



## USING VARIANCE NEIGHBOURHOOD SEARCH TO OPTIMIZE THE BUS WAITING ALLOCATION PROBLEM IN A MULTI-FLOOR BUS STATION

Yiyo KUO\*, Cheng Yan HUA

*Dept of Industrial Engineering and Management, Ming Chi University of Technology, New Taipei City, Taiwan*

Received 9 June 2015; revised 14 February 2016; accepted 21 July 2016

**Abstract.** This research deals with a multi-floor bus station, which provides services for a large number of passengers. The bus station has a limited platform capacity and there is no temporary parking lot for buses. When a large number of buses move into one floor of the station, buses cannot move smoothly and may not even be able to move at all. When a floor is full, buses waiting outside cannot enter, and buses inside cannot move out. It is fortunate that the station is designed as a multi-floor structure. When a bus is scheduled to move onto a floor, which has no more space for parking, it can move to another floor temporarily to wait. This research proposes the use of integer-programming to optimize the assignment of temporary waiting floor for all incoming buses in order to minimize the maximum delay. A Variance Neighbourhood Search (VNS) is proposed to solve the problem. The results show that when temporary waiting on another floor is permitted, the total time delay can be reduced by up to 47.41%.

**Keywords:** bus station, bus waiting allocation, variance neighbourhood search.

### Introduction

With the fast development of urbanization, more and more people live in the city and move in and out through the traffic centers. As a result, the space in traffic centers becomes overcrowded. Therefore, some cities have developed new traffic centers or rebuilt the old ones as multi-floor stations.

This research focuses on a multi-floor bus station, which still faces the challenge of insufficient space. The buses pass in and out of the bus station from long distances outside the city. Therefore, the station is designed as the final or initial stop for all bus schedules. When a bus arrives at the bus station, it always drives to the scheduled floor for alighting and boarding. The earlier the buses arrive, the longer they will wait in the station. During holidays, the station is always full of buses. If the scheduled floor is full of buses, buses outside cannot move in and buses inside cannot easily move out. Blockages inside the bus station can influence the outside traffic and the departure time of the buses can be seriously delayed.

In the transportation industry, drivers are not, traditionally, closely and directly supervised. Typically, drivers spend their entire work shift in the field-serving customers, and have only infrequent contact with their supervisors (Dessouky *et al.* 2003). However, the technology is

changing rapidly. Transport companies can receive a lot of real time data from all their vehicles by coupling them with wireless communication global-positioning-systems and other location technologies (Dessouky *et al.* 1999). Therefore, in the bus station, the control center can count the number of buses on each floor; know the departure time of each bus inside and their departure floors. They can even estimate the arrival times of buses (Yu *et al.* 2011), which will arrive within the next half hour or hour. If the floor that an arriving bus is scheduled to depart from is too crowded, and the arriving bus can be diverted to wait temporarily on another floor. Thus, more buses can move in without making any floor too crowded and the total delay of departures can be reduced.

Efficient real-time station control strategies can help maintain the smooth operation of the entire bus transport system. Station control strategies includes stop-skipping strategies, short-turn strategies, express strategies, dead-heading strategies, and holding strategies, which are the most popular strategies and are frequently used by public transit operators to reduce passenger waiting times and prevent the clustering of vehicles along a route (Yu, Yang 2009; Tirachini *et al.* 2011; Lo, Chang 2012). A stop-skiping strategy scheme allows the bus to skip one or more

\*Corresponding author. E-mail: [yiyo@mail.mcut.edu.tw](mailto:yiyo@mail.mcut.edu.tw)

stops to reduce its travel time when a bus is late and behind schedule (Liu *et al.* 2013). A short-turning strategy consists of selecting a portion of the fleet to serve short cycles on those route sections exhibiting high demand (Cortés *et al.* 2011). Express strategies serve only one section of their route, and then they proceed without stopping until reaching either the terminal or a pre-specified zone where the service is re-established (Tirachini *et al.* 2011). Deadheading strategies allow a vehicle to run empty from a terminal, skipping several stops before it starts its new service. The deadheading strategy must determine the dispatch time of the deadheaded vehicle and the beginning stop of the new service (Yu *et al.* 2012). These four strategies focus on buses routing or stop selection problems. They deal with the whole transit system, and are intended to provide a better service for all passengers. The status of the stations or terminals are ignored.

A holding strategy is one of the most common real-time control strategies in any transit operation. It is used to delay bus movements deliberately when a vehicle is ahead of schedule (Zolfaghari *et al.* 2004). A holding strategy not only reduces headway variance and average waiting time of passengers, but also increases the travel time of passengers on board and the total bus cycle time. There is some literature related to holding strategies, such as Dessouky *et al.* (1999), which deals with timed transfer at terminals by evaluating the performance of dispatching rules with Intelligent Transportation Systems (ITS) compared with those without ITS. When ITS are taken into consideration, eight holding strategies are tested through a simulation. The performance measures that are studied include total passenger delay and number of passengers missing their connections.

