

Public Transport Fleet Scheduling for Minimizing Total Transfer Waiting Time

Farzaneh Nasirian & Mohammad Ranjbar*

Farzaneh Nasirian, Department of Industrial Engineering, Faculty of Engineering, Ferdowsi University of Mashhad
Mohammad Ranjbar, Department of Industrial Engineering, Faculty of Engineering, Ferdowsi University of Mashhad

KEYWORDS

*Transportation,
Scheduling,
scatter search
metaheuristic.*

ABSTRACT

Public transportation has been one of the most important research fields in the last two decades. The purpose of this paper is to create a schedule for public transport fleets, such as buses and metro trains, with the goal of minimizing the total transfer waiting time. We extend previous research studies in the field of transit schedule by considering headways of each route as decision variables. In this paper, we formulate the problem as a mixed integer linear programming model and solve it using ILOG CPLEX solver. For large-scale test instances, we develop a metaheuristic based on the scatter search algorithm to obtain good solutions in reasonable CPU run times. Finally, in the computational section, the efficiency of the proposed model and developed algorithm is compared with that of the existing results in the literature on a real railway network.

© 2017 IUST Publication, IJIEPR. Vol. 28, No. 2, All Rights Reserved

1. Introduction

Due to the high availability and low travel price of public transport, passengers have a significant tendency to use these fleets. To encourage people to use public transport, we should organize the fleets to minimize passengers waiting time and increase their convenience to arrive at their destinations. The necessity for making a satisfying situation for passengers based on the available budget brings about six substantial research areas as follows (Haghani and Shafahi [1]): Transit network design, frequency setting

problem, timetabling design, vehicle scheduling, crew assignment, fleet maintenance scheduling.

In the first category, which is the strategic one, two types of problems are studied. The first one is called Transit Network Design Problem (TNDP) in which routes of vehicles on the given networks are designed. The second category is Transit Network Design and Frequency Setting Problem (TNDFSP) determining fleet routes and service frequency of each route through minimizing the operation costs, number of transfer stations, and total passenger travel times. Due to their high complexity, these problems have received numerous attention, e.g., by Chakroborty and Wivedi [2], Fan and Mumford [3], and Nikolić and Teodorović [4]. The results of all methods are reported and compared in the Nikolić and Teodorović [4]. Farahani et al. [5]

* Corresponding author: *Mohammad Ranjbar*

Email: *m_ranjbar@um.ac.ir*

Received 6 March 2017; revised 29 May 2017; accepted 15 July 2017

and Lusby et al. [6], respectively, express more information about urban network design and railway network design.

Even though service frequency of each route can be determined along with the transit network design, some actors, like variable demand pattern, compel the authorities to determine them based upon a different approach. This approach is obtained by solving the Frequency Setting Problem (see Desaulniers & Hickman [7], Guihaire & Hao [8], and Ibarra-Rojas et al. [9]). The primary goal is to determine the frequency of each route to maximize service quality with considering the given transit network design (see Yu et al. [10]).

The third category is about fleet timetabling which is a conversion of service frequency of routes into a schedule. The output is a set of departure times showing at what time each station is served during the given time period. If both frequencies and departure times are considered as decision variables, the problem is called Transit Network Scheduling Problem (TNSP), which is classified as a NP-Hard problem (see Guihaire and Hao [11]). The purpose of TNSP is to determine a schedule with minimized waiting time, maximized simultaneous arrivals, or minimized fleet size, etc. For the first time, Ceder et al. [12] introduced the concept of synchronization for TNSP. They proposed a model to create a timetable for a given network of buses to maximize the number of simultaneous arrivals at transfer stations with considering headways and the difference of two subsequent vehicles' departure times in each route as decision variables. The definition of synchronization was improved by Shafahi and Khani [13] in which the minimization of the total transfer waiting time is considered instead of maximizing simultaneous arrivals. In the fourth category, the purpose is to find an assignment of trips to vehicles, such that each vehicle performs a feasible sequence of trips and the overall cost is minimized. In addition, in some papers like Anupam et al. [14], transportation problem with fuzzy-stochastic costs has been studied. Liu and Ceder [15] developed a bi-objective bi-level model using deficit function approach for integrated public transport timetable synchronization and vehicle scheduling demand assignment. Bunte and Kliwer [16] addressed a comprehensive review of vehicle scheduling. Moreover, Daraio et al. [17] presented a critical review with directions for future research about

efficiency and effectiveness of already research paper in the urban public transport sector.

The common goal of the fifth category is to find the minimum number of drivers to cover a set of pieces of work. This field of study directly depends on policies of transportation companies; consequently, each company has its own crew scheduling software. Ciancio et al. [18] developed an integrated algorithm for shift scheduling problems for local public transport companies. They studied two shift-scheduling problems and developed local search and simulated annealing used to analyze the solution space. Finally, in the sixth category, vehicles are scheduled for daily inspections and maintenances and assigned to the limited existing facilities to minimize the amount of times that vehicles are pulled out of their services (See Haghani and Shafahi [1]).

The contributions of this article are twofold: (1) creating a schedule for public transport fleet with the goal of minimizing the total transfer waiting time as a mixed integer linear programming model; (2) We also develop a metaheuristic based on a scatter search algorithm to find good solutions for large-scale test instances in a reasonable CPU run time. In the former model, the heterogeneous headways at each service cycle, the period in which a vehicle serves all the stations of its route and returns to its starting point, are considered as decision variables, and stations have limited capacity.

The remainder of this paper is organized as follows. Problem statement and formulation are presented in Section 2. The scatter search algorithm and its operators are described in Section 3. Moreover, computational results are shown in Section 4. Finally, conclusions and future research opportunities are presented in Section 5.

2. Problem Statement and Formulation

The objective of the proposed problem is to minimize the total transfer waiting time at transfer stations. Since we consider heterogeneous headways as decision variables for each service cycle, arrival times of the vehicles at transfer stations are the outputs of the model.

2-1. Problem statement

The main assumptions considered in the model are as follows:

- The transit network is given as an input.
- Due to the various distributions of some parameters, such as number of passengers, travel times, and minimum and maximum

allowable headways in different time of a day, we have to model the problem for a known period of time in which these parameters are assumed to be constant.

- Number of passengers being transferred between each pair of routes is given. It is assumed to have a unique distribution under the studied period.
- Dwell time at a station is a constant and known value.
- Walking times between two transfer stations are known. Therefore, this includes an average amount of time that each passenger needs to get off the vehicle and walks towards another one.
- Headways are heterogeneous and considered as decision variables for each service cycle. The amount of headway is bounded to a minimum value regarding the existing fleet size as well as a maximum limit to avoid passengers' dissatisfaction. Shafahi and Khani [13] considered constant headway for both directions of each route that simplifies the model noticeably. Under this condition, only the departure time of the first vehicle of each route is a decision variable. In addition, Ceder et al. [12] considered heterogeneous headways which are constant in all service cycles. We relax these assumptions and suppose that headways at both directions of routes are heterogeneous in all service cycles.

