A Memetic Algorithm to Optimize Bus Timetable with Unequal Time Intervals

Pengfei Gao
School of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China
gpf17861@163.com

Xingquan Zuo
School of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China
zuoxq@bupt.edu.cn

Qiang Bian
General Department
Aerospace Science and Industry Smart Industry Development Co., Ltd.
Beijing, China
bianqiang@aiidc.com.cn

Xinchao Zhao
School of Science
Beijing University of Posts and Telecommunications
Beijing, China
xcbupt@126.com

ABSTRACT

An appropriate bus timetable is vital for bus enterprises to improve service quality and save operational cost. Most existing literature on bus timetable optimization divide a whole day into several periods and assume that departure time intervals are the same for each period. As passengers' flow varies over time in a period, giving the same time interval for each period cannot really meet the demand of passengers. In this paper, we study the optimization of bus timetable with unequal time intervals. Aiming at characteristics of this problem, a memetic algorithm is devised that combines a genetic algorithm with elite strategy and a local search. To handle infeasible solutions, a repair method is proposed to repair solutions that do not meet the constraint. A new metric reflecting the degree of bus carrying capability matching passengers' need is introduced. The metric together with passengers' waiting time is used to evaluate a bus timetable. Experiments show that compared to the actually used timetables, timetables optimized by the proposed approach are able to save about 4.54-12.84% cost and 11.87-37.76% passengers' waiting time.

CCS CONCEPTS

• **Mathematics of computing → Combinatorial optimization**; • **Computing methodologies → Planning and scheduling**; • **Applied computing → Operation research**;

KEYWORDS

Bus timetable, Genetic algorithm, Local search, Memetic algorithm.

ACM Reference format:
Pengfei Gao, Xingquan Zuo, Qiang Bian and Xinchao Zhao. 2019. A Memetic Algorithm to Optimize Bus Timetable with Unequal Time Intervals. In *GECCO '19 Companion: Genetic and Evolutionary Computation Conference Companion, July 13–17, 2019, Prague, Czech Republic*. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3319619.3326844.

1 INTRODUCTION

Public transportation is an important part of urban transportation systems. The bus timetable is directly related to the quality of service and operational cost of bus enterprises. An appropriate timetable is able to meet passengers' need, save operational cost and promote the service level of bus companies [1-4].

Bus timetable optimization is to determine the bus headway (departure time interval). Existing approaches on the optimization of bus timetables typically divide the time of a day into multiple periods, and consider the vehicles departs with an equal time interval (fixed departure frequency) in each period [7-15]. It means that each period in the timetable has the same time interval and time intervals of different periods are different.

Bus timetable is based on passenger's flow of a bus line. The adjacent two times in the timetable have a large (small) time interval if passengers' flow at that time is small (large). Those typical approaches facilitate the optimization of timetables since the number of decision variables (time intervals) to be optimized are quite limited. However, giving the same time interval (departure frequency) for each period cannot really reflect the demand of passengers, since passengers' flow in a period usually changes over time. That will result in the increase of operational cost and prolonging waiting time of passengers. In addition, it is not easy to divide a whole day into several reasonable time periods to make each period has similar passengers' flow.

In this paper, we propose an approach to optimize bus timetables with unequal time intervals (bus headway). Optimizing bus timetables with unequal intervals is much more complex than the optimization of timetables with equal ones. A memetic algorithm combining a genetic algorithm and a local search is devised for the problem. To handle infeasible solutions generated during the search procedure, a solution repair method is proposed to repair infeasible solutions to make them become feasible (satisfy the constraint).

In existing literature, the operational cost of bus companies and the passengers' waiting time are two common metrics to evaluate a bus timetable [8-14]. In practice, the departure frequency (time interval) is usually determined by vehicles' carrying capability of satisfying the need of passengers. That is, the departure frequency is expected to be as smaller as possible under the condition of satisfying the need of passengers' comfort and waiting time. A small departure frequency means saving the operational cost. It is quite a reasonable method to making bus timetable, but there are no literature using it as a metric to optimize bus timetable. In this paper, we introduce this metric as an optimization objective to evaluate the quality of timetable.

The paper is organized as follows. Section 2 reviews the related literature. Section 3 presents the problem of bus timetable optimization. The memetic algorithm is presented in Section 4. Section 5 gives experimental results. Finally, conclusions are drawn in Section 6.

2 RELATED WORK

Bus timetable making is based on passengers' flows, which are much different in different periods within a day. The passengers' flow is large in rush (peak) hours and small in off-peak hours. A small time interval (high departure frequency) should be given for large passengers' flow, and a large time interval (low departure frequency) for small passenger's flow.