Hall *et al.* (2001) proposed control policies to minimize transfer time under stochastic conditions. They tried to determine how long a bus should be held at a transfer stop in anticipation of the arrival of passengers from connecting bus lines. Dessouky *et al.* (2003) compare seven holding strategies at a terminal. They also develop methods to forecast bus arrival times, which are most accurate for lines with long headways, which is usually the case in timed transfer systems. These methods are tested in simulations, which demonstrate that technology is most advantageous when the schedule slack is close to zero, when the headway is large, and when there are many connecting buses. In general, holding strategies take the arrival times of buses, and the transit information of all passengers, to maximize the transit efficiency of passengers. However, there is no related research, which takes the capacity of bus stations into consideration. Research related to the capacity of terminal bus stations is rare in the literature. Yan and Chen (2002) used a measure of bus station capacity as the maximum number of buses that can be held in the station during a specific time window. Based on station capacity and other constraints, they developed a model of inter-city bus carriers in timetable setting and bus routing/scheduling problems. They propose a Lagrangian heuristic and a flow decomposition algorithm to solve the problem efficiently.

The present research deals with a multi-floor bus station whose capacity is limited. For each floor, if the number of buses is greater than the capacity, all the buses on the floor will be affected, and buses outside cannot enter the station. However, buses are allowed to drive to another floors of the station which has spare capacity, temporarily. Integer-programming is proposed for the bus waiting allocation problem. This kind of research has not been conducted before. Moreover, Variance Neighbourhood Search (VNS) is a metaheuristic, which uses systematic changes of neighbourhood within a possibly randomized local search algorithm to produce a simple and effective metaheuristic for combinatorial and global optimization (Hansen *et al.* 2010). In this research, VNS is used for optimizing the integer-programming.

The remainder of this paper is organized as follows. In Section 1, the problem of bus waiting allocation in a multi-floor bus station is introduced. Then a mathematical programming model is provided. In Section 2, a VNS is proposed for solving the mathematical programming provided in Section 1. In Section 3, a case study of Taipei Bus Station is presented. Finally, conclusions are drawn in last section.

## 1. Problem description

This research deals with a multi-floor bus station. The station is always the first or last stop on any trip. Therefore, transit is not an important problem. All floors are different in their capacity. If the number of buses on a floor exceeds the limitation, no bus will be able to move easily. This would mean that buses inside cannot move out and buses outside cannot move in. When a bus arrives at the station, and the number of buses on its planned departure floor is equal to the capacity limitation, the bus can wait on another floor until the number of buses on its planned departure floor reduces. When all floors are full of buses (no capacity remains) the newly arrived buses are not permitted to enter, but passengers on the bus can alight outside the station. If a bus arrives to its planned departure floor early, it cannot depart before its planned departure time. Moreover, preparation time is required after it arrives at its platform and before it can depart.

It is not easy to estimate the travel times of all buses accurately. This research assumes that the control center of the bus station can use advanced technology to estimate the arrival of any bus within 30 minutes. The object of this research is to minimize the total delay in departure times by optimizing the bus waiting floor allocation.

In order to formulate this problem, following notations are used:

- $j$  – index of bus;
- $t$  – index of time interval;
- $f$  – index of floor;
- $J$  – total number of buses;
- $T$  – total number of time intervals;
- $F$  – total number of floors;
- $AT_j$  – the time interval that bus  $j$  arrives at the station;

$DT_j$  – the time interval that bus  $j$  is planned to depart;  
 $PT$  – the number of time intervals required for a bus preparing to leave after it moves to the platform;

$RC_{jf}$  – the remaining capacity of floor  $f$  in time interval  $t$ ;  
 $E$  – a number, and  $0 < E < 1$ ;

$AI_{jt} = \begin{cases} 1, & \text{if time interval } t \text{ is later than bus } j \text{ arrives;} \\ 0, & \text{otherwise;} \end{cases}$

$BP_{jt} = \begin{cases} 1, & \text{if time interval } t \text{ is earlier than bus } j \text{ arrives} \\ & \text{at its departure platform;} \\ 0, & \text{otherwise;} \end{cases}$

$BO_{jt} = \begin{cases} 1, & \text{if time interval } t \text{ is earlier than bus} \\ & j \text{ actually departs;} \\ 0, & \text{otherwise;} \end{cases}$

$AP_{jt} = \begin{cases} 1, & \text{if time interval } t \text{ is later than bus } j \text{ arrives at} \\ & \text{its departure platform;} \\ 0, & \text{otherwise;} \end{cases}$