- We consider an interval for the travel time of the vehicles between transfer stations. This interval is determined based on the minimum and maximum allowable speed and average traffic on that part. This assumption helps us to set heterogeneous headways at each transfer station. If two sequential transfer stations are so close that makes heterogeneous headway impossible, lower and upper bounds of travel time should be the same. In this case, we have a constant headway between the two considered stations.
- Due to the limited space at transfer stations, number of vehicles stopping at the station simultaneously should be restricted. Thus, we consider a limited capacity for each transfer station.
- Vehicles should serve the stations located on both directions of their routes at each service cycle. As Fig. 1 illustrates, if the vehicle of route starts its work from terminal number 1 (D1), it would serve A, B, D2, C, E, F, and D1 stations, respectively. Terminals are the final stations at each direction in which the driver has a respite before starting to serve the other direction and all passengers should get off the vehicles. We take into account the transfer stations, capacitated stations, and corresponding terminals on the considered route as stations in the rest of the paper.

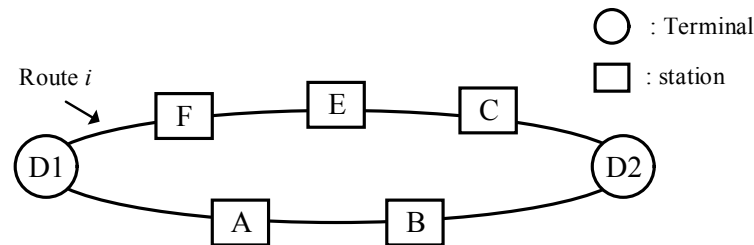


Fig. 1. Service cycle

- We suppose that there are two types of vehicle assignments, which we consider in our model. All vehicles of a route can be assigned to one terminal or split equally between two terminals of its route, see Fig. 2. In state (a), vehicles initiate their work from one direction of the route (first or return direction), while in state (b), they start to give the service from

both directions. In state (a), passengers of one direction get the service after a longer waiting time compared with state (b) at the first service cycle. In the next service cycles, the passengers do not face such waiting times because the vehicles would be scheduled to traverse the route in an allowable headway interval.

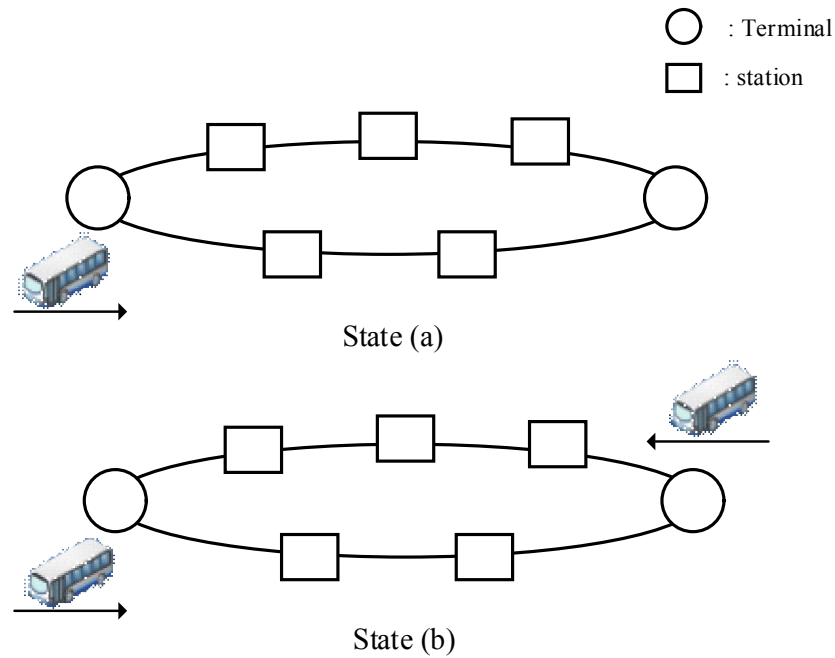


Fig. 2. Two states for initiating the service

- Some of the vehicles might not serve a number of stations in their last service cycle during the considered time period; consequently, the remaining stations would be scheduled with the policy of the next time period. We call the passengers of these stations as non-served passengers. For the last time period, the vehicles are responsible to serve the remaining stations with the current policy till they arrive at one of the terminals of their routes.
- Since the service quality is variable during each time period of a day, we have to maximize the number of served passengers in the considered time period. In order to achieve this goal, we define a virtual vehicle for each route. If any passenger is assigned to this vehicle, it implies that he/she does not get service in the related time period. So, for any assignment of this virtual vehicle, a penalty is considered in the objective function of the model.

2-2. Variables and parameters

We define the following parameters and notations to formulate our proposed problem.

R : Set of routes in the transit network;
 $R = \{1, 2, \dots, |R|\}$ with indices i, j .

TS : Set of all stations in the network with indices s, s' .

N_i : Set of the vehicles of route i which has to be scheduled during the time period with indices a, b . Element $N_i(|N_i|)$ is the last vehicle of route i in the considered period; therefore, we set the index of virtual vehicle as $N_i(|N_i|) + 1$.

C : Set of all possible transfer patterns:
 $C = \{(i, j, s, s') \mid i, j \in R; s, s' \in TS\}$. Transfer pattern (i, j, s, s') represents a situation in which passengers of route i get off at station s to take the service of route j at station s' .

$SR(s)$: Set of routes that serve station s .

ST_{ia} : Set of all stations that the a th vehicle of route i serves during the time period;
 $ST_{ia}(v) \in TS$ represents the v th element of set ST_{ia} . This parameter can be determined based on the current schedule of the vehicles or be approximated with the average speed of the vehicles to get which stations they could serve during the time period. We need to define this parameter due to the assumption that some of the vehicles might not serve a number of stations in their last service cycle during the considered period.

fs_i : Number of stations located on the first direction of route i .

bs_i : Number of stations located on the return direction of route i .

nv_i : Number of available vehicles in route i . For simplification, we assume that this number is even. After one service cycle, index of the a th vehicle of route i will be changed to $a + nv_i$. This assumption impedes the need to define another vehicle index for service cycle. For instance, if the number of available vehicles in route i is ten, indices 4, 14, and 24 are related to the fourth vehicle of route i in the first, second, and third service cycles, respectively.

$P_{ijss'}$: Number of passengers in route i getting off at station s to take the service of route j at station s' .

st_{is} : Average dwell time of a vehicle of route i at station s . Dwell time at terminals is the average amount that the related agency determines; dwell time would be higher than low traffic ones for the high traffic routes.

$wt_{ss'}$: Average walking time between two stations s and s' .

$d_{is}^{\min}$: Minimum travel time of a vehicle of route i to station s from its preceding station.

$d_{is}^{\max}$: Maximum travel time of a vehicle of route i to station s from its preceding station.

$h_i^{\min}$: Minimum allowable headway at route i .

$b_{is} = \max(h_i^{\min}, st_{is})$. This parameter is used in the model instead of $h_i^{\min}$ to avoid simultaneous presence of two subsequent vehicles of route i at station s .

$h_i^{\max}$: Maximum allowable headway at route i .

C_s : Capacity of station s .

PE : Penalty for non-served passengers.

Parameters N_i , ST_{ia} , $P_{ijss'}$, st_{is} , $d_{is}^{\min}$, $d_{is}^{\max}$, $h_i^{\min}$, and $h_i^{\max}$ are those changed during each time period of a day. Some of these parameters have direct effect on each other. For instance, in

the low-demand time periods, $h_i^{\min}$ and $h_i^{\max}$ are increased to have bigger headways and subsequently dwell time at terminals should also be increased to let the vehicles stay for a longer time at the terminals.