Existing studies typically divide a day into several periods and give a fixed departure time interval for each period, based on the assumption that passengers' flow is the same at every moment in this period [5-15]. Lampkin *et al.* [5] designed a heuristic to provide a route network and service frequencies to maximize the service quality of passengers. Four methods were presented by Ceder *et al.* [6] to obtain the bus frequency: two methods are based on point check (maximum load) data and the other two use ride check (load profile) data. Oudheusden *et al*. [7] developed an integer program model and two heuristics to design the departure frequency: one is based on a linear program and the other is a straightforward derivation of common bus operation practice. Sun *et al.* [8] provided a departing time interval control model, considering generalized trip cost of passengers and bus enterprises' operational cost. Luhua *et al.* [9] used the sum of the operation cost and passengers' cost as the objective function, and presented a genetic algorithm to optimize bus timetable, with consideration of the influence of signal lamp on the waiting time. Tang *et al.* [11] presented a timetable optimization model based on the trade-off between the cost of bus operation and benefit of passengers and used a quantum genetic algorithm to solve the model. Zhu *et al.* [12] presented an integer program model and used a branch and bound algorithm to solve it. Qian *et al.* [13] designed a hybrid algorithm combining a genetic algorithm and a tabu search for the problem. Dong *et al.* [14] presented a departure time interval transition method, which considers passengers' travel demand and traffic congestion. Ceder *et al.* [15] presented a methodology to approach even-headway and even-load timetables by utilizing different bus sizes. The methodology uses a graphical heuristic to examine different strategies during the optimization of timetables. About methods lead to the problem of how to divide reasonable periods in a day. There are researches on the time periods division. For example, Big *et al.* [16] proposed an algorithm based on GPS data to partition bus operating hours into several time intervals.

Instead of giving a fixed departure frequency for each period, some studies consider a day as one period and give the same departure frequency for a day [10]. Yu *et al.* [10] presented a parallel genetic algorithm combined with a coarse-grained strategy and a local search to optimize a bus timetable with equal headway.

There are studies that do not divide a day into periods and optimize the timetable with unequal time intervals. That is, the time interval may be different for each pair of adjacent times in bus timetable. Such studies are scarce, and we find two literatures on this topic. They optimize a short segment (one or two hours) of a bus timetable, and do not optimize the bus timetable in a whole day. Sun *et al.* [17] proposed a heuristic algorithm for flexible timetable optimization. A hybrid vehicle size model is used to tackle the demand fluctuations in transit operations. They only select one hour of peak and one hour of off-peak to verify their approach. Aiming to minimize passenger waiting time and maximize the number of passengers a bus carries, Li *et al.* [18] proposes two algorithms, a hybrid particle swarm optimization and a genetic algorithm to optimize the bus timetable. Only one hour of the timetable is optimized. There exist quite limited researches on the optimization of bus timetable with unequal time interval. Existing researches consider the optimization of a small segment of a bus timetable and do not optimize the whole timetable, which significantly simply the problem and may not be effective for a real-world bus timetable optimization.

3 THE BUS TIMETABLE OPTIMIZATION PROBLEM

Generally, a bus line has two control points (a starting station and a terminal station), each of which has a departure timetable. Thus, a bus line has two timetables: one is for upward direction and the other for the downward direction. Optimizing a bus timetable is to determine the time interval of each pair of adjacent times in the timetable. In other words, it is to determine the departure frequency of vehicles. Bus timetable making is an important issue for bus enterprises since a reason timetable is able to save the operational cost and meanwhile improve the service quality.

The optimization of the bus timetable of a bus line is based on the requirement of passengers taking the bus line. Such requirement can be reflected by passengers' flows amongst each pairs of stations in the bus line, which is denoted by an origin-destination (OD) matrix of passengers' flows. The bus timetable optimization is based on the OD matrix of passengers' flows. The timetable should have a large (small) time interval at the time when the passengers' flow is small (large). As the passengers' flow of a bus line may vary and therefore may be different at different moments. Correspondingly, time intervals at different moments in the bus timetable should not be equal. In this paper, we consider the optimization of the bus timetable with unequal time intervals.

4 PROPOSED SOLUTION APPROACH

Compared to the optimization of bus timetable with equal time intervals, optimizing bus timetable with unequal time intervals is more complex due to the sharp increase of decision variables. For the timetable with equal time intervals, a day is partitioned into several periods, and the number of time intervals to be optimized equals the number of periods. However, for the bus timetable with unequal time intervals, the number of time intervals to be determined equals the number of times in the timetable, which is much greater than the partitioned periods. Due to the huge solution space of such problem, an effective search algorithm is needed to solve it.

Memetic algorithm (MA) is a type of optimization algorithms that combine evolutionary algorithms and local searches. It uses a population-based evolutionary algorithm to perform global search to explore excellent regions in the solution space, and uses an individual-based local search to find good solutions in local regions. MA been proved to be very effective for both of continuous and combinatorial optimization problems.

In this paper, we propose a bus timetable optimization approach based on a memetic algorithm (BTOA-MA) to optimize a bus timetable with unequal time intervals. A genetic algorithm with an elite strategy is used to perform global search, and a local search is devised and embedded into the genetic algorithm. The flow chart of BTOA-MA is shown in Fig. 1. First, each individual in the initial population $P(1)$ is generated randomly. The population consists of N individuals. Then, calculate objective function values of individuals in current population $P(g)$. The objective function calculation can be found in Section 4.1. Subsequently, the selection, crossover, and mutation operations are conducted on the population. If the number of generations, g, is a multiple of L_f, then the local search is performed. It means that the local search is conducted every L_f generations of population evolution. Above procedure is repeated until the maximum number of generations, G_{max}, is reached.