$Y_{jf} = \begin{cases} 1, & \text{if bus } j \text{ is planned to depart from floor } f; \\ 0, & \text{otherwise.} \end{cases}$

Decision variables:

$IT_j$  – the time interval that bus  $j$  drives into the station;

$OT_j$  – the time interval that bus  $j$  actually departs;

$X_{jf} = \begin{cases} 1, & \text{if bus } j \text{ is allocated to wait temporarily on floor } f; \\ 0, & \text{otherwise.} \end{cases}$

This research aims to minimize the total time interval delay  $TD$  by optimizing the bus waiting floor allocation. The mathematical model can be described as follows:

min  $TD$ :

$$TD = \sum_{j=1}^J OT_j - DT_j; \tag{1}$$

$$IT_j \geq AT_j, j = 1, 2, \dots, J; \tag{2}$$

$$OT_j \geq DT_j, j = 1, 2, \dots, J; \tag{3}$$

$$OT_j \geq IT_j + PT, j = 1, 2, \dots, J; \tag{4}$$

$$(E - AI_{jt}) \cdot (t - IT_j - E) \leq 0, j = 1, 2, \dots, J; \\ t = 1, 2, \dots, T; \tag{5}$$

$$(BP_{jt} - E) \cdot (t - OT_j + PT - E) \leq 0, j = 1, 2, \dots, J; \\ t = 1, 2, \dots, T; \tag{6}$$

$$\sum_{f=1}^F X_{jf} = 1, j = 1, 2, \dots, J; \tag{7}$$

$$(E - BO_{jt}) \cdot (OT_j - t + E) \leq 0, j = 1, 2, \dots, J; \\ t = 1, 2, \dots, T; \tag{8}$$

$$(E - AP_{jt}) \cdot (t - OT_j + PT - E) \leq 0, j = 1, 2, \dots, J; \\ t = 1, 2, \dots, T; \tag{9}$$

$$RC_{jf} \geq \sum_{j=1}^J AI_{jt} \cdot BP_{jt} \cdot X_{jf} + BO_{jt} \cdot AP_{jt} \cdot Y_{jf}, \\ j = 1, 2, \dots, J; t = 1, 2, \dots, T; \tag{10}$$

$$IT_j = \text{positive integer}, j = 1, 2, \dots, J; \tag{11}$$

$$OT_j = \text{positive integer}, j = 1, 2, \dots, J; \tag{12}$$

$$X_{jf} = \{0,1\}, j = 1, 2, \dots, J; f = 1, 2, \dots, F. \tag{13}$$

The right hand side of Equation (1) is used to calculate the number of time intervals that each bus is delayed and the whole of Equation (1) is used to calculate total delay of all buses. When all floors are full of buses (no capacity remains) the newly arrived buses are not permitted to drive into the station. Equation (2) ensures that buses cannot drive into the station before they arrive, and Equation (3) ensures that buses cannot drive out of the station before their planned departure time. After all buses move into the station, preparation time is required for a bus to depart, and the corresponding constraint is shown in Equation (4). Equation (5) is used to determine if time interval  $t$  is later than the time interval when the bus  $j$  drives into the station. Equation (6) is used to determine if time interval  $t$  is earlier than the arrival of bus  $j$  at its departure platform. Equation (7) is used to determine whether bus  $j$  is allocated to wait on floor  $f$  and to ensure that only one floor can be allocated for each bus. Equation (8) is used to determine if time interval  $t$  is earlier than bus  $j$  actually departs. Equation (9) is used to determine if time interval  $t$  is later than the arrival of bus  $j$  at its departure platform. The remaining capacity of each floor for every time interval is a constraint for the bus allocation as shown in Equation (10). Equations (11)–(13) specify the bounds on each decision variable.

## 2. Proposed methodology

VNS is a meta-heuristic based upon systematic changes of neighbourhoods that combine a descent phase, to find a local minimum, and a perturbation phase, to emerge from the corresponding valley (Hansen *et al.* 2010). In the descent phase, the proposed VNS initially generates a pool of solutions. Then a solution will be selected from the solution pool based on its performance. Based on the selected solution, a new solution is generated by one of several strategies. The better a solution is, the higher the probability of it being selected to generate new solutions. As regards the perturbation phase, a worse solution still has a chance of being selected. The new solution will replace the worst solution in the original solution pool. Finally, when the search process is terminated, the best solution in the final solution pool will be the final solution.

The procedure of the proposed methodology for the bus waiting allocation is illustrated in Figure 1. The procedure comprises four phases: generating an initial population, selecting a solution from the solution pool, generating a new solution and terminating the search procedure. The details of these phases are described in the following sections.