The decision variables of our model are as follows:

AT_{ias} : Arrival time of the a th vehicle of route i at station s .

$w_{iajss'}$: Waiting time of the passengers of the a th vehicle of route i who get off at station s to get on the service of route j at station s' . If the vehicle of route j is present when passengers arrive at station s' , i.e., $AT_{ias} + wt_{ss'}$, they get on the service without waiting time. In addition, if the vehicle of route j arrives at station s' after the passengers do, the difference equals $AT_{jbs'} - AT_{ias} - wt_{ss'}$, and is considered as the waiting time.

$m_{iajbs'}$: A binary variable that takes 1 if passengers of the a th vehicle of route i getting off at station s are served by the b th vehicle of route j at station s' ; otherwise, it takes zero.

$y_{iajbs'}$: A non-negative variable that takes the value of $AT_{jbs'}$ if passengers of the a th vehicle of route i getting off at station s are served by the b th vehicle of route j at station s' ; otherwise, it takes zero.

x_{iajbs}^1 : A binary variable that gets 1 if the a th vehicle of route i and the b th vehicle of route j are available at station s at the same time.

x_{iajbs}^2 : A binary variable that gets 1 if the b th vehicle of route j departs from station s before the a th vehicle of route i arrives at this station.

2-3. Problem formulation

Our developed model for public transport fleet scheduling reads as follows:

$$\min \sum_{\forall(i,j,s,s') \in C} \sum_{a \in N_i; s \in ST_{ia}} P_{ijss'} \times w_{iajss'} \quad (1)$$

$$s.t.: \sum_{b \in \{N_j \cup N_j(|N_j|+1) \mid s' \in ST_{jb}\}} m_{iajbs'} = 1 \quad \forall(i,j,s,s') \in C; a \in \{N_i \mid s \in ST_{ia}\} \quad (2)$$

$$y_{iajbs'} \leq AT_{jbs'} \quad \forall(i,j,s,s') \in C; a \in \{N_i \mid s \in ST_{ia}\} \quad (3)$$

$$b \in \{N_j \cup N_j(|N_j|+1) \mid s' \in ST_{jb}\}$$

$$y_{iajbss'} \geq AT_{jbs'} - M(1 - m_{iajbss'}) \quad \forall (i, j, s, s') \in C; a \in \{N_i | s \in ST_{ia}\} \quad (4)$$

$$b \in \{N_j \cup N_j(|N_j|) + 1 | s' \in ST_{jb}\}$$

$$y_{iajbss'} \leq Mm_{iajbss'} \quad \forall (i, j, s, s') \in C; a \in \{N_i | s \in ST_{ia}\} \quad (5)$$

$$b \in \{N_j \cup N_j(|N_j|) + 1 | s' \in ST_{jb}\}$$

$$\sum_{b \in \{N_j \cup N_j(|N_j|) + 1 | s' \in ST_{jb}\}} y_{iajbss'} + st_{js'} - AT_{ias} - wt_{ss'} \geq 0 \quad \forall (i, j, s, s') \in C; a \in \{N_i | s \in ST_{ia}\} \quad (6)$$

$$w_{iajss'} \geq \sum_{b \in \{N_j | s' \in ST_{jb}\}} y_{iajbss'} - AT_{ias} - wt_{ss'} \quad \forall (i, j, s, s') \in C; a \in \{N_i | s \in ST_{ia}\} \quad (7)$$

$$w_{iajss'} \geq PEm_{iaj(N_j(|N_j|)+1)ss'} \quad \forall (i, j, s, s') \in C; a \in \{N_i | s \in ST_{ia}\} \quad (8)$$

$$d_{is}^{\min} \leq AT_{ils} \leq d_{is}^{\min} + h_i^{\max} \quad \forall i \in R; s = ST_{il}(1) \quad (9)$$

$$d_{is}^{\min} + st_{is'} \leq AT_{ias} - AT_{ias'} \leq d_{is}^{\max} + st_{is'} \quad \forall i \in R; \forall a \in N_i; s = ST_{ia}(v+1) \quad (10)$$

$$s' = ST_{ia}(v); v \in \{1, \dots, |ST_{ia}| - 1\}$$

$$d_{is}^{\min} + st_{is'} \leq AT_{i(a+nv_i)s} - AT_{ias'} \leq d_{is}^{\max} + st_{is'} \quad \forall i \in R; a \in \{a \geq 1 | a + nv_i \in N_i\} \quad (11)$$

$$s = ST_{ia}(1); s' = ST_{ia}(|ST_{ia}|)$$

$$AT_{i(N_i(|N_i|)+1)s} = M \quad \forall i \in R; \forall s \in ST_{i(N_i(|N_i|)+1)} \quad (12)$$

$$AT_{jbs} - AT_{ias} \leq st_{is} + M(1 - x_{iajbs}^1) \quad \forall s \in TS; i, j \in \{SR(s) | j \neq i\} \quad (13)$$

$$b \in \{N_j | s \in ST_{jb}\}; a \in \{N_i | s \in ST_{ia}\}$$

$$AT_{jbs} - AT_{ias} \geq -st_{js} - M(1 - x_{iajbs}^1) \quad \forall s \in TS; i, j \in \{SR(s) | j \neq i\} \quad (14)$$

$$b \in \{N_j | s \in ST_{jb}\}; a \in \{N_i | s \in ST_{ia}\}$$

$$AT_{jbs} - AT_{ias} \leq -st_{js} - \varepsilon + M(1 - x_{iajbs}^2) \quad \forall s \in TS; i, j \in \{SR(s) | j \neq i\} \quad (15)$$

$$b \in \{N_j | s \in ST_{jb}\}; a \in \{N_i | s \in ST_{ia}\}$$

$$AT_{jbs} - AT_{ias} \geq st_{is} + \varepsilon - M(x_{iajbs}^1 + x_{iajbs}^2) \quad \forall s \in TS; i, j \in \{SR(s) | j \neq i\} \quad (16)$$

$$b \in \{N_j | s \in ST_{jb}\}; a \in \{N_i | s \in ST_{ia}\}$$

$$\sum_{j \in \{SR(s) | j \neq i\}} \sum_{b \in \{N_j | s \in ST_{jb}\}} x_{iajbs}^1 \leq C_s - 1 \quad \forall s \in TS; \forall i \in SR(s); a \in \{N_i | s \in ST_{ia}\} \quad (17)$$

$$AT_{ias} \text{ Integer} \quad \forall i \in R; \forall a \in N_i; \forall s \in ST_{ia} \quad (18)$$

$$x_{iajbs}^1, x_{iajbs}^2 \in \{0, 1\} \quad \forall s \in TS; i, j \in \{SR(s) | j \neq i\}; \quad (19)$$

$$a \in \{N_i | s \in ST_{ia}\}; b \in \{N_j | s \in ST_{jb}\}$$

$$w_{iajss'}, y_{iajbss'} \geq 0 \quad \forall (i, j, s, s') \in C; a \in \{N_i | s \in ST_{ia}\} \quad (20)$$

$$b \in \{N_j \cup N_j(|N_j|) + 1 | s' \in ST_{jb}\}$$

$$m_{iajbss'} \in \{0, 1\} \quad \forall (i, j, s, s') \in C; a \in \{N_i | s \in ST_{ia}\} \quad (21)$$

$$b \in \{N_j \cup N_j(|N_j|) + 1 | s' \in ST_{jb}\}$$

The objective function of the model minimizes the total waiting time and penalty for non-served passengers, presented in (1). Constraints (2) to (6) determine the nearest vehicle of route j which could serve passengers of the a th vehicle of route i at station s' . There is a possibility that the vehicle cannot serve some stations in its last

service cycle, during the considered time period. Therefore, we limit set C by defining $a \in \{N_i | s \in ST_{ia}\}$ and $b \in \{N_j | s' \in ST_{jb}\}$ to impose constraints on the vehicles serving stations s and s' . By constraint (2), one vehicle of route j should serve the passengers of the

a th vehicle of route i at station s' . So, if the b th vehicle of route j serves the passengers of the a th vehicle of route i , $m_{iajbss'}$ will be equal to 1 and makes constraint (5) redundant; $y_{iajbss'}$ will take $AT_{jbs'}$ value through constraints (3) and (4).