The genetic algorithm adopts an elite strategy to prevent the population degeneration. An elite individual is the best individual found so far. If the best individual in the current population is better than the elite individual, then replace the elite individual with the best individual; otherwise, replace the worst individual in the current population with the elite individual. Note that the local search is only applied to the elite individual, not to each individual in the population, to improve the search efficiency.

During the search procedure, some individuals that do not meet the problem's constraint may be generated. Those individuals are considered as infeasible solutions. A repair method is proposed by adjusting the departing times in the timetable to make infeasible individuals become feasible.

The evaluation function will be introduced next, followed by the genetic operations and the local search.

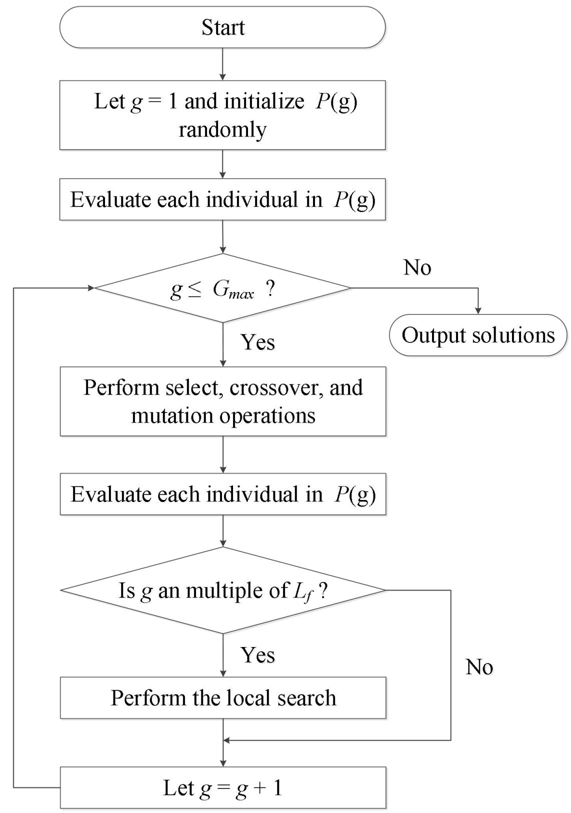

Figure 1: Flow chart of BTOA-MA

4.1 Evaluation function design

Evaluation function is a key issue for a meta-heuristic to optimization a bus timetable. In this paper, a new metric based on carrying capability is introduced to evaluate the bus timetable of a bus line.

4.1.1 Carrying capability to match passengers' flow. Assume that a bus line consists of S stations and that the distance between station i and j be $L_{i,j}$, $i,j \in \{1, 2, \ldots, S\}$ and $i \neq j$. Suppose that the bus line uses vehicles of same size and the number of seats in each vehicle is C. The minimum unit in a bus timetable is minute, and one vehicle must depart at each departure time in the timetable. A timetable must contain the time when the service begins (ends) in a day, to ensure the service quality.

The carrying capacity of a vehicle is defined as the travel distance of all passengers on that vehicle. For a bus timetable Z with M times, assume that the mth time in the timetable is Z_m, $m \in \{1, 2, \ldots, M\}$. The carrying capability of a vehicle departing at each time of Z_m is given by

$$E_m = \alpha \times L \times C \tag{1}$$

where L is the length of the bus line; α is a coefficient reflecting the comfort level of passengers in the vehicle and is given 1.5 here.

Suppose that there are K times in the actually used timetable. Let T_k be the kth time in the timetable, $k \in \{1, 2, \ldots, K\}$, and let d_{ki} (h_{ki}) be the number of passengers getting on (off) the vehicle departing from T_k at station i. From station (*i-1*) to i, the number of passengers on the vehicles that actually depart in the interval [Z_{m-1}, Z_m] can be calculated by

$$O_{mi} = \sum_{k \epsilon \{k | Z_{m-1} \leq T_k < Z_m\}} \sum_{j=1}^{i} \left(d_{kj} - h_{kj}\right) \tag{2}$$

The total travel distance of passengers on the vehicles departing in the period of [Z_{m-1}, Z_m] can be calculated by

$$O_m = \sum_{i=1}^{S-1} L_{i,i+1} O_{mi} \tag{3}$$

Eq. (3) reflects the actual need of passengers on carrying capability in the period of [Z_{m-1}, Z_m]. Recall that E_m is the carrying capability of a vehicle. Since one vehicle departs in [Z_{m-1}, Z_m], E_m is the carrying capability provided by the timetable in the period. The different between O_m and E_m is expected to be as smaller as possible. It means that we expect using a timetable with the minimum number of departure times to satisfy the need of passengers taking buses. A small number of departure times in a timetable indicates low operation cost. Thus, the following metric in (4) is suggested to evaluate a bus timetable.

$$O = \sum_{m}^{M-1} E_m - O_m \tag{4}$$

To demonstrate the actual need of passengers on carrying capability and carrying capability provided by a bus timetable, the service time of a real-world bus line (upward direction of the bus line 18 in experiment part) is partitioned into time slices of half an hour. Using O_m in (3), the actual need of carrying capability in each time slice $[t_a, t_b]$ can by calculated by

$$\sum_{m \epsilon \{k | t_a \leq Z_k < t_b\}} O_m \tag{5}$$

Similarly, we can calculate the carrying capability provided by the timetable in the slice $[t_a, t_b]$.

$$\sum_{m \epsilon \{k | t_a \leq Z_k < t_b\}} E_m \tag{6}$$

Figure 2 shows the actual need on carrying capability in (5) and the carrying capability provided by the timetable in (6) for each time slice (half an hour). The horizontal ordinate represents the time slices. The value of carrying capability is marked in the beginning of each time slice. For example, the carrying capability provided by the timetable from 6:00 AM to 6:30 AM is 4563.8, and the needed carrying capability in this period is 938.61.

