ASM Sc. J., 12, Special Issue 1, 2019 for IQRAC2018, 150-173

Multiple Tabu Search for Multiobjective Urban Transit Scheduling Problem

Vikneswary Uvaraja[1], Lai Soon Lee [1,2,*], and Nor Aliza Ab. Rahmin[1]

[1] *Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.*
[2] *Laboratory Computational Statistics and Operations Research, Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.*

**Corresponding author: lls@upm.edu.my*

The study of urban public transportation is essential for building an efficient transit system that can minimize the traffic congestion, reduce pollutions and increase the mobility of a community. Urban Transit Scheduling Problem (UTSP) considers the process of creating timely transit schedules that includes bus and drivers assignment based on the users' and operators' requirements. It is necessary to achieve a tradeoff between the interest of users and operators which lead to multiobjective nature of UTSP. This paper studies multiobjective UTSP consisting of frequency setting, timetabling, simultaneous bus and driver scheduling by applying a Multiple Tabu Search (MTS) algorithm. In addition, a multiobjective set covering model is also adapted by including some real-world restrictions to find adequate number of buses and drivers. The MTS algorithm is tested on benchmark instances from Mandl's Swiss Network. The computational results shown that the algorithm able to produce comparable results for most cases from the literature.
Keywords: Urban Transit, Scheduling, Multiobjective, Tabu Search.

I. Introduction

The development of urban public transit technology has major impact on the size of a city and its population and vice versa. The lack of systematic public transportation system will always lead to various conflicts such as traffic congestion and air pollution. Nevertheless, it is often difficult to improve the public transport system since many criteria need to be taken into consideration which includes passenger's and operator's preferences. It is necessary to include both of their perspectives in order to increase the ridership with minimal use of resources. Some of the major disadvantages of urban public transportation are longer waiting time and its non-availability to destination at desired time.

This type of problem can be classified as Urban Transit Scheduling Problem (UTSP) which involves the process of deriving optimal schedules for buses and drivers with respect to the passenger's demand and operator cost. UTSP can be divided into frequency setting, timetabling, vehicle and crew scheduling (Ceder, 2002). Generally, these problems are tackled consecutively due to its complexity. UTSP is a complex NP-hard problem and many approaches have included mathematical models to formulate the problem and solve it quantitatively through optimization. Recently, researchers are turning to the study of multiobjective UTSP which is related to the interaction of passengers' preferences and the industry regulation to provide more choices for decision makers. The process of finding a best solution from the perspective of both stakeholders is quite difficult as it depends on the requirement of decision makers.

Metaheuristic have been a popular method for solving this hard combinatorial optimization problem due to its capability of finding near optimal solutions in reasonable time (Nikolić and Teodorović, 2014). However, there is no such application of Multiple Tabu Search (MTS) in multiobjective UTSP are known to date. Thus, MTS with systematic neighborhood selection approach is developed by modifying the initialization process and incorporating intensification and diversification to find optimal solutions for solving multiobjective UTSP. The proposed algorithm is verified and validated using benchmark data sets to generate competitive solutions with previously published results.

In this study, a multiobjective mixed integer programming model that the minimize number of buses, total waiting times and overcrowding is formulated to find suitable frequency for each of the routes studied. The model is further extended including timeslots to set the frequencies during peak and off-peak hours throughout the time period. On the other hand, a set covering model is adapted from Zuo et al. (2015) to minimize the number of buses and drivers simultaneously for constructing optimal schedules.

Over the years, UTSP has been solved by different approaches based on the problem's complexity and the development of computer technology. At the early stage, researchers have tackled only the frequency or headways optimization problem. Beginning from 1976, timetabling problem has been introduced to produce efficient schedules and followed by vehicle and crew scheduling approach. Due to the scope of this paper, only the application of TS in UTSP is dicussed in this section.

Cavique et al. (1999) discussed two heuristic approaches for crew scheduling problem to minimize the crews required to cover a predefined timetable under contractual rules. The first algorithm uses strategic oscillation procedure and the second applied block partitions and matching algorithm in Tabu subgraph ejection chain where the latter gives better performance. Gomes et al. (2006) extended it to balance the number of drivers and cover crews since the non-driving periods are usually longer than total duty time. A Lisbon Underground case study is used to test effectiveness of the TS algorithm.

The first study to consider multiobjective bus driver scheduling problem based on TS and Genetic Algorithm (GA) is Lourenço et al. (2001). A TS with three types of neighbourhood selection and optimized intensification strategy is applied to find the best solutions. Based on the results, the TS outperformed the other methods as linear programming in term of cost, time and quality. Moreover, Ruisanchez and Ibeas (2012) constructed a bi-level optimization model to assign optimal bus sizes and frequencies to public transport routes. The upper level problem minimizes the cost of users and operators while the lower level model solved public transport assignment model subject to capacity constraint. The basic TS algorithm can converge quickly to optimal solution than Hooke-Jeeves algorithm (HJ) when compared using real-world problem. Similarly, Giesen et al. (2016) also proposed a TS algorithm combined with aspiration plus strategy for solving multiobjective transit frequency optimization. The objectives are to minimize the total travel time and operator cost. The proposed algorithm improved the current solution in term of total travel time and fleet size.

The article is organized as follows. Section 2 explains the concept of multiobjective UTSP which includes the models formulation. Section 3 explains the development of MTS for both frequency setting and bus and driver scheduling problem. Section 4 describes the computational experiments and presents the results for benchmark data sets. Finally, Section 5 presents conclusion and suggestions for future research.

II. Multiobjective Urban Transit Scheduling Problem

UTSP is a multiobjective problem in nature as it must considers the preferences of both users and operators which are always conflicting to each other. Passengers would choose to travel quickly from their origin to destination with shorter waiting time and high comfort. On the contrary, operators would try to minimize their operational cost in term of the number of buses and drivers required which consequently might reduce the quality of service. This problem can be solved either by setting weights to the objectives according to its relative importance to obtain a single solution or handling them simultaneously to produce various solutions with different tradeoff levels.

Many models combine these objectives into single function under the resources constraints such as fleet size (Scheele, 1980; Constantin and Florian, 1995) and bus loading (Verbas and Mahmassani, 2015; Li et al., 2013). As for the second case, an optimal solution is chosen from the set of non-dominated solutions which can also be obtain by executing the single objective model consecutively using different weights of the objectives and constraints values. On the other hand, the studies on multiobjective optimization to produce Pareto optimal solutions in a single execution are increasing in order to explore different range of solutions and observe the relation between the objectives for long-term planning (Fedorko and Weiszer, 2012; Zuo et al., 2015; Prata, 2016).

A. Frequency Optimization

The frequency optimization problem can be formulated as a bilevel procedure to show the interaction between transit users and operators. Initially, the frequency is assigned based on the available resources and the users planned their travel according to the service design. Then, the transit operator improved the service frequency by observing the users' travel pattern. The first procedure describes about passenger assignment method that represents their route choice behaviour based on the Mandl's transportation network design which includes fixed demand and deterministic travel times for both direction of an origin-destination (OD) pair. The second stage discusses the frequency optimization procedure by MTS. The optimal set of frequency is used to update the initial frequency and the procedure is reiterated until convergence pattern of the frequency set is observed.

In this study, two cases are considered to find the optimal frequency for each route. The first case assumed that the demand and frequency of a route are constant throughout the time period studied whereas for the second case, the problem is extended to vary the demands between each OD pair and frequencies of the routes according to specific timeslot. The notations to represent the passenger assignment and frequency optimization models are shown below.

Two passenger assignment methods are adopted from Baaj and Mahmassani (1991) and Afandizadeh et al. (2013). Generally, a passenger can use at most two transfers to travel between their OD pair. It is expected that the passenger will first select a direct path without any transfer to reach their destination. When no direct path is available, the passengers prefer to travel by one transfer. If there is no one transfer path provided for the OD pair, the path with two transfers is chosen. The demand is usually considered unsatisfied when more than two transfers are needed for travelling. The difference between the two methods is the allocation of passengers to the routes when multiple paths of equal preference exist between their origin and destination.

d_{ij}	number of passengers travelling between nodes i and j
$d_{ij,s}$	number of passengers travelling between nodes i and j in timeslot s
f_k	frequency of route k
$f_{k,s}$	frequency of route k in timeslot s
f_min	minimum frequency for a time period
f_max	maximum frequency for a time period
l_k	layover time of route k
Q_k	maximum load (passengers) of route k
$Q_{k,s}$	maximum load (passengers) of route k in timeslot s
$t^p_{i,j}$	total travel time of a passenger between nodes i and j on path p
$t^p_{i,j,s}$	total travel time of a passenger between nodes i and j on path p in timeslot s
$t^p_{invt,ab}$	in-vehicle travel time between nodes a and b for path p
$t^p_{wt,a}$	waiting time of a passenger at node a for path p
$t^p_{tt,b}$	transfer time (penalty per transfer) of a passenger at node b for path p
$t^p_{invt,ab,s}$	in-vehicle travel time between nodes a and b in timeslot s for path p
$t^p_{wt,a,s}$	waiting time of a passenger in timeslot s at node a for path p
$t^p_{tt,b,s}$	transfer time (penalty per transfer) of a passenger in timeslot s at node b for path p
t_k	vehicle travel time of route k
u_p	utility function represent the probability for each of the path p to be chosen
CAP	seating capacity of a bus
DWP	dwell time for a passenger
LF	load factor of a bus
R	set of bus routes
R_{ab}	set of potential routes between node a and b
T	time horizon
TS	set of timeslots
N	set of nodes in transit network

Based on Baaj and Mahmassani (1991), when direct paths are available between OD pair, the in-vehicle travel time of each route is computed and a filtering process is invoked such that any route with in-vehicle travel time more than 50% (or a pre-specified threshold) of the minimum value is rejected.Then, the demands are distributed to the surviving routes using ‘frequency share’ rule: a route carries a part of the flow equal to the ratio of its frequency to the sum of the frequencies of all acceptable routes.

If multiple one transfer or two transfer routes are found, the total travel time of a passenger at each possible path that sum up the in-vehicle travel time, waiting time and transfer time between origin, i and destination, j is calculated. Equations (1) and (2) find the total travel time for one and two transfers respectively. The passenger's waiting time at the origin and transfer node are assumed to be half of the headway of a route k. The transfer nodes are defined as n and n' in the equations.

Next, a filtering procedure similar to the zero-transfer is applied such that all paths whose total travel time are higher than 10% (or a pre-specified threshold) of the minimum value offered by any paths between that pair of nodes are rejected. The detail explanation of this passenger assignment process is stated in Nikolić and Teodorović (2014).

