.

For A–C transfers, flight arrival time information is used to make the planning. The service company adds a fixed amount of time to the arrival time to account for walking and luggage collection to calculate the ready time of the passengers. Hence, the ready time of passengers denotes the time when they are out of the terminal building and ready for the transfer. On the other hand, as a company rule, A–C passengers should not wait more than a certain amount of time after their ready time. As a consequence, the transfer of an A–C passenger should start within a time window.

On the other hand, for C–A transfers, flight departure time information is used for the planning. As a regulation of the airline companies, passengers should arrive at the airport before a certain amount of time before their departure time. For domestic flights that are considered in the course of this study this time is set as 1 hour. This means that, each C–A passenger has a deadline to reach the airport. On the other hand, in order not to make the passengers wait long for their flights at the airport, they cannot be picked up too early. As a consequence, similar to the A–C case, the C–A transfers should also be started within a certain time window. Passengers with the same flight arrival or departure time have identical transfer time windows, and hence, these passengers are assumed to be identical in the planning.

The reservations are collected at least 1 day before the date of flights from the passengers. The transfer requests of some customers can be rejected due to unavailability

Figure 1. An illustration of the ASBSP.

of capacity. Also, the transfer requests can be rejected to avoid transfers with underutilised vehicles. This way the transfer costs can be reduced. For the confirmed reservations, the transfer start time is notified to the passengers.

The vehicles have a limited seat capacity. This capacity is renewed when the vehicle delivers the passengers to the airport or to the city centre. After completing a transfer in each direction, the vehicle can either take (after waiting some time, if necessary) new passengers and transfer in the opposite direction or it can return back empty to transfer new passengers in the same direction. We assume that the cost of a loaded trip is more than the cost of an empty trip because of the increased fuel consumption due to the load. The problem is to schedule the vehicles and to assign the passengers to the vehicles to maximise the total profit.

In the following section, we develop a mixed integer linear programming formulation based on the PBPMS formulations. However, the model size increases rapidly with respect to the number of passengers. As already mentioned, the passengers that have the same flight arrival or departure times are assumed to be identical. As a consequence, there are many identical passengers in the considered problem. Making use of this property, we develop an alternative formulation based on the VRP and grouping of the passengers in Section 2.2.

2.1. Parallel batch processor machine scheduling formulation

Here we first develop a formulation based on the PBPMS. In the PBPMS formulations, the jobs are assigned to batches and the batches are assigned to the parallel machines. The passengers in the current problem are similar to the jobs in the PBPMS and the vehicles are similar to the machines. The batches correspond to trips and the transfer times correspond to the processing times. The set of passengers is denoted by $\mathcal{N} = \{1, \ldots, N\}$. This set includes all passengers in the A–C or C–A directions. The set of homogeneous fleet of vehicles is denoted by $\mathcal{M}$.

Each vehicle has a capacity of w seats. C–A passengers must reach to the airport at least s minutes before their departure time to complete their check-in operations. The maximum allowable waiting time for A–C (C–A) passengers is denoted by e_1 (e_2) minutes after their ready time. Transfer fare is denoted by v. Travel time in each direction is equal and denoted by p. This is a realistic assumption for the ASBSP since the airports are usually outside the city centres with strong highway connections that reduce uncertainties such as traffic congestion that affect the transfer time. Direction of passenger $j \in \mathcal{N}$ is denoted by binary parameter a_j. If passenger j is in A–C direction, then $a_j = 1$ and if it is in C–A direction, $a_j = 0$.

Π_j denotes the arrival (departure) time of passenger j that is in A–C (C–A) direction. Let r_j be the ready time of passenger j. The ready time depends on the direction of transfer. If $a_j = 1$, then $r_j = \Pi_j$. If $a_j = 0$, then the ready time of the passenger is determined by subtracting the check-in time, the transfer time, and allowable waiting time from the departure time, that is, $r_j = \Pi_j - s - p - e_2$. Let D_j be the latest time that the transfers can be started. This also depends on the direction of the transfer. If $a_j = 1$, then $D_j = \Pi_j + e_1$; otherwise, $D_j = \Pi_j - s - p$. All transfers must start in between r_j and D_j.

We define set of trips at each vehicle as $\mathcal{H}_{||} = \{1, \ldots, H_k\}$, where H_k is the maximum number of trips that a vehicle can perform in a day. Since the transfer time for one trip is p minutes, then $H_k \leq (\Pi_{j_1} - \Pi_{j_2})/p$, where Π_{j_1} and Π_{j_2} represent the latest and earliest flight arrival/departure times, respectively, in the considered day. Since all vehicles and their transfer times are identical to each other, H_k is the same for all vehicles. Therefore, we can drop the k index and use $\mathcal{H}$ to represent the set of trips.

We summarise all the notation for sets, parameters and decision variables in Table 1.

Using this notation, we develop the following mixed integer linear programming formulation for the ASBSP.

MIP1:

$$\max \sum_{j\in\mathcal{N}} \sum_{h\in\mathcal{H}} \sum_{k\in\mathcal{M}} v x_{jhk} - \sum_{h\in\mathcal{H}} \sum_{k\in\mathcal{M}} c_1 z_{hk} - \sum_{h\in\mathcal{H}} \sum_{k\in\mathcal{M}} c_2 y_{hk} \tag{1}$$

subject to

$$\sum_{h\in\mathcal{H}} \sum_{k\in\mathcal{M}} x_{jhk} \leq 1 \quad \forall j \in \mathcal{N} \tag{2}$$

$$\sum_{j\in\mathcal{N}} x_{jhk} \leq w \quad \forall h \in \mathcal{H},\ k \in \mathcal{M} \tag{3}$$

$$x_{ihk} + x_{jhk} \leq 1 \quad \forall i,j \in \mathcal{N},\ h \in \mathcal{H},\ k \in \mathcal{M} : a_i \neq a_j,\ i \neq j \tag{4}$$

$$x_{jhk} \leq z_{hk} \quad \forall j \in \mathcal{N},\ h \in \mathcal{H},\ k \in \mathcal{M} \tag{5}$$

$$z_{hk} \leq z_{(h-1)k} \quad \forall h \in \mathcal{H},\ k \in \mathcal{M} : h > 1 \tag{6}$$

$$R_{hk} \geq r_j x_{jhk} \quad \forall j \in \mathcal{N},\ h \in \mathcal{H},\ k \in \mathcal{M} \tag{7}$$

$$C_{hk} \geq R_{hk} + p - B(1 - z_{hk}) \quad \forall h \in \mathcal{H},\ k \in \mathcal{M} \tag{8}$$

$$C_{hk} \geq C_{(h-1)k} + p + p y_{hk} - B(1 - z_{hk}) \quad \forall h \in \mathcal{H},\ k \in \mathcal{M} : h > 1 \tag{9}$$

Table 1. Summary of the notation.

Sets	
$\mathcal{N}$	set of passengers
$\mathcal{H}$	set of trips
$\mathcal{M}$	set of vehicles
Parameters	
w	seat capacity of each vehicle
s	check-in and allowance time before their departure times for the C–A passengers
e_1, e_2	the maximum allowable waiting times for the A–C and C–A passengers, respectively
v	transfer fare per passenger
p	transfer time in each direction
a_j	binary parameter; 1 (0) if passenger j travels in A–C (C–A) direction
r_j	ready time for passenger j
D_j	due date for passenger j
c_1, c_2	loaded and empty transfer costs, respectively
B	big number. B must be larger than the number of minutes in a day ($B \geq 1440$)
Binary variables	
x_{jhk}	1 if passenger $j \in \mathcal{N}$ is assigned to trip $h \in \mathcal{H}$ in vehicle $k \in \mathcal{M}$, 0 otherwise
z_{hk}	1 if trip $h \in \mathcal{H}$ in vehicle $k \in \mathcal{M}$ is used, 0 otherwise
y_{hk}	1 if there is an empty travel before trip $h \in \mathcal{H}$ in vehicle $k \in \mathcal{M}$, 0 otherwise
Continuous variable	
R_{jk}	ready time of trip $h \in \mathcal{H}$ in vehicle $k \in \mathcal{M}$
C_{jk}	completion time of trip $h \in \mathcal{H}$ in vehicle $k \in \mathcal{M}$

$$C_{hk} \leq D_j + p + B(1 - x_{jhk}) \quad \forall j \in \mathcal{N},\ h \in \mathcal{H},\ k \in \mathcal{M} \tag{10}$$

$$y_{hk} \geq x_{i(h-1)k} + x_{jhk} - (1 - |a_i - a_j|) \quad \forall i, j \in \mathcal{N},\ h \in \mathcal{H},\ k \in \mathcal{M} : h > 1 \tag{11}$$

$$x_{jhk}, y_{hk}, z_{hk} \in \{0, 1\} \quad \forall j \in \mathcal{N},\ h \in \mathcal{H},\ k \in \mathcal{M} \tag{12}$$

$$R_{hk}, C_{hk} \geq 0 \quad \forall h \in \mathcal{H},\ k \in \mathcal{M}. \tag{13}$$

The objective (1) maximises the total profit. The first term of the objective function calculates the total ticket revenue from the transferred passengers. The second term calculates the cost of loaded trips and the third term calculates the cost of the empty trips. Constraints (2) assure that each passenger can be assigned to at most one trip and vehicle. The total number of passengers in the same trip of the same vehicle cannot exceed the vehicle capacity as satisfied by Constraints (3). The passengers that travel in opposite directions cannot be assigned to same trip in the same vehicle. This is satisfied by Constraints (4). Constraints (5) guarantee that the passengers cannot be assigned to the unused trips of the vehicles. Constraints (6) satisfy the sequence of the trips and guarantee that if an earlier trip of a vehicle is not used then, none of the later trips can be used. A trip cannot start if all of the passengers assigned to that trip is not ready. Therefore, the ready time of a trip is equal to the maximum of the ready times of the passengers that are assigned to that trip. This is satisfied by Constraints (7). The completion time of a trip is defined as the time of arrival to the destination. According to this definition, the completion time must be later than the ready time of the trip plus the transfer time as satisfied by Constraints (8). Similarly, the completion time must be later than the completion time of the previous trip of the vehicle plus any necessary empty travel time in between these two trips plus the transfer time of the considered trip. This is satisfied by Constraints (9). On the other hand, the passengers cannot be kept waiting more than the allowable waiting time. Constraints (10) guarantee this by limiting the completion time of each trip depending on the due dates of individual passengers assigned to that trip. Constraints (11) identify if there is an empty travel between two consecutive trips of the same vehicle. If passenger i of trip $h-1$ and passenger j of trip h of the same vehicle have the same transfer direction, then there will be an empty travel between these two consecutive trips. Constraints (12) and (13) are the integrality and non-negativity constraints, respectively.

We tested this model with smaller instances containing 180 passengers and 4 vehicles using CPLEX 12.8. Even after 12 hours, the gap value was more than 300%. This formulation deals with each passenger individually. In the real-life instances that we used for numerical studies, there are approximately 1700 passengers in a single day. This results in huge model size. This proved that MIP1 cannot provide solutions in reasonable times even for small-size instances. As a consequence, in the following section, we develop an alternative formulation that uses unique features of the ASBSP.

2.2. Vehicle routing formulation

In this section, we formulate the problem as a variation of the VRP. As mentioned earlier, the passengers of the same direction that have the same ready times are identical. To reduce problem complexity, we group these identical passengers altogether and name them as a 'cohort' instead of handling them individually. Hence, each passenger is a member of one cohort. The set of all cohorts is denoted by $\mathcal{J} = \{0, \ldots, J\}$ where '0' and 'J' are dummy cohorts indicating the first and last visited cohorts that are used in the VRP formulation. For each cohort, the number of passengers, the direction of transfer and the ready times are already known. The number of passengers in cohort j is named as the size of cohort j and denoted by Q_j. This number is determined by analysing the historical data that include the total number of passengers requesting transfer for each flight. The real life instances contain approximately 1700 passengers which can be combined into approximately 75 cohorts. The exact number of passengers in each cohort varies but the average is approximately 23 passengers.

The passengers of the same cohort can be split and transferred by different vehicles. It is also possible to assign passengers from different cohorts with agreeable ready times to the same trip. The fundamental reason of dividing or grouping different cohorts in this way is to utilise the seat capacity more efficiently.

Let g_{ij} be a binary indicator that denotes whether cohort $i \in \mathcal{J}$ and cohort $j \in \mathcal{J}$ can be grouped or not. Grouping of cohorts depends on the direction of the cohorts and their ready times. If the difference between ready times of cohorts i and j in C–A (A–C) direction is less than e_2 (e_1) minutes, the cohorts can be assigned to the same group, that is, $g_{ij} = 1$, otherwise $g_{ij} = 0$.