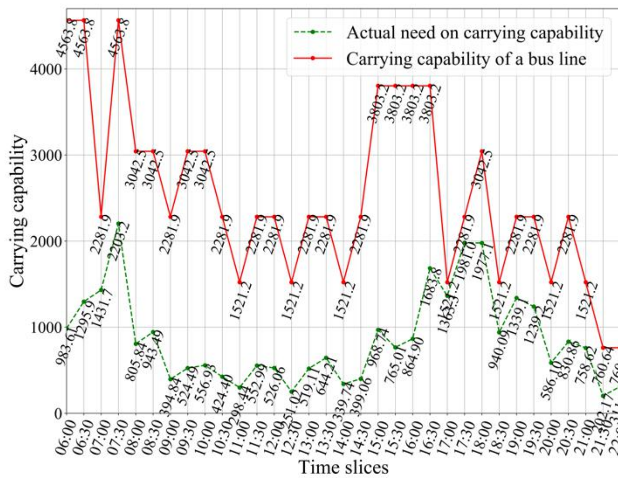

Figure 2: Actual need of passengers on carrying capability vs. carrying capability provided by a bus timetable.

Figure 2 shows that the carrying capability provided by the bus timetable and the needed carrying capability are much different. It means that the bus timetable may contain more departure times (more trips) than actual need, thereby wasting the operational cost. Through optimizing the time intervals of a bus timetable, we may make the carrying capability of timetable match the actual need on carrying capability.

4.1.2 Passengers' waiting time. Passengers' waiting time refers to the time of passengers waiting for the next bus. For a vehicle departing from the time Z_m in a bus timetable, the number of passengers to get on the vehicle at station i can be calculated by the number of passengers that get on the vehicles that depart (according the actual timetable) in the period [Z_{m-1}, Z_m]. Thus, the number of passengers getting on the vehicle at station i is:

$$q_{mi} = \sum_{k \epsilon \{k | Z_{m-1} \leq T_k < Z_m\}} d_{ki} \tag{7}$$

Assume that passengers randomly arrive at a station, following the uniform distribution. The average waiting time of passengers at station i equals $(Z_{m-1} - Z_m)/2$. Thus, the sum of waiting time of passengers for the vehicle at station i is:

$$P_{mi} = \frac{q_{mi}(Z_m - Z_{m-1})}{2} \tag{8}$$

The total waiting time of passengers for the vehicle (departing from Z_m) at all station can be calculated by

$$P_m = \sum_{i=1}^{S-1} P_{mi} \tag{9}$$

The total waiting time of all passengers is computed by

$$P = \sum_{m=1}^{M} P_m \tag{10}$$

4.1.3 Fitness function. The metric of carrying capability expects a timetable to have a small number of departure times to satisfy the need of passengers taking buses, to save the operation cost. The metric of waiting time expects short waiting time of passengers (more departure times in a timetable are preferred) to ensure the service quality. We hope a timetable is able to meet the need of passengers and meanwhile make the waiting time as smaller as possible. Thus, the two metrics are linearly combined to form the following fitness function.

$$F = F_{max} - (w_p P + w_o O) \tag{11}$$

$$s.t. \quad E_m - O_m > 0\ , m \in \{1, 2, ..., M\} \tag{12}$$

where w_p and w_o are weights for waiting time and carrying capabi lity and $w_p + w_o = 1$; the constraint(12) means that the carrying c apacity of a bus line must meet the need of passengers on the carr ying capacity; F_{max} is a given positive value to make a solution w ith smaller metrics have a larger fitness function value.

4.2 Genetic algorithm

The algorithm's outline is shown in Figure 1. The detailed operations are introduced below.

4.2.1. Solution encoding and decoding. A binary solution coding is suggested to represent a solution (timetable), as shown in Figure 3. Each gene is a binary number, corresponding to a minute during the service time. For the coding in Figure 3, the service time are from 6:00 to 22:00. There are total 961 minutes in this period, such that the coding length is 961.

The value of a gene is "1" if the timetable contains the time represented by the gene. For example, the two genes corresponding to 6:00 and 6:03 are marked with "1". It means the two times are in the timetable and there are two trips departing from them. If the value of a gene is "0", then the time corresponding the gene is not in the timetable. That is, no trip departs from the time.

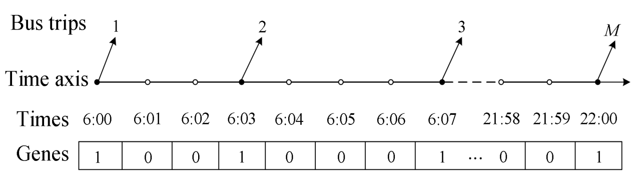

Figure 3: Solution coding

Such solution coding can be easily decoded into a bus timetable, which contains all times marked in "1" in the coding. The number of times in the timetable equals the number of "1" in the solution coding.