Based on Afandizadeh et al. (2013), if multiple direct paths exist, each path has a possibility to be chosen based on its route frequency. Frequency share rule is applied to allocate the demands based on the respective frequency of each route on the chosen paths. Alternatively, the passengers are assigned according to the total travel time utility in case there are multiple paths exist with one or two transfers from the origin to destination. The travel time utility is calculated from the formula of Logit model as shown in equation (3) where the total travel time is equal to equation (1) or (2) according to the number of transfer.

$$\begin{aligned} t^p_{i,j} &= t^p_{invt,ij} + t^p_{wt,ij} + t^p_{tt,ij}, \\ &= [t^p_{invt,in} + t^p_{invt,nj}] + [t^p_{wt,i} + t^p_{wt,n}] + t^p_{tt,n}, \\ &= [t^p_{invt,in} + t^p_{invt,nj}] + [(\frac{T}{2\sum_{k\in R_{in}} f_k})^p \\ &+ (\frac{T}{2\sum_{k\in R_{nj}} f_k})^p] + t^p_{tt,n}. \end{aligned} \tag{1}$$

$$\begin{aligned} t^p_{i,j} &= t^p_{invt,ij} + t^p_{wt,ij} + t^p_{tt,ij}, \\ &= [t^p_{invt,in} + t^p_{invt,nn'} + t^p_{invt,n'j}] \\ &+ [t^p_{wt,i} + t^p_{wt,n} + t^p_{wt,n'}] + [t^p_{tt,n} + t^p_{tt,n'}], \\ &= [t^p_{invt,in} + t^p_{invt,nn'} + t^p_{invt,n'j}] \\ &+ [(\frac{T}{2\sum_{k\in R_{in}} f_k})^p + (\frac{T}{2\sum_{k\in R_{nn'}} f_k})^p \\ &+ (\frac{T}{2\sum_{k\in R_{n'j}} f_k})^p] + [t^p_{tt,n} + t^p_{tt,n'}]. \end{aligned} \tag{2}$$

$$u_p = \frac{e^{-(t^p_{ij})}}{\sum_{p \in P} e^{-(t^p_{ij})}}. \tag{3}$$

This procedure is extended by including timeslots in order to differentiate peak and off-peak hours. During the assignment process, the passengers are allocated based on the frequencies of the routes at that time period. First, the frequencies of the routes and the demand for every OD pair are divided proportionally according to peak and off-peak hours. The demands on the peak hours are assumed to be double the demands on off-peak hours.The total travel time of a passenger at each timeslot for every possible path from their origin to destination is calculated. Likewise, equations (4) and (5) are applied to find the total travel time of the path with one and two transfers respectively.

$$\begin{aligned} t^p_{i,j,s} &= t^p_{invt,ij,s} + t^p_{wt,ij,s} + t^p_{tt,ij,s}, \\ &= [t^p_{invt,in,s} + t^p_{invt,nj,s}] + [t^p_{wt,i,s} + t^p_{wt,n,s}] \\ &+ t^p_{tt,n,s}, \\ &= [t^p_{invt,in,s} + t^p_{invt,nj,s}] + [(\frac{T}{2\sum_{k \in R_{in}} f_{k,s}})^p \\ &+ (\frac{T}{2\sum_{k \in R_{nj}} f_{k,s}})^p] + t^p_{tt,n,s}. \end{aligned} \tag{4}$$

$$\begin{aligned} t^p_{i,j,s} &= t^p_{invt,ij,s} + t^p_{wt,ij,s} + t^p_{tt,ij,s}, \\ &= [t^p_{invt,in,s} + t^p_{invt,nn',s} + t^p_{invt,n'j,s}] \\ &+ [t^p_{wt,i,s} + t^p_{wt,n,s} + t^p_{wt,n',s}] \\ &+ [t^p_{tt,n,s} + t^p_{tt,n',s}], \\ &= [t^p_{invt,in,s} + t^p_{invt,nn',s} + t^p_{invt,n'j,s}] \\ &+ [(\frac{T}{2\sum_{k \in R_{in}} f_{k,s}})^p + (\frac{T}{2\sum_{k \in R_{nn'}} f_{k,s}})^p \\ &+ (\frac{T}{2\sum_{k \in R_{n'j}} f_{k,s}})^p] + [t^p_{tt,n,s} + t^p_{tt,n',s}]. \end{aligned} \tag{5}$$

All routes are initialized with similar frequency before the passenger's demand is assigned based on the route choice. After the passenger assignment process, the maximum load of each route k is obtained from its list of link flows and used as the input for the frequency optimization procedure. The objectives are the minimization of total number of buses, passengers waiting time and overcrowding in the bus. Most of the papers in the literature have used these objectives for optimizing the frequency. The passenger assignment procedure is conducted to find the total waiting time of all passengers.

minimize

$$F_1 = \sum_{k \in R} [\frac{2t_k f_k}{T}], \tag{6}$$

$$F_2 = \sum_{i \in N} \sum_{j \in N} [d_{ij} \frac{T}{2\sum^{k \in R_{ij}} f_k}], \tag{7}$$

$$F_3 = \sum_{k \in R} [Q_k - (CAP(LF) f_k)], \tag{8}$$

subject to

$$f_{min} \leq f_k \leq f_{max} \quad \text{for all} \quad k \in R. \tag{9}$$

Equation (6) calculates the number of buses needed for each route that obtained by dividing the total round trip time with the time horizon. Equation (7) measures the total waiting time for all the passengers. Equation (8) determines the total number of passengers that exceed the maximum capacity of the bus. Equation (9) ensures that frequency of each route is within the lower and upper boundary value. Consequently, the previous model is further extended by incorporating timeslots in order to find the frequency of a route in a specific time period based on the variable demands as follows:

minimize

$$F_1 = \sum_{k \in R} [max_{s \in TS}(\frac{2t_k f_{k,s} + Q_{k,s} DWP + l_k f_{k,s}}{T})], \tag{10}$$

$$F_2 = \sum_{i \in N} \sum_{j \in N} \sum_{s \in TS} [d_{ijs} \frac{T}{2\sum_{k \in R_{ij}} f_{k,s}}], \tag{11}$$

$$F_3 = \sum_{k \in R} \sum_{s \in TS} [Q_{k,s} - (CAP(LF) f_{k,s})], \tag{12}$$

subject to

$$f_{min} \leq f_k \leq f_{max} \quad \text{for all} \quad k \in R. \tag{13}$$

Equation (10) determines the maximum number of buses needed among all the timeslots for each route. The round trip time includes dwell time and layover time. Equation (11) measures the total waiting time for all the passengers in every timeslot while equation (12) calculates the total number of passengers that exceed the maximum capacity of the bus in all timeslots. Equation (13) shows the constraint on the frequency of each route.

B. Bus and Driver Scheduling

The proposed procedure for solving these problems is inspired from Zuo et al. (2015) who tackled the vehicle scheduling problem. The solution approach is revised by incorporating the elements for bus driver scheduling. It is assumed that the drivers are assigned to the same bus throughout their working time to ensure systematic assignment process. A driver is assigned to the departure times based on several work load rules such as total work duration, maximum working period without break and maximum break duration which are set as follows with its value in the bracket:

D_{break}: maximum break duration (1 hour),
D_{work}: maximum working duration without break (4 hours),
D_{work}: total work duration (9 hours).

Each route has two control points (CP_1 and CP_2) which represent the first and last nodes of the route at where the drivers can take a break for a specific duration. There are two scenarios considered in this study for solving bus and driver scheduling problem. The first scenario favors passengers requirement by assigning equal departure times to both CP_s while the second scenario consider operators preference by allocating the time at CP_1 only. The solution procedure for both scenario is similar except the formation of blocks varies when choosing the departure times.

The scheduling process can be subdivided into three stages. Initially, a set of blocks is produced to cover all the departure times. Then a subset of candidate blocks is selected from them which minimize the objective functions. Later, all the candidate blocks are reconstructed to further reduce the objective functions values. The process is repeated for all the routes. Let A_1^k and A_2^k be the set of departure time of the route k from both CP_1 and CP_2 respectively. The set of independent blocks are grouped as B^k where q is the total number of blocks and b_y^k is the yth block of B^k.

By referring to Figure 1, let trip 1 starts from a_x at CP_1 and ends with w_x at CP_2. For the first scenario, the departure time of trip 2 will be t_b which is the next available time in the list of A_2^k after $w_x + O_k^s$, where O_k^s is the one-way travel times of route k at timeslot s that includes layover time and dwell time. Meanwhile, for the second case, t_a is the departure time for trip 2 such that $t_a = w_x + O_k^s$. Note that, $t_b \geq t_a$. The next trip also continues to find the time in similar manner to complete the block. Moreover, the drivers break time also included in the block after D_{maxb} without considering the location.

Based on Figure 2, suppose w_x is the departure time after several trips and $w_x - a_x \geq D_{maxb}$, then the next trip must be equal or higher than $w_x + D_{break}$. Likewise, t_b and t_a are the succeeding trip for scenario 1 and 2 respectively. The maximum duration of a block is always checked before choosing the consequent trips. The construction of a block terminates if the departure time of the trip is more than D_{work} for a short block and $2D_{work}$ for a long block.

A set covering model is altered to represent the multiobjective problem and evaluates the objective values in order to produce optimal blocks. One of the advantages of set covering formulation is that it can generate a set of buses that assigned to different trips based on the travel times such that all the trips are covered with minimum cost. Moreover, it allows some of the trips to be covered by more than one bus which consequently provides

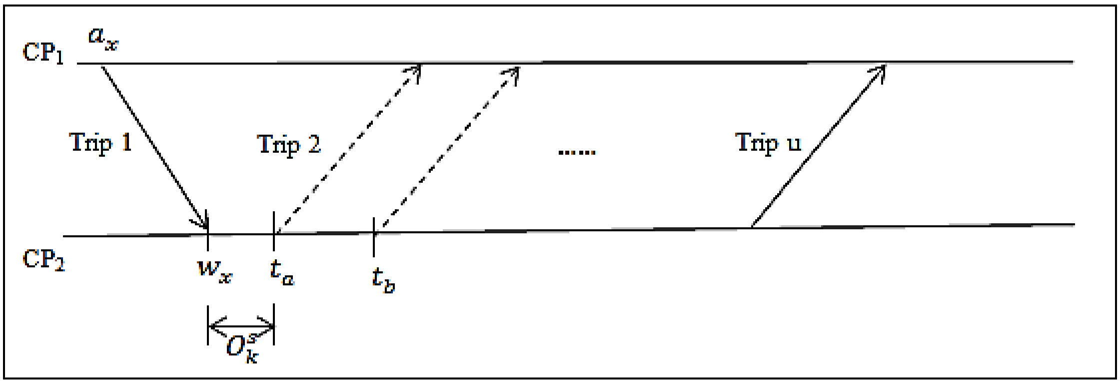

Figure 1: Construction of a Vehicle Block

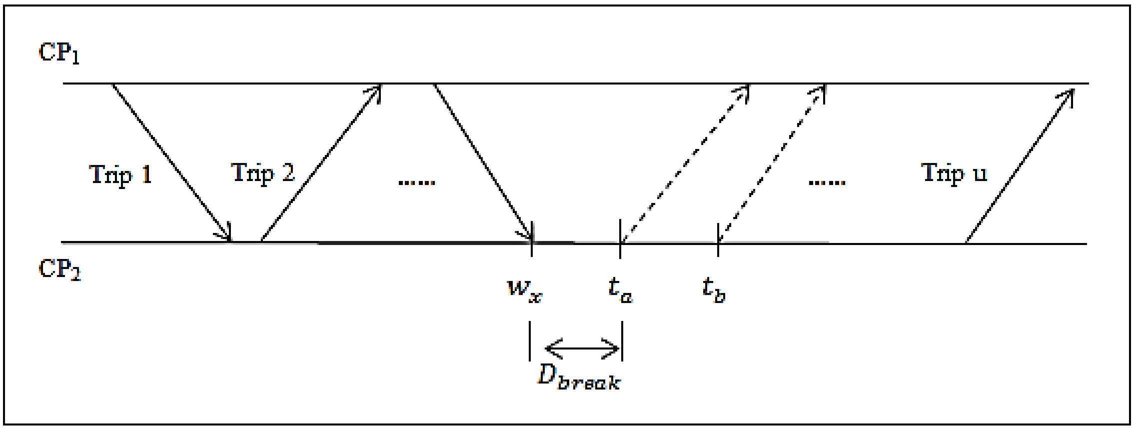

Figure 2: Choosing Departure Times after Break

various combinations of trips for each bus. The set covering model for bus and driver scheduling is presented as follows:

minimize

$$F_l = \sum_{y=1}^{q} z_y, \tag{14}$$

$$F_2 = \sum_{y=1}^{q} C_y z_y, \tag{15}$$

subject to

$$\sum_{y=1}^{q} v_{xy} z_y \geq 1, \tag{16}$$

$z_y = \{0,1\}$ $\quad C_y = \{1,2\}$ $\quad v_{xy} = \{0,1\}$ $\quad y = 1, ..., q.$

where,

$$v_{xy} = \begin{cases} 1,\text{if block } y \text{ has a trip starting from } x\text{th} \\ \quad \text{departure time} \\ 0,\text{otherwise,} \end{cases}$$

$$z_y = \begin{cases} 1,\text{if block y is in the solution} \\ 0,\text{otherwise,} \end{cases}$$

$$z_y = \begin{cases} 1,\text{if } b_y \text{ is a short block} \\ 2,\text{if } b_y \text{ is a long block.} \end{cases}$$