Since passengers of the a th vehicle of route i arrive at station s' at $AT_{ias} + wt_{ss'}$ time instant, they should be served by the first vehicle of route j departing from station s' after the passengers' arrival time. This issue is shown by constraint (6). If passengers wait at station s' to get service, the term $\sum_{b \in \{N_j | s' \in ST_{jb}\}} y_{iajbss'} - AT_{ias} - wt_{ss'}$ will

take a positive value. In this case, since the objective function is to minimize the total waiting time, constraint (7) will be converted to an equality one. If passengers take service without any waiting time, the aforementioned term gets a negative value and $w_{iajbss'}$ takes zero as its minimum amount.

If passengers of the a th vehicle of route i at station s' are not served by vehicles of route j during the considered time period, they are assigned to the virtual vehicle and $m_{iaj(N_j(|N_j|)+1)ss'}$ gets the value of 1, and through constraint (8), penalty of PE is applied to the objective function. Constraint (9) limits the initial time of the first service cycle in each route to $h_i^{\max}$. By eliminating this constraint, more synchronization may happen, but service quality might be decreased due to the long delay of some routes for starting their services. We set $h_i^{\max}$ as an upper bound in Constraint (9) such as Ceder et al. [12] and Shafahi and Khani [13], but it could be changed based on the relevant company decision.

$$\begin{aligned} AT_{ias} \leq AT_{jbs} \leq AT_{ias} + st_{is} &\Rightarrow 0 \leq AT_{jbs} - AT_{ias} \leq st_{is} \\ \begin{cases} AT_{jbs} \leq AT_{ias} \\ AT_{jbs} + st_{js} \geq AT_{ias} \end{cases} &\Rightarrow \begin{cases} AT_{jbs} - AT_{ias} \leq 0 \\ AT_{jbs} - AT_{ias} \geq -st_{js} \end{cases} \Rightarrow -st_{js} \leq AT_{jbs} - AT_{ias} \leq 0 \end{aligned}$$

Thus, if $AT_{jbs} - AT_{ias}$ takes a value from interval $[-st_{js}, st_{is}]$, the b th vehicle of route j and the a th vehicle of route i will meet each other at station s .

III. If the b th vehicle of route j arrives at station s after the departure time of the

The Vehicle travel time between two sequential stations is restricted by a minimum and a maximum amount in constraint (10). After one service cycle, index of the a th vehicle of route i will be changed to $a + nv_i$; $AT_{i(a+nv_i)s}$, used in constraint (11), describes the arrival time of the a th vehicle of route i at station s in the next service cycle. Since the initial time of each service cycle of vehicle depends on the closure time of the prior one, constraint (11) correlates these two terms.

The arrival times of virtual vehicle at all transfer stations on its route are determined by constraint (12). Parameter M is considered as a big number, such that arrival times of virtual vehicles are large enough to cover all non-served passengers. The concept of limited capacity of stations has been presented in the developed model through constraints (13) to (17). Suppose that the a th vehicle of route i is available at station s . Then, three following cases may happen:

- I. If the b th vehicle of route j departs from station s before the arrival time of the a th vehicle of route i , we have $AT_{jbs} + st_{js} < AT_{ias}$, and consequently $AT_{jbs} - AT_{ias} < -st_{js}$.
- II. If the b th vehicle of route j is available at station s , we may face two cases. In the first one, the b th vehicle of route j arrives at station s after the arrival time of the a th vehicle of route i . In the second one, the b th vehicle of route j arrives at station s before the arriving of the a th vehicle of route i and it departs from the station after that. Under this condition, we have the following relations:

a th vehicle of route i , we have $AT_{jbs} > AT_{ias} + st_{is}$ and as a result $AT_{jbs} - AT_{ias} > st_{is}$.

If x_{iajbss}^1 gets 1, the second condition occurs via constraints (13) and (14). In addition, if x_{iajbss}^2 takes value of 1, the first condition happens

through constraint (15); if x_{iajbs}^1 and x_{iajbs}^2 both get zero at once, the third condition happens by constraint (16). Since constraints (14) and (15) contradict each other, x_{iajbs}^1 and x_{iajbs}^2 will never get 1 at the same time. Parameter ε is a small number that we use to avoid strict inequality. The number of vehicles simultaneously available with the a th vehicle of route i at station s are bounded by the right-side expression of constraint (17). Since the a th vehicle of route i is not enumerated, capacity of station s is decreased by one unit in constraint (17). Finally, type and range of variables are defined in constraints (18) to (21). Arrival time is

considered as an integer variable to be applicable in the real world conditions.

As already mentioned, we have two types for vehicle assignment to the terminals of its route, which affect the initial service cycle. If all the vehicles of a route are assigned to one terminal, initiating their work from one direction of the route, constraint (22) needs to be added to the former model, while if the vehicles of a route are split equally between two terminals, initiating their work from both directions, constraints (23) to (28) should be appended to the prior model instead of constraint (22). These constraints control the headways at both directions of routes in all service cycles.

$$b_{is} \leq AT_{i(a+1)s} - AT_{ias} \leq h_i^{\max} \quad \forall i \in R; \forall a \in \{a \geq 1 | a, a+1 \in N_i\} \quad (22)$$

$$s = ST_{i(a+1)}(v); v \in \{1, \dots, |ST_{i(a+1)}|\}$$

$$d_{is}^{\min} \leq AT_{i(1+\frac{nv_i}{2})s} \leq d_{is}^{\min} + h_i^{\max} \quad \forall i \in R; s = ST_{i(1+\frac{nv_i}{2})}(1) \quad (23)$$

$$b_{is} \leq AT_{i(a+1)s} - AT_{ias} \leq h_i^{\max} \quad \forall i \in R; a \in [1+k, \min(\frac{nv_i}{2} + k, N_i(|N_i|)) - 1]; a+1 \in N_i \quad (24)$$

$$\forall k \in \{0, \frac{nv_i}{2}, \dots, \left\lceil \frac{2N_i(|N_i|)}{nv_i} \right\rceil \times \frac{nv_i}{2}\};$$

$$s = ST_{i(a+1)}(v); v \in \{1, \dots, |ST_{i(a+1)}|\}$$

$$b_{is} \leq AT_{i(a-nv_i+1)s} - AT_{ias} \leq h_i^{\max} \quad \forall i \in R \quad (25)$$