4.2.2 Genetic operations. As the solution coding is a binary coding, which can be handled by many genetic operations. In this paper, we simply adopt the operations of single point crossover and random mutation [19]. The roulette selection is used to perform the selection operation.

4.2.3 Repair infeasible solutions. According to constraint (12), a bus timetable's carrying capability must satisfy the actual need of passengers on carrying capability to ensure the service quality. For an individual generated during the search, it may not satisfy the constraint. An individual (solution) is infeasible if it does not satisfy (12). Discarding infeasible solutions directly would decrease the search efficiency, thus that we propose a method to repair infeasible solutions to make them be feasible.

The constraint (12) shows that E_m must be greater than O_m for each time m in a bus timetable Z. The timetable (solution) is infeasible if there is a time m where $E_m < O_m$. For example, Figure 4 is a part of bus timetable Z. T_1-T_6 are the times in the actual timetable, and Z_1-Z_4 are the times in timetable Z. In this example, suppose that E_1 (E_2, E_3) is greater than O_1 (O_2, O_3) but E_4 is smaller than O_4. The solution shown in Figure 4 is infeasible because $E_4 < O_4$. To repair this solution, we move the departure time Z_4 to the middle point between T_5 and T_6 to make Z_4 only cover T_5, instead of T_5 and T_6. Thus, O_4 is reduced to make it smaller than E_4. Each time m in the timetable is checked to observe whether $E_m > O_m$. If there are time m that does not satisfy $E_m > O_m$, then the time m is moved forward by above method. After the forward adjustment of departure times, there may be one or more T_k that are at the end of timetable and do not be covered by any Z_m. In this case, one or more times are inserted into the timetable Z to cover them.

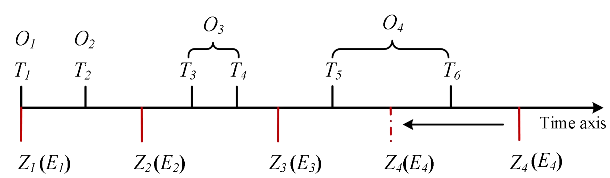

Figure 4: An example to repair an infeasible solution.

4.3 Local search

A local search (LS) is devised and embedded into the genetic algorithm. It uses the same solution coding and evaluation function as the genetic algorithm. It is applied to the elite individual of the genetic algorithm every L_f generations.

The LS consists of two mutation operators: One is random mutation operator and the other is reverse operator [20]. In the early stage of population evolution (the number of generations is less than R_{iter}), the random mutation operator is used to perform local search. As individuals in the population of early stage have low fitness function values, the operator is to explore the solution space in a large range. In the latter stage of population evaluation (the number of generations is greater than R_{iter}), the LS is carried out by the reverse operator. Pseudocode of LS is given below.

```
Input: the elitist solution S_e; the number of generation g.
Output: the updated S_e.
    Let S = S_e.
    For i = 1 to L_max do
       If (g< R_iter), then
           Randomly create 20 neighbors of S by the
           random mutation.
       Else
           Randomly create 20 neighbors of S by the reverse
           mutation.
```

```
    End if
    Find the best solution S_best from the set of neighbors.
    Let S = S_best.
    If (S_best is better than S_e), then
        S_e = S_best.
    End if
    i = i + 1.
End for
Output S_e.
```

4.3.1 Random mutation operator. Each gene of an individual mutates according to a mutation probability of 0.01. If a gene mutates, its value is changed to 1(0) if its original value is 0(1).

4.2.2 Reverse operator. First, a random integer, R, in the range of [0, 60] is generated. Two integers I_1 and I_2 are randomly generated in $[0, |Z| - R]$. Then, all the genes between position $I_1(I_2)$ and position $I_1 + R$ $(I_2 + R)$ in the individual are reversed with the probability of 0.5. Finally, the genes between I_1 and $(I_1 + R)$ are swapped with those between I_2 and $(I_2 + R)$.

5 EXPERIMENTAL RESULTS

BTOA-MA is applied to 4 real-world bus timetables in a city, China. Its results are compared against the actually used timetables and results obtained by a genetic algorithm.

5.1 Problem and Algorithm parameters

The numbers of four bus lines are 18, 115, 29 and 38. Each bus line has two timetables: one is for upward direction and the other for the downward one. As passengers' flow of two directions are different, two timetables of each bus line need to be optimized separately.

Real world data of those bus lines on one day is collected. The information of those bus lines is shown in Table 1. The "Length" means the total length of a bus line; "Stations" is the number of stations in a bus line; "Passengers" is the total number of passengers getting on the vehicles of a bus line on that day. The number of seats, C, of each vehicle is 31.

Parameters of BTOA-MA are presented in Table 2. P_c and P_{mu} are crossover and mutation probabilities of the genetic algorithm, respectively.