Equations (14) and (15) minimize the number of buses and drivers respectively and equation (16) ensures that every departure time is covered by at least one trip. v_{xy} is a binary matrix that records the availability of departure time in a block whereas the binary decision variable z_y shows the presence of blocks in the solution. The parameter C_y defines the number of drivers for each block.

The proposed solution approach differs from the literature as it does not alter the existing departure times in the optimal blocks to include more times that are unable to cover in the first phase. Despite that, this research generates extra vehicle blocks for the time that cannot be covered in the blocks created from initial departure times. This is because the adjustment of departure times may affect the headway and layover time determined earlier which consequently have an impact on robustness of the schedule.

III. Multiple Tabu Search

MTS has been first developed by Pothiya et al. (2006) to design the optimal fuzzy logic proportional integral (PI) controller. The conventional TS method might require longer computational time to find expected solution if the initial solution is further from the promising space. Thus, MTS algorithm highlights the problem and helps to guide the search to the optimal region in less computational time.

MTS algorithm begins by generating several initial solutions to increase the possibility of reaching the optimal region quickly. Then, adaptive search mechanism is applied to alter the step size of the neighborhood accordingly during the search and multiple TS algorithms are performed sequentially according to its initial solution. Consequently, the new starting solutions for next iteration are obtained by crossover mechanism. The process is restarted when the search reached local optimal solution and stopped after the termination criterion is satisfied.

In this study, the proposed MTS algorithm functions distinctly for discrete optimization (bus and driver scheduling) and continuous optimization (frequency setting) problem. The performance of MTS is greatly depends on the initial solution as it affects the computational time for the solution to converge. The proposed MTS algorithm works with multiple initial solutions such that each of them is selected from different feasible domain. Explicitly, the search space is divided into a number of domains and each of the domains is allocated to different range of values. This can help to speed up the search and examine the search space precisely to find better solutions. We used variable precision value to create a flexible neighborhood structure since constant step size might not be able to move the current solution efficiently. So, the adaptive search mechanism is applied to find step sizes for locating the neighborhood of the current solution.

The use of single or multiple tabu lists depends on the types of move involved during the search that directly influenced by type of the problem. There are several elements added or dropped at the same time to create the neighborhood of current solution. Therefore, this study applied two-dimensional tabu lists with same tabu tenure to record the elements with their positions in the list. This approach inhibits repeated moves and enables the search to explore variety of solution using organized memory structure.

Moreover, intensification is executed if there is no promising solution available and diversification occurs when intensification is not possible to be implemented. This is because the normal search procedure in the proposed MTS is good enough to analyze a portion of the whole neighborhood since it is divided earlier. Additionally, there is no need to spend extra time to examine the region that already visited previously. On the other hand, aspiration criteria allows tabu moves if it yields the best results and termination criteria stop the search if there is no progress in the objective values after a fixed number of iterations.

A. MTS for Frequency Optimization

Initially, the range of frequency for each route is divided into m number of domains using equation (17) and the procedure to set the individual range is explained as follows:

$$dif = \frac{f_{max} - f_{min} + 1}{m}. \tag{17}$$

Step 1: Let $k = 1$.
Step 2: Set $f_{1,min} = f_{min}$ and $f_{m,max} = f_{max}$. Find the interval dif for the route k.
Step 3: Let $n = 1$, calculate $f_{n,max}$ $=f_{n,min} + dif - 1$ and $f_{n+1,min} = f_{n,max} + 1$.
Step 4: Repeat Step 3 until $n = m - 1$.
Step 5: Repeat Steps 2-4 for all route $k \in R$.

Each of the routes is assigned to the random frequencies within their boundaries. The initial solutions are represented as a vector of $X_n^0 = f_{n1}^0, f_{n2}^0, \cdots, f_{mw}^0$, such that w is the number of routes. The neighborhoods of the current solutions are formed by increasing and decreasing the frequencies based on the step size values which are obtained randomly by Equation (18). The step size reduces as the number of iteration increases. This approach can produce more accurate solution in minimal computational time. Let $\triangle_n^0 = \triangle_{n1}^0, \triangle_{n2}^0, \cdots, \triangle_{mw}^0$ represents the vector of step size for the multiple solutions.

$$\triangle_n^0 = K \times rand() \times (f_{n,max} - f_{n,min}). \tag{18}$$

The weight factor, K is calculated as follows:

$$K = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times iter, \tag{19}$$

where, rand() is a random value between the interval (0,1], w_{max} is the maximum weight, w_{min} is the minimum weight, $iter_{max}$ is the maximum iteration and $iter$ is the current iteration. In this study, w_{max}=1.0, w_{min}=0.2 and $iter_{max}$=100.

Each solution in the neighborhood is checked for its feasibility, tabu restriction and dominance respectively. It is kept in a set of non-dominated solutions if it satisfy all the criteria. The set is always updated by removing the worse solutions each time after a solution is added. Then, a new solution is randomly selected from the set to be assigned as next current solution. All other non-dominated solutions are stored in intermediate-term memory for intensification. When there is no dominated solution available, intensification process is conducted by choosing a solution from intermediate-term memory. Alternatively, if the intermediate-term memory is empty, the least worst solution is selected which means if some of the objective values of the trial solution dominate the corresponding values of the current solution while some are not. Otherwise, diversification process is initiated to find the next current solution at under-explored areas by equation (20). Similarly, when there is no feasible solution in the neighborhood, the search is restarted from the feasible region. The MTS is reiterated until there is no improvement in the best known solution for certain consecutive iterations. The main framework of the proposed MTS for frequency optimization is shown in Algorithm 1.

$$\text{Current solution} = (2 \times \text{trial solution}) - \text{current solution}. \tag{20}$$

Once the frequency of each route is obtained, the headway at each timeslot of a route is calculated by dividing the time period studied with the route's frequency at a timeslot which consequently used to determine the departure times.

B. MTS for Bus and Driver Scheduling

This scheduling procedure is conducted consecutively based on the number of routes. The pseudocode of the proposed MTS for bus and driver scheduling is given in Algorithm 2. At first, all the departure times are divided into several domains to start the MTS. The initialization process begins by setting all the departure times within the domain to state 0, and then a departure time, with state 0 is randomly selected from the set. A vehicle block b_y that can cover the time is also randomly chosen and every departure time in the block is assigned to state 1. The process continues until all the departure times are covered and the list of blocks selected are denoted as the initial solutions. The initial solution can be represented as $X^0 = b_1^0, b_2^0, \cdots, b_c^0$

such that each vehicle block covers a subset of departure times, $b^0 = h_1^0, h_2^0, \cdots, h_e^0$. X is denoted as a collection of sets (vehicle blocks), c indicates number of blocks included and e indicates the total departure times covered by each block.

Three types of neighborhood moves namely add, drop and swap are considered for finding the possible solutions. A block is allowed to be inserted if it covers a departure time that is not available in the current solution. The unavailability of acceptable solutions in the neighborhood and intermediate-term memory induce the diversification process. A new solution is formed by swapping a block from the least worst solution with another block which is rarely used and not in the set. This allows us to introduce new variation in the set and find good solutions. If neither of the solutions in the neighborhood covers all the departure times, the constraint handling procedure is developed to modify the neighborhood and generate feasible solutions. For each solution in the neighborhood, the blocks that contains the uncovered departure times are added into the solution together with another new block which is not included in the current solution and this trial solution.

Finally, all the solutions from each domain are collected and since each departure time can be covered by more than one block, its replicates are removed from any blocks randomly. This reduces some trips in the blocks but it creates a free time between the consecutive trips and cause imbalance in the utilization of the buses. Thus, all the blocks are reconstructed to minimize further the number of buses and drivers and also maintain the continuity of service for all buses during their working period. Let b_y be the first block in the solution X, the procedure is applied as follows:

Step 1: Check whether the minimum working time for the block b_y is achieved. If yes, continue to next block and restart this procedure. Else, go to Step 2.

Step 2: Trips from other blocks are inserted into the appropriate segments of the current block by considering the round-trip time and meal break time of the drivers. The trips are added from various blocks until the maximum working time of the block is fulfilled. Let $y = y + 1$.

Step 3: If $y = c$, then stop; otherwise, go back to Step 1.

IV. Computational Experiments

The computational experiments are conducted for both frequency optimization and bus and driver scheduling problem.

A. Experimental Design

The efficiency of the proposed MTS algorithm is tested on a transit network from Mandl's Swiss Network. It represents a small transit network with 15 nodes, 21 undirected edges and 15570 total passengers demand (see Figure 3). The MTS algorithm is coded in ANSI-C language and executed on 2.30 GHz Intel(R) Core(TM) i3-2350M CPU with 2GB of RAM under Windows 7 operating system. The parameters for UTSP and MTS used in this study are given in Table 1. A computational experiment is conducted for several route sets and compared with the solutions from previous literature.

B. Experimental Results

1. Bus Frequency Optimization

The effectiveness of the proposed MTS algorithm for frequency setting problem is evaluated using the following performance metrics as suggested by Arbex and da Cunha (2015).