If a vehicle visits cohorts i and j consecutively and if these cohorts are in opposite directions, then there is no empty travel in between. In this case, the transfer cost is c_1. However, if the consecutive cohorts are in the same direction and their ready times do not agree, so that they cannot be assigned to the same group ($g_{ij} = 0$), then the vehicle makes an empty return after completing the transfer of cohort i to start the transfer of cohort j. Then, the transfer cost includes a loaded transfer and an empty transfer. The total cost is $c_1 + c_2$.

Different than MIP1, we define the following binary variables in order to formulate our problem: u_k indicates if vehicle k is used for transfers or not and useful for calculating the objective function. y_{jk} takes the value of 1 if cohort j is assigned to the vehicle k, and 0 otherwise. x_{ijk} indicates if cohort i is assigned just before cohort j in vehicle k. b_{ijk} indicates if cohorts i and j are in the same trip in vehicle k. Furthermore, $l_{ijk} = x_{ijk}(1 - b_{ijk})$ is used for linearisation of the multiplication of decision variables and takes the value of 1 if cohort i is scheduled just before cohort j in vehicle k on different trips. This is necessary to calculate the travel costs in the objective function. t_{jk} is a continuous variable identifying the transfer starting time of cohort j in vehicle k. f_{jk} denotes the number of passengers of cohort j that are assigned to vehicle k. h_{ijk} denotes the number of passengers of cohort j that are assigned to the same vehicle with cohort i in vehicle k.

Table 2 summarises the updated or additional notation that is needed for the new formulation.

We develop the following mixed integer linear programming formulation for the ASBSP.

MIP2:

$$\max \sum_{j=1}^{J-1} \sum_{k\in\mathcal{M}} v f_{jk} - \sum_{i=1}^{J-1} \sum_{j=1}^{J-1} \sum_{k\in\mathcal{M}} |a_i - a_j| c_1 x_{ijk} - \sum_{i=1}^{J-1} \sum_{j=1}^{J-1} \sum_{k\in\mathcal{M}} (1 - |a_i - a_j|)(c_1 + c_2) l_{ijk}$$

$$- \sum_{k\in\mathcal{M}} c_1 u_k \tag{14}$$

subject to

$$\sum_{i=0:i\neq j}^{J-1} x_{ijk} = y_{jk} \quad \forall\, j \in \mathcal{J},\ k \in \mathcal{M} : j > 0 \tag{15}$$

$$\sum_{j=1:j\neq i}^{J} x_{ijk} = y_{ik} \quad \forall\, i \in \mathcal{J},\ k \in \mathcal{M} : i < J \tag{16}$$

$$b_{ijk} \le y_{jk} \quad \forall\, i,j \in \mathcal{J},\ k \in \mathcal{M} \tag{17}$$

$$b_{ijk} \le y_{ik} \quad \forall\, i,j \in \mathcal{J},\ k \in \mathcal{M} \tag{18}$$

$$\sum_{k\in\mathcal{M}} f_{jk} \le Q_j \quad \forall\, j \in \mathcal{J} : 0 < j < J \tag{19}$$

$$f_{jk} \le B_1 y_{jk} \quad \forall\, j \in \mathcal{J},\ k \in \mathcal{M} : 0 < j < J \tag{20}$$

$$f_{jk} \ge y_{jk} \quad \forall\, j \in \mathcal{J},\ k \in \mathcal{M} \tag{21}$$

$$y_{jk} \le u_k \quad \forall\, k \in \mathcal{M} \tag{22}$$

$$b_{ink} \ge b_{ijk} + b_{jnk} - 1 \quad \forall\, i,j,n \in \mathcal{J},\ k \in \mathcal{M} : a_i = a_j = a_n,\ 0 < i \neq j \neq n < J \tag{23}$$

$$h_{ijk} \le f_{jk} \quad \forall\, i,j \in \mathcal{J}, k \in \mathcal{M} : 0 < i \neq j < J \tag{24}$$

$$h_{ijk} \le B_1 b_{ijk} \quad \forall\, i,j \in \mathcal{J},\ k \in \mathcal{M} : i \neq j \tag{25}$$

$$h_{ijk} \ge f_{jk} - B_1(1 - b_{ijk})$$

Table 2. Additional notation for the VRP formulation.

Sets	
$\mathcal{J}$	set of cohorts
$\mathcal{M}$	set of vehicles
Parameters	
J	the number of cohorts, $J = \lvert\mathcal{J}\rvert$
Q_j	the number of passengers in cohort $j \in \mathcal{J}$. Also named as the size of cohort j
g_{ij}	binary parameter; 1 if cohort $i \in \mathcal{J}$ and cohort $j \in \mathcal{J}$ can be groped, 0 otherwise
B_1, B_2	big numbers
Binary variables	
u_k	1 if vehicle $k \in \mathcal{M}$ is used for transportation, 0 otherwise
x_{ijk}	1 if cohort $i \in \mathcal{J}$ is assigned just before cohort $j \in \mathcal{J}$ in vehicle $k \in \mathcal{M}$, 0 otherwise
y_{jk}	1 if cohort $j \in \mathcal{J}$ is assigned to the vehicle $k \in \mathcal{M}$, 0 otherwise
b_{ijk}	1 if passengers of cohort $i \in \mathcal{J}$ and cohort $j \in \mathcal{J}$ are in the same trip in vehicle $k \in \mathcal{M}$, 0 otherwise
l_{ijk}	variable for linearisation of $x_{ijk}(1 - b_{ijk})$. It takes the value of 1 if cohort $i \in \mathcal{J}$ is followed by cohort $j \in \mathcal{J}$ in vehicle $k \in \mathcal{M}$ but they are in different trips
Continuous variable	
t_{jk}	the transfer starting time of cohort $j \in \mathcal{J}$ in vehicle $k \in \mathcal{M}$
Integer variables	
f_{jk}	the number of passengers of cohort $j \in \mathcal{J}$ that are assigned to vehicle $k \in \mathcal{M}$
h_{ijk}	variable for linearisation of $f_{jk} b_{ijk}$. It represents the number of passengers of cohorts $j \in \mathcal{J}$ that are assigned to the same trip with cohort $i \in \mathcal{J}$ in the same vehicle $k \in \mathcal{M}$

$$\forall\, i,j \in \mathcal{J},\; k \in \mathcal{M} : 0 < i \neq j < J \quad (26)$$

$$\sum_{j=1:j\neq i}^{J-1} h_{ijk} + f_{ik} \leq w$$
$$\forall\, i \in \mathcal{J},\; k \in \mathcal{M} : 0 < i < J \quad (27)$$

$$x_{jik} \leq (1 - b_{ijk}) \quad \forall\, i,j \in \mathcal{J},\; k \in \mathcal{M} : r_i < r_j \quad (28)$$

$$t_{jk} + B_2(1 - x_{ijk}) \geq t_{ik} + (2 - |a_i - a_j|)p(1 - b_{ijk})$$
$$\forall\, i,j \in \mathcal{J},\; k \in \mathcal{M} : 0 < j < J \quad (29)$$

$$t_{jk} \geq r_j \quad \forall\, j \in \mathcal{J},\; k \in \mathcal{M} : 0 < j < J \quad (30)$$

$$t_{jk} \leq D_j \quad \forall\, j \in \mathcal{J},\; k \in \mathcal{M} : 0 < j < J \quad (31)$$

$$b_{ijk} \leq g_{ij} \quad \forall\, i,j \in \mathcal{J},\; k \in \mathcal{M} \quad (32)$$

$$l_{ijk} \leq x_{ijk} \quad \forall\, i,j \in \mathcal{J},\; k \in \mathcal{M} \quad (33)$$

$$l_{ijk} \leq (1 - b_{ijk}) \quad \forall\, i,j \in \mathcal{J},\; k \in \mathcal{M} \quad (34)$$

$$l_{ijk} \geq x_{ijk} - b_{ijk} \quad \forall\, i,j \in \mathcal{J},\; k \in \mathcal{M} \quad (35)$$

$$x_{jjk} = 0 \quad \forall\, j \in \mathcal{J},\; k \in \mathcal{M} : 0 < j < J \quad (36)$$

$$b_{ijk} = b_{jik} \quad \forall\, i,j \in \mathcal{J},\; k \in \mathcal{M} \quad (37)$$

$$x_{ijk}, b_{ijk}, l_{ijk}, y_{jk} \in \{0,1\} \quad \forall\, i,j \in \mathcal{J},\; k \in \mathcal{M} \quad (38)$$

$$t_{jk} \geq 0, h_{ijk}, f_{jk} \in \mathbb{Z}^+ \quad \forall j \in \mathcal{J},\; k \in \mathcal{M}. \quad (39)$$

The objective (14) maximises the profit calculated by subtracting the total cost from the total ticket revenue. The first term of the objective function calculates the total revenue attained from all transported passengers. The second term is the cost of a loaded trip which is immediately followed by a loaded trip in the opposite direction. The third term is the cost of a loaded trip which is followed by an empty trip. If cohort i is followed by cohort j and if they are not on the same trip ($l_{ijk} = 1$), and if these cohorts have the same direction ($1 - |a_i - a_j| = 1$), then the vehicle makes a loaded trip to transport cohort i which is followed by an empty trip to return back for cohort j. The total cost of this loaded and empty trips is $c_1 + c_2$ as included in the third term of the objective function. Since the last loaded trip of the day of each vehicle is followed by a dummy cohort and since this is not included in the second or third terms, the last term is included. This term calculates the cost of the last loaded trip of the day for each of the used vehicles. Constraints (15) assure that every cohort that is assigned to a vehicle has a predecessor, except dummy cohort 0, which is assigned as a first cohort to all vehicles. Similarly, dummy cohort J is assigned as a last cohort to all vehicles. Each cohort that is assigned to a vehicle, except cohort J, must have one successor as satisfied by Constraints (16). Constraints (17) and (18) satisfy the relation between the variables b_{ijk} and y_{jk}. According to this, if a cohort is not assigned to a vehicle, then it cannot be in the same trip with another cohort on that vehicle. The total number of passengers assigned to all vehicles from cohort j can be at most the total number of passengers in cohort j as satisfied by Constraints (19). If a cohort is not assigned to vehicle k, then no passengers of cohort j can be assigned to this vehicle as stated in Constraints (20). In these constraints, B_1 is a large number which should be larger than the total number of passengers that can be assigned from one cohort to one vehicle. Therefore $B_1 \geq \min\{w, Q_j\}$.

Unless at least one passenger is assigned to a vehicle, this cohort cannot be assumed to be assigned to this vehicle. This condition is satisfied by Constraints (21). Constraints (22) ensure that, if some passengers are assigned to a vehicle, then this vehicle is used for transportation. u_k variables determined by these constraints are used in the objective function to calculate the total cost.

Constraints (23) determine all cohorts that are assigned to the same trip of the same vehicle. According to this, if cohorts i and j and cohorts j and n are assigned to the same trip of the same vehicle then this indicates that cohorts i and n are also in the same trip. This constraint works with Constraints (24)–(27) to calculate the total number of passengers that are assigned to the same trip so that the vehicle capacity is not exceeded. Constraints (24)–(26) linearise $h_{ijk} = f_{jk}b_{ijk}$. If at least one of them is zero, then as a consequence of Constraints (24) and (25), $h_{ijk} = 0$. On the other hand, if both of them are greater than zero, then as a consequence of Constraints (24) and (26), $h_{ijk} = f_{jk}$. Constraints (27) use h_{ijk} variables to ensure that the total number of passengers that are assigned to the same trip of the same vehicle from all cohorts cannot exceed the vehicle capacity.

Constraints (28) and (29) determine the sequence of cohorts and their starting times. If cohorts i and j are assigned to the same trip, Constraints (28) ensure that the cohort with an earlier ready time is not scheduled later than the other cohort. These constraints basically prevent the sub-tours of the cohort sequence that are assigned to the same trip. On the other hand, Constraints (29) handle the cohorts that are assigned to different trips. In these constraints, B_2 represents a big number which must be larger than the available total time in minutes in a single day ($B_2 \geq 1440$). If cohorts i and j are not assigned to the same trip ($1 - b_{ijk} = 1$) and if these cohorts are following each other ($1 - x_{ijk} = 0$), then the starting time of cohort j must be greater than the starting time of cohort i plus the necessary travel time from cohort i to cohort j. If these two cohorts are on the same direction ($|a_i - a_j| = 0$), then one loaded and one empty trip is needed in between ($2p$). However, if they are on opposite directions only one loaded trip is necessary (p).