### 2.1. Generating the initial population

In order to diversify the searching space of the proposed VNS, this research adopted the concept of Genetic Algorithms to generate an initial population of possible solutions. The proposed methodology for generating the first solution assumes that all buses can move into the bus station once they arrive; with the required preparation time

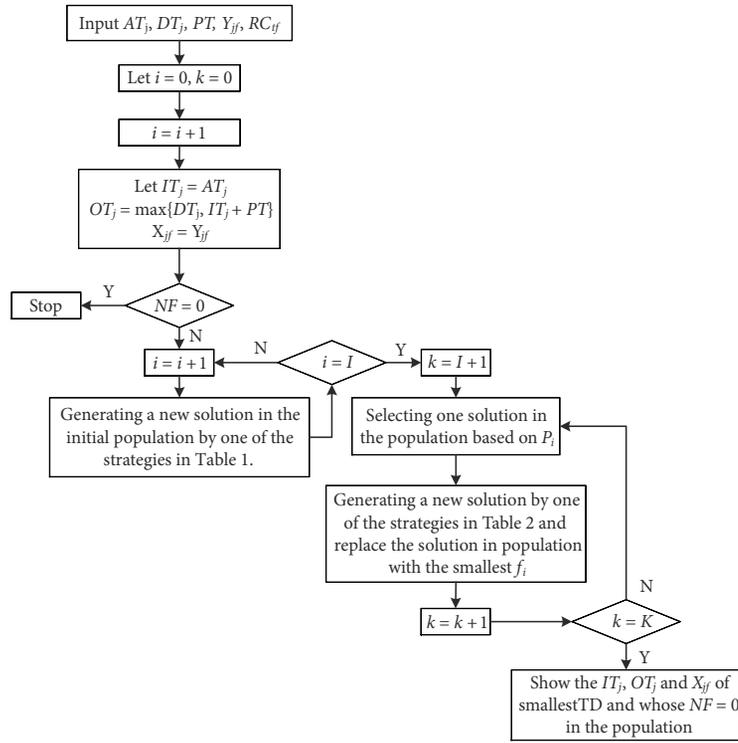


Figure 1. The procedure of the proposed methodology

interval, all buses can depart at the planned departure time; and all buses are allocated to wait on the floor that they are planned to depart from. Therefore, for the first solution is  $IT_j = AT_j$ ,  $OT_j = \max\{DT_j, IT_j + PT_j\}$  and  $X_{jf} = Y_{jf}$ . It should be noted that if the first solution is feasible, that means the remaining capacities of the floors that the buses are planned to depart from are enough, and all buses can move to their departure floor immediately when they arrive. Consequently, all buses can depart as earlier as possible. Thus, the first solution is the optimal solution, if it is feasible. The evaluation of whether the solution is feasible or not will be addressed in Section 2.2.

Based on the first solution, the proposed methodology generates the remaining solutions of the initial population by multiple strategies. When generating a new solution for the initial population, one bus  $j$  is selected at random and one of five strategies is executed as shown in the Table 1. The initial population is completed when  $I$  solutions (including the first solution) are generated.

**2.2. Selecting solutions**

The objective of the proposed approach is to minimize the total time delay of all buses,  $TD$ . A solution with a smaller  $TD$  means better performance. However, not all generated solutions are feasible. Infeasible solutions violate the limitation of Equation (10). The right hand side of Equation (10) is the required capacity of floor  $f$  in time interval  $t$ . Let  $IF_{tf} = \max\{0, AI_{jt} \cdot BP_{jt} \cdot X_{jf} + BO_{jt} \cdot AP_{jt} \cdot Y_{jf} - RC_{jf}\}$ .

$IF_{tf}$  indicates that there is insufficient capacity on floor  $f$  in time interval  $t$ . Then the total shortfall in capacity,  $NF$ ,

Table 1. Strategies for generating initial population of solutions

No	Strategy	Description
1	Let $IT_j = IT_j + 1$ , $OT_j = \max\{IT_j + PT, OT_j\}$	Let the selected bus move into the station later by one time interval
2	Let $IT_j = IT_j + 2$ , $OT_j = \max\{IT_j + PT, OT_j\}$	Let the selected bus move into the station later by two time interval
3	Let $OT_j = OT_j + 1$	Let the selected bus depart later by one time interval
4	Let $OT_j = OT_j + 2$	Let the selected bus depart later by two time intervals
5	Select one $X_{jf} = 0$ randomly, let it be 1 and all others be 0.	Change the bus waiting floor

can be calculated by Equation (14):

$$NF = \sum_{t=1}^T \sum_{f=1}^F IF_{tf}. \tag{14}$$

A solution with smaller  $NF$  means the solution is less infeasible. Only when  $NF = 0$  is the solution feasible. For a solution  $i$ , the proposed methodology combines  $TD$  and  $NF$  into one value  $Z_p$ , which can be calculated by Equation (15):

$$Z_i = a \cdot \frac{1}{NF + 1} + b \cdot \frac{1}{TD + 1}. \tag{15}$$

Equation (15) can be divided into two parts. The first part,  $\frac{1}{NF + 1}$ , presents the level feasibility of a solution.