- Total number of buses
- Total waiting times

Algorithm 1 Multiple Tabu Search for Frequency Optimization

X_i	set of respective frequencies
x	current solution
x^*	best-known solution
k	iteration
$N(x)$	neighbourhood of x
F^*	objective value of x^*
T	tabu list

BEGIN
Variables and functions declaration
Passenger assignment procedure
Divide the range of frequency into m domains
For $i = 1$ **to** m
  select an initial $x \in X_i$ and set $x^* = x, F^* = f(x^*), T = \emptyset, k = 0$.
  while there is no improvement in x^* until certain iteration.
    set $k = k + 1$ and generate $N(x)$ by increasing and decreasing the frequency based on adaptive
    search mechanism.
    evaluate every solutions in $N(x)$
      If there is no feasible solution
        do **intensification** and if intermediate term memory is empty, **restart** the process.
      else
      choose randomly the best x from an admissible subset (non-dominated solution) of $N(x)$ which
      contains non-tabu moves or moves allowed by aspiration criteria. Update the subset.
      If the subset is empty,
        do **intensification** and **if** intermediate term memory is empty, select the least worst
        solution. **If** there is none, do **diversification**.
    if $F(x) < F^*$, then set $x^* = x$ and $F^* = F(x)$. Update T and intermediate term memory.
  end while
  Record the best solution from each domain
End for
Return **BEST**
END

Algorithm 2 Multiple Tabu Search for Bus and Driver Optimization

X_i	set of respectively vehicle blocks
x	current solution (a set of blocks)
x^*	best-known solution (a set of blocks)
k	iteration
$N(x)$	neighbourhood of x
F^*	objective value of x^*
T	tabu list

BEGIN
Variables and functions declaration
Divide the set of departure times into m domains.
For $i = 1$ **to** m
  select an initial $x \in X_i$ and set $x^* = x, F^* = f(x^*), T = \emptyset, k = 0$.
  while there is no improvement in x^* until certain iteration.
    set $k = k + 1$ and generate $N(x)$ by adding, dropping and swapping the blocks based on systematic
    neighborhood search mechanism.
    evaluate every solutions in $N(x)$
      If there is no feasible solution
        do constraint handling procedure.
      else
      choose randomly the best x from an admissible subset (non-dominated solution) of $N(x)$ which
      contains non-tabu moves or moves allowed by aspiration criteria. Update the subset.
      If the subset is empty,
        do **intensification** and **if** intermediate term memory is empty, select the least worst
        solution. **If** there is none, do **diversification**.
    if $F(x) < F^*$, then set $x^* = x$ and $F^* = F(x)$. Update T and intermediate term memory.
  end while
  Record the best solution from each domain
End for
Reconstruction mechanism
Return **BEST**
END

Table 1: Parameter Configuration for UTSP

Description	Value
Transfer penalty for one transfer	5 minutes
Maximum number of transfers allowed	2 transfer/passenger
Bus capacity	50 passenger (40 seating capacity)
Load factor	1.25
Time horizon	1080 minutes (18 hours)
Minimum allowable frequency of buses on any route	18 (1 per hour)
Maximum allowable frequency of buses on any route	360 (20 per hour)
Maximum number of iteration without improvement of the solution	100
Tabu list size	2× number of routes
Number of domains	10

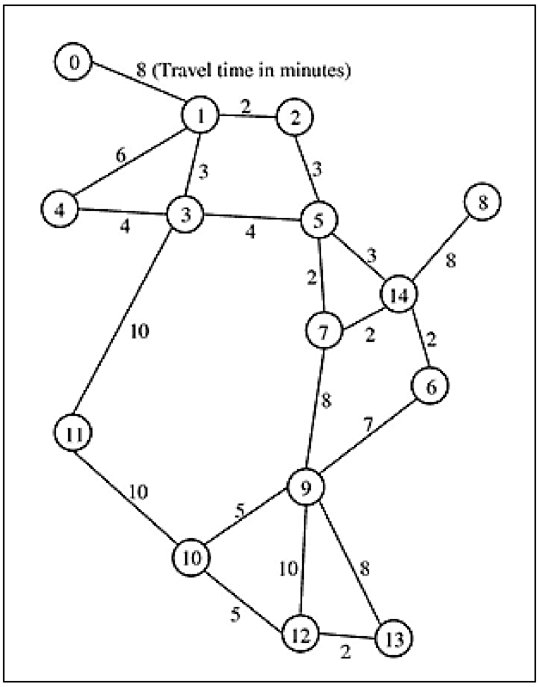

Figure 3: Mandl's Swiss Network

- Average route headways
- Maximum route headways

The proposed MTS algorithm is compared to different algorithms: heuristic by Mandl (1980), Baaj and Mahmassani (1991), Shih and Mahmassani (1994); genetic algorithm with ant-system (GA-AS) by Bagloee and Ceder (2011); bee colony optimization (BCO) by Nikolić and Teodorović (2014); genetic algorithm (GA) by Arbex and da Cunha (2015); memetic algorithm (MA) by Zhao et al. (2015); and differential evolution (DE) by Buba and Lee (2018). These researchers also contributed their route sets for this study together with other authors such as Chakroborty (2003), Mumford (2013), Chew et al. (2013). Every researcher has their own optimal sequence of the nodes in the route set. Therefore, the solution of the frequency setting problem is based on the route set given. The solution with lower number of buses, lesser total waiting time, lesser average and maximum headways without overcrowding is considered as the best.

The ranking of the performance metrics is categorized such that number of buses is the most important followed by total waiting times, average route headway and maximum route headway consecutively. The solution with lower number of buses is given the highest priority to be chosen as the best. Note that, the average values for number of buses are round up to the nearest whole number.

The proposed algorithm is tested with the respective passenger assignment methods and transfer penalties used by the previous authors for all the route sets. For each route, the proposed algorithm is conducted for 10 runs to check the robustness of the algorithm based on the parameter configuration in Table 1. Based on the Tables 2 - 7, the first column indicates the source of the route sets to the benchmark data. The second column indicates the algorithms used to obtain the optimal results. The next five columns specified the performance metrics values mentioned earlier. Additionally, overcrowding is included to show the acceptance of solution obtained.

In the case of 4 routes (see Table 2), the proposed MTS algorithms improved the number of buses, total waiting times, average route headways and maximum route headways as compared to the values produced by other authors for all the route sets except Mandl (1980). For the route set from Mandl (1980), although the number of buses is significantly lesser but the total waiting time is higher. The results from the two passenger assignment models are presented for comparison since both of the models are utilized in the past literature. We can also observe that there is no significant difference in the solution from the two models such that they are comparable to each other. Note that although Chakroborty (2003) has published the route sets for 4, 6, 7 and 8 routes but only route 4 is included for comparison because there is no previously published result available considering the aforementioned performance criteria for 6, 7 and 8 routes.

For 5 routes, the proposed MTS algorithm is compared with the result only from Arbex and da Cunha (2015) as it is the only study that produced 5 routes for Mandl's network. From Table 3, MTS algorithm produces better results in term of total number of buses, maximum route headway and average route headway. In the case of 6 routes, the MTS algorithms outperformed all the previously published results but the average route headway for Mumford (2013) is higher since the number of buses is lower. For the routes in Baaj and Mahmassani (1991), two passenger assignment method using multinomial logit model and frequency share rule are applied to compare with the respective solutions from literature. The proposed MTS algorithm produced superior results when compared to the existing solutions correspondingly. Besides, for the route sets in Chew et al. (2013) and Arbex and da Cunha (2015), the total waiting times for the best values are higher than the average values since the waiting time can be increased by reducing the number of buses.

In the case of 7 routes, Table 4 shows that the proposed MTS algorithm improves the value of Arbex and da Cunha (2015) for their own route sets and the routes from Nikolić and Teodorović (2013). The proposed MTS algorithm outperformed the solutions from Arbex and da Cunha (2015) for the route set of Chew et al. (2013) and the solutions of Buba and Lee (2018). Besides, our solutions are comparable to Nikolić and Teodorović (2014) for both operator and passenger and to Arbex and da Cunha (2015) for the Mumford (2013) routes but not in the Pareto sense because the number of buses for the average solution is higher. Similarly, the results from MTS algorithm is equivalent to the values from Baaj and Mahmassani (1991). Most of studies have higher total waiting times for average values as compared to the best values due to lower number of buses.

In the case of 8 routes (see Table 5), the proposed MTS algorithm improved the previous solutions for the routes from all the studies except Arbex and da Cunha (2015). The number of buses produced by the proposed MTS algorithms are higher but the average and maximum route headway are lower for the routes in Arbex and da Cunha (2015).Alternatively, the best value of total waiting time for the route set are higher as compared to the average value. In addition, the average solutions of MTS algorithm are equivalent to Nikolić and Teodorović (2014) and Zhao et al. (2015) as the total number of buses is higher even though the total waiting times are lesser.

Furthermore, our proposed MTS algorithm generate best results as compared to Arbex and da Cunha (2015) for 9 and 11 routes (see Table 6). However, for 11 routes, the best value of total waiting time is greater than the average value as the number of buses is reduced. For route set of size 10, the results found are equivalent as it able to reduce the average and maximum route headway although the total buses are increased. Based on Table 7, for 12 routes, our results are preferable as compared to Bagloee and Ceder (2011), Nikolić and Teodorović (2014)-operator and Arbex and da Cunha (2015) for their respective route sets. Moreover, the results in this study are relatively comparable with Arbex and da Cunha (2015) for the routes from Bagloee and Ceder (2011) and with Nikolić and Teodorović (2014)-passenger and Buba and Lee (2018) for their own route sets.

In order to further validate and verify our proposed algorithms, the algorithm is tested using an extended model considering different demands and travel times throughout the time horizon as in peak and off-peak hours. Each of the time intervals between 5.00 am to 11.00 pm are indicated as a timeslot (see Table 8). The parameters are set as in Table 1 and the multinomial logit model is applied for passenger assignment procedure. The route sets from Buba and Lee (2018) are experimented and the results for 4 routes of the extended model which includes the average solutions of every domain are presented in Table 9.

Generally, the total buses are dependent on round trip time of a bus. Thus, the number of buses required is quite high as compared to the previous model because the layover time and dwell time are added to the travel time of a bus which increase the round trip time for a trip. Since the frequencies for peak and off-peak hours are not the same, the total waiting times also increases. The existence of overcrowding in the first two domain is caused by insufficient number of bus trip that unable to carry all the passengers at specific timeslot in some of the routes.

The solution from domain 3 which needs lesser number of buses is chosen to study the bus and crew scheduling problem and to build the schedules. This is because there is no overcrowding in the results starting from domain 3 and thus the solutions from domain 3 to 10 are more preferable.

2. Bus and Driver Scheduling

For bus and driver scheduling, we have studied two different scenarios regarding the departure times at the origin and destination of a route. The first scenario allocate same departure times at both terminals which favors the travelling passengers. This is to ensure the headways are equal during the time period for both terminals. Conversely, the second scenario assigns the departure times only for the starting terminal.

Each scenario is performed for 10 runs and the average and best solution among them are recorded. The input data such as frequency and one-away travel time of the routes are obtained from frequency setting problem. Based on the Tables 10 and 11, the number of buses and drivers required for consecutive trips from one terminal to another are listed according to the number of routes.

The total buses and drivers are increase as the number of routes increase since each route need to be assigned different buses and drivers. Furthermore, the average and best solution from both tables are almost equal to each other which shows the robustness of the algorithm.

By comparing both scenarios, the total buses and drivers are higher for scenario 1 than scenario 2 because the departure times to be covered are increased as it considers both starting and ending terminals. The bus and driver schedule of scenario 1 for the 4 routes from Buba and Lee (2018) are displayed in Table 12-15. It shows the bus numbers and drivers that cover the respective departure times at the first stop (origin) and last stop (destination) of a route.