Constraints (30) and (31) are time window constraints. A cohort cannot start earlier than its ready time and its transfer should start before its due date. Constraints (32) state that if cohorts cannot be grouped (since they are in opposite directions or since their time windows do not overlap), then they cannot be assigned to the same vehicle. Constraints (33)–(35) are used for linearisation of $x_{ijk}(1 - b_{ijk})$, which are necessary to eliminate the empty transfer cost from the objective function when the cohorts are in the same trip. Constraints (36) ensure that no cohort can be a predecessor or a successor of itself. Constraints (37) guarantee that the b_{ijk} variables are symmetric. Finally, Constraints (38) and (39) are the integrality and non-negativity constraints.

In the following, we identify two valid inequalities for MIP2 to increase its computational efficiency and bounds. We add these inequalities to MIP2 to obtain MIP3. In Section 4, we perform computational experiments with these two formulations to test the effect of these inequalities on the performance.

$$x_{ijk} \leq 0 \quad \forall i,j \in \mathcal{J}, k \in \mathcal{M} : (a_i = a_j \,\&\, r_i > r_j) \text{ or } (a_i = 1 \,\&\, a_j = 0 \,\&\, r_i > r_j + e_2 - p) \text{ or } (a_i = 0 \,\&\, a_j = 1 \,\&\, r_i > r_j + e_1 - p) \quad (40)$$

$$b_{ink} \leq b_{ijk} \quad \forall i,j,n \in \mathcal{J}, k \in \mathcal{M} : (g_{ij} = g_{in} = 1) \,\&\, (r_i < r_j < r_n) \quad (41)$$

The first of these inequalities is rather trivial. If cohorts i and j are in the same direction and $r_i > r_j$, then there is an optimal solution in which cohort i is not scheduled earlier than cohort j. On the other hand, if they are on the opposite directions, if after starting the transfer of cohort i, it cannot reach cohort j within its allowable waiting time, then it cannot be scheduled earlier than cohort j.

In the following proposition, we prove that inequality (41) is valid in the sense that it does not change the optimal objective function value.

Proposition 2.1: *Inequality (41) does not change the optimal objective function value of MIP2.*

Proof: For a given instance of the ASBSP consider three arbitrary cohorts on the same direction ($a_i = a_j = a_n$), where $r_i < r_j < r_n$. In the following, we will prove the proposition for these arbitrary selected three cohorts which will imply that Inequality (41) is valid for all combinations of three cohorts that satisfy the grouping condition.

If in the optimal solution of MIP2 $b_{ink} = 0$ or $b_{ink} = b_{ijk} = 1$, then Inequality 41 is also satisfied and the same solution is still optimal. Therefore, let us consider the remaining case and assume that there is an optimal solution to MIP2 in which $b_{ink} = 1$ but $b_{ijk} = 0$. This implies that, a number of passengers from cohort n is grouped with cohort i and they are transferred altogether.

Table 3. Allowable start times of the cohorts for the proof of Proposition 2.1.

	Cohort i	Cohort j	Cohort n
Original	$[r_i, D_i]$	$[r_j, D_j]$	$[r_n, D_n]$
Optimal MIP2	$[\boldsymbol{r_n}, D_i]$	$[r_j, D_j]$	$[r_n, D_n]$
Modified	$[\boldsymbol{r_j}, D_i]$	$[\boldsymbol{r_n}, D_j]$	$[r_n, D_n]$

Note that, in the optimal solution some of these cohorts may also be rejected (not transferred). However, in this proof we assume that all three cohorts are transferred which is the more complicated case and which does not cause a loss of generality. Let the transfer start times of these cohorts in the optimal solution be $t^*_{ik_1}$, $t^*_{jk_2}$ and $t^*_{nk_3}$ on vehicles k_1, k_2 and k_3, respectively. As a consequence of the grouping of passengers, the transfer of cohort i cannot start earlier than the ready time of cohort n, but it must still be earlier than the due date of cohort i. That is, $r_n \leq t^*_{ik_1} \leq D_i$. On the other hand, the ready times and due dates of cohorts j and n remain the same. These are shown in the first two rows of Table 3, which presents the transfer start time windows for the three cohorts as given in the original problem and in the optimal solution.

Now let us modify the optimal solution to yield a new solution. In this modification, we keep everything the same but instead of shifting passengers from cohort n to i, we shift the same number of passengers from cohort j to i and, in sequence, from cohort n to j. This change has the same effect with the previous solution in terms of the number of passengers in each grouped cohort. However, their transfer start time windows change as presented in the last row of Table 3. Let t'_{ik_1}, t'_{jk_2}, and t'_{nk_3} denote the transfer start times of this modified solution. As it can be seen in the second and third rows of this table, there is no change in the time window of cohort n. That is, vehicle k_3 can still transfer this cohort with the same start time $r_n \leq t'_{ik_3} = t^*_{ik_3} \leq D_n$.

If in the optimal solution $r_n \leq t^*_{jk_2} \leq D_j$, vehicle k_2 can still transfer the modified cohort j with the same start time, $t'_{jk_2} = t^*_{jk_2}$. Also, since $r_j < r_n \leq t^*_{ik_1} \leq D_i$, vehicle k_1 can still transfer the modified cohort i with the same start time $t'_{ik_1} = t^*_{ik_1}$. As a result, the objective function value of the modified solution is not worse than the initial optimal solution.

On the other hand, if $r_j \leq t^*_{jk_2} < r_n$, then vehicle k_2 can no longer have the same transfer start time for the modified cohort j. This may cause a change or an infeasibility in the subsequent transfers of this vehicle. In this case, in order not to cause a deterioration in the objective value, we modify the solution by exchanging the vehicles of cohorts i and j. Since the vehicles are identical, such an exchange does not affect the feasibility and the cost. Hence, vehicle k_1 transfers cohort j with $t'_{jk_1} = t^*_{jk_2}$, and in the rest of the schedule vehicle k_1 follows the optimal route of k_2. On the other hand, since $b_{ink} = 1$ in the optimal solution, it means that cohorts i and n can be grouped, that is, $r_n \leq D_i$. This implies that if $r_j \leq t^*_{ik_1} < r_n$, then $t'_{ik_2} = t^*_{ik_1}$. Else if $r_n \leq t^*_{ik_1} \leq D_i$, then $t'_{ik_2} < t^*_{ik_1}$. That is, vehicle k_2 does start the transfer of modified cohort i earlier than its optimal value. Being early does not cause any infeasibility in the rest of the schedule when vehicle k_2 follows the optimal route of k_1. This proves the proposition. ■

Note that, this proposition guarantees the equality of the optimal objective function values, not the solutions. In fact, Inequality (41) may eliminate some of the alternative optimal solutions, which may speed up the search process.

Although these cuts improve the efficiency of the developed formulation by improving the upper and lower bounds as shown in Section 4, it is still not satisfactory to attain optimal solutions for even moderate size problem instances in plausible times. Therefore, we develop a heuristic algorithm for this complex problem in the following section.

3. Hybrid metaheuristic

As it is discussed in the previous sections, the ASBSP contains several unique features. Thus none of the earlier algorithms developed for other problems can be used directly to solve the ASBSP. The ASBSP includes several complex decisions such as scheduling the vehicles and assigning passengers to these vehicles. As it is shown in Section 4, finding the optimal solution with mathematical programming formulations requires long computational time. There are many metaheuristic algorithms that can be adapted to solve the ASBSP. However, since the problem is complex, we need an efficient algorithm that can search the feasible region quickly. For this purpose, we develop a hybrid metaheuristic algorithm that combines multi-start scheme, initialisation and grouping procedures, simulated annealing, and large neighbourhood search heuristic. These methods performed quite well on a variety of other optimisation problems (Koç, Jabali, and Laporte 2018). The algorithm starts with an initial solution. The neighbours of a current solution are searched as candidates for the next iteration. To escape from local optima, poor candidate solutions are also allowed according to a probabilistic criterion (see, e.g. Dowsland and Thompson 2012).

The steps of the proposed heuristic are presented in Algorithm 1. In this algorithm, $iterLimit_1$ is the number of times that the algorithm is restarted, and $iterLimit_2$ is the number of maximum iterations allowed. rnd_1, rnd_2 and rnd_3 are randomly generated numbers (integer, continuous and integer, respectively). $\hat{T}$ and T are the initial and current temperatures, respectively, in simulated annealing. β is the cooling speed. ω^*, ω_n and ω_c denote the best, new and current solutions, respectively. $f(\omega^*)$, $f(\omega_n)$ and $f(\omega_c)$ are the objective function values corresponding to these three solutions.

The initialisation procedure (Algorithm 2) generates an initial feasible solution (ω_c). Then, 'add', 'exchange #1', 'first remove then add' and 'create dummy routes' operators are applied to ω_c in sequence to generate feasible neighbours. If none of these operators generate a feasible neighbour, then a random integer, $rnd_3 \in [1, 3]$ is generated. Depending on this random integer one of the remove operators #1, #2, or exchange operator #2 is applied to ω_c. If ω_n is infeasible after the exchange operator #2, then the move is reversed and remove operator #1 is applied. The algorithm does not allow infeasible solutions. A new solution must satisfy the capacity and the time-windows constraints. After finding a feasible solution in a neighbourhood, none of the neighbourhoods is used. In other word, while more than one operator can be tried to generate a feasible solution, only one operator is applied on solution ω_c. The objective function value of the feasible solution is calculated, and then compared with the best feasible solution.

If the new solution has a better objective function value than the current and/or the best solutions, then they are updated. Otherwise, a continuous random number, rnd_2, is generated to decide whether to select the worsening neighbour as the next current solution or not. The temperature is cooled down at the end of each iteration to reduce the probability of accepting non-improving candidates through the end of the search. When the number of iterations reaches to the maximum iteration limit, the hybrid algorithm terminates and returns the best found solution ω^*.

In what follows, we will successively describe the grouping procedure in Section 3.1; initialisation procedure in Section 3.2; and finally the operators to generate neighbours in Section 3.3.

3.1. Grouping procedure

Two important decisions need to be made jointly in the ASBSP: (i) splitting/grouping the cohorts and (ii) routing the vehicles. The former one is important to better utilise

Algorithm 1 General framework of the hybrid meta-heuristic

```
Input: Grouped cohorts and an initial feasible solution
for 1 to iterLimit_1 do
    Initialisation (Algorithm 2)
    for 1 to iterLimit_2 do
        ω_n ⟵ ω_c + Add operator
        if ω_n is infeasible then
            ω_n ⟵ ω_c + Exchange operator #1
        end if
        if ω_n is infeasible then
            ω_n ⟵ ω_c + First remove then add operator
        end if
        if ω_n is infeasible then
            ω_n ⟵ ω_c + Create dummy routes operator
        end if
        if ω_n is infeasible then
            generate rnd_3 ∈ [1, 3]
            if rnd_3 = 1 then
                ω_n ⟵ ω_c + Remove operator #1
            end if
            if rnd_3 = 2 then
                ω_n ⟵ ω_c + Remove operator #2
            end if
            if rnd_3 = 3 then
                ω_n ⟵ ω_c + Exchange operator #2
                if ω_n is infeasible then
                    ω_n ⟵ ω_c + Remove operator #1
                end if
            end if
        end if
        if f(ω_c) < f(ω_n) then
            ω_c ⟵ ω_n
            f(ω_c) ⟵ f(ω_n)
        else
            generate rnd_2 ∈ [0, 1]
            γ ⟵ e^(−((f(ω_c) − f(ω_n)) / T))
            if rnd_2 ≤ γ then
                ω_c ⟵ ω_n
                f(ω_c) ⟵ f(ω_n)
            end if
        end if
        if f(ω*) < f(ω_c) then
            ω* ⟵ ω_c
            f(ω*) ⟵ f(ω_c)
        end if
        T ⟵ β × T
        if T < α × T̂ then
            T ⟵ T̂
        end if
    end for
end for
Output: ω*
```

the vehicle capacities and reduce the number of empty seats. Since the number of passengers in some cohorts could exceed the vehicle capacity, w, we need to split the cohorts that exceed the vehicle capacity into sub-cohorts such that they satisfy the capacity. On the other hand, if the demand of the cohorts or sub-cohorts is less than w, they can be re-grouped with other cohorts or sub-cohorts to satisfy the capacity. Note that, only the cohorts which are in the same direction and whose time window $[r_j, D_j]$ overlap can be grouped. Additionally, more than two cohorts can also be grouped. If n cohorts having time windows $[r_j, D_j]$ are grouped to be transferred with the same vehicle, the new time window of the new grouped cohort is determined as $[\max_j\{r_j\}, \min_j\{D_j\}]$. This is a consequence of the allowable waiting times of the grouped passengers.

In order to effectively split and group the cohorts, we introduce the following two alternative grouping algorithms.