Since  $NF \geq 0$  and the smaller  $NF$  the less infeasible, the value of  $\frac{1}{NF+1}$  is smaller than or equal to 1. The second part of Equation (15) is  $\frac{1}{TD+1}$ . Like  $NF$ ,  $TD \geq 0$  and the smaller  $TD$  the better the solution. Therefore, the value of  $\frac{1}{TD+1}$  is smaller or equal to 1. The two parts of Equation (15) are weighted by  $a$  and  $b$  respectively. In this research the sum of  $a$  and  $b$  is 1 and  $0 \leq a \leq 1, 0 \leq b \leq 1$ . A solution with higher  $Z_i$  means its performance is better.

When the performance of all solutions in the population has been calculated, the proposed methodology adopts the roulette-wheel principle (Goldberg 1989) to select one solution to generate a new solution. If there are  $I$  solution in the population, the probability  $P_i$  that the solution  $i$  will be selected is shown in Equation (16):

$$P_i = \frac{Z_i}{\sum_{i=1}^I Z_i} \tag{16}$$

Based on the probability calculated by Equation (16), a better solution would have a higher probability of being selected. However, it should be noted that a solution with a worse performance still has a chance of being selected.

### 2.3. Generating new solutions

When a solution is selected, this research proposes five strategies for generating new solutions. When generating a new solution, one bus  $j$  is selected at random and one of five strategies is executed as shown in Table 2. This is similar to VNS, which uses multiple strategies to search for neighbourhood solutions. In the proposed methodology, the generated new solution will replace the solution in the original population with the worst performance, as indicated by  $Z_i$ .

### 2.4. Terminating criteria

The entire searching process is terminated, when the maximal number of solutions  $K$ , are generated (including the solutions in the initial population). After the searching process is terminated, the final solution will be selected from the final population and the solution with the smallest  $TD$  and  $NF = 0$  will be selected for the final solution.

## 3. Empirical illustrations

A case study of the Taipei Bus Station (Taiwan) is adopted for the empirical illustration. Taipei Bus Station is a main transportation hub for over 50 bus routes to eastern, central, and southern Taiwan, with a daily volume of approximately 2500 scheduled buses, serving over 45000 passengers daily (Cheng *et al.* 2012). In 2013, approximately 2800 buses were scheduled each weekend. All buses arrive at or depart from the station for long distance transportation. Therefore, buses may arrive much earlier or later than is planned due to the traffic jams, which are common. Due to the limitation of available space, Taipei Bus Station is built as a multi-floor station. Even so, the capacity is not always sufficient for the demand. For example, during the Chinese New Year, passengers who take the buses from other cities and want to arrive at Taipei Bus Station are only allowed to alight outside the station to keep the number of buses in the station as small as possible. However, this causes the traffic around the station to become worse. Therefore, the waiting allocation decision is a very important issue for Taipei Bus Station.

Cheng *et al.* (2012) introduced the structure of the Taipei Bus Station as follows. The concourse on the ground floor of this bus terminal has ticket counters and a bus information center as well as several gift and souvenir shops. The central areas of the second, third, and fourth floors are passenger waiting rooms. For example, a bird’s eye view of the third floor is illustrated in Figure 2. Buses drive around the waiting rooms within a bus lane inside the building. Each floor has 16 platforms surrounding the waiting room. Passengers pass through the gates between the waiting room and the platform to board buses or debus. Buses enter the building and drive to the second floor lane, and exit the station from the third floor taking a route to the ground floor or exiting directly onto an elevated expressway from the third floor via a connecting overpass.

Figure 2 shows that buses can drive to the third floor from the second and fourth floors and buses can drive to the fourth floor from the third floor. However, buses cannot drive to the second floor from the third floor. This means that if a bus is planned to depart from the second floor, it cannot be allocated to wait on the third or fourth floors, as the bus cannot drive back to the second floor.

Table 2. Strategies for generating new solutions

No	Strategy	Description
1	Let $IT_j = IT_j + 1$ , $OT_j = \max\{IT_j + PT, OT_j\}$	Let the selected bus move into the station later by one time interval
2	Let $IT_j = \max\{IT_j - 1, AT_j\}$ , $OT_j = \max\{IT_j + PT, OT_j\}$	Let the selected bus move into the station earlier by one time interval
3	Let $OT_j = OT_j + 1$	Let the selected bus delay departure by one time interval
4	Let $OT_j = \max\{OT_j - 1, IT_j + PT, DT_j\}$	Let the selected bus depart earlier by one time interval
5	Select one $X_{ff} = 0$ randomly, let it be 1 and all others be 0.	Change the bus waiting floor