Based on Table 12, total number of buses and drivers needed for the first route are 26 and 37 respectively. The total frequency

Table 2: Comparison Results for 4 Routes

Source of route sets	Algorithm	Buses	Total waiting times (min)	Overcrowding	Average route headways (min)	Maximum route headways (min)
Mandl (1980)	[1]	103^a	n/a	n/a	3.48^a	6.67^a
	[2]	99^b	18194^b	n/a	n/a	n/a
	[3]	99^b	n/a	n/a	n/a	n/a
	[9]	$54^a(55)^b$	$27959^a(24666)^b$	$0^a(0)^b$	$3.08^a(3.05)^b$	$3.14^a(3.11)^b$
	[10]	$54^a(54)^b$	$27563^a(24746)^b$	$0^a(0)^b$	$3.00^a(3.07)^b$	$3.14^a(3.14)^b$
Chakroborty (2003)	[1]	105^a	n/a	n/a	4.06^a	9.00^a
	[9]	81^a	19724^a	0^a	3.39^a	3.51^a
	[10]	80^a	19247^a	0^a	3.30^a	3.36^a
Mumford (2013)	[1]	86^a	n/a	n/a	4.43^a	9.33^a
	[9]	79^a	22281^a	0^a	3.82^a	3.95^a
	[10]	79^a	22110^a	0^a	3.78^a	3.93^a
Chew et al. (2013)	[1]	87^a	n/a	n/a	3.64^a	5.11^a
	[9]	87^a	19119^a	0^a	3.44^a	3.58^a
	[10]	86^a	19440^a	0^a	3.50^a	3.61^a
Nikolić and Teodorović (2013)	[1]	94^a	n/a	n/a	3.41^a	4.32^a
	[9]	87^a	18871^a	0^a	3.42^a	3.56^a
	[10]	86^a	18767^a	0^a	3.42^a	3.59^a
Arbex and da Cunha (2015)	[1]	79^a	n/a	n/a	4.60^a	8.60^a
	[9]	76^a	20857^a	0^a	3.78^a	3.94^a
	[10]	76^a	20780^a	0^a	3.77^a	3.93^a
Nikolić and Teodorović (2014)-passenger	[4]	94^b	21147^b	n/a	n/a	n/a
	[9]	88^b	19656^b	0^b	3.38^b	3.49^b
	[10]	88^b	19489^b	0^b	3.35^b	3.47^b
Nikolić and Teodorović (2014)-operator	[1]	67^b	26057^b	n/a	n/a	n/a
	[3]	67^b	n/a	n/a	n/a	n/a
	[9]	54^b	25040^b	0^b	4.29^b	4.48^b
	[10]	54^b	24711^b	0^b	4.24^b	4.50^b
Buba and Lee (2018)	[5]	95^b	24098^b	n/a	n/a	n/a
	[9]	87^b	21117^b	0^b	3.38^b	3.53^b
	[10]	86^b	21095^b	0^b	3.41^b	3.61^b

Note: n/a=not available; a=multinomial logit model; b=frequency share rule
[l]:Arbex and da Cunha (2015); [2]:Mandl (1980); [3]:Zhao et al. (2015); [4]:Nikolić and Teodorović (2014); [5]:Buba and Lee (2018); [6]:Baaj and Mahmassani (1991); [7]:Shih and Mahmassani (1994); [8]:Bagloee and Ceder (2011); [9]:proposed MTS(average); [10]:proposed MTS(best)

Table 3: Comparison Results for 5 and 6 Routes

Source of route sets	Algorithm	Buses	Total waiting times (min)	Overcrowding	Average route headways (min)	Maximum route headways (min)
5 routes						
Arbex and da Cunha (2015)	[1]	75^a	n/a	n/a	5.39^a	9.56^a
	[9]	66^a	25660^a	0^a	5.17^a	5.45^a
	[10]	64^a	26262^a	0^a	5.26^a	5.48^a
6 routes						
Baaj and Mahmassani (1991)	[6]	89^b	20920^b	n/a	n/a	n/a
	[1]	87^a	n/a	n/a	4.11^a	11.33^a
	[3]	89^b	n/a	n/a	n/a	n/a
	[9]	$76^a(76)^b$	$20862^a(19655)^b$	$0^a(0)^b$	$3.36^a(3.36)^b$	$3.46^a(3.45)^b$
	[10]	$76^a(76)^b$	$20782^a(19558)^b$	$0^a(0)^b$	$3.35^a(3.35)^b$	$3.44^a(3.44)^b$
Shih and Mahmassani (1994)	[7]	84^b	20058^b	n/a	n/a	n/a
	[3]	84^b	n/a	n/a	n/a	n/a
	[9]	82^b	19957^b	0^b	3.06^b	3.13^b
	[10]	82^b	19869^b	0^b	3.04^b	3.09^b
Mumford (2013)	[1]	98^a	n/a	n/a	5.06^a	8.00^a
	[9]	88^a	18469^a	0^a	5.08^a	5.31^a
	[10]	88^a	18433^a	0^a	5.08^a	5.24^a
Chew et al. (2013)	[1]	110^a	n/a	n/a	4.86^a	8.00^a
	[9]	104^a	16626^a	0^a	4.37^a	4.59^a
	[10]	101^a	17457^a	0^a	4.56^a	4.74^a
Nikolić and Teodorović (2013)	[1]	102^a	n/a	n/a	5.25^a	10.29^a
	[9]	100^a	18178^a	0^a	4.58^a	4.83^a
	[10]	100^a	18147^a	0^a	4.52^a	4.70^a
Nikolić and Teodorović (2014)-passenger	[4]	99^b	21766^b	n/a	n/a	n/a
	[9]	98^b	19800^b	0^b	3.78^b	3.91^b
	[10]	98^b	19697^b	0^b	3.76^b	3.87^b
Nikolić and Teodorović (2014)- operator	[4]	66^b	31500^b	n/a	n/a	n/a
	[3]	66^b	n/a	n/a	n/a	n/a
	[9]	61^b	26162^b	0^b	4.30^b	4.52^b
	[10]	61^b	25946^b	0^b	4.27^b	4.43^b
Arbex and da Cunha (2015)	[1]	77^a	n/a	n/a	6.42^a	9.56^a
	[9]	66^a	22994^a	0^a	6.06^a	6.55^a
	[10]	63^a	26728^a	0^a	6.31^a	6.71^a
Buba and Lee (2018)	[5]	92^b	24705^b	n/a	n/a	n/a
	[9]	85^b	23117^b	0^b	5.07^b	5.34^b
	[10]	85^b	23091^b	0^b	5.04^b	5.57^b

(number of departure time) at each terminal is 99 with the average headway of 11.79 minutes. Each departure time is assigned to a specific bus and driver according to the travel time and working period. The frequency of the route at peak hours is 7 with 8.57 minutes of interval between each departure whereas at off-peak hours, the frequency is 4 with 15

Table 4: Comparison Results for 7 Routes

Source of route sets	Algorithm	Buses	Total waiting times (min)	Overcrowding	Average route headways (min)	Maximum route headways (min)
Mumford (2013)	[1]	102[a]	n/a	n/a	5.32[a]	6.80[a]
	[9]	105[a]	15996[a]	0[a]	5.11[a]	5.42[a]
	[10]	101[a]	16446[a]	0[a]	5.25[a]	5.51[a]
Chew et al. (2013)	[1]	110[a]	n/a	n/a	4.86[a]	8.00[a]
	[9]	112[a]	16350[a]	0[a]	4.32[a]	4.54[a]
	[10]	108[a]	17077[a]	0[a]	4.50[a]	4.74[a]
Nikolić and Teodorović (2013)	[1]	98[a]	n/a	n/a	7.00[a]	17.5[a]
	[9]	82[a]	21501[a]	0[a]	6.12[a]	6.60[a]
	[10]	82[a]	21458[a]	0[a]	6.13[a]	6.63[a]
Arbex and da Cunha (2015)	[1]	77[a]	n/a	n/a	7.58[a]	12.8[a]
	[9]	76[a]	22323[a]	0[a]	6.18[a]	6.74[a]
	[10]	76[a]	21955[a]	0[a]	6.12[a]	6.84[a]
Baaj and Mahmassani (1991)	[6]	82[b]	22804[b]	n/a	n/a	n/a
	[3]	82[b]	n/a	n/a	n/a	n/a
	[9]	72[b]	23893[b]	0[b]	3.03[b]	3.11[b]
	[10]	71[b]	23901[b]	0[b]	3.04[b]	3.20[b]
Nikolić and Teodorović (2014)- passenger	[4]	99[b]	23157[b]	n/a	n/a	n/a
	[9]	100[b]	19446[b]	0[b]	4.33[b]	4.55[b]
	[10]	96[b]	20169[b]	0[b]	4.44[b]	4.70[b]
Nikolić and Teodorović (2014)- operator	[4]	63[b]	35481[b]	n/a	n/a	n/a
	[3]	63[b]	n/a	n/a	n/a	n/a
	[9]	67[b]	31984[b]	0[b]	6.10[b]	6.45[b]
	[10]	63[b]	33700[b]	0[b]	6.38[b]	6.75[b]
Buba and Lee (2018)	[5]	90[b]	25587[b]	n/a	n/a	n/a
	[9]	90[b]	26293[b]	0[b]	6.17[b]	6.47[b]
	[10]	90[b]	25584[b]	0[b]	6.01[b]	6.67[b]

Table 5: Comparison Results for 8 Routes

Source of route sets	Algorithm	Buses	Total waiting times (min)	Overcrowding	Average route headways (min)	Maximum route headways (min)
Baaj and Mahmassani (1991)	[6]	77[b]	27064[b]	n/a	n/a	n/a
	[1]	78[b]	n/a	n/a	4.65[b]	10.67[b]
	[3]	77[b]	n/a	n/a	n/a	n/a
	[9]	73[a](73)[b]	23656[a](21967)[b]	0[a](0)[b]	4.26[a](4.28)[b]	4.52[a](4.54)[b]
	[10]	73[a](72)[b]	23596[a](22200)[b]	0[a](0)[b]	4.23[a](4.31)[b]	4.52[a](4.60)[b]
Shih and Mahmassani (1994)	[7]	68[b]	26455[b]	n/a	n/a	n/a
	[3]	68[b]	n/a	n/a	n/a	n/a
	[9]	62[b]	23479[b]	0[b]	5.02[b]	5.16[b]
	[10]	62[b]	23227[b]	0[b]	4.98[b]	5.14[b]
Mumford (2013)	[1]	101[a]	n/a	n/a	6.91[a]	15.00[a]
	[9]	97[a]	18582[a]	0[a]	6.12[a]	6.69[a]
	[10]	96[a]	18510[a]	0[a]	6.09[a]	6.59[a]
Chew et al. (2013)	[1]	88[a]	n/a	n/a	9.67[a]	31.00[a]
	[9]	86[a]	24170[a]	0[a]	6.05[a]	6.54[a]
	[10]	86[a]	23843[a]	0[a]	6.02[a]	6.51[a]
Nikolić and Teodorović (2013)	[1]	104[a]	n/a	n/a	9.66[a]	29.00[a]
	[9]	95[a]	19954[a]	0[a]	6.15[a]	6.66[a]
	[10]	95[a]	19880[a]	0[a]	6.15[a]	6.79[a]
Nikolić and Teodorović (2014)- passenger	[4]	99[b]	24726[b]	n/a	n/a	n/a
	[9]	95[b]	21045[b]	0[b]	6.04[b]	6.50[b]
	[10]	95[b]	20804[b]	0[b]	5.96[b]	6.35[b]
Nikolić and Teodorović (2014)- operator	[4]	63[b]	34931[b]	n/a	n/a	n/a
	[3]	63[b]	n/a	n/a	n/a	n/a
	[9]	65[b]	25741[b]	0[b]	5.90[b]	6.49[b]
	[10]	65[b]	25479[b]	0[b]	5.87[b]	6.51[b]
Arbex and da Cunha (2015)	[1]	69[a]	n/a	n/a	7.02[a]	10.33[a]
	[9]	73[a]	23547[a]	0[a]	6.04[a]	6.48[a]
	[10]	71[a]	23803[a]	0[a]	6.11[a]	6.84[a]
Buba and Lee (2018)	[5]	94[b]	25487[b]	n/a	n/a	n/a
	[9]	94[b]	25397[b]	0[b]	6.11[b]	6.61[b]
	[10]	93[b]	25083[b]	0[b]	6.11[b]	6.79[b]