Grouping algorithm #1: The idea behind this algorithm is to make the size of each cohort equal to w as much as possible. For this purpose, the cohorts in each direction (A–C or C–A) are considered separately. Starting from the earliest cohort to the latest one in the considered direction, the algorithm does the following sequentially. If the cohort has a size of more than w, divide it into sub-cohorts that have a size equal to w. If the total number of passengers in this cohort is not a multiple of w, the size of the last sub-cohort becomes less than w. If this is the situation or if the size of the original cohort is less than w, then try to assign necessary number of passengers from the next cohort to this one to make its size equal to w. For instance, let i be the sub-cohort such that $Q_j < w$ and let j be the next cohort in sequence. Then, passengers can be assigned from j to i if $|r_i - r_j| \leq e_1$ when i and j are both in A–C direction and $|r_i - r_j| \leq e_2$ when they are both in C–A direction. Note that, after grouping these cohorts, we need to update the sizes, release times and the due dates of the cohorts as follows: $Q_i = Q_i + \min\{(w - Q_i), Q_j\}$, $r_i = \max\{r_i, r_j\}$, $D_i = \min\{D_i, D_j\}$ and $Q_j = Q_j - \min\{(w - Q_i), Q_j\}$. If even after assigning all passengers of j, the size of i is still less than w, then try to assign passengers from the next cohort in the sequence. If there are no other cohorts that satisfy the release time condition, then set cohort i as it is and move to the next cohort. When all the cohorts in one direction are completed, repeat the same procedure with the cohorts in the other direction.

Grouping algorithm #2: This second version also considers the cohorts in different directions separately. Starting from the earliest cohort to the latest one in the considered direction, the algorithm does the following sequentially: If the size of the cohort is larger than w, it is divided into sub-cohorts having a size of w. If the size of the original cohort is less than w or if after splitting there appears a sub-cohort that has a size less than w, then try to assign some passengers from the feasible alternatives to this one using some probability. Let the considered sub-cohort be numbered as cohort i. Then, for all cohorts j with a release time less than or equal to D_i (assume there are $\hat{n}$ such cohorts), calculate the probability of moving k passengers from cohort j to cohort i. Let this probability be denoted as p_{ijk}. This probability is higher when j is consecutive to i. It is also higher if k makes the number of passengers in i closer to w, and/or it makes the number of passengers in j closer to a multiple of w. To satisfy these, the following formulas are used:

$$p_{ijk} = \frac{A}{B+C} \tag{42}$$

Notations A, B and C are calculated as follows:

$$A = \frac{u^{(w-|h-k|)} + u^{(w-|k-\eta_j|)}}{v^j} \tag{43}$$

$$B = \frac{u^{(w-|h-0|)} + u^{(w-|0-\eta_j|)}}{v} \tag{44}$$

$$C = \sum_{j=1}^{\hat{n}} \sum_{k=1}^{h} \frac{u^{(w-|h-k|)} + u^{(w-|k-\eta_j|)}}{v^j} \tag{45}$$

In these formulas, u and v are weight coefficients used to prioritise the assignment of passengers from a closer cohort j and assigning k passengers that makes the total number of passengers closer to w in source and destination cohorts. After preliminary analyses, we used 1.5 for both of these coefficients. h denotes the missing number of passengers in cohort i to reach w and $w - |h-k|$ calculates the missing number of passengers after the assignment of k passengers is made. Similarly, η_j denotes the excess number of passengers in cohort j so that the total number of passengers in this cohort is not equal to a multiple of w. This is calculated as $\eta_j = Q_j \pmod{w}$. Then, $w - |k - \eta_j|$ calculates the excess number after k passengers are removed from this cohort. Note that, it is also possible to not to assign any passengers to i and keep it as it is. For this purpose, $k = 0$ is used in A, and B is added separately to C in order not to count this situation multiple times (i.e. assigning zero cohorts from cohort j or cohort $j+1$ or any other cohort are the same).

After calculating these probabilities, using a tournament selection rule, the number of passengers assigned from cohort j to cohort i is calculated, the number of passengers in i and j are updated and the algorithm proceeds to the next cohort in sequence until all cohorts are processed. The procedure is then repeated for the cohorts in the opposite direction.

Note that, after these grouping algorithms the cohorts may be split or grouped with others. Therefore, they have modified sizes. At the end of this phase, if a cohort has a new size $Q_j^{new} < (c_1/v)$, it has more transfer cost than its revenue. In other words, covering such cohorts do not improve the objective function. Therefore, it is immediately rejected and it is not considered in the routing step.

3.2. Initialisation procedure

After grouping cohorts, we generate an initial feasible solution for the ASBSP to be used in the hybrid algorithm. We introduce three initialisation algorithms to enhance searching space to generate diverse solutions at each step. Each algorithm guarantees that all generated solutions are feasible. We take the solution which yields the best objective function value among these algorithms as the initial solution. The steps of the initialisation procedure are presented in Algorithm 2.

Algorithm 2 Initialisation procedure

```
Input: Sets 𝒥 and ℳ, and related parameters
change Random generation seeds
generate rnd_1 ∈ [1, 2]
if rnd_1 = 1 then
    Grouping algorithm #1
     Initialisation algorithm #1
     Initialisation algorithm #2
     Initialisation algorithm #3
end if
if rnd_1 = 2 then
    Grouping algorithm #2
     Initialisation algorithm #1
     Initialisation algorithm #2
     Initialisation algorithm #3
end if
Output: ω_c
```

Initialisation algorithm #1: We first rank the cohorts in ascending order according to their release times. One day is divided into a total of 24 one-hour time intervals. Then, the total number of passengers within each time interval is determined. The busiest time interval is selected as the one which has the largest total number of passengers. The algorithm starts to assign the cohorts in the busiest time interval to the vehicles. The cohorts in this interval are assigned starting from the earliest release time and assigned to the first available vehicle. When all the cohorts in this busiest interval are processed, the corresponding transfers of the vehicles are fixed. Then, the cohorts prior to this busiest interval are processed. These cohorts are assigned to the vehicles starting from the latest release time to the earliest in such a way that the vehicles would be available for their corresponding fixed transfers. Each cohort is processed one by one and is assigned to the first available vehicle considering transfer timing of the vehicles. The cohorts after the busiest interval are processed similarly starting from the earliest one to the latest and again by assigning to the first available vehicle. If there is more than one available vehicle, selection is made randomly.

Initialisation algorithm #2: We first rank the cohorts in ascending order according to their release times. Starting from the first cohort, the cohorts are assigned to the first available vehicle. For this purpose, the vehicles are ranked in ascending order of their indices. Starting from the first vehicle, the possible departure time of the vehicle at the demand point is calculated. This is done by adding the necessary travel time from its last position after transferring the last cohort assigned to this vehicle to the location of the new cohort. If the due date of the considered cohort is greater than or equal to the possible departure time of the vehicle, then the cohort is assigned to this vehicle. If the cohort cannot be assigned to none of the vehicles, then this cohort is not covered. Note that, this algorithm tries to increase the utilisation of the vehicles as much as possible. However, this may end up with more empty travels.

Initialisation algorithm #3: To minimise the number of empty travels we use modified version of the above algorithm. In this version, we also start with ranking the cohorts in ascending order of their release times. Then starting from the first cohort, we determine the vehicle to be assigned by calculating the ready time of the vehicle at the demand location. Let i be the last cohort in vehicle k and j be the next cohort to be assigned. The ready time of the vehicle for cohort j is equal to $\min\{t_{ik} + p + p(1 - |a_i - a_j)\}$. Different than Initialisation algorithm #2, in this version the priority is given to a vehicle that is less used and that avoids empty travel.

3.3. Operators

Hybrid algorithm requires a neighbourhood definition to generate candidate solutions for the next iteration. The neighbourhood definition is one of the most important factors in hybrid algorithm that affects the solution quality. In the following, we propose seven neighbourhood operators that are applied in sequence in Algorithm 1. Note that, these operators do not make any changes on grouping cohorts.

Add operator: Identify the cohort with the maximum size from the uncovered (not transported) cohorts. If there is more than one such cohort, select one of them randomly. Consider each vehicle and try to add the selected cohort to one of the vehicles. Add the cohort to the first vehicle that provides a feasible solution. If the cohort can be added to one of the vehicles, its new objective function value is calculated. Else, if it cannot be added to none of the vehicles, the algorithm continues with the next uncovered cohort. If none of the cohorts can be added to none of the vehicles, then move to the next operator.

Exchange operator #1: Randomly select cohort i from the covered cohorts. For all remaining covered cohorts j, calculate the following value:

$$\begin{aligned}\delta_{ij} &= n_1|Q_i - Q_j| + n_2|r_i - r_j| + n_3|t_i - t_j| \\ &\quad + n_4(m_{i,\sigma(i-1)} + m_{i,\sigma(i+1)} + m_{j,\sigma(j-1)} + m_{j,\sigma(j+1)} \\ &\quad - m_{j,\sigma(i-1)} - m_{j,\sigma(i+1)} - m_{i,\sigma(j-1)} - m_{i,\sigma(j+1)}) \quad (46)\end{aligned}$$

where t_i denotes the departure time of i, $m_{i,\sigma(i-1)} = p|a_i - a_{\sigma(i-1)}|$ is the time distance between cohorts i and j, $\sigma_{(i-1)}$ and $\sigma_{(i+1)}$ denote the cohorts that immediately precede and succeed cohort i, respectively, and $n_1, n_2, n_3, n_4 \in [0, 1]$ are weighting coefficients. The first three terms of δ_{ij} measures the closeness of cohorts i and j to each other in terms of their size, release time and departure times. If they are closer, the probability of having a feasible solution after the exchange increases. The last term calculates the gain after the exchange in terms of the travelled distance. Figure 2 illustrates the exchange operator. The algorithm exchanges cohort i with cohort j that is feasible and that gives the minimum δ_{ij} value. If it is not possible to find a feasible solution, then move to the next operator.

Exchange operator #2: Similar to the previous one, this operator also exchanges two cohorts assigned to different vehicles. All possible cohort pairs are considered for this exchange operation. The feasibility after the exchange of the selected cohort pair is investigated. Also the contribution of this exchange in terms of the profit is calculated. The feasible pair of cohorts that provides the maximum gain on profit is exchanged.

First remove then add operator: Calculate the following function for each covered cohort:

$$\begin{aligned}\Omega_i = \vartheta_1(w - Q_i) + \vartheta_2(2 - |a_i - a_{\sigma(i-1)}| \\ - |a_i - a_{\sigma(i+1)}|) - \vartheta_3 c_i \quad (47)\end{aligned}$$

where c_i is the number of times that cohort i is used for removal in the previous iterations. ϑ_1, ϑ_2 and ϑ_3 are weighting coefficients where $\vartheta_1 + \vartheta_2 + \vartheta_3 = 1$. Ω_i calculates an index value for each cohort. This index value is higher: (i) if the size of i is lower, (ii) if there is empty

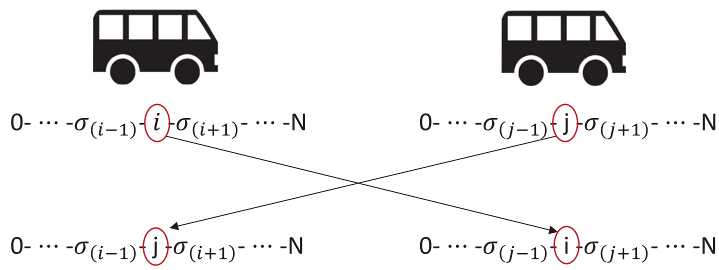

Figure 2. Exchange operator.

travel between this cohort and its predecessor and/or successor, and (iii) if it is not already used for remove operation many times previously. The cohort with the maximum Ω_i value (break ties arbitrarily) is removed from the corresponding vehicle. It is then inserted into the best, in terms of profit, feasible position by trying all feasible insertion options.

Create dummy routes: Generate $\hat{M} = \lceil \frac{\textit{number of uncovered cohorts}}{\Theta} \rceil$ dummy vehicles where Θ is a user-defined parameter that determines the number of dummy vehicles to be generated. The algorithm proceeds by constructing dummy feasible routes using these dummy vehicles and uncovered cohorts. The dummy route that has the maximum profit is then replaced with the vehicle that has the lowest profit.

Remove operator #1: For each covered cohort, use the following function to calculate the probability of removal where ε_1 and ε_2 are weighting parameters with $\varepsilon_1 + \varepsilon_2 = 1$.

$$\gamma_i = \exp\left(-\left(\varepsilon_1\left(1 - \frac{Q_i}{w}\right) + \varepsilon_2 \frac{(2 - |a_i - a_{\sigma(i-1)}| - |a_i - a_{\sigma(i+1)}|)}{2}\right)\right) \quad (48)$$

This function assigns a larger probability for the cohorts that has smaller size with respect to the vehicle capacity (first term in the exponent) and to the cohorts that causes an empty transfer in the route (second term). Then, using these probabilities with a roulette wheel selection procedure one of the cohorts is selected and removed.