In evaluation function (11), parameters w_p and w_o serve to balance two metrics of carrying capability and waiting time. We determine the two parameters based on the proportion of carrying capability (waiting time) in the sum of total carrying capability and waiting time, to make them be consistent with their magnitudes in the actual situation. w_p and w_o are calculated as follows:

$$w_p = \frac{\sum_m^{M-1} O_m}{\sum_m^{M-1} (O_m + P_m)} \tag{13}$$

$$w_0 = \frac{\sum_m^{M-1} P_m}{\sum_m^{M-1} (O_m + P_m)} \tag{14}$$

Table 1: Information of bus lines.

Lines	Direction	Service time	Length (km)	Stations	Passengers
18	Up	06:00-23:00	16.358	33	7739
	Down	06:45-22:00	16.958	33	6818
115	Up	06:20-22:00	16.622	36	4968
	Down	06:30-22:00	17.998	35	4515
29	Up	06:00-23:05	21.805	35	5896
	Down	06:40-23:40	21.480	36	5082
38	Up	06:20-22:00	14.972	27	4485
	Down	06:30-22:00	16.286	28	3889

Table 2: Parameters of BTOA-MA.

Parameters	Values	Parameters	Values
N	50	P_{mu}	0.002
P_c	0.5	R_{iter}	800
L_f	20	L_{max}	11
G_{max}	5000		

5.2 Result analysis

BTOA-MA is coded in C++ language. Experiments are conducted on a PC with 3.20 GHz CPU and 8G RAM. 30 independent runs of BTOA-MA are done.

Solutions obtained by BTOA-MA is compared against the actually used timetables of the four bus lines. The local search is removed from BTOA-MA to form a genetic algorithm based approach (GA). The GA adopts the same solution coding, evaluation function, genetic operators, repair method and algorithm parameters as BTOA-MA. The results of BTOA-MA are compared with those of GA to observe the effect of memetic algorithm. To make fair comparison, the GA is given the same number of object function evaluations (5000 $\times$ 50=25000) as BTOA-MA.

Tables 3 and 4 present experimental results obtained by BTOA-MA, GA and actually used timetables for upward and downward directions of the four bus lines. The best results are in bold type. The third column is the average fitness value. The fourth column is the average number of times in the timetable. Fifth column is the average departure interval in the timetable. Column six represents the average carrying capacity matching (metric O) in (4). Column seven is the average total waiting time. The eighth column represents the average of passengers' waiting time. The last column contains weight values w_p and w_0. The percentage in brackets means the percentage reduction in the corresponding metrics.

Tables 3 and 4 show that BTOA-MA is able to produce solutions with the best average fitness values amongst the three approaches. Compared to the actually used timetables, BTOA-MA reduces the metric O by 7.26-19.43%.MA reduces the metric

Table 3: Experimental results of BTOA-MA, GA, and the actual timetable for the upward direction.

Lines		Fitness	No. of times	Interval (min)	Capability (metric *O*)	Total waiting time (min)	Waiting time (min)	Weights
18	Actual	43604.69	110	8.81	54764	35054.50	4.53	w_p=0.6097
	GA	37194.36	96.50 (12.27%)	10.06	44496 (18.75%)	32520.38 (7.23%)	4.20	
	BTOA-MA	**34344.33**	98.16 (10.76%)	9.88	45764 (16.44%)	**27034.35 (22.88%)**	**3.49**	w_o=0.3903
115	Actual	35400.54	75	12.70	40806.77	31258.50	6.29	w_p=0.5662
	GA	33394.02	68.37 (8.84%)	13.96	35680 (12.56%)	31642.82 (-1.22%)	6.37	
	BTOA-MA	**30952.17**	68 (9.33%)	14.04	35396 (13.26%)	**27547.25 (11.87%)**	**5.54**	w_o=0.4338
29	Actual	37600.13	118	8.69	80173	24558.50	4.17	w_p=0.7655
	GA	34697.07	107.67 (8.75%)	9.61	69696 (13.07%)	23975.70 (2.37%)	4.07	
	BTOA-MA	**32431.89**	109.70 (7.03%)	9.43	71758 (10.50%)	**20385.05 (17.00%)**	**3.46**	w_o=0.2345
38	Actual	30987.45	80	11.78	38984	25712.00	5.73	w_p=0.6025
	GA	26401.18	69.80 (12.75%)	13.67	31882 (18.22%)	22784.96 (11.38%)	5.08	
	BTOA-MA	**24243.89**	70.13 (12.34%)	13.60	32114 (17.62%)	**19051.30 (25.91%)**	**4.25**	w_o=0.3975

Table 4: Experimental results of BTOA-MA, GA, and the actual timetable for the downward direction.