Table 6: Comparison Results for 9,10 and 11 Routes

Source of route sets	Algorithm	Buses	Total waiting times (min)	Overcrowding	Average route headways (min)	Maximum route headways (min)
9 routes						
Arbex and da Cunha (2015)	[1]	66[a]	n/a	n/a	9.14[a]	20.00[a]
	[9]	58[a]	35226[a]	0[a]	7.51[a]	8.26[a]
	[10]	58[a]	34381[a]	0[a]	7.34[a]	8.06[a]
10 routes						
Arbex and da Cunha (2015)	[1]	72[a]	n/a	n/a	9.44[a]	20.00[a]
	[9]	81[a]	23990[a]	0[a]	7.43[a]	8.16[a]
	[10]	81[a]	23903[a]	0[a]	7.36[a]	7.88[a]
11 routes						
Arbex and da Cunha (2015)	[1]	68[a]	n/a	n/a	8.76[a]	14.4[a]
	[9]	70[a]	26032[a]	0[a]	7.38[a]	8.01[a]
	[10]	67[a]	26863[a]	0[a]	7.56[a]	8.12[a]

Table 7: Comparison Results for 12 Routes

Source of route sets	Algorithm	Buses	Total waiting times (min)	Overcrowding	Average route headways (min)	Maximum route headways (min)
Bagloee and Ceder (2011)	[1]	78^a	n/a	n/a	7.23^a	11.33^a
	[8]	87^b	24951^b	n/a	n/a	n/a
	[9]	$73^a(73)^b$	$22901^a(19827)^b$	$0^a(0)^b$	$7.40^a(7.41)^b$	$8.30^a(8.25)^b$
	[10]	$73^a(70)^b$	$22520^a(20576)^b$	$0^a(0)^b$	$7.24^a(7.63)^b$	$7.83^a(8.71)^b$
Arbex and da Cunha (2015)	[1]	73^a	n/a	n/a	10.58^a	15.33^a
	[9]	63^a	31512^a	0^a	10.01^a	11.83^a
	[10]	62^a	31162^a	0^a	10.01^a	12.13^a
Nikolić and Teodorović (2014)-passenger	[4]	98^b	23867^b	n/a	n/a	n/a
	[9]	96^b	24746^b	0^b	6.09^b	6.70^b
	[10]	95^b	24675^b	0^b	6.09^b	6.71^b
Nikolić and Teodorović (2014)-operator	[4]	65^b	36051^b	n/a	n/a	n/a
	[9]	52^b	35215^b	0^b	10.13^b	11.94^b
	[10]	52^b	33828^b	0^b	10.01^b	11.61^b
Buba and Lee (2018)	[5]	88^b	25670^b	n/a	n/a	n/a
	[9]	73^b	42878^b	0^b	10.02^b	11.79^b
	[10]	72^b	42468^b	0^b	10.31^b	12.13^b

Table 8: Category of each Timeslot

Timeslot	Time interval	Category
1	0500-0600	Off-peak
2	0600-0700	Off-peak
3	0700-0800	Peak
4	0800-0900	Peak
5	0900-1000	Peak
6	1000-1100	Off-peak
7	1100-1200	Off-peak
8	1200-1300	Peak
9	1300-1400	Peak
10	1400-1500	Off-peak
11	1500-1600	Off-peak
12	1600-1700	Peak
13	1700-1800	Peak
14	1800-1900	Peak
15	1900-2000	Peak
16	2000-2100	Off-peak
17	2100-2200	Off-peak
18	2200-2300	Off-peak

Table 9: Results Obtained for Extended Model for 4 Routes

Domain	Number of buses	Total waiting times (min)	Overcrowding
1	17	260559	6034
2	31	139851	731
3	44	99934	0
4	58	76488	0
5	70	62809	0
6	83	52725	0
7	97	45625	0
8	109	40419	0
9	122	36097	0
10	136	29406	0

Table 10: Results for scenario 1 (Buba and Lee, 2018)

Routes	Algorithm	Total buses	Total crews
4	Average	110	154
	Best	106	153
6	Average	151	207
	Best	148	203
7	Average	195	254
	Best	191	254
8	Average	206	250
	Best	202	268
12	Average	258	355
	Best	256	349

Table 11: Results for scenario 2 (Buba and Lee, 2018)

Routes	Algorithm	Total buses	Total crews
4	Average	103	136
	Best	98	134
6	Average	137	185
	Best	134	179
7	Average	175	227
	Best	165	226
8	Average	185	242
	Best	179	239
12	Average	240	312
	Best	234	310

minutes of interval.

Table 13 shows that route 2 require 40 buses and 58 drivers to cover all the departure times between 5.00 am and 11.00 pm. The total departure times at each terminal is 126 with the average headway of 9.34 minutes. The frequency and headway of the route at peak hours is 9 and 6.67 minutes respectively. Meanwhile, the frequency and headway of the route at off-peak hours are 5 and 12 minutes correspondingly.

Table 14 indicates the buses and drivers as-

Table 12: Schedule of Route 1 from 4 Routes

Origin to Destination			Destination to Origin			Origin to Destination			Destination to Origin		
Time	Bus No.	Driver No.	Time	Bus No.	Driver No.	Time	Bus No.	Driver No.	Time	Bus No.	Driver No.
5:00	1	1	5:00	9	15	13:51	25	36	13:51	11	19
5:15	2	2	5:15	18	28	14:00	7	11	14:00	23	34
5:30	3	4	5:30	10	17	14:15	22	33	14:15	2	2
5:45	25	36	5:45	4	6	14:30	6	9	14:30	12	21
6:00	19	29	6:00	1	1	14:45	11	19	14:45	25	36
6:15	18	28	6:15	2	2	15:00	23	34	15:00	7	11
6:30	8	13	6:30	3	4	15:15	2	3	15:15	22	33
6:45	11	19	6:45	25	36	15:30	12	21	15:30	6	9
7:00	1	1	7:00	19	29	15:45	21	32	15:45	11	19
7:09	2	2	7:09	18	28	16:00	7	11	16:00	8	14
7:17	3	4	7:17	8	13	16:09	22	33	16:09	2	3
7:26	5	7	7:26	6	9	16:17	6	9	16:17	12	21
7:34	25	36	7:34	11	19	16:26	10	18	16:26	21	32
7:43	22	33	7:43	21	32	16:34	13	22	16:34	14	23
7:51	19	29	7:51	1	1	16:43	8	14	16:43	7	11
8:00	18	28	8:00	2	2	16:51	20	30	16:51	15	24
8:09	8	13	8:09	3	4	17:00	2	3	17:00	16	25
8:17	6	9	8:17	5	7	17:09	5	8	17:09	6	10
8:26	11	19	8:26	25	36	17:17	3	5	17:17	9	16
8:34	17	26	8:34	22	33	17:26	14	23	17:26	13	22
8:43	1	1	8:43	19	29	17:34	7	11	17:34	8	14
8:51	2	2	8:51	18	28	17:43	15	24	17:43	20	30
9:00	3	4	9:00	8	13	17:51	16	25	17:51	2	3
9:09	5	7	9:09	6	9	18:00	6	10	18:00	5	8
9:17	25	36	9:17	11	19	18:09	12	21	18:09	10	18
9:26	22	33	9:26	17	26	18:17	13	22	18:17	14	23
9:34	19	29	9:34	23	34	18:26	8	14	18:26	7	11
9:43	4	6	9:43	1	1	18:34	20	30	18:34	15	24
9:51	8	13	9:51	3	4	18:43	17	27	18:43	16	25
10:00	6	9	10:00	5	7	18:51	5	8	18:51	6	10
10:15	11	19	10:15	20	30	19:00	10	18	19:00	12	21
10:30	17	26	10:30	22	33	19:09	14	23	19:09	13	22
10:45	18	28	10:45	2	2	19:17	24	35	19:17	3	5
11:00	5	7	11:00	6	9	19:26	15	24	19:26	20	31
11:15	9	15	11:15	25	36	19:34	16	25	19:34	17	27
11:30	22	33	11:30	17	26	19:43	2	3	19:43	5	8
11:45	2	2	11:45	8	13	19:51	12	21	19:51	10	18
12:00	1	1	12:00	19	29	20:00	13	22	20:00	14	23
12:09	25	36	12:09	11	19	20:15	7	11	20:15	15	24
12:17	17	26	12:17	23	34	20:30	17	27	20:30	16	25
12:26	8	13	12:26	2	2	20:45	26	37	20:45	12	21
12:34	21	32	12:34	12	21	21:00	10	18	21:00	24	35
12:43	6	9	12:43	5	7	21:15	11	20	21:15	7	11
12:51	26	37	12:51	4	6	21:30	2	3	21:30	17	27
13:00	11	19	13:00	25	36	21:45	3	5	21:45	26	37
13:09	23	34	13:09	7	11	22:00	14	23	22:00	13	22
13:17	2	2	13:17	22	33	22:15	7	12	22:15	11	20
13:26	12	21	13:26	21	32	22:30	16	25	22:30	2	3
13:34	5	7	13:34	6	9	22:45	20	31	22:45	3	5
13:43	4	6	13:43	26	37						

signed at every departure time in route 3. The schedule consists of 21 buses and 31 drivers. The time intervals between every departure time are set at 6.67 minutes for peak hours and 12 minutes for off-peak hours. Thus, the number of bus departure for peak hours and off-peak hours are 9 and 5 respectively. The average headway of the route is 9.34 minutes.

Similarly, Table 15 presents the sequence of 126 departure times at each terminal of route 4. A total of 19 buses and 27 drivers are allocated to the departure times. The frequency and headway of the route at peak hours are fixed as 9 and 6.67 minutes. During off-peaks hours, the frequency and headway are scheduled at 5 and 12 minutes respectively.