Remove operator #2: Calculate Ω_i for each covered cohort i using Equation (47). Remove the cohort with the maximum Ω_i value from the corresponding vehicle. Break ties arbitrarily.

4. Computational experiments and analyses

In this section, we present the results of our computational experiments. The hybrid algorithm was implemented in Java. All experiments were conducted on a server with dual 2.00 GHz Intel Xeon E5-2665 processors and 128 GB RAM. We used CPLEX 12.8 with its default settings as the optimiser to solve the mathematical formulation. Two hours of time limit was imposed in CPLEX to solve the instances.

The remainder of this section is organised as follows. We first describe our benchmark instances in Section 4.1. We then present the parameter tuning for the proposed heuristic in Section 4.2. We present the results obtained on benchmark instances in Section 4.3. We finally present the results of our analysis on the effects of several parameters on the ASBSP in Section 4.4.

4.1. Benchmark instances

We have generated our benchmark instances by using the real life data of an airport shuttle company that operates in Ankara, Turkey, between the city centre and the Ankara Esenboğa Airport. In total, we generated 64 benchmark instances for the ASBSP, which can be accessed at http://dx.doi.org/10.17632/tt6zhyr3n9.

The company serves for the passengers of a specific domestic airlines. Past data including the flight arrival and departure times as well as the total number of passengers in each flight requesting shuttle service are available. The company has 71 vehicles and the number of cohorts in a day is in the range [74, 78]. The total number of passengers in 1 day is in the range [1600,1800]. To test the performance of the proposed heuristic and compare with the mathematical models, we developed benchmark instances of various sizes from this real life data. Note that, the MIP models were unable to solve actual real-life problem instances and did not provide useful gap values. Therefore, we have developed 48 medium-size instances in which we considered 8, 18 and 71 for the number of vehicles ($|\mathcal{M}|$) and 20 and 40 for the number of cohorts (J). We randomly selected four separate days with a total of 1706, 1708, 1718 and 1719 passengers. We used the days with a total of 1708 and 1718 passengers to generate the data for $J = 20$ cohorts and the remaining ones for $J = 40$ cohorts. We randomly eliminated cohorts from the real data until the number of cohorts is reduced to the desired value. We then considered two alternatives: in the first one, we discarded the number of passengers of the eliminated cohorts. In the second one, we kept the original passenger number and distributed the passengers of the eliminated cohorts to the remaining cohorts in proportion to their original passenger numbers. Finally, we considered 7 and 15 for the vehicle capacity (w). Note that, we entitled each instance as $J\_|\mathcal{M}|\_w\_\sum_{j\in\mathcal{J}} Q_j$. As an example 20_8_15_531 denotes an instance with 20 cohorts, 8 vehicles with vehicle capacity 15 and a total of 531 passengers. However, since even most of these medium-size instances could not be solved in the given time limit, we also generated 16 smaller instances which include 10 and 15 cohorts (J), 4 and 8 vehicles ($|\mathcal{M}|$) with a vehicle capacity of 7 and 15. As a result, we obtained the small and medium-size instances listed in Tables A1 and A2, respectively, in the Appendix.

We also set the following parameter values from the real data: The latest time the passengers are supposed to be at the airport before the departure time (s) is set to 60 minutes. The transfer fare (v) is set to 25 monetary unit. The travel time between each direction p is set to 60 minutes. The maximum waiting time for passengers travelling from airport to city centre (e_1) is set to 15 minutes. The maximum waiting time for passengers travelling from city centre to airport (e_2) is set to 90 minutes. The transfer cost (c_1) is set to 70 monetary unit. The empty transfer cost (c_2) is set to 50 monetary unit. Their effects on the results are also analysed later in this section.

4.2. Parameter tuning

Our metaheuristic algorithm combines different methods and neighbourhood definitions which are configured by several correlated parameters. In this section, we conduct an extensive parameter tuning experiment to determine the best parameter combination for the hybrid algorithm. Each experiment corresponds to a parameter combination.

For this purpose, we randomly select the following eight instances: 20_8_15_531, 20_18_15_531, 20_71_7_531, 20_71_15_531, 40_8_7_853, 40_8_7_1706, 40_8_15_1706 and 40_18_7_1706. After our preliminary tests, the following parameters are selected for detailed analysis: for the cooling speed, β, we try two values, 0.95 and 0.99; for the initial temperature, $\hat{T}$, we try 10, 000 and 100, 000; for the number of iterations allowed in the hybrid algorithm ($iterLimit_2$), we try 500, 1000 and 2000; for the iteration limit to generate initial feasible solutions ($iterLimit_1$), we try 50, 75, 100, 150, 200, 250 and 300. Values for the other parameters that are used in the neighbourhood operators in the hybrid algorithm are given in Tables 4, 5, and 6, respectively. The rest of the hybrid metaheuristic parameter values Θ, α, u and v are set to 5, 0.2, 1.5 and 1.5, respectively after some preliminary tests.

Table 4. Alternatives for the parameters in the first remove then add operator.

Alternative	ϑ_1	ϑ_2	ϑ_3
1	0.3	0.3	0.4
2	0.3	0.4	0.3
3	0.4	0.3	0.3

Table 5. Alternatives for the parameters in the remove operator #1.

Alternative	ε_1	ε_2
1	0.3	0.7
2	0.4	0.6
3	0.5	0.5

Table 6. Alternatives for the parameters in the exchange operator #1.

Alternative	n_1	n_2	n_3	n_4
1	0.2	0.2	0.5	0.5
2	0.3	0.3	0.5	0.5
3	0.4	0.4	0.5	0.5

In total, 2268 parameter combinations are used on all 8 instances which makes 18,144 runs. The main objective of these runs is to determine the best parameter values that yield near optimal solutions for any problem instances without repeating the calibration process. In this sense, our parametric heuristic is a robust solution method. Since the heuristic is stopped with an iteration limit, the running time of the algorithm may vary for different problem instances. Therefore, to select the best parameter combination we need to consider the trade-off between the running time and the percent deviation from the optimal. These are depicted in Figure 3 for each combination. Since the time and gap values are two conflicting criteria, there is no single best parameter combination. We select the four combinations that are highlighted in this figure as the promising candidates in terms of solution time and quality. Table 7 presents the values of the parameters used in these four combinations and Table 8 presents the %GAP and solution time (Time) for the selected eight instances attained by these four combinations (C-1, C-2, C-3, C-4).

4.3. Results obtained on instances

This section presents the performance comparison of MIP2, MIP3 and the proposed hybrid metaheuristic. All four parameter combinations of the hybrid metaheuristic are compared with the formulations solved by CPLEX. All results are summarised in Table 9. The detailed results

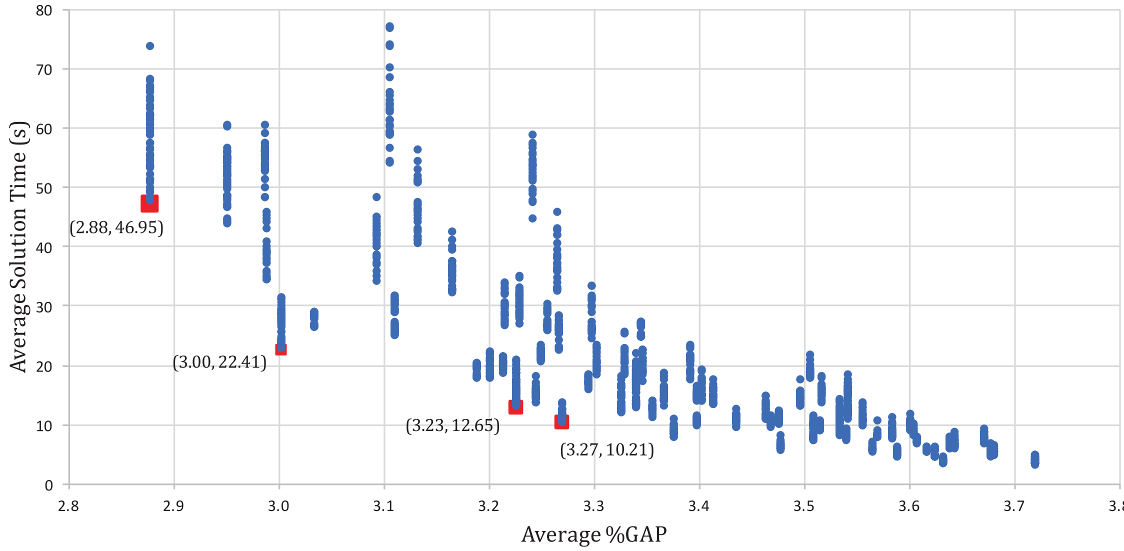

Figure 3. Average solution time vs. %GAP for the tested parameter combinations.

Table 7. Values of the parameters in the selected four combinations.

Combination	β	T	$iterLimit_1$	$iterLimit_2$	ϑ_1	ϑ_2	ϑ_3	ε_1	ε_2	n_1	n_2	n_3	n_4
C-1	0.95	10,000	250	2000	0.4	0.3	0.3	0.3	0.7	0.2	0.2	0.5	0.5
C-2	0.95	10,000	250	1000	0.3	0.4	0.3	0.5	0.5	0.5	0.5	0.2	0.2
C-3	0.95	10,000	250	500	0.3	0.4	0.3	0.3	0.7	0.5	0.5	0.2	0.2
C-4	0.95	10,000	200	500	0.3	0.4	0.3	0.4	0.6	0.3	0.3	0.4	0.4

Table 8. Detailed results for the selected four parameter combinations.

	C-1		C-2		C-3		C-4	
Instance	%GAP	Time (s)	%GAP	Time (s)	%GAP	Time (s)	%GAP	Time (s)
20_8_15_531	4.7	9.6	4.7	6.1	4.7	3.4	4.7	3.5
20_18_15_531	2.7	49.7	2.7	21.7	2.7	13.8	2.7	11.6
20_71_7_531	1.8	111.6	1.8	51.4	1.8	31.1	2.0	24.8
20_71_15_531	1.7	4.3	1.7	2.7	1.7	2.6	1.7	2.0
40_8_7_853	3.0	37.5	3.5	15.8	3.5	8.4	3.7	7.3
40_8_7_1706	3.2	51.6	3.7	23.1	5.2	12.7	5.2	10.6
40_8_15_1706	3.6	34.8	3.6	15.1	3.6	8.3	3.6	7.0
40_18_7_1706	2.3	76.5	2.3	43.3	2.6	21.0	2.6	14.9
Average	2.9	46.9	3.0	22.4	3.2	12.7	3.3	10.2

Note: * Compared with the best objective function value provided by MIP2 and MIP3.

are presented in Tables A1 and A2 for small and medium-size benchmark instances, respectively, in the Appendix. In all these tables, the '%GAP' columns display the relative percent deviation of the results from the lower bound. For MIP2 and MIP3, these values are the reported values of CPLEX. If the solver stops due to the time limit (represented as *TL* in the table), they take nonzero values. For the heuristics, these columns are the percent deviations of the heuristic from the best objective value found by MIP2 or MIP3. The 'Profit' column in Tables A1 and A2 displays the objective value.

Both MIP2 and MIP3 found the optimal solution in 14 out of the 16 small-size instances. In all instances, MIP2 and MIP3 provided the same upper bound. The average gap of MIP3 seems to be higher than MIP2. However, this is because of one instance (15_8_15_615). In all instances for which the optimal solution could be found the solution time of MIP3 is less than MIP2 and the solution time is improved by 61.5% on the average. For these small-size instances, the heuristic combinations C-1, C-2, and C-3 missed the optimal solution in only four of the instances, whereas C-4 missed in five of the instances. The maximum percentage error over all combinations and all instances is 1.1%. The minimum and maximum of the average solution time of all instances are provided by C-4 and C-1 as 3.2 s and 10.3 s, respectively. These results prove the efficiency of the proposed heuristics on small-size instances.

As it is shown in Table 9, the gap values of MIP2 and MIP3 have increased significantly. Table A2 shows that MIP2 and MIP3 could find the optimal solution only in 5 of the 48 medium-size instances within the

Table 9. Average comparative results on the ASBSP benchmark instances.