$$\forall a \in \{nv_i, 2nv_i, \dots, \left\lceil \frac{N_i(|N_i|)}{nv_i} \right\rceil \times nv_i | a, (a-nv_i+1) \in N_i\}$$

$$s = ST_{i(a-nv_i+1)}(v); v \in \{fs_i + 2, \dots, |ST_{i(a-nv_i+1)}|\}$$

$$b_{is} \leq AT_{i(a+1)s} - AT_{ias} \leq h_i^{\max} \quad \forall i \in R \quad (26)$$

$$\forall a \in \{nv_i, 2nv_i, \dots, \left\lceil \frac{N_i(|N_i|)-1}{nv_i} \right\rceil \times nv_i | a, a+1 \in N_i\}$$

$$s = ST_{i(a+1)}(v); v \in \{1, \dots, \min(|ST_{i(a+1)}|, fs_i + 1)\}$$

$$b_{is} \leq AT_{i(a+1)s} - AT_{ias} \leq h_i^{\max} \quad \forall i \in R \quad (27)$$

$$\forall a \in \{\frac{nv_i}{2}, \frac{3nv_i}{2}, \dots | a \leq N_i(|N_i|)-1 \& a, a+1 \in N_i\}$$

$$s = ST_{i(a+1)}(v); v \in \{bs_i + 2, \dots, |ST_{i(a+1)}|\}$$

$$b_{is} \leq AT_{i(a+1+nv_i)s} - AT_{ias} \leq h_i^{\max} \quad \forall i \in R \quad (28)$$

$$\forall a \in \{\frac{nv_i}{2}, \frac{3nv_i}{2}, \dots | a \leq N_i - nv_i - 1 \& a, a+1+nv_i \in N_i\}$$

$$s = ST_{i(a+1+nv_i)}(v); v \in \{1, \dots, \min(|ST_{i(a+1+nv_i)}|, bs_i + 1)\}$$

In constraint (22), the possible range of values for $AT_{i(a+1)s} - AT_{ias}$, indicating headways of two subsequent vehicles, is determined.

We assume that, in the second state, vehicles are assigned to both directions of the routes equally and start their services from two terminals. Indices of the assigned vehicles to the first direction are $1, 2, \dots, \frac{nv_i}{2}$, which will be increased nv_i units after each service cycle. Correspondingly, indices of the assigned vehicles to the return direction are $\frac{nv_i}{2} + 1, \dots, nv_i$.

Therefore, $1 + \frac{nv_i}{2}$ indicates the index of vehicle starting the first service cycle from the return direction. Constraint (23) limits the initial time of the first service cycle starting from the return direction of each route to $h_i^{\max}$, similar to constraint (9) implemented on the first direction. For the second state, when vehicles are assigned equally to both directions, the headways among

vehicles of each direction are determined with constraint (24). However, this constraint is not enough by itself; we define the rest of the constraints, i.e., (25), (26), (27) and (28), to put a restriction on headway between the first vehicle of one direction with the last vehicle of another one. Fig. 3 shows six vehicles in a route where three of them have been assigned to each direction. The possible range of values for headways between the first vehicle of the return direction and the last vehicle of the first direction, like vehicles number four and three in Fig. 3, is determined at stations on the first direction through constraint (25). Their headways at stations on the return direction are also controlled by constraint (26). Index of the first vehicle of the return direction is increased in the next service cycle and this is the reason we show this concept by using two separate constraints. Similarly, constraints (27) and (28) arrange the headways between the first vehicle of the first direction and the last vehicle of the return direction, such as vehicles number one and six in Fig. 3, at stations on the first and return directions, respectively.

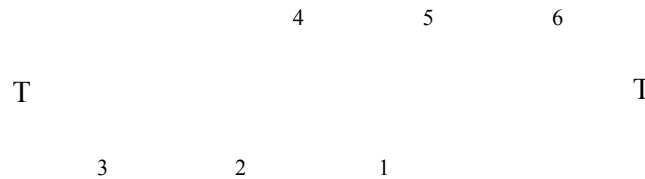


Fig. 3. Vehicle assignment to both of the first and return directions

However, we can run our proposed model for all time periods to find the optimal schedule in each one, leading to a suboptimal solution except the situation in which we consider the day as one period. Suppose that we have five time periods in a day and the first one has been scheduled. Our model is able to schedule the second time period without dissociation via setting the determined schedule as an input and changing some of the parameters mentioned in Section 2.3.

3. Scatter Search Algorithm

Complexity of the developed mixed integer linear programming models in Section 2 is intensified

by increasing the number of elements of set C , transfer patterns. Therefore, we develop a metaheuristic solution method based on the scatter search (SS) algorithm to solve the problem for real-life networks.

3-1. General overview

Scatter search is an evolutionary algorithm that constructs new solutions by combining the existing ones in a systematic method. For a general introduction, we refer readers to Laguna and Martí [19]. Algorithm. 1 shows the main structure of our algorithm.

-
1. Construct an initial population P with size of $|P|$. Set $StopCount = 0$
- While** ($StopCount = 0$) **do**
2. Build $RefSet$ by using the reference set building method.
 3. Generate $Newsubsets$ with the subset generation method. Make $P = \emptyset$.
- While** ($Newsubsets \neq \emptyset$) **do**
-

4. Select new σ in *Newsubsets*.
5. Apply solution combination method to σ in order to obtain r new solutions and add them to P .
If the new generated solutions have better objective values compared with their parents, apply improvement method over them and replace them with the improved ones.
6. Delete σ from subsets.

end while

7. If the termination criterion is met, set $StopCount = 1$.

end while

Algorithm. 1. General structure of the scatter search algorithm

In the first step, we generate an initial population P with $|P|$ different solutions randomly. In the second step, we construct the reference set, shown as *RefSet*, including $b_1 + b_2$ solutions.

The first b_1 solutions are the ones with the best objective function values and the last b_2 solutions are the ones with high diversity. Next, we generate *NewSubsets* in which each element σ contains two elements of *RefSet* solutions. The two solutions of each set are combined to generate r new solutions, added to population P . If any new solution has better objective function value than its parents do, the improvement method is applied to it and the obtained solution

is replaced with it. If the termination criteria, considered as a CPU run time, are met, the algorithm stops; otherwise, the algorithm restarts from step 2 with building a new *RefSet*.

a. Solution representation

The solution representation has a major impact on the efficiency of metaheuristic methods. Since the arrival times of vehicles to stations are the main decision variables in our problem, they are used as solution representation. Fig. 4 shows solution representation as an array in which AT_{ias} indicates arrival times of all vehicles of route i at their stations.

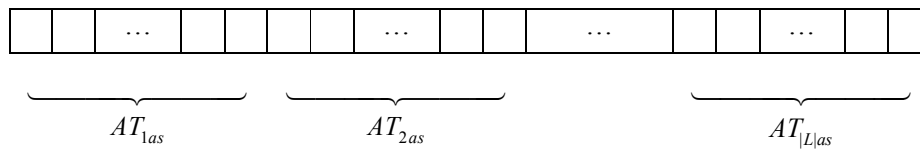


Fig. 4. Solution Representation

3-2. Construction of initial population P

The first step in SS algorithm is constructing an initial population. For this purpose, we generate each solution as represented in the previous section, randomly. The advantage of the random generation method is that all parts of the solution space have an equal chance to be investigated. The process of random generation is designed, such that the feasibility is guaranteed. In this process, the constraints, such as headways, travel times, etc., are considered while the cells of a solution representation are filled.