Lines		Fitness	No. of times	Interval (min)	Capability (metric *O*)	Total waiting time (min)	Waiting time (min)	Weights
18	Actual	40381.47	109	8.34	60885	30209.50	4.43	w_p=0.6684
	GA	33955.45	93.83 (13.91%)	9.86	48137 (20.94%)	26919.97 (10.89%)	3.95	
	BTOA-MA	**31617.16**	95 (12.84%)	9.74	49057 (19.43%)	**22965.22 (23.79%)**	**3.37**	w_o=0.3316
115	Actual	34043.51	72	13.00	43864	27815.00	6.16	w_p=0.6119
	GA	31801.57	67.27 (6.57%)	14.05	39902 (9.03%)	26663.62 (4.14%)	5.91	
	BTOA-MA	**29676.40**	68.16 (5.33%)	13.85	40656 (7.32%)	**22712.82 (18.34%)**	**5.03**	w_o=0.3881
29	Actual	35201.71	116	8.74	86246	22112.00	4.35	w_p=0.7959
	GA	31657.98	108.90 (6.12%)	9.46	78155 (9.38%)	19734.23 (10.75%)	3.88	
	BTOA-MA	**29640.19**	110.73 (4.54%)	9.30	79987 (7.26%)	**16729.42 (24.34%)**	**3.29**	w_o=0.2041
38	Actual	33520.62	84	11.45	48957	25483.00	6.55	w_p=0.6576
	GA	27258.69	73.03 (13.06%)	13.27	40652 (16.96%)	20284.88 (20.40)	5.22	
	BTOA-MA	**25040.89**	75.70 (9.88%)	12.79	42672 (12.84%)	**15860.82 (37.76%)**	**4.08**	w_o=0.3424

O by 7.26-19.43%. Metric *O* represents the matching degree of actual passengers' flow. Smaller values of metric *O* mean better matching of actual passengers' flow. It indicates that the timetables (solutions) found by BTOA-MA can save operation cost, i.e., have smaller number of departure times than the actual timetables. We can observe that the average numbers of departure times in the timetables obtained by BTOA-MA are smaller than those of actual timetables.

Solutions(timetables) obtained by BTOA-MA shorten the average passengers' waiting time by 11.87-37.76% compared to the actual timetables.Note that the reduction of waiting time is not achieved by increasing the departing times. Instead, the number of departure times in the obtained solutions is slightly smaller than that of actual timetables.

Solutions produced by BTOA-MA and GA have the similar values of metric O (the similar number of departure times), but solutions found by BTOA-MA have much shorter waiting time than that of GA. The passengers' waiting time of BTOA-MA is 12.90-17.36% shorter than that of GA. With similar average number of departure times in timetables, the average waiting time of BTOA-MA is 0.5-1.2 minutes shorter than that of GA.

Evolutionary curves of 30 runs of BTOA-MA and GA for bus line 18 are shown in Figure 5. BTOA-MA converges more quickly and obtains smaller fitness function values than GA. In addition, each run of BTOA-MA achieves similar fitness function values, which means the performance of BTOA-MA is quite stable.

The bus line 18 is chosen to illustrate the matching of carrying capability of the optimized timetable on passenger's flow. Figure 6 shows carrying capability of the timetable obtained by BTOA-MA and the actual need of carrying capability. Compared to carrying capability of the actual timetable (shown in Figure 2), carrying capability of the optimized timetable more matches the actual need. It means that the optimized timetable can use less departure times (lower bus frequency) to meet passengers' requirement, thereby reducing the operation cost. Metric *O* reflects the difference between carrying capability of a bus timetable and the needed carrying capability. Metric *O* of the optimized timetable is 47918.65, smaller than that of actual bus timetable (54764.47).

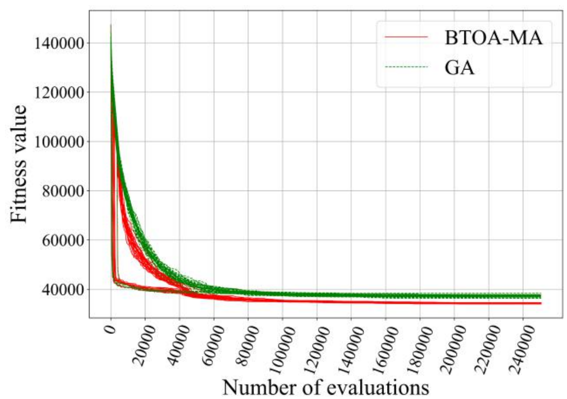

Figure 5: Evolutionary curves of BTOA-MA and GA

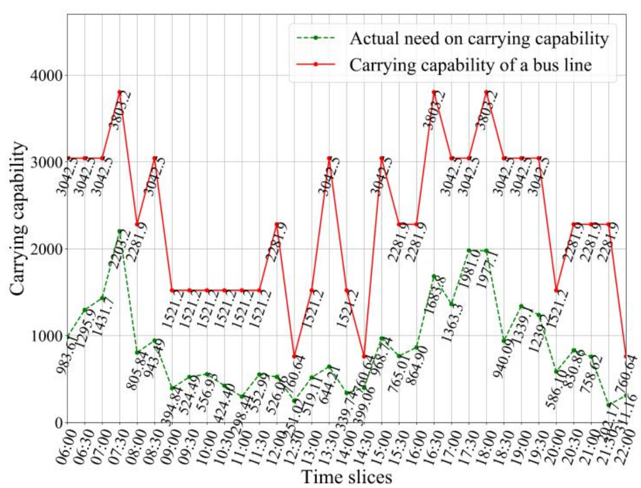

Figure 6: Actual need of passengers on carrying capability vs. carrying capability provided by the optimized timetable of upward direction of bus line 18.