	MIP2	MIP3	C-1		C-2		C-3		C-4	
$\|\mathcal{J}\|$	%GAP	%GAP	%GAP	Time (s)	%GAP	Time (s)	%GAP	Time (s)	%GAP	Time (s)
10	0.0	0.0	0.1	7.7	0.1	4.7	0.1	3.0	0.2	2.5
15	0.3	0.5	0.1	13.0	0.1	7.2	0.2	4.4	0.2	3.8
20	29.3	15.9	0.8	280.0	0.8	108.0	0.9	53.7	0.9	65.0
40	36.7	29.2	2.0	595.5	2.0	289.6	2.2	132.8	2.2	269.9

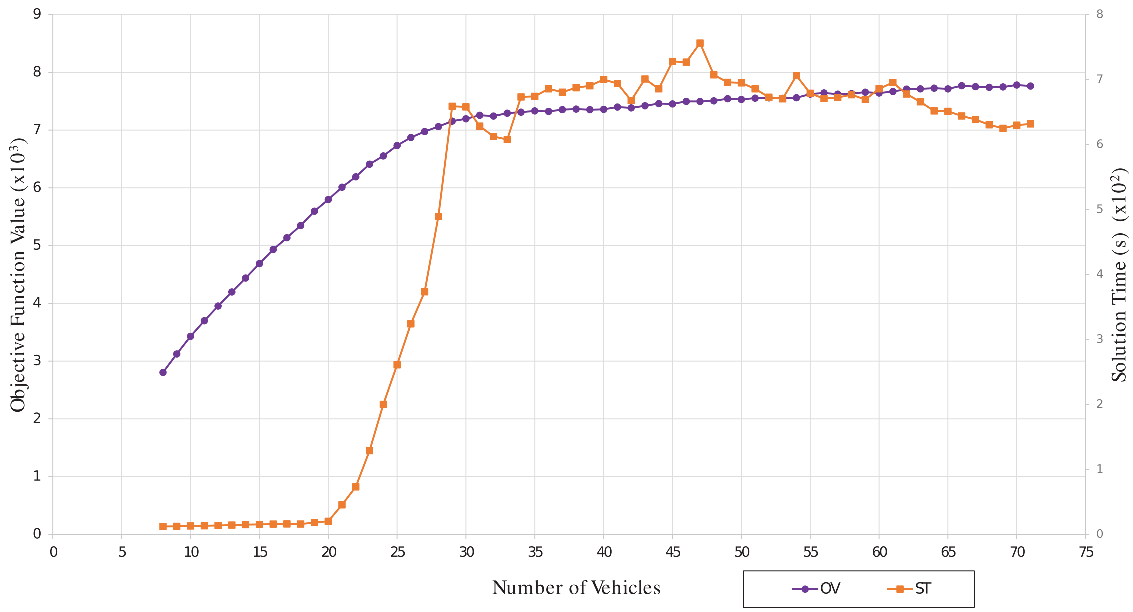

Figure 4. The average objective function value and solution time corresponding to the number of vehicles.

given time limit. In 17 of the instances, MIP3 provided a better objective function value than MIP2 and in 31 of the instances it provided a better gap value. On the other hand, MIP2 provided better objective function value in 7 of the instances and provided a better gap in 8 instances.

On average, the gap values for C-1, C-2, C-3 and C-4 appeared to be 1.4, 1.4, 1.5 and 1.5, respectively. In particular, C-1 obtained a better average %GAP value than the other combinations, but required more solution time since its $iterLimit_2$ value is higher. C-3 and C-4 have similar average %GAP value, but the former requires almost half the solution time than the latter. The average and maximum solution times of the combinations differ from each other but all of them are reasonable for the available planning time in the considered problem. Furthermore, this solution time remains stationary when the number of vehicles is more than 40 as discussed later in this section and depicted in Figure 4.

4.4. Analysis of the effects of several parameters on the ASBSP

In this section, our aim is to provide insights about the problem and to quantify the effects of the parameters on the results by making what-if analysis. We analyse the impacts of different problem parameters on the results. These impacts can be on several performance measures related to the operation of the shuttle service system. These measures include the objective function value (*Profit*); solution time (*Time*); percentage of transferred (covered) passengers (*%PT*); percentage of empty transfer time (total empty transfer time/total service time) (*%ET*); percentage utilisation of the vehicles (*%U*); total transfer time per vehicle (*TTV*); average idle time between the first and last transfers of the vehicles (*WTV*); average normalised waiting time (waiting time/maximum allowable waiting time) per passenger in the A–C direction (*%WTAC*), and in the CA direction (*%WTCA*). Note that, the only objective in the ASBSP is the minimisation of the profit. All others that are listed here are some side performance measures and used to provide managerial insights.

To this end, we conduct extensive analysis on real case scenarios which consist of all 7 days of a randomly selected week. These instances can also be accessed at http://dx.doi.org/10.17632/tt6zhyr3n9. All runs are taken with the heuristic using parameter combination C-3. Table 10 presents the number of cohorts and the total number of passengers for the real case scenarios. The

Table 10. Real case instances.

Instance	J	$\sum_{j\in\mathcal{J}} Q_j$
Monday	77	1794
Tuesday	78	1710
Wednesday	78	1708
Thursday	75	1664
Friday	74	1707
Saturday	77	1653
Sunday	75	1672

Table 11. Scenarios for sensitivity analysis in real case instances.

Parameter	Values
Number of Vehicles	20, 30, 40,71
(Seat capacity, loaded cost, empty cost)	(7, 70, 50), (15, 100, 80)
Transfer fare (monetary unit)	15, 25
Transfer time (min)	45, 60
Max waiting time, A–C (min)	15, 30
Max waiting time, C–A (min)	60, 90

results given in Tables 12–17 are the average values over 7 days.

We analyse the effects of the number of vehicles ($|\mathcal{M}|$), vehicle capacity (w), transfer fare (v), travel time between each direction (p), the allowable waiting time for the passengers in the A–C direction (e_1) and in the C–A direction (e_2). Note that, the transfer costs are associated with the vehicle capacity and therefore, we adjusted the transfer costs depending on the vehicle capacity. The cost for a larger capacity vehicle is assumed to be higher.

To decide which values to be used for $|\mathcal{M}|$, we conducted a preliminary test in which we demonstrated the effect of $|\mathcal{M}|$ on the profit and solution time of the hybrid heuristic. We increase $|\mathcal{M}|$ from 8 to 71 one by one and solved all seven problem instances. Figure 4 depicts the average objective function value and solution time obtained for different $|\mathcal{M}|$ values. As expected, when the number of vehicles increases, the profit increases, as well. Because more passengers are scheduled. However, as more and more passengers are covered with the increased $|\mathcal{M}|$, the marginal contribution of each additional vehicle reduces after some critical point. As a consequence, the objective function in Figure 4 appeared to be an approximately two-piece function. The critical point is approximately when $|\mathcal{M}| = 30$. At this point, most of the passengers are covered. For each additional vehicle, the planning of the vehicles becomes more important and critical. Hence the problem is more complicated which is reflected in the solution time. For small values of $|\mathcal{M}|$, the solution time is very small. However, it increases rapidly between 20 and 30. Beyond 30, the solution time fluctuates around the same level. This is an important observation that shows that the solution time of the heuristic does not increase indefinitely with the problem size and there is a limit on the maximum solution time.

Currently, the company operates with 71 vehicles. However, due to these observations, we decided to use 20, 30, 40 and 71 vehicles to conduct our analysis. The values of all other parameters that are used in our tests are summarised in Table 11. In total, we used 128 combinations of parameters on 7 real-case instances which makes 896 runs.

4.4.1. The effect of changing the number of vehicles ($|\mathcal{M}|$)

We now investigate the effects of changing the number of vehicles ($|\mathcal{M}|$). Table 12 summarises these effects on the mentioned performance measures. As it is mentioned earlier, the profit and the solution time increase rapidly when the number of vehicles is increased from 20 to 30. However, the marginal effect of additional vehicles reduces for $|\mathcal{M}| \geq 30$. The reason behind this can be seen in the column of percentage of transferred passengers, *%PT*. When $|\mathcal{M}| = 30$, 97.5% of all passengers are already covered. When the number of vehicles is increased to 71, this percentage can only increase to 98.1%. Correspondingly, the profit also increases slightly with this change.

On the other hand, *%ET, TTV* and *WTV* decrease when the number of vehicles is increased. Empty travels decrease, since with the existence of more vehicles, there is a higher probability of finding a vehicle in the required direction instead of making an empty transfer of a car which is in the opposite direction. When there are 20 vehicles, the total empty and loaded transfer time for all the vehicles are 7960 hours, which makes 398.0 hours per vehicle. When the number of vehicles is increased to 71, the total empty and loaded transfer time for all the vehicles are 6333.2 hours, which makes 89.2 hours per vehicle. As it can be seen, total transfer time is reduced. This is due to the reduction in empty travels from 10.5% to 1.9%. The reduction in TTV is due to this reduction in total time as well as the increase in the number of vehicles. WTV, which measures the idle time between the first and last transfer of the vehicles, also decreases when the number of vehicles is increased. Together with the reduction in TTV, this reduction shows that the heuristic tries to utilise the same vehicles as long as possible. A new vehicle is used only whenever it is necessary and after its use, the vehicle is sent to rest again.

It is observed that the utilisation of the vehicles (*%U*) does not change too much with the number of vehicles. It means that, whenever a vehicle is used for transfer, it is mostly fully loaded. This is a direct consequence of the objective function and the grouping phase of the heuristic. Note that, %WTAC values decrease with the number of vehicles but %WTCA moves in the opposite direction. Additionally, %WTAC values are much smaller than

Table 12. Effect of the number of vehicles on performance measures.

$\lvert\mathcal{M}\rvert$	Profit	Time (s)	%PT	%ET	%U	TTV	WTV	%WTAC	%WTCA
20	17,227.2	132.9	94.1	10.5	94.9	398.0	531.0	7.1	76.1
30	18,258.6	461.3	97.5	5.6	94.6	272.2	523.1	5.5	82.2
40	18,424.8	542.4	97.8	4.0	94.4	190.8	507.0	4.5	84.2
71	18,707.8	530.3	98.1	1.9	94.4	89.2	355.7	3.4	88.0

Table 13. Effect of the vehicle capacity on performance measures.

w	Profit	Time (s)	%PT	%ET	%U	TTV	WTV	%WTAC	%WTCA
7	15,074.1	628.3	94.9	7.9	97.3	323.7	532.3	6.1	80.1
15	21,235.1	205.1	98.8	3.0	91.9	151.5	426.1	4.1	85.2

Table 14. Effect of the transportation price on performance measures.

v	RD (%)	Profit	Time (s)	%PT	%ET	%U	TTV	WTV	%WTAC	%WTCA
15	–	9,953.5	379.7	95.4	4.4	95.6	228.1	476.9	4.7	83.5
25	0	26,355.8	453.7	98.3	6.5	93.5	247.0	481.5	5.6	81.8
25	5	25,119.9	415.0	98.6	6.1	93.3	231.5	477.3	4.7	84.3
25	10	23,931.7	384.9	98.9	5.4	92.7	222.2	473.0	4.7	84.6
25	15	22,598.4	355.6	99.1	5.1	92.1	210.1	460.3	4.6	85.6
25	20	20,965.2	339.7	99.3	4.6	90.0	199.5	466.5	4.1	85.3

Table 15. Effect of the transportation time on performance measures.

p	Profit	Time (s)	%PT	%ET	%U	TTV	WTV	%WTAC	%WTCA
45	18,141.6	437.8	97.2	6.1	94.5	205.3	506.0	3.5	85.3
60	18,167.7	395.6	96.5	4.8	94.6	269.8	452.4	6.8	80.0

Table 16. Effect of the allowable waiting time in A–C direction on performance measures.

e_1	Profit	Time (s)	%PT	%ET	%U	TTV	WTV	%WTAC	%WTCA
15	18,153.2	405.9	96.8	5.5	94.7	237.4	482.3	4.8	82.9
30	18,156.1	427.5	96.9	5.5	94.5	237.7	476.1	5.5	82.4

Table 17. Effect of the allowable waiting time in C–A direction on performance measures.

e_2	Profit	Time (s)	%PT	%ET	%U	TTV	WTV	%WTAC	%WTCA
60	18,122.6	429.1	96.6	5.3	94.6	239.3	471.4	5.3	81.3
90	18,186.7	404.4	97.1	5.6	94.6	235.9	487.0	5.0	83.9

%WTCA values. There are several reasons for these outcome. First of all, the maximum allowable waiting times in these two directions are very different from each other (15 min vs. 90 min). Second, in the heuristic, the passengers are tried to be grouped with other passengers that have closer ready times. Finally, our heuristic algorithm tries to utilise a vehicle as much as possible. Therefore, whenever the vehicle and the passengers are ready, the transfer starts immediately. In this way, the heuristic tries to reduce the idle time of the vehicles and make more transfers with each used vehicle. This is in favour of reducing the waiting time in the A–C direction. However, it inversely affects the waiting time in the C–A direction as can be seen in the results. Note that, the only objective in the model was to maximise the profit, and increasing the utilisation of a vehicle improves this objective.