3-3. Reference set building method

RefSet includes solutions based on both quality and diversity. Construction of *RefSet* starts with the selection of the best solution of P . This solution is added to *RefSet* and deleted from P . The next best solution x in P is chosen only if $MD_x \geq a_1$, where MD_x is the minimum distance of solution x with the current solutions in

RefSet and a_1 is a threshold distance. Distance between two solutions x and y is calculated as follows:

$$D(x, y) = \sum_{i \in L} \sum_{a \in N_i} \sum_{s \in ST_{ia}} \frac{|AT_{ias}^x - AT_{ias}^y|}{\sum_{i \in L} \sum_{a \in N_i} |ST_{ia}|}, \quad (29)$$

where AT_{ias}^x represents the arrival time of the a th vehicle of route i at station s in solution x . This process is repeated until b_1 solutions are selected. We follow the same strategy for selecting the last b_2 solutions by choosing the best solution with $MD_x \geq a_2$ where a_2 is also a threshold distance which is greater than a_1 . If no qualified solution is found, we complete *RefSet* with randomly generated solutions based on the method used for the initial population.

3-4. Newsubsets generation method

Before the combination of two solutions, *Newsubsets* should be determined. The combination of each pair of elements of the first b_1 solutions of *RefSet* and the combination of each element of the first b_1 solutions with each element of the last b_2 solutions of *RefSet*

construct *Newsubsets*. So, *Newsubsets* has $b_1 \times (b_1 - 1) + b_1 \times b_2$ elements.

3-5. Combination method

As illustrated in Fig. 5, for combination of two solutions, we select two routes randomly and exchange the schedules of these routes in the considered parent solutions. The advantage of this process is that all generated new solutions are feasible.

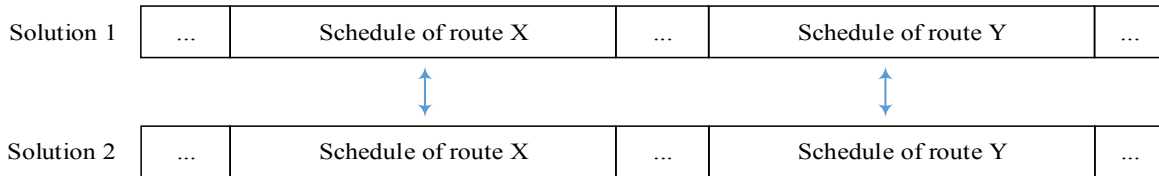


Fig. 5. Illustration of combination method

3-6. Improvement method

Improvement method is a local search procedure exploring high-quality solutions. Since for each (i, a, j, s, s') , waiting times are determined through $AT_{jbs'} - AT_{ias} - wt_{ss'}$, by decreasing $AT_{jbs'}$ and increasing AT_{ias} , waiting times could be reduced. In the first step of this improvement method, all (i, a, j, s, s') are sorted in an ascending order based on their waiting times. This helps us to first modify the quadruplets having large waiting times. It should be noticed that decreasing of $AT_{jbs'}$ or increasing of AT_{ias} may increase waiting times of some other passengers. Thus, for each selected quadruplet τ with non-zero waiting time, $AT_{jbs'}$ is decreased and AT_{ias} is increased in an iterative process where, in each iteration, only one of the variables is changed into one unit, such that the solution remains feasible. Finally, the best-found solution is selected.

4. Computational Results

In this section, we compare the performances of our proposed model, described in Section 2, and

the SS algorithm with those of existing models in the literature and a real railway network. We use ILOG CPLEX 12.3 and a PC with Corei5-2.67 GHz and 4.00 GB RAM to run our models and algorithm.

4.1. Comparative computational results

Our first proposed model is implemented on two test instances generated by Shafahi and Khani [13] and Ceder et al. [12]. Fig. 6 illustrates the first test instance in which we have four routes with two transfer stations. All the assumptions and parameters values considered for this test instance are taken from Shafahi and Khani [13]. In addition, the minimum and maximum allowable headways are assumed to be 5 and 10 minutes, respectively, based on the constant amounts given in Shafahi and Khani [13]. The purpose is the construction of a schedule for a one-hour horizon, such that the total transfer waiting time is minimized. The definition of waiting time in the model of Shafahi and Khani [13] contains dwell time, which is a bit different from our definition. Hence, for a fair comparison, we used their description for computing waiting times.

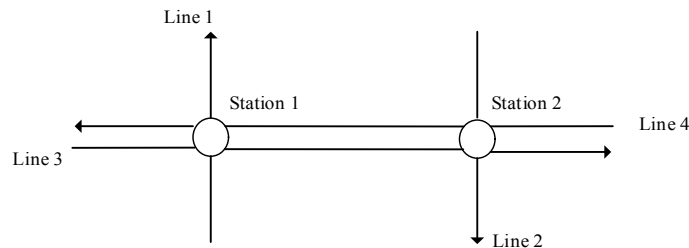


Fig. 6. The first instance in Shafahi and Khani [13]

Comparative results of the first test instance based on two different conditions, i.e., variable and constant headways, are shown in Table 1. The obtained waiting times for variable and constant headways are 4540 and 4550 minutes, respectively. As expected, variable headways'

assumption improves the schedule, but it also increases the CPU running time. These outcomes also show the results reported by Shafahi and Khani [13] in which the obtained total waiting time is 4860 minutes, which is not an optimal solution.

Tab. 1. Comparison result with Shafahi and Khani [13]

	Shafahi and Khani [13]	The proposed model (heterogeneous headways)	The proposed model (constant headways)
Total waiting time (minutes)	4860	4540	4550
CPU run time (seconds)	<1	15.94	2.43

It should be mentioned that vehicles of each route might have different traveling times to each station. Therefore, there is a possibility that some passengers do not get their services at the last service cycle in the considered time period. We try to minimize number of such non-served passengers in the proposed model. As shown in

Table 2, number of non-served passengers in Shafahi and Khani [13] is 310 while this value is 160 for our proposed model. In addition to 48.3% increase in the number of served passengers, our proposed model found a schedule decreasing the total waiting time around 6.5%.

Tab. 2. Number of non-served passengers

	Shafahi and Khani [13]				Our model (heterogeneous headways)			
	route				route			
route	1	2	3	4	1	2	3	4
1	---	---	0	0	---	---	0	0
2	---	---	0	0	---	---	0	0
3	40	10	---	---	0	10	---	---
4	250	10	---	---	150	0	---	---

Moreover, there are four test examples in Ceder et al. [12] in which minimum and maximum headways, travel times, and the number of vehicles are given and the objective function is the maximization of simultaneous arrivals in order to minimize passengers waiting time with the assumption of heterogeneous headways. In studies of Ceder et al. [12], some of the parameters, such as number of transferring passengers, dwell times, and walking times, are not defined. Thus, for a fair comparison, we set these parameters to 1, 0, 0, respectively, similar

to the work of Shafahi and Khani [13]. The results of running the proposed model with the objective function of minimizing total transfer waiting time are shown in Tables 3 and 4. These results indicate that maximizing simultaneous arrivals does not lead to minimum waiting time. This happens because by increasing simultaneous arrivals in one station, number of zero waiting times will be increased in that station, but this event does not necessarily lead to less waiting time at other stations.