Note that BTOA-MA does not find solutions (timetables) with zero values of metric *O* (it means that the carrying capability matches the actual need exactly). This is because another objective (passengers' waiting time) is also optimized. The decrease of metric *O* will lead to less number of departure times in the timetable (save the operation cost) but will increase the waiting time of passengers, which reflects the quality of service. Zero values of metric *O* would greatly prolong the waiting time, which is unacceptable from the respective of service quality. For example, even there is only one passenger at a station, it is not allowed to let the passenger wait for one hour to take a bus. Metric *O* and waiting time are two conflict objectives, such that we combine them together to balance the operation cost and the service quality.

6 CONCLUSIONS

In this paper, we propose a memetic algorithm to optimize the bus timetable with unequal time intervals. The memetic algorithm consists of a genetic algorithm and a local search with a reverse operator. A repair method is proposed to deal with infeasible solutions. A new metric reflecting the carrying capability of a bus timetable is introduced to evaluate a solution (timetable). This metric together with the metric of passengers' waiting time are used to balance the operation cost and quality of service.

The proposed approach is applied to four real-world bus lines. Experiments show that the approach is able to generate bus timetables with less departure times (less operation cost) and smaller passengers' waiting time compared to the actually used timetables.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China under Grant 61873040.

REFERENCES

[1] Wu, Y., Yang, H., Tang, J., & Yu, Y. 2016. Multi-objective re-synchronizing of bus timetable: model, complexity and solution. *Transportation Research Part C: Emerging Technologies 67*, (2016), 149-168.

[2] Yao-Cheng, H. E., Ming, Y. E., & Ming, Y. 2011. Research on dynamic bus scheduling based on passenger flow fluctuation. *Manufacture Information Engineering of China*, (2011).

[3] Sun, C., Zhou, W., & Wang, Y. 2008. Scheduling combination and headway optimization of bus rapid transit. *Journal of Transportation Systems Engineering and Information Technology* 8, 5(2008), 61-67.

[4] Jin-Bao, Y., Bao-Zhen, Y., Zhi-Hong, Y., & Bin, Y. 2012. A bus headway optimization model with dual-population genetic algorithm. *Journal of Shenzhen University science and engineering* 29, 6(2012).

[5] Lampkin, W., & Saalmans, P. D. 1967. The design of routes, service frequencies, and schedules for a municipal bus undertaking: a case study. *Journal of the Operational Research Society* 18, 4(1967), 375-397.

[6] Ceder, A. 1984. Bus frequency determination using passenger count data. *Transportation Research, Part A 18*, 5-6(1984), 439-453.

[7] Oudheusden, D. L. V., & Zhu, W. 1995. Trip frequency scheduling for bus route management in bangkok. *European Journal of Operational Research* 83, 3(1995), 439-451.

[8] Sun, W. S. W., Wang, X. W. X., & Qiao, G. Q. G. 2007. Study on departing time interval control of bus dispatching. *Journal of Hebei University of Technology*, (2007), 1-5.

[9] Luhua, S., Yin, H., & Xinkai, J. 2011. Study on method of bus service frequency optimal model based on genetic algorithm. *Procedia Environmental Sciences*, (2011).

[10] Yu, B., Yang, Z., Sun, X., Yao, B., Zeng, Q., & Jeppesen, E. 2011. Parallel genetic algorithm in bus route headway optimization. *Applied Soft Computing* 11, 8(2011), 5081-5091.

[11] Tang, X. L., & Yang, S. H. 2013. An improved bus timetable scheduling model using quantum genetic algorithm based on penalty strategy. *Applied Mechanics and Materials*, 253-255(2013), 1406-1409.

[12] Zhu, X. W., Meng, X. F., & Zhang, M. M. 2014. Optimizing departure interval for bus dispatching system based on comprehensive objective model. *Cota International Conference of Transportation Professionals*, (2014).

[13] Qian, Z., Feng-Lian, W., & Ju, L. 2015. A bus headway optimization model based on genetic taboo algorithm. *Journal of Transport Science and Engineering*, (2015).

[14] Hong-Zhao, D., Juan-Juan K., Qing-Hui, L. 2016. A bus departure time interval transition model considering traffic congestion. *Journal of Transportation Systems Engineering and Information Technology* 16, 3(2016), 101-106.

[15] Ceder, A., Hassold, S., & Dano, B. 2013. Approaching even-load and even-headway transit timetables using different bus sizes. *Public Transport*, (2013).

[16] Bie, Y., Gong, X., & Liu, Z. 2015. Time of day intervals partition for bus schedule using gps data. *Transportation Research Part C: Emerging Technologies* 60, (2015), 443-456.

[17] Sun, D., Xu, Y., Peng, Z. 2015. Timetable optimization for single bus line based on hybrid vehicle size model. *Journal of Traffic and Transportation Engineering (English Edition)* 2, 3(2015), 179-186.

[18] Li, J., Hu, J., & Zhang, Y. 2017. Optimal combinations and variable departure intervals for micro bus system. *Tsinghua Science and Technology* 03, (2017), 58-68.

[19] Holland, J. H. 1992. *Adaptation in Natural and Artificial System*. Adaptation in natural and artificial systems. MIT Press.

[20] Tao, G., & Michalewicz, Z. 1998. Inver-over operator for the TSP. *International Conference on Parallel Problem Solving from Nature*. (1998). Springer, Berlin, Heidelberg.