4.4.2. The effect of changing the vehicle capacity (w)

In practice, the vehicle capacity has a huge impact on shuttles routing decisions. Since it increases the number of passengers scheduled, it could change all solutions. Currently, the shuttle transfers in the considered system are being performed by vehicles with 7 seats. To analyse the impact of increasing the vehicle capacity, we consider 15-seat vehicles. However, the increase of the seat capacity requires a larger vehicle with higher loaded and empty transfer costs. Therefore, we update the original costs of 70 and 50 for loaded and empty transfers, respectively, as 100 and 80 for $w = 15$.

Increasing the vehicle capacity improves the profit by 40.9%. Table 13 summarises the results for the effect of changing the vehicle capacity. As can be seen in this table, vehicle capacity has a significant effect on %ET, TTV and

WTV. The utilisation of the vehicles slightly reduced by this change.

4.4.3. The effect of changing the transfer fare per passenger (v)

The transfer price per passenger is another important parameter of the problem. We expect that the total profit increases when the price increases. We use two different prices (15 and 25) per passenger to quantify the effect of such a change. However, increasing the transfer fare may inversely affect the demand, which in turn may reduce the total profit. The reduction in demand depends on the price elasticity of the customers. Thus, to quantify the integrated impact of these conflicting changes, we consider various rates of change in demand. More specifically, let *RD* denote the percentage reduction in demand. We assume that the demand is not affected in the baseline scenario ($RD = 0$). Whereas in other scenarios we assume that the total demand decreases at the rates indicated in the second column of Table 14. This table summarises the results for the effect of changing the transportation price per passenger.

As expected, profit is very sensitive to the price. An increase in the price from 15 to 25 increases the profit by 164.8% in the base scenario. Beside the multiplier effect of this price on the profit, another reason of this increase is the increase in the covered passengers, %PT as can be seen in Table 14. When $v = 15$, it is not profitable to transfer less than five passengers when $w = 7$ and less than seven passengers when $w = 15$. However, when $v = 25$, these numbers reduce to three and four passengers, respectively. As a consequence, percentage of covered passengers increased when the price is increased and this boosted the increase in the profit even further. On the other hand, transferring less number of passengers in one trip reduced the utilisation of the vehicles.

As it can be seen, the reduction rate of demand has an almost linear effect on the profit. According to this, even with a demand reduction rate of 60%, it is still more profitable to increase the transfer fare to 25. This is due to the current excessive demand and high profit per passenger. Also, as expected, the percentage of covered passengers increases while the utilisation of the vehicles decreases with the effect of the decrease in demand.

4.4.4. The effect of transportation time (p)

The current travel time in the real data is 60 minutes. In this section, we analyse the effect of transfer time on the results. Table 15 summarises the effect of reducing the transportation time to 45 minutes. As it can be seen, there is no significant effect on the performance measures when the transportation time is reduced. As expected, the total travel time decrease, waiting time between the transfers and the percentage of empty travels increase. There is no significant change in the profit.

4.4.5. The effects of changing the allowable waiting times in A–C (e_1) and C–A (e_2) directions

Passengers can request a transfer in the A–C or C–A direction. In the A–C direction, after landing to the airport the passengers become ready for transfer. However, in order to better utilise the vehicle capacity, the transfer may be delayed to wait for passengers from later flights. Since, this waiting time is a source of dissatisfaction on the users, the maximum allowable waiting time is limited in the real practice. Currently, a maximum waiting time of 15 minutes is allowed. Increasing this value may help to group passengers from different flights and increase vehicle utilisation. To see its effect on the results we consider increasing this value to 30 minutes. A similar limit applies to the passengers on the C–A direction. These passengers want to reach to their flights. However, they do not want to go too early and wait longer in the airport. However, as it is the case in the real practice, these passengers have more flexibility than the passengers on the opposite direction. Therefore, a larger allowable waiting limit is used in this direction. Currently, this value is 90 minutes and we consider reducing it to 60 minutes.

Tables 16 and 17 summarise the effects of e_1 and e_2 parameters on the results, respectively, on the considered performance measures. In these tables, the maximum allowable waiting time of the passengers in the A–C (C–A) direction is denoted with *MTAC* (*MTCA*). As it can be seen, these parameters seem not to effect results. This is because, the utilisation of the vehicles are already very high even with small allowable waiting time limit. Since increasing the limits may lead to dissatisfaction, this result suggests not to increase the limits in the current practice.

5. Conclusion

We have introduced the airport shuttle bus scheduling problem (ASBSP) as a new practical scheduling variant. We have presented two alternative mixed integer programming formulations and proposed valid inequalities for the latter one to increase its efficiency. We have also developed a hybrid metaheuristic to solve the ASBSP. This hybrid metaheuristic effectively combines multi-start, grouping, simulated annealing and large neighbourhood search procedures. We have performed extensive analyses on problem parameters using real-life benchmark instances of the ASBSP. We have investigated the effects of several parameters, such as vehicle capacity, transportation cost per passenger, transportation time between directions and allowable waiting time at the airport or at the city centre.

For the ASBSP, 16 small-size and 48 medium-size instances were generated based on real data. We have compared the results of our method with CPLEX. The hybrid metaheuristic obtains comparable solutions in shorter time than CPLEX. The process times of the proposed algorithm make it more useful in practical applications.

Future research directions include developing other heuristic/metaheuristic solution procedures and comparing them with our heuristic algorithm using the provided online benchmark instances. The stochastic version of the problem can be considered where the transfer times are assumed to be random variables. The release times can also be considered as random variables due to the delays of the flights and delays associated with the procedures in the terminal buildings such as luggage collection, walking and security checks. Another interesting future research direction is to consider dynamic transfer requests where the customers make reservations for the transfers at least 1 day before but at different times. A response regarding the approval or rejection of the request must be given at the time of reservation. This may require the integration of the ASBSP with revenue management and online data analysis and planning. More pickup and drop-off points both at the airport and in the city centre can be considered instead of using a single pick up point in both directions. In this case, the problem includes the routing of the vehicles between these points and also the clustering of the customers that will be transferred by the same vehicle. Otherwise, if customers going for different regions of a large city are assigned to the same vehicle, the total transfer time of the customers may be very large. This is a practical and more complex problem. Last but not the least, the ASBSP can be modified so that the transfer cost depends on the vehicle speed. In these models, vehicle speeds can also be considered as decision variables.

Acknowledgments

The authors gratefully acknowledge funding provided by the Scientific Research Projects Coordination Unit of Social Sciences University of Ankara under grant SBF-2019-190.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Nihat Öner is a Ph.D. candidate in Industrial Engineering at TOBB University of Economics and Technology. Before starting his degree, he earned his B.Sc. and M.Sc. at the same university in Turkey. During his M.Sc., he worked as project assistant funded by the Scientific and Technological Research Council of Turkey grant. He received an outstanding success scholarship from the same university. He is interested in cooperative game theory, cooperative logistics, routing problems, network optimization and heuristic algorithms.

Hakan Gultekin received the B.Sc., M.Sc. and Ph.D. degrees in industrial engineering from Bilkent University, Ankara, Turkey, in 2000, 2002 and 2007, respectively. He has been an associate professor with the Department of Mechanical and Industrial Engineering, Sultan Qaboos University, Muscat, Oman, since September 2018. He has also been affiliated with the Department of Industrial Engineering, TOBB University of Economics and Technology, Ankara, since 2007. Before joining the TOBB University of Economics and Technology, he visited the University of Liège, Belgium, for his postdoctoral studies. His research interests include scheduling, optimisation modelling, and exact and heuristic algorithm development, especially for problems arising in communication and logistics systems, modern manufacturing systems, energy systems, and wireless sensor networks.

Çağrı Koç is an Associate Professor in Department of Business Administration at Social Sciences University of Ankara (ASBU). Prior to joining the ASBU in 2017, he worked as a Postdoctoral Fellow at HEC Montreal and at CIRRELT. He received his Ph.D. degree (2015) in Operational Research/Management Science from the Southampton Business School of University of Southampton. He is the recipient of the Doctoral Award from the Operational Research Society (2015), the Scientific and Technological Research Council of Turkey (TÜBİTAK) CAREER Award (2019) and Postgraduate Publication Awards from the University of Southampton. His research mainly focuses on the application of mathematical and metaheuristic optimization techniques to transportation, logistics and supply chain management problems, more specifically to vehicle routing and scheduling problems.

ORCID

Hakan Gultekin http://orcid.org/0000-0002-5416-3249
Nihat Öner http://orcid.org/0000-0002-8067-224X
Çağrı Koç http://orcid.org/0000-0002-7377-204X

References

Agatz, N., A. Erera, M. Savelsbergh, and X. Wang. 2012. “Optimization for Dynamic Ride-Sharing: A Review.” *European Journal of Operational Research* 223: 295–303.

Antalya Shuttle. 2020. *Antalya Transfer Club*, <https://www.antalyatransferclub.com/>.

Cakici, E., J. M. Scott, W. F. John, and H. N. Geismar. 2013. “Batch Scheduling on Parallel Machines with Dynamic Job Arrivals and Incompatible Job Families.” *International Journal of Production Research* 51: 2462–2477.

Chen, H. P., B. Du, and Q. H. George. 2010. “Metaheuristics to Minimise Makespan on Parallel Batch Processing Machines with Dynamic Job Arrivals.” *International Journal of Computer Integrated Manufacturing* 23: 942–956.

Diri Kenger, Z., Ç. Koç, and E. Özceylan. 2020. "Integrated Disassembly Line Balancing and Routing Problem." *International Journal of Production Research*, in press.

Dong, G., J. Tang, K. K. Lai, and Y. Kong. 2011. "An Exact Algorithm for Vehicle Routing and Scheduling Problem of Free Pickup and Delivery Service in Flight Ticket Sales Companies Based on Set-Partitioning Model." *Journal of Intelligent Manufacturing* 22: 789–799.

Dowsland, K. A., and J. M. Thompson. 2012. "Simulated Annealing." In *Handbook of Natural Computing*, 1623–1655. Berlin: Springer.

Furuhata, M., M. Dessouky, F. Ordóñez, M. E. Brunet, X. Wang, and S. Koenig. 2013. "Ridesharing: The State-of-the-Art and Future Directions." *Transportation Research Part B* 57: 28–46.

Ikura, Y., and M. Gimple. 1986. "Efficient Scheduling Algorithms for A Single Batch Processing Machine." *Operations Research Letters* 5: 61–65.

Koç, Ç., O. Jabali, and G. Laporte. 2018. "Long-Haul Vehicle Routing and Scheduling with Idling Options." *Journal of the Operational Research Society* 69: 235–246.

Koç, Ç., and G. Laporte. 2018. "Vehicle Routing with Backhauls: Review and Research Perspectives." *Computers & Operations Research* 91: 79–91.

Koç, Ç., G. Laporte, and İ. Tükenmez. 2020. "A Review on Vehicle Routing with Simultaneous Pickup and Delivery." *Computers & Operations Research* 122: 104–987.

KPMG. 2018. *The Aviation Industry Leaders Report 2018*. < https://home.kpmg.com/ie/en/home/insights/2018/01/airlines.html > .

Mathirajan, M., and A. Sivakumar. 2006. "A Literature Review, Classification and Simple Meta-Analysis on Scheduling of Batch Processors in Semiconductor." *International Journal of Advanced Manufacturing Technology* 29: 990–1001.

Perez-Gonzalez, P., V. Fernandez-Viagas, M. Z. García, and J. M. Framinan. 2019. "Constructive Heuristics for the Unrelated Parallel Machines Scheduling Problem with Machine Eligibility and Setup Times." *Computers & Industrial Engineering* 131: 131–145.

Potts, C. N., and M. Y. Kovalyov. 2000. "Scheduling with Batching: A Review." *European Journal of Operational Research* 120: 228–249.

Savelsbergh, M., and T. Van Woensel. 2016. "City Logistics: Challenges and Opportunities." *Transportation Science* 50: 579–590.

Shahvari, O., and R. Logendran. 2017. "A Bi-Objective Batch Processing Problem with Dual-Resources on Unrelated-Parallel Machines." *Applied Soft Computing* 61: 174–192.

Song, D. P., J. Dinwoodie, and M. Roe. 2007. "Integrated Vehicle Fleet-Sizing, Leasing and Dispatching Policy in a Shuttle Service System." *International Journal of Logistics: Research and Applications* 10: 29–40.

Tang, J., Y. Yu, and J. Li. 2015. "An Exact Algorithm for the Multi-Trip Vehicle Routing and Scheduling Problem of Pickup and Delivery of Customers to the Airport." *Transportation Research Part E* 73: 114–132.