Tab. 3. Comparative result based on Ceder et al. [12]

	Example 1		Example 2		Example 3	
	Our proposed model	Ceder et al. [12]	Our proposed model	Ceder et al. [12]	Our proposed model	Ceder et al. [12]
Total waiting time (minutes)	21	21	30	46	62	66
Simultaneous arrivals	8	4	14	7	12	6

The fourth example of Ceder et al. [12] is a real network in Tel Aviv in which the purpose is scheduling the fleet in a 2-hour and 54-minute time interval. Table 4 shows the result of our proposed model on this example in which the best obtained solution is reported after two hours.

This result shows that our developed model can effectively schedule the fleet, but its running time is highly encouraging us to develop a metaheuristic algorithm to find a good solution in a reasonable time.

Tab. 4. Comparative results based on the fourth example of Ceder et al. [12]

	Ceder et al. [12]	Shafahi and Khani [13]	The proposed model
Total waiting time (minutes)	5340	1176	288
Simultaneous arrivals	240	257	378

4-2. Evaluation of the scatter search performance

In order to evaluate the performance of our developed SS algorithm, we used Tehran Urban Railway Network (TURN) shown in Fig. 7. The TURN consists of five different routes and eight transfer intersections that totally contains 27 different transfer stations.

The purpose of our proposed model is to reschedule the fleets at the peak hours of an ordinary day, from 5:30 to 8:00 am. A number of

available metro trains in each route and current headways at the peak hours are given in Tab. 5. It should be mentioned that parameters N_i and ST_{ia} are determined for the considered time period based on the current schedule of the TURN. More information about the structure of the TURN and its current schedule is retrieved from Tehran Urban and Suburban railway Operation Co. website [20].

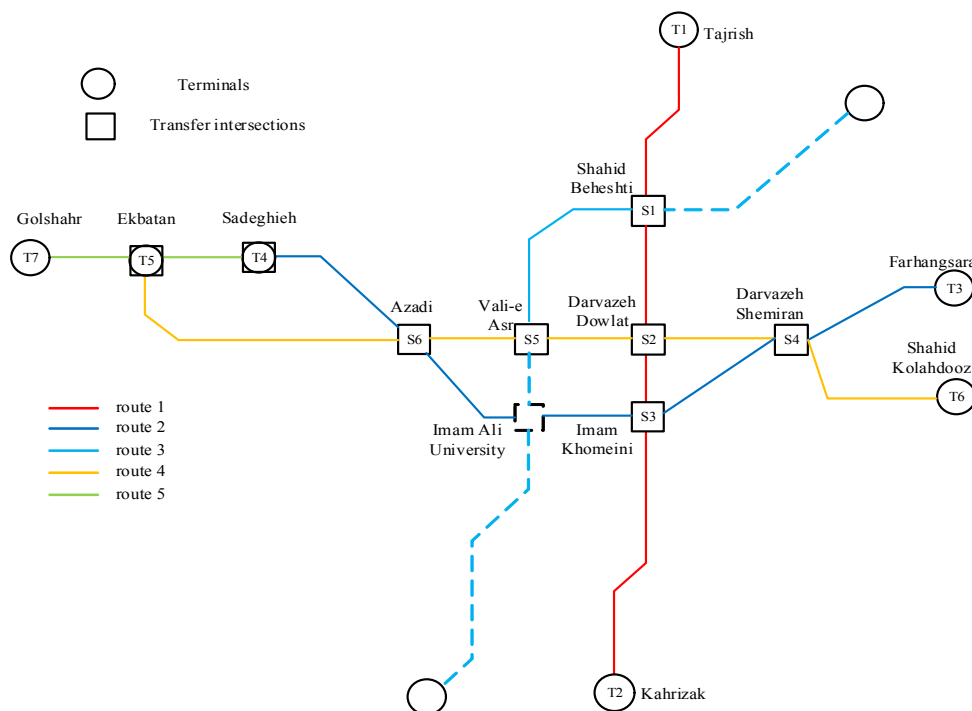


Fig. 7. Tehran Urban Railway Network

Tab. 5. Number of metro trains and headways of TURN

	Number of Metro trains	Headways at the peak hours
route 1	32	4
route 2	24	4
route 3	4	10
route 4	14	7
route 5	16	10

In order to apply our developed SS, we should set five parameters of this algorithm. For this purpose, we consider three values for each parameter, determined by experimental analysis, given in Tab. 6. The best obtained values for

these parameters are found by L27 matrix of Taguchi method, leading to $|P| = 240$; $b_1 = 10$; $b_2 = 12$; $a_1 = 0.3$; $a_2 = 2$; $r = 2$

Tab. 6. Possible values for parameters

	$ P $	b_1	b_2	a_1	a_2	r
Level 1	120	8	8	0.3	1.3	2
Level 2	180	10	10	0.6	1.6	6
Level 3	240	12	12	1	2	10

The results of implementing our proposed model using CPLEX and SS algorithm are indicated in Table. 7. Due to the unreasonable run time of the CPLEX for solving the TURN, we restricted our

report to 24 hours CPU run time. The best solution obtained by using the SS algorithm is also reported based on one-hour CPU run time, considered as its termination criterion.

Table. 7. Comparative results for the TURN

	Current Schedule	Schedule obtained using CPLEX	Schedule obtained using SS
Objective value (minutes)	153273.8	89797.4	66749
Total transfer waiting time (minutes)	60273.8	73897.4	43699
Number of non-served passengers	1860	849	461

The objective value of the current schedule of TURN includes the total waiting time and penalty of non-served passengers. As a sample of timetable, Fig. 8 shows an example timetable obtained by taking the schedule of the trains on route number 2 during the considered time period using the SS algorithm. With respect to Fig. 7, each train of route number 2 should serve eight transfer stations on its route at both directions. For instance, the first train of route number 2 in the first direction should serve S4, S8, S6, T4, S6, S8, S4 and T3 stations, respectively, at service cycles. Arrival times of the first train to the stations of its first and second service cycles are shown in Fig. 8 (each station is shown by a different symbol); the first train will visit S4 station at the second service cycle after 138 minutes (at 7:48 am). Since we have 24 trains in

this route, twelve of them should start their services from return direction. Thus, the first train of route number 2 in return direction serves S6, S8, S4, T3, S4, S8, S6, and T4 stations, respectively, at service cycles. Arrival time of this train to S6 station is 9 minutes after the start time (at 5:39 am).

As indicated in Table 7, the best obtained solutions from CPLEX and SS for the TURN are 89797.4 and 66749, respectively. These results show that our developed model and algorithm can well improve the current situation. Moreover, the performance of SS is much better than that of CPLEX. Although the number of served passengers in the schedule obtained from SS approach has increased to around 75% compared with the current schedule, the total waiting time has decreased around 27%.