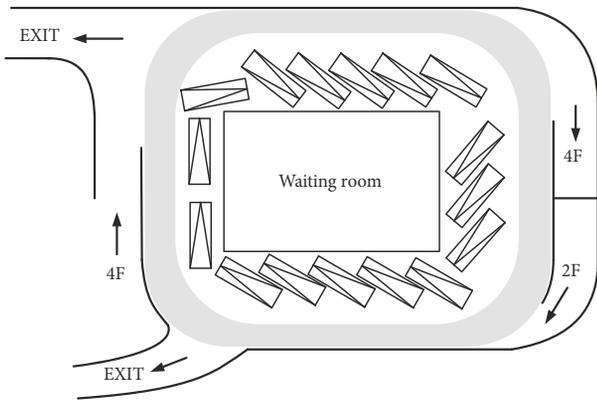


Figure 2. The third floor layout of the Taipei Bus Station

Therefore, the buses, which are planned to depart from the second floor, are ignored in this case study.

More than ten bus companies schedule their buses to provide transportation services in Taipei Bus Station. The statuses of all the buses in the station are collected by a Radio Frequency Identification (RFID) system. The information is visualized in the control center, as illustrated in Figure 3. Through the interface of the monitoring system, the control center can know the locations, licenses and owning companies, arrival time, departure time and duration of stay in the station of all buses. Therefore, it is easy to know the number of buses on each floor and to estimate the remaining capacity of each floor over several time intervals.

The capacity limitations of third and fourth floor are 25 and 20 buses respectively. This means that when there are more than 16 buses waiting on the same floor, only 16 buses at most can stop at their platforms and wait, the remainder have to drive around the waiting rooms. The current strategy for arriving buses is to drive directly to the floor that they are planned to depart from, no matter how many buses are already waiting on that floor. If too

many buses drive around the waiting room on a floor, the spaces between buses become too small to drive easily. Thus, the buses inside cannot drive out easily and buses outside cannot drive in easily. This reduces the efficiency of the bus station.

The proposed integer-programming solution aims to optimize the allocation of temporary waiting floors to all newly arrived buses every 20 minutes, as the control center can estimate the arrival times and departure times exactly for the next 20 minutes. The time interval is two minutes. This research generates test problems using three parameters – number of buses  $\# buses$ , average departure time  $ADT$  and standard deviation of departure time  $S$ . Number of buses means the number of buses that arrive in the next 20 minutes, and need to be allocated to a temporary waiting floor. The greater the number of buses, the more crowded the bus station.  $ADT$  means the average planned departure time from now for the buses that arrive in the next 20 minutes. In this case study,  $PT = 0$ . That means 16 minutes are required for a bus to prepare to leave after it arrives at its platform. If the average  $ADT$  is smaller, more buses would be delayed.  $S$  is the standard deviation of  $ADT$ . The test problems are generated based on the first three columns in Table 3. In Table 3, it can be judged that delays are most serious when  $S$ ,  $\# buses$  and  $ADT$  are 3, 20 and 7 respectively.

The proposed methodology is developed using the C# programming language and tested on a Core i5 2.90 GHz personal computer. In the proposed VNS, four parameters are required to be decided. They are population size  $I$ , maximum number of solutions  $K$ , the weights in Equation (15)  $a$  and  $b$ . The setting of these parameters will affect the performance of the proposed VNS. This research firstly tests different population sizes for the problem so that the  $S$ ,  $\# buses$  and  $ADT$  are set as 3, 20 and 7. Each population size is tested ten times. The average convergence for different population sizes are shown in Figure 4.



Figure 3. The interface of the monitoring system (third floor) (in Chinese)