V. Conclusion

In this research, urban transit scheduling problem which consist of frequency setting, timetabling and simultaneous bus and driver

Table 13: Schedule of Route 2 from 4 Routes

Origin to Destination			Destination to Origin			Origin to Destination			Destination to Origin		
Time	Bus No.	Driver No.	Time	Bus No.	Driver No.	Time	Bus No.	Driver No.	Time	Bus No.	Driver No.
5:00	1	1	5:00	37	52	13:53	25	34	13:53	39	55
5:12	30	41	5:12	12	16	14:07	38	54	14:07	27	37
5:24	20	27	5:24	28	38	14:12	4	7	14:12	1	2
5:36	22	30	5:36	29	39	14:24	11	15	14:24	14	19
5:48	2	3	5:48	5	9	14:36	17	23	14:36	22	30
6:00	3	5	6:00	34	48	14:48	36	51	14:48	2	3
6:12	25	34	6:12	39	55	15:00	6	10	15:00	9	13
6:24	4	7	6:24	1	1	15:12	39	55	15:12	38	54
6:36	24	33	6:36	20	27	15:24	1	2	15:24	4	7
6:48	17	23	6:48	22	30	15:36	14	19	15:36	11	15
7:00	5	9	7:00	2	3	15:48	22	31	15:48	17	23
7:07	34	48	7:07	3	5	16:00	2	4	16:00	36	51
7:13	6	10	7:13	30	41	16:07	9	13	16:07	6	10
7:20	39	55	7:20	25	34	16:13	38	54	16:13	39	56
7:27	37	52	7:27	32	45	16:20	15	21	16:20	19	26
7:33	1	1	7:33	4	7	16:27	4	8	16:27	20	28
7:40	20	27	7:40	24	33	16:33	16	22	16:33	21	29
7:47	7	11	7:47	33	46	16:40	11	15	16:40	14	19
7:53	22	30	7:53	17	23	16:47	17	24	16:47	22	31
8:00	8	12	8:00	40	57	16:53	3	6	16:53	10	14
8:07	2	3	8:07	5	9	17:00	36	51	17:00	2	4
8:13	3	5	8:13	34	48	17:07	18	25	17:07	23	32
8:20	9	13	8:20	6	10	17:13	34	49	17:13	9	13
8:27	25	34	8:27	39	55	17:20	39	56	17:20	25	35
8:33	10	14	8:33	37	52	17:27	19	26	17:27	15	21
8:40	4	7	8:40	1	1	17:33	20	28	17:33	4	8
8:47	24	33	8:47	20	27	17:40	21	29	17:40	16	22
8:53	32	45	8:53	7	11	17:47	31	43	17:47	11	15
9:00	17	23	9:00	22	30	17:53	22	31	17:53	17	24
9:07	36	51	9:07	8	12	18:00	27	37	18:00	3	6
9:13	5	9	9:13	2	3	18:07	2	4	18:07	36	51
9:20	34	48	9:20	3	5	18:13	23	32	18:13	18	25
9:27	6	10	9:27	9	13	18:20	13	18	18:20	38	54
9:33	39	55	9:33	25	34	18:27	25	35	18:27	39	56
9:40	37	52	9:40	10	14	18:33	15	21	18:33	19	26
9:47	1	1	9:47	4	7	18:40	4	8	18:40	20	28
9:53	31	43	9:53	24	33	18:47	14	19	18:47	21	29
10:00	7	11	10:00	32	45	18:53	33	47	18:53	31	44
10:12	12	16	10:12	17	23	19:00	17	24	19:00	12	17
10:24	8	12	10:24	36	51	19:06	3	6	19:06	27	37
10:36	9	13	10:36	6	10	19:13	40	58	19:13	1	2
10:48	30	41	10:48	27	37	19:20	18	25	19:20	23	32
11:00	20	27	11:00	28	38	19:27	38	54	19:27	13	18
11:12	22	30	11:12	7	11	19:33	16	22	19:33	37	53
11:24	2	3	11:24	5	9	19:40	19	26	19:40	15	21
11:36	36	51	11:36	34	48	19:47	26	36	19:47	35	50
11:48	25	34	11:48	39	55	19:53	21	29	19:53	14	19
12:00	4	7	12:00	1	1	20:00	29	40	20:00	22	31
12:07	24	33	12:07	20	27	20:12	30	42	20:12	2	4
12:13	17	23	12:13	22	30	20:24	34	49	20:24	40	58
12:20	7	11	12:20	37	52	20:36	39	56	20:36	38	54
12:27	5	9	12:27	2	3	20:48	20	28	20:48	4	8
12:33	10	14	12:33	26	36	21:00	14	19	21:00	16	22
12:40	6	10	12:40	36	51	21:12	22	31	21:12	17	24
12:47	39	55	12:47	25	34	21:24	2	4	21:24	30	42
12:53	27	37	12:53	38	54	21:36	1	2	21:36	34	49
13:00	1	1	13:00	4	7	21:48	38	54	21:48	39	56
13:07	14	19	13:07	30	41	22:00	4	8	22:00	21	29
13:13	20	27	13:13	33	46	22:12	33	47	22:12	14	20
13:27	37	52	13:27	7	11	22:24	17	24	22:24	22	31
13:33	2	3	13:33	5	9	22:36	40	58	22:36	2	4
13:40	8	12	13:40	40	57	22:48	13	18	22:48	37	53
13:47	9	13	13:47	6	10						

scheduling is studied. A mixed integer multi-objective model is developed to find optimal frequencies for the routes by minimizing the number of buses, passenger's waiting times and overcrowding. The model is further extended by including timeslots to find the frequencies during peak and off-peak hours throughout the time period. The MTS algorithm is

Table 14: Schedule of Route 3 from 4 Routes

Origin to Destination			Destination to Origin			Origin to Destination			Destination to Origin		
Time	Bus No.	Driver No.	Time	Bus No.	Driver No.	Time	Bus No.	Driver No.	Time	Bus No.	Driver No.
5:00	1	1	5:00	14	19	13:47	20	29	13:47	6	9
5:12	12	16	5:12	11	15	13:53	10	14	13:53	4	6
5:24	15	21	5:24	18	26	14:07	12	16	14:07	7	11
5:36	14	19	5:36	1	1	14:12	21	31	14:12	18	26
5:48	5	7	5:48	12	16	14:24	1	2	14:24	3	5
6:00	18	26	6:00	2	3	14:36	6	9	14:36	20	29
6:12	1	1	6:12	19	27	14:48	7	11	14:48	21	31
6:24	6	9	6:24	20	29	15:00	3	5	15:00	1	2
6:36	2	3	6:36	18	26	15:12	20	29	15:12	6	9
6:48	16	23	6:48	1	1	15:24	21	31	15:24	16	23
7:00	20	29	7:00	6	9	15:36	1	2	15:36	3	5
7:07	4	6	7:07	5	7	15:48	6	10	15:48	20	30
7:13	18	26	7:13	2	3	16:00	19	28	16:00	21	31
7:20	1	1	7:20	16	23	16:07	3	5	16:07	1	2
7:27	3	5	7:27	15	21	16:13	13	17	16:13	8	12
7:33	6	9	7:33	20	29	16:20	20	30	16:20	6	10
7:40	17	24	7:40	4	6	16:27	9	13	16:27	7	11
7:47	2	3	7:47	18	26	16:33	21	31	16:33	19	28
7:53	16	23	7:53	1	1	16:40	1	2	16:40	17	24
8:00	15	21	8:00	3	5	16:47	8	12	16:47	13	17
8:07	20	29	8:07	6	9	16:53	6	10	16:53	20	30
8:13	4	6	8:13	17	24	17:00	7	11	17:00	9	13
8:20	18	26	8:20	2	3	17:07	19	28	17:07	2	4
8:27	1	1	8:27	16	23	17:13	17	25	17:13	1	2
8:33	3	5	8:33	15	21	17:20	10	14	17:20	8	12
8:40	6	9	8:40	20	29	17:27	20	30	17:27	6	10
8:47	17	24	8:47	4	6	17:33	9	13	17:33	7	11
8:53	2	3	8:53	18	26	17:40	2	4	17:40	19	28
9:00	16	23	9:00	1	1	17:47	1	2	17:47	17	25
9:07	15	21	9:07	3	5	17:53	8	12	17:53	10	14
9:13	20	29	9:13	6	9	18:00	6	10	18:00	20	30
9:20	4	6	9:20	17	24	18:07	13	17	18:07	9	13
9:27	18	26	9:27	2	3	18:13	19	28	18:13	2	4
9:33	12	16	9:33	16	23	18:20	7	11	18:20	5	8
9:40	3	5	9:40	15	21	18:27	10	14	18:27	8	12
9:47	6	9	9:47	20	29	18:33	20	30	18:33	6	10
9:53	17	24	9:53	4	6	18:40	9	13	18:40	13	17
10:00	2	3	10:00	18	26	18:47	2	4	18:47	19	28
10:12	16	23	10:12	12	16	18:53	5	8	18:53	7	11
10:24	11	15	10:24	14	19	19:00	8	12	19:00	10	14
10:36	20	29	10:36	6	9	19:07	6	10	19:07	20	30
10:48	1	1	10:48	16	23	19:13	13	17	19:13	9	13
11:00	14	19	11:00	11	15	19:20	17	25	19:20	1	2
11:12	4	6	11:12	7	11	19:27	7	11	19:27	5	8
11:24	16	23	11:24	1	1	19:33	10	14	19:33	8	12
11:36	5	7	11:36	14	19	19:40	14	20	19:40	15	22
11:48	15	21	11:48	4	6	19:47	9	13	19:47	13	17
12:00	1	1	12:00	16	23	19:53	1	2	19:53	17	25
12:07	6	9	12:07	20	29	20:00	5	8	20:00	7	11
12:13	13	17	12:13	10	14	20:12	8	12	20:12	10	14
12:20	7	11	12:20	12	16	20:24	19	28	20:24	9	13
12:27	18	26	12:27	21	31	20:36	17	25	20:36	1	2
12:33	16	23	12:33	1	1	20:48	20	30	20:48	6	10
12:40	20	29	12:40	6	9	21:00	21	31	21:00	19	28
12:47	10	14	12:47	13	17	21:12	1	2	21:12	14	20
12:53	12	16	12:53	7	11	21:24	6	10	21:24	20	30
13:00	21	31	13:00	18	26	21:36	19	28	21:36	8	12
13:07	1	1	13:07	3	5	21:48	14	20	21:48	1	2
13:13	6	9	13:13	20	29	22:00	20	30	22:00	6	10
13:20	4	6	13:20	10	14	22:12	9	13	22:12	19	28
13:27	7	11	13:27	12	16	22:24	1	2	22:24	14	20
13:33	18	26	13:33	21	31	22:36	6	10	22:36	20	30
13:40	3	5	13:40	1	1	22:48	13	18	22:48	9	13

developed by modifying the initialization process and incorporating intensification and diversification to guide the search effectively in order to obtain better solution. On the other hand, a set covering model is adapted from Zuo et al. (2015) for bus and driver scheduling by considering the working and break duration. Moreover, a reconstruction