Thomasson, O., M. Battarra, G. Erdoğan, and G. Laporte. 2018. "Scheduling Twin Robots in a Palletising Problem." *International Journal of Production Research* 56: 518–542.

Toth, P., and D. Vigo. 2014. *Vehicle Routing: Problems, Methods, and Applications*. MOS-SIAM Series on Optimization, Philadelphia.

Trindade, R. S., O. C. B. de Araùjo, M. H. C. Fampa, and F. P. Müller. 2018. "Modelling and Symmetry Breaking in Scheduling Problems on Batch Processing Machines." *International Journal of Production Research* 56: 7031–7048.

Turkish Airlines. 2020. *Shuttle Service*. < https://shuttle.turkishairlines.com/about-shuttle#about > .

Xiao, Z., and H. Shao. 2013. "Two-Dimensional Scheduling: A Review." *Research Journal of Applied Sciences, Engineering and Technology* 6: 1566–1572.

Yu, J. M., R. Huang, and D. H. Lee. 2017. "Iterative Algorithms for Batching and Scheduling to Minimise the Total Job Tardiness in Two-Stage Hybrid Flow Shops." *International Journal of Production Research* 55: 3266–3282.

Zhang, J., X. Yao, and Y. Li. 2020. "Improved Evolutionary Algorithm for Parallel Batch Processing Machine Scheduling in Additive Manufacturing." *International Journal of Production Research* 58: 2263–2282.

Appendix

Table A1. Comparative results on small-size ASBSP benchmark instances.

	MIP2			MIP3			C-1		C-2		C-3		C-4	
Instance	Profit	%GAP	Time (s)	Profit	%GAP	Time (s)	%GAP	Time (s)	%GAP	Time (s)	%GAP	Time (s)	%GAP	Time (s)
10_4_7_186	1,500	0.0	4.5	1,500	0.0	2.5	0.0	4.7	0.0	3.4	0.0	2.3	0.0	1.8
10_4_15_186	3,415	0.0	10.2	3,415	0.0	1.7	0.0	5.2	0.0	4.1	0.0	2.6	0.0	2.0
10_4_7_294	1,815	0.0	35.0	1,815	0.0	1.7	0.0	6.7	0.0	4.1	0.0	2.8	0.0	2.3
10_4_15_294	4,000	0.0	16.9	4,000	0.0	3.0	0.0	5.6	0.0	3.3	0.0	2.1	0.8	2.8
10_8_7_166	2,290	0.0	18.2	2,290	0.0	10.0	1.1	14.8	1.1	8.6	1.1	4.7	1.1	4.0
10_8_15_166	3,120	0.0	34.4	3,120	0.0	24.6	0.0	3.3	0.0	2.5	0.0	1.5	0.0	1.2
10_8_7_349	3,465	0.0	5.5	3,465	0.0	0.9	0.0	7.6	0.0	4.7	0.0	4.0	0.0	2.5
10_8_15_349	6,285	0.0	8.2	6,285	0.0	6.6	0.0	13.6	0.0	7.2	0.0	4.2	0.0	3.4
15_4_7_645	2,040	0.0	30.4	2,040	0.0	2.4	0.0	11.1	0.0	6.1	0.0	3.8	0.0	3.3
15_4_15_645	6,130	0.0	75.7	6,130	0.0	4.5	0.0	6.9	0.0	4.1	0.0	2.6	0.0	2.2
15_4_7_985	2,120	0.0	23.8	2,120	0.0	2.4	0.0	15.7	0.0	8.7	0.0	4.9	0.0	4.4
15_4_15_985	6,540	0.0	64.6	6,540	0.0	3.6	0.0	8.5	0.0	4.9	0.0	3.0	0.0	3.4
15_8_7_615	5,165	1.7	*TL*	5,165	1.5	*TL*	0.0	13.3	0.0	6.9	0.0	4.0	0.0	3.7
15_8_15_615	11,810	0.6	*TL*	11,810	2.2	*TL*	0.2	15.1	0.2	8.4	0.5	6.1	0.5	4.5
15_8_7_985	5,810	0.0	189.1	5,810	0.0	31.3	0.3	20.0	0.3	11.0	0.3	6.5	0.3	5.4
15_8_15_985	16,355	0.0	682.7	16,355	0.0	366.3	0.5	13.2	0.5	7.6	0.5	4.6	0.5	3.6
Average	–	0.1	–	–	0.2	–	0.1	10.3	0.1	6.0	0.1	3.7	0.2	3.2

Note: * Compared with the best objective function value provided by MIP2 and MIP3.

Table A2. Comparative results on medium-size ASBSP benchmark instances.

	MIP2			MIP3			C-1		C-2		C-3		C-4	
Instance	Profit	%GAP	Time (s)	Profit	%GAP	Time (s)	%GAP	Time (s)	%GAP	Time (s)	%GAP	Time (s)	%GAP	Time (s)
20_8_7_531	2,665	89.9	*TL*	2,665	20.5	TL	0.0	11.4	0.0	6.6	0.0	3.7	0.0	5.7
20_8_7_1708	2,900	86.0	*TL*	2,900	19.3	TL	1.0	27.0	1.0	14.1	1.0	8.2	1.0	11.9
20_8_15_531	8,045	25.0	*TL*	8,045	18.7	TL	4.7	9.6	4.7	6.1	4.7	3.4	4.7	3.5
20_8_15_1708	9,320	63.2	*TL*	9,320	13.0	TL	0.1	14.8	0.1	7.9	0.1	4.7	0.1	9.9
20_8_7_462	5,835	0.0	21.1	5,835	0.0	1.7	0.6	17.3	0.6	9.5	0.6	5.3	0.6	12.3
20_8_7_1718	8,780	0.0	3.7	8,780	0.0	1.2	0.3	40.9	0.3	21.3	0.3	11.6	0.3	28.4
20_8_15_462	8,855	0.3	*TL*	8,855	0.5	TL	0.0	67.8	0.0	33.7	0.0	16.7	0.0	33.1
20_8_15_1718	23,755	0.0	1.5	23,755	0.0	1.3	0.0	25.9	0.8	13.9	0.8	7.6	0.8	17.8
20_18_7_531	5,040	81.9	*TL*	5,260	63.1	TL	1.4	12.2	1.6	7.4	1.6	3.9	1.6	6.4
20_18_7_1708	6,100	145.0	*TL*	6,100	72.5	TL	0.5	31.4	0.5	15.8	0.5	8.8	0.5	20.2
20_18_15_531	10,480	12.5	*TL*	10,480	12.0	TL	2.7	49.7	2.7	21.7	2.7	13.8	2.7	11.6
20_18_15_1708	19,470	54.2	*TL*	19,470	39.2	TL	0.3	17.2	0.3	9.2	0.3	5.2	0.3	8.6
20_18_7_462	6,490	1.1	*TL*	6,490	1.1	TL	0.0	236.1	0.0	118.5	0.0	51.4	0.1	108.5
20_18_7_1718	17,545	0.0	4.3	17,545	0.0	3.2	0.9	50.9	0.9	26.6	0.9	14.3	0.9	24.3
20_18_15_462	8,855	1.1	*TL*	8,855	1.1	TL	0.0	58.6	0.0	28.8	0.0	13.5	0.0	30.8
20_18_15_1718	33,505	0.0	8.2	33,505	0.0	89.4	0.4	115.4	0.4	78.7	0.4	67.4	0.4	125.4
20_71_7_531	7,855	40.5	*TL*	7,855	32.9	TL	1.8	111.6	1.8	51.4	1.8	31.1	2.0	24.8
20_71_7_1708	19,375	69.1	*TL*	19,375	57.8	TL	1.4	35.6	1.4	19.2	1.4	10.2	1.4	22.7
20_71_15_531	10,610	14.1	*TL*	10,680	12.3	TL	1.7	4.3	1.7	2.7	1.7	2.6	1.7	2.0
20_71_15_1708	34,350	14.8	*TL*	34,600	12.3	TL	0.3	728.5	0.3	323.5	0.3	154.2	0.3	209.8
20_71_7_462	6,490	1.8	*TL*	6,490	1.8	TL	0.0	4.2	0.0	2.6	0.0	1.7	0.0	3.1
20_71_7_1718	25,370	0.3	*TL*	25,370	0.3	TL	1.0	4,237.8	1.0	1,405.1	1.4	724.9	1.2	598.6
20_71_15_462	8,855	2.0	*TL*	8,855	2.1	TL	0.0	3.5	0.0	2.0	0.0	1.4	0.0	2.2
20_71_15_1718	34,455	0.1	*TL*	34,455	0.1	TL	0.0	809.1	0.0	364.8	0.0	123.1	0.0	238.5
40_8_7_938	5,590	125.6	*TL*	5,975	80.0	TL	1.8	22.7	1.8	12.0	2.8	6.8	2.8	11.3
40_8_7_1719	5,760	159.3	*TL*	6,085	90.8	TL	2.7	35.0	2.7	18.3	2.7	9.9	2.7	18.9
40_8_15_938	15,365	26.6	*TL*	15,470	24.4	TL	4.1	14.7	4.1	8.4	4.1	4.7	4.1	9.4
40_8_15_1719	18,185	69.3	*TL*	19,155	52.0	TL	5.2	21.4	5.2	12.1	5.3	6.5	5.3	10.6
40_8_7_853	8,880	9.7	*TL*	8,880	11.5	TL	3.0	37.5	3.5	15.8	3.5	8.4	3.7	7.3
40_8_7_1706	10,085	20.1	*TL*	10,140	13.7	TL	3.2	51.6	3.7	23.1	5.2	12.7	5.2	10.6
40_8_15_853	16,600	6.6	*TL*	16,600	5.5	TL	3.0	38.9	2.9	25.4	2.8	20.7	2.8	36.5
40_8_15_1706	27,380	9.5	*TL*	27,400	8.7	TL	3.6	34.8	3.6	15.1	3.6	8.3	3.6	7.0
40_18_7_938	10,910	61.4	*TL*	11,100	54.5	TL	1.9	27.0	1.9	14.3	1.9	8.1	1.9	12.0
40_18_7_1719	12,280	118.3	*TL*	13,050	93.0	TL	−0.5	43.0	−0.5	22.4	−0.5	11.6	−0.5	20.0
40_18_15_938	18,605	13.6	*TL*	18,775	13.1	TL	1.6	246.1	1.6	124.7	1.6	60.8	1.6	130.6
40_18_15_1719	31,080	22.7	*TL*	31,380	21.1	TL	2.1	23.4	2.1	12.4	2.1	7.0	2.1	12.7
40_18_7_853	12,255	18.6	*TL*	12,230	17.4	TL	3.3	439.7	3.3	277.6	3.3	168.0	3.3	308.5
40_18_7_1706	19,450	26.0	*TL*	19,530	17.6	TL	2.3	76.5	2.3	43.3	2.6	21.0	2.6	14.9
40_18_15_853	16,720	8.7	*TL*	16,790	8.1	TL	0.7	220.7	0.7	110.9	1.3	55.4	0.7	108.0
40_18_15_1706	34,015	7.1	*TL*	34,015	6.4	TL	0.8	516.2	0.8	328.0	0.8	178.6	0.8	314.9
40_71_7_938	13,775	40.3	*TL*	13,720	40.8	TL	1.7	1,165.9	1.7	538.3	1.7	238.6	1.7	369.8
40_71_7_1719	25,285	39.3	*TL*	24,390	46.2	TL	1.27	4,425.1	1.1	2,101.4	1.7	907.3	1.7	1,711.1
40_71_15_938	18,700	14.0	*TL*	18,445	15.7	TL	1.2	56.7	1.2	19.4	1.2	7.7	1.2	14.8
40_71_15_1719	34,610	13.8	*TL*	34,390	14.5	TL	0.2	826.4	0.2	395.5	0.2	174.3	0.2	438.1
40_71_7_853	12,410	26.9	*TL*	12,385	26.9	TL	2.3	914.1	1.9	409.9	2.3	166.4	1.9	416.2
40_71_7_1706	25,225	23.2	*TL*	25,245	20.8	TL	2.0	4,153.5	2.3	1,994.2	2.7	922.5	2.9	2,067.7
40_71_15_853	16,720	10.1	*TL*	16,735	9.6	TL	0.4	4.3	0.4	2.5	0.4	1.8	0.4	3.6
40_71_15_1706	34,010	9.7	*TL*	33,940	9.7	TL	−0.1	896.0	−0.1	425.3	−0.1	179.2	−0.1	422.4
Average	–	33.0	–	–	22.5	–	1.4	437.7	1.4	198.8	1.5	93.2	1.5	167.4

Note: * Compared with the best objective function value provided by MIP2 and MIP3.