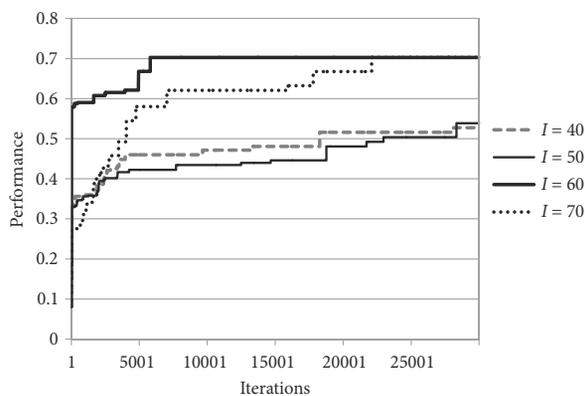


Figure 4. Convergence for different population sizes

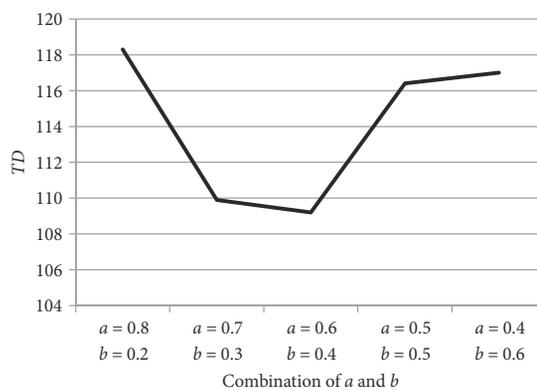


Figure 5. The performance for different combinations of  $a$  and  $b$

The results in Figure 4 are based on  $K = 30000$ ,  $a = 0.7$  and  $b = 0.3$ . It is found that when the population size is 60, the convergence speed is the fastest and the method can converge within 20000 iterations. Consequently, this research set  $I$  and  $K$  as 60 and 20000 respectively to test different combinations of  $a$  and  $b$  based on the same test problem for ten times. The average results are illustrated in Figure 5. It is found that the performance is the best when  $a = 0.6$  and  $b = 0.4$ . Accordingly,  $I$ ,  $K$ ,  $a$  and  $b$  are set as 60, 30000, 0.6 and 0.4 respectively.

The results are shown in Table 3. All optimal solutions are found within 2 seconds. A scenario called ‘Current Strategy’, in which all buses wait on the floor that they are planned to depart from, is used for comparison. In the Current Strategy,  $X_{jf}$  is not a decision variable and equal to  $Y_{jf}$  for  $j = 1, 2, \dots, J$ ;  $f = 1, 2$ . If there is no remaining capacity on the floor that they are planned to depart from, then the bus cannot drive into the station.

In Table 3, there are a total of 18 scenarios. For each scenario, 5 test problems are generated at random. One of the test problems and its corresponding results are shown in Table 4. It shows that there is one bus ( $j = 1$ ) waiting temporarily on a floor that they are not planned to depart

Table 3. The computational results

S	# buses	ADT [time intervals]	TD [time intervals]		Improvement
			Current strategy	Proposed methodology	
1.5	20	7	138.4	124.0	10.40%
		9	100.2	82.0	18.16%
		11	74.8	53.8	28.07%
	15	7	80.2	78.6	2.00%
		9	56.8	40.8	28.17%
		11	25.8	21.6	16.28%
	10	7	54.8	44.0	19.71%
		9	25.6	24.8	3.13%
		11	27.0	14.2	47.41%
3.0	20	7	142.8	129.2	9.52%
		9	100.0	84.0	16.00%
		11	81.8	65.4	20.05%
	15	7	101.6	81.4	19.88%
		9	58.2	54.2	6.87%
		11	42.0	29.6	29.52%
	10	7	68.2	49.6	27.27%
		9	53.8	35.4	34.20%
		11	22.8	19.2	15.79%

Table 4. An example of a test problem and its corresponding results

	Test problem				Corresponding results				
	$Y_{jf}$		$AT_j$	$DT_j$	$X_{jf}$		$IT_j$	$OT_j$	$OT_j - DT_j$
	$f = 1$	$f = 2$			$f = 1$	$f = 2$			
$j = 1$	0	1	1	12	1	0	1	12	0
$j = 2$	0	1	1	12	0	1	1	12	0
$j = 3$	0	1	1	10	0	1	2	10	0
$j = 4$	0	1	2	10	0	1	2	10	0
$j = 5$	1	0	2	12	1	0	2	12	0
$j = 6$	1	0	2	11	1	0	2	11	0
$j = 7$	0	1	2	15	0	1	3	11	1
$j = 8$	1	0	4	11	1	0	4	12	1
$j = 9$	1	0	5	10	1	0	5	13	3
$j = 10$	1	0	5	10	1	0	5	13	3
$RC_{t1} = (0, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 25)$									$TD = 8$
$RC_{t2} = (0, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 21, 21, 21)$									-

from and the total time interval delay is 8. In Table 3 the average results for each scenario are presented. It is found that, whether the Current Strategy or the proposed methodology is used, if there are more buses with a shorter time to departure from the current time, then buses will face more serious delays. Moreover, the proposed methodology can reduce the total delay for buses in every scenario.

It is worth noting that the proposed integer-programming formulation is an NP-hard problem. Any increase in the number of time intervals or floors will increase the problem size and may make the problem become computationally prohibitive. Nevertheless, this research successfully formulated the bus waiting allocation problem for a multi-floor bus station. A practical case study illustrated the effectiveness of the proposed method.

### Conclusions and discussion

This research deals with a multi-floor bus station, which has insufficient space. In order to optimize space utilization between different floors, an integer-programming model is proposed to minimize the total delay. The integer-programming model allows buses to wait on a floor, which is different from the floor that they are planned to depart from. Taipei Bus Station is used as an illustration for the case study. A VNS is proposed to solve the problem. Unlike general VNS, the proposed VNS adopts the concept of population in GA, and then selects a solution from the population to generate neighbourhood solutions. The empirical results showed the effectiveness of the proposed methodology for a practical application. It can potentially increase customer satisfaction. However, it would also increase the number of occasions on which a bus was required to move from one floor to another, and hence increase the emission of waste gas. This may also be a future research opportunity for further investigation.