Table 15: Schedule of Route 4 from 4 Routes

Origin to Destination			Destination to Origin			Origin to Destination			Destination to Origin		
Time	Bus No.	Driver No.	Time	Bus No.	Driver No.	Time	Bus No.	Driver No.	Time	Bus No.	Driver No.
5:00	14	19	5:00	4	5	13:47	12	17	13:47	5	6
5:12	17	23	5:12	12	17	13:53	4	5	13:53	17	23
5:24	19	26	5:24	13	18	14:07	13	18	14:07	19	26
5:36	4	5	5:36	14	19	14:12	16	21	14:12	9	12
5:48	12	17	5:48	17	23	14:24	15	20	14:24	8	10
6:00	13	18	6:00	19	26	14:36	5	6	14:36	10	13
6:12	14	19	6:12	4	5	14:48	9	12	14:48	16	21
6:24	5	6	6:24	12	17	15:00	8	10	15:00	15	20
6:36	19	26	6:36	13	18	15:12	10	13	15:12	5	6
6:48	9	12	6:48	14	19	15:24	16	21	15:24	9	12
7:00	12	17	7:00	5	6	15:36	15	20	15:36	8	10
7:07	10	13	7:07	11	15	15:48	5	7	15:48	10	13
7:13	13	18	7:13	19	26	16:00	17	24	16:00	16	21
7:20	14	19	7:20	9	12	16:07	8	10	16:07	18	25
7:27	1	1	7:27	2	3	16:13	7	9	16:13	3	4
7:33	5	6	7:33	12	17	16:20	10	14	16:20	5	7
7:40	11	15	7:40	10	13	16:27	1	1	16:27	6	8
7:47	19	26	7:47	13	18	16:33	16	21	16:33	17	24
7:53	9	12	7:53	14	19	16:40	18	25	16:40	8	10
8:00	2	3	8:00	1	1	16:47	3	4	16:47	7	9
8:07	12	17	8:07	5	6	16:53	5	7	16:53	10	14
8:13	10	13	8:13	11	15	17:00	6	8	17:00	15	20
8:20	13	18	8:20	19	26	17:07	19	27	17:07	1	2
8:27	14	19	8:27	9	12	17:13	8	10	17:13	18	25
8:33	1	1	8:33	2	3	17:20	7	9	17:20	3	4
8:40	5	6	8:40	12	17	17:27	10	14	17:27	5	7
8:47	11	15	8:47	10	13	17:33	15	20	17:33	6	8
8:53	19	26	8:53	13	18	17:40	1	2	17:40	19	27
9:00	9	12	9:00	14	19	17:47	18	25	17:47	8	10
9:07	2	3	9:07	1	1	17:53	3	4	17:53	7	9
9:13	17	23	9:13	5	6	18:00	5	7	18:00	10	14
9:20	10	13	9:20	11	15	18:07	17	24	18:07	16	21
9:27	13	18	9:27	19	26	18:13	19	27	18:13	1	2
9:33	4	5	9:33	9	12	18:20	6	8	18:20	18	25
9:40	1	1	9:40	2	3	18:27	7	9	18:27	3	4
9:47	5	6	9:47	17	23	18:33	10	14	18:33	5	7
9:53	11	15	9:53	10	13	18:40	16	21	18:40	17	24
10:00	19	26	10:00	13	18	18:47	1	2	18:47	19	27
10:12	9	12	10:12	4	5	18:53	18	25	18:53	6	8
10:24	12	17	10:24	1	1	19:00	3	4	19:00	7	9
10:36	10	13	10:36	5	6	19:07	5	7	19:07	10	14
10:48	14	19	10:48	9	12	19:13	17	24	19:13	16	21
11:00	1	1	11:00	12	17	19:20	8	10	19:20	15	20
11:12	11	15	11:12	10	13	19:27	6	8	19:27	18	25
11:24	8	10	11:24	14	19	19:33	7	9	19:33	3	4
11:36	12	17	11:36	1	1	19:40	10	14	19:40	11	16
11:48	13	18	11:48	19	26	19:47	16	21	19:47	17	24
12:00	14	19	12:00	8	10	19:53	15	20	19:53	8	10
12:07	5	6	12:07	12	17	20:00	18	25	20:00	6	8
12:13	2	3	12:13	15	20	20:12	3	4	20:12	7	9
12:20	19	26	12:20	13	18	20:24	17	24	20:24	16	21
12:27	9	12	12:27	16	21	20:36	8	11	20:36	15	20
12:33	8	10	12:33	14	19	20:48	6	8	20:48	5	7
12:40	12	17	12:40	5	6	21:00	16	21	21:00	17	24
12:47	4	5	12:47	2	3	21:12	11	16	21:12	10	14
12:53	13	18	12:53	19	26	21:24	5	7	21:24	6	8
13:00	16	21	13:00	9	12	21:36	17	24	21:36	18	25
13:07	14	19	13:07	8	10	21:48	10	14	21:48	3	4
13:13	5	6	13:13	12	17	22:00	7	9	22:00	5	7
13:20	17	23	13:20	4	5	22:12	18	25	22:12	17	24
13:27	19	26	13:27	13	18	22:24	3	4	22:24	10	14
13:33	9	12	13:33	16	21	22:36	5	7	22:36	1	2
13:40	8	10	13:40	14	19	22:48	17	24	22:48	16	22

mechanism is introduced in the procedure to further reduce the number of buses and drivers required to build schedules for all the routes.

The effectiveness of the proposed MTS algorithm is tested on Mandl's benchmark datasets using the route sets published in the literature. Two passenger assignment methods based on frequency share rule and multinomial logit model are used accordingly to determine the values of performance indicators as applied in the previous studies. The computational results show that the frequency set obtained improves the number of buses and total waiting times in most of the cases from the existing literature. Moreover, the extended model which consider different demands and travel times during the time period, produced higher number of buses and longer waiting times since it includes dwell time and layover time that increases the round-trip time of a bus.

On the other hand, for bus and driver scheduling, the proposed MTS algorithm is evaluated in two different scenarios reflecting the preference of users and operators to reduce the total buses and drivers for each route set. The first scenario considering users' perspective increases the number of buses and drivers required as there are more departure times to be covered within the time horizon. The second scenario assign the departure times at starting terminal of a route only that can reduces the operational cost. The solutions from the two scenarios are compared in term of the total number of buses and drivers required to cover the departure times within the time horizon. The analysis shows significant saving in the number of buses and drivers for the second scenario favoring operators as compared to the first scenario.

As future work, the proposed model can be extended to find the departure times at every stops of a route to build systematic schedules that can minimize passengers waiting times at the stops. Besides that, UTSP can be tackled with variable fleet size to be assigned accordingly to peak and off-peak hours in order to reduce the frequency and number of buses needed. Moreover, variable demand at different places and days can be considered to represent real world situation. The uncertainty events also can be incorporated to produce more robust schedules.

VI. Acknowledgement

This research is supported by Fundamental Research Grant Scheme (FRGS) 01-01-16-1867FR (Ministry of Higher Education, Malaysia). The authors are grateful to two anonymous reviewers and the editor for their detailed and constructive comments, which have helped us to improve the paper significantly.

References

[1] S. Afandizadeh, K. Hassan, and K. Narges. Bus fleet optimization using genetic algorithm a case study of mashhad. *International Journal of Civil Engineering*, 11(1):43–52, 2013.

[2] R. O. Arbex and C. B. da Cunha. Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm. *Transportation Research Part B: Methodological*, 81:355–376, 2015.

[3] M. H. Baaj and H. S. Mahmassani. An ai-based approach for transit route system planning and design. *Journal of Advanced Transportation*, 25(2):187–209, 1991.

[4] S. A. Bagloee and A. A. Ceder. Transit-network design methodology for actual-size road networks. *Transportation Research Part B: Methodological*, 45(10): 1787–1804, 2011.

[5] A. T. Buba and L. S. Lee. A differential evolution for simultaneous transit network design and frequency setting problem. *Expert Systems with Applications*, 106:277–289, 2018.

[6] Luis Cavique, Rego César, and l Themido Isabe. New heuristic algorithms for

the crew scheduling problem. In *Meta-Heuristics*, pages 37–47. New York: Springer US, 1999.

[7] A. Ceder. Urban transit scheduling: framework, review and examples. *Journal of Urban Planning and Development*, 128(4):225–244, 2002.

[8] P. Chakroborty. Genetic algorithms for optimal urban transit network design. *Computer-Aided Civil and Infrastructure Engineering*, 18(3):184–200, 2003.

[9] J. S. C. Chew, L. S. Lee, and H. V. Seow. Genetic algorithm for biobjective urban transit routing problem. *Journal of Applied Mathematics*, 2013.

[10] I. Constantin and M. Florian. Optimizing frequencies in a transit network: A nonlinear bi-level programming approach. *International Transactions in Operational Research*, 2(2):149–164, 1995.

[11] G Fedorko and M Weiszer. Configuring the parameters of multi-objective evolutionary algorithm for integrated timetabling and vehicle scheduling in public transport. In *Congress Proceeding*, pages 435–440. CLC, 2012.

[12] R. Giesen, H. Martínez, A. Mauttone, and M. E. Urquhart. A method for solving the multi-objective transit frequency optimization problem. *Journal of Advanced Transportation*, 50(8):2323–2337, 2016.

[13] M. C. Gomes, L. Cavique, and I. Themido. Bivariate extreme analysis of (olympic) swimming data. *Annals of Operations Research*, 144(1):111–132, 2006.

[14] Y. Li, W. Xu, and S. He. Expected value model for optimizing the multiple bus headways. *Applied Mathematics and Computation*, 219(11):5849–5861, 2013.

[15] H. R. Lourenço, J. P Paixão, and R. Portugal. Multiobjective metaheuristics for the bus driver scheduling problem. *Transportation Science*, 35(5):331–343, 2001.

[16] C.E. Mandl. Evaluation and optimization of urban public transportation networks. *European Journal of Operational Research*, 5(6):396–404, 1980.

[17] Christine L Mumford. New heuristic and evolutionary operators for the multi-objective urban transit routing problem. In *Evolutionary Computation (CEC)*, pages 939–946. IEEE, 2013.

[18] M. Nikolić and D. Teodorović. Transit network design by bee colony optimization. *Expert Systems with Applications*, 40:5945–5955, 2013.

[19] M. Nikolić and D. Teodorović. A simultaneous transit network design and frequency setting: Computing with bees. *Expert Systems with Applications*, 41 (16):7200–7209, 2014.

[20] S. Pothiya, I. Ngamroo, S. Runggeratigul, and P. Tantaswadi. Design of optimal fuzzy logic based pi controller using multiple tabu search algorithm for load frequency control. *International Journal of Control, Automation, and Systems*, 4 (2):155–164, 2006.

[21] B. D. A. Prata. A multiobjective metaheuristic approach for the integrated vehicle and crew scheduling. *Journal of Transport Literature*, 10(2):10–14, 2016.

[22] F. Ruisanchez and A. Ibeas. Design of a tabu search algorithm for assigning optimal bus sizes and frequencies in urban transport services. *Journal of Advanced Transportation*, 46(4):366–377, 2012.

[23] S. Scheele. A supply model for public transit services. *Transportation Research Part B*, 14:133–146, 1980.

[24] Mao-Chang Shih and Hani S Mahmassani. A design methodology for bus transit networks with coordinated operations. Technical report, 1994.

[25] I. Ö. Verbas and H. S. Mahmassani. Exploring trade-offs in frequency allocation in a transit network using bus route patterns: Methodology and application to large-scale urban systems. *Transportation Research Part B: Methodological*, 81: 577–595, 2015.

[26] H. Zhao, W. Xu, and R. Jiang. The memetic algorithm for the optimization of urban transit network. *Expert Systems with Applications*, 42(7):3760–3773, 2015.

[27] X. Zuo, C. Chen, W. Tan, and M. Zhou. Vehicle scheduling of an urban bus line via an improved multiobjective genetic algorithm. *IEEE Transactions on Intelligent Transportation Systems*, 16(2): 1030–1041, 2015.