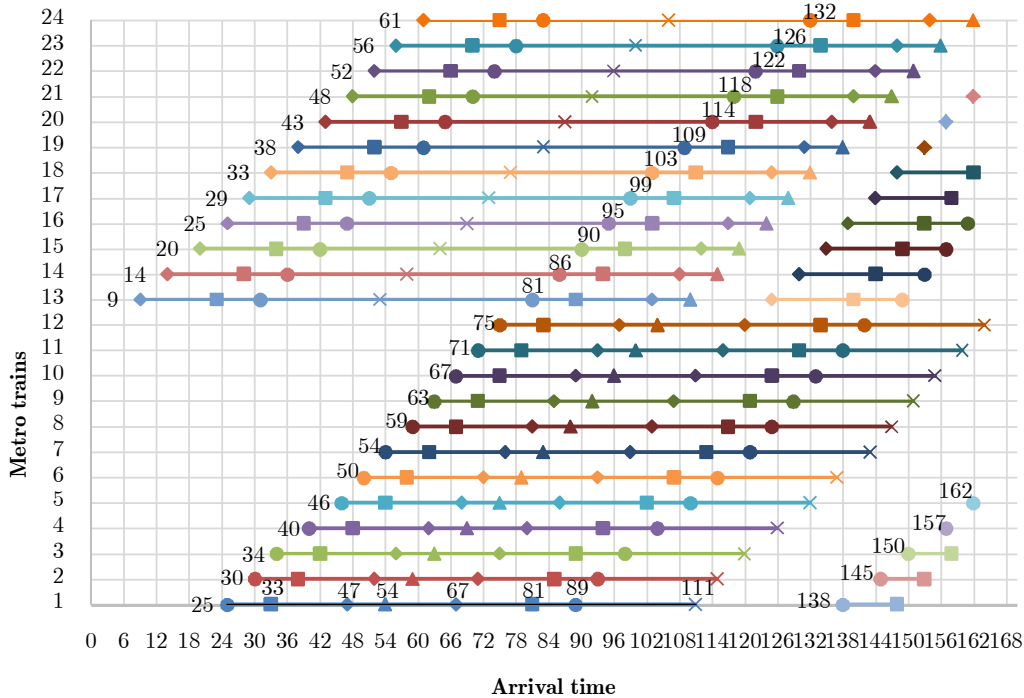


Fig. 8. Metaheuristic timetable of metro trains in route number 2 during the considered time period

5. Conclusion

In this paper, we formulated a public transport fleet scheduling problem as a mixed integer linear programming model in order to minimize the total transfer waiting time. For large networks, it is not possible to achieve optimal solutions using ILOG CPLEX in an acceptable CPU run time. Therefore, we developed a scatter search algorithm to obtain good solutions for real-life networks. The performance of our developed model has been analyzed based on the test instances of Ceder et al. [12], Shafahi and Khani [13] and Tehran Urban Railway Network. We show our results and model using 8 figures and 7 tables. Our solution approach shows great improvement compared with the current schedule of TURN. Since the number of available vehicles in each route has a direct effect on headways and waiting times, performance of the model could be improved even more with also considering the number of vehicles as decision variables. Transit scheduling is an operational decision, which has to be done several times in a day. Therefore, developing a heuristic method for achieving the best solutions in a short time and also an optimal number of vehicles are important questions to investigate in future studies.

References

- [1] Haghani, A., Y. Shafahi, *Bus maintenance systems and maintenance scheduling: model formulations and solutions*. Transportation Research: Part A, Vol. 36, (2002), pp. 453–482.
- [2] Chakroborty, P., T. Wivedi, *Optimal Route Network Design for Transit Systems Using Genetic Algorithms*. Engineering Optimization, Vol. 34, (2002), pp. 83–100.
- [3] Fan, L., C.L. Mumford, *A metaheuristic approach to the urban transit routing problem*. Journal of Heuristics, Vol. 16 (2010), pp. 353–372.
- [4] Nikolić, M., D. Teodorović, *Transit network design by Bee Colony Optimization*. Expert Systems with Applications, Vol. 40, (2013), pp. 5945–5955.
- [5] Farahani, R.Z., E. Miandoabchi, W.Y. Szeto, H. Rashidi, *A review of urban transportation network design problems*. European Journal of Operational Research, Vol. 229, (2013), pp. 281–302.
- [6] Lusby, R.M., J. Larsen, M. Ehrgott, D. Ryan, *Railway track allocation: models and*

- methods*. OR Spectrum, Vol. 33, (2011), pp. 843-883.
- [7] Desaulniers, G., M.D. Hickman, M.D. *Public Transit*. In Public Transition, (2007), pp. 69-127.
- [8] Guihaire, V., J.K. Hao, *Transit network timetabling and vehicle assignment for regulating authorities*. Computers and Industrial Engineering, Vol. 59, (2010), pp. 16-23.
- [9] Ibarra-Rojas, O.J., F. Delgado, R. Giesen, J.C. Munoz, *Planning, operation, and control of bus transport systems: A literature review*. Transportation Research: Part B, Vol. 77, (2015), pp. 38-75.
- [10] Yu, B., Z. Yang, X. Sun, B. Yao, Q. Zeng, E. Jeppesen, E. *Parallel genetic algorithm in bus route headway optimization*. Applied Soft Computing, Vol. 11, (2011), pp. 5081-5091.
- [11] Guihaire, V., J.K. Hao, *Transit network design and scheduling: A global review*. Transportation Research: Part A, Vol. 42, (2008), pp. 1251-1273.
- [12] Ceder, A., B. Golany, O. Tal, *Creating bus timetables with maximal synchronization*. Transportation Research: Part A, Vol. 35, (2001), pp. 913-928.
- [13] Shafahi, Y., A. Khani, *A practical model for transfer optimization in a transit network: Model formulations and solutions*. Transportation Research: Part A, Vol. 44, (2010), pp. 377-389.
- [14] Anupam, O., D. Barun, K.M. Shyamal, M. Manoranjan, *A transportation problem with fuzzy-stochastic cost*. Applied Mathematical Modelling, Vol. 38, (2014), pp. 1464-1481.
- [15] Liu, T., A. Ceder, *Integrated Public Transport Timetable Synchronization and Vehicle Scheduling with Demand Assignment: A Bi-objective Bi-level Model Using Deficit Function Approach*. Transportation Research Procedia, Vol. 23, (2017), pp. 341-361.
- [16] Bunte, S., N. Klierer, *An overview on vehicle scheduling models*. Public Transport, Vol. 1, (2009), pp. 299-317.
- [17] Daraio, C., M. Diana, F. Costa, C. Leporelli, G. Mateucci, A. Nastasi, *Efficiency and effectiveness in the urban public transport sector: A critical review with directions for future research*. European Journal of Operational Research, Vol. 248, (2016), pp. 1-20.
- [18] Ciancio, C., D. Lagana, R. Musmanno, F. Santoro, *An integrated algorithm for shift scheduling problems for local public transport companies*, Omega, In Press, (2017).
- [19] Laguna, M., R. Marti, *Scatter Search - Methodology and Implementations in C 1st ed.*, C. Kluwer Academic Publishers, Boston, (2003).
- [20] Tehran Urban and Suburban railway Operation Co., Tehran Metro line schedule. Available at: <http://metro.tehran.ir/Default.aspx?tabid=304> [Accessed May 28, 2014].

Follow This Article at The Following Site

Nasirian F, Ranjbar M. Public transport fleet scheduling for minimizing total transfer waiting time. IJIEPR. 2017; 28 (2) :113-128 URL: <http://ijiepr.iust.ac.ir/article-1-721-en.html>
DOI: 10.22068/ijiepr.28.2.113