Today, multi-floor bus stations are not very common in cities in the world. The application of the proposed methodology is possibly rather limited. However, this research shows an opportunity for improving the performance of a traffic center based on location technologies. The proposed methodology can not only reduce the delay of buses but also provide more accurate information about departure times for passengers. This approach can integrate many management issues to improve customer satisfaction for all kinds of traffic centers, and the results may not be limited to multi-floor traffic centers.

### References

- Cheng, Y.-H.; Chang, H.-P.; Yan, J.-W. 2012. Temporal variations in airborne particulate matter levels at an indoor bus terminal and exposure implications for terminal workers, *Aerosol and Air Quality Research* 12(1): 30–38. <https://doi.org/10.4209/aaqr.2011.06.0085>
- Cortés, C. E.; Jara-Díaz, S.; Tirachini, A. 2011. Integrating short turning and deadheading in the optimization of transit services, *Transportation Research Part A: Policy and Practice* 45(5): 419–434. <https://doi.org/10.1016/j.tra.2011.02.002>
- Dessouky, M.; Hall, R.; Nowroozi, A.; Mourikas, K. 1999. Bus dispatching at timed transfer transit stations using bus tracking technology, *Transportation Research Part C: Emerging Technologies* 7(4): 187–208. [https://doi.org/10.1016/S0968-090X\(99\)00019-4](https://doi.org/10.1016/S0968-090X(99)00019-4)
- Dessouky, M.; Hall, R.; Zhang, L.; Singh, A. 2003. Real-time control of buses for schedule coordination at a terminal, *Transportation Research Part A: Policy and Practice* 37(2): 145–164. [https://doi.org/10.1016/S0965-8564\(02\)00010-1](https://doi.org/10.1016/S0965-8564(02)00010-1)
- Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional. 432 p.
- Hall, R.; Dessouky, M.; Lu, Q. 2001. Optimal holding times at transfer stations, *Computers & Industrial Engineering* 40(4): 379–397. [https://doi.org/10.1016/S0360-8352\(01\)00039-0](https://doi.org/10.1016/S0360-8352(01)00039-0)
- Hansen, P.; Mladenović, N.; Pérez, J. A. M. 2010. Variable neighbourhood search: methods and applications, *Annals of Operations Research* 175(1): 367–407. <https://doi.org/10.1007/s10479-009-0657-6>
- Liu, Z.; Yan, Y.; Qu, X.; Zhang, Y. 2013. Bus stop-skipping scheme with random travel time, *Transportation Research Part C: Emerging Technologies* 35: 46–56. <https://doi.org/10.1016/j.trc.2013.06.004>
- Lo, S.-C.; Chang, W.-J. 2012. Design of real-time fuzzy bus holding system for the mass rapid transit transfer system, *Expert Systems with Applications* 39(2): 1718–1724. <https://doi.org/10.1016/j.eswa.2011.08.111>
- Tirachini, A.; Cortés, C. E.; Jara-Díaz, S. R. 2011. Optimal design and benefits of a short turning strategy for a bus corridor, *Transportation* 38(1): 169–189. <https://doi.org/10.1007/s11116-010-9287-8>
- Yan, S.; Chen, H.-L. 2002. A scheduling model and a solution algorithm for inter-city bus carriers, *Transportation Research Part A: Policy and Practice* 36(9): 805–825. [https://doi.org/10.1016/S0965-8564\(01\)00041-6](https://doi.org/10.1016/S0965-8564(01)00041-6)
- Yu, B.; Lam, W. H. K.; Tam, M. L. 2011. Bus arrival time prediction at bus stop with multiple routes, *Transportation Research Part C: Emerging Technologies* 19(6): 1157–1170. <https://doi.org/10.1016/j.trc.2011.01.003>
- Yu, B.; Yang, Z. 2009. A dynamic holding strategy in public transit systems with real-time information, *Applied Intelligence* 31(1): 69–80. <https://doi.org/10.1007/s10489-007-0112-9>
- Yu, B.; Yang, Z.; Li, S. 2012. Real-time partway deadheading strategy based on transit service reliability assessment, *Transportation Research Part A: Policy and Practice* 46(8): 1265–1279. <https://doi.org/10.1016/j.tra.2012.05.009>
- Zolfaghari, S.; Azizi, N.; Jaber, M. Y. 2004. A model for holding strategy in public transit systems with real-time information, *International Journal of Transport Management* 2(2): 99–110. <https://doi.org/10.1016/j.ijtm.2005.02.001>