Available online www.jsaer.com

Journal of Scientific and Engineering Research, 2021, 8(7):105-112

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

To meet the multi-constraint conditions of bus timetable preparation method research

Yuan Fu, Xiaolin Wang, Ziwen Song, Shulong Dong, Chenchen Li

School of Transportation and Vehicle Engineering, Shandong University of Technology, Zibo 255000, China

Abstract The main part of the thesis is based on the multi-constraint conditions of the various stages of the timetable preparation model ideas and the preparation of procedures, and finally complete the bus route departure timetable preparation, and the model of various ways to select, and finally form the actual timetable, the conclusion is that the existing prepared timetable efficiency is higher, is the minimum cost to meet the conditions of the scientific timetable preparation program.

Keywords regular bus scheduling, platform scheduling schedule preparation, time period division, Python programming

Introduction

With the development of China's modernization, the promotion of regional urbanization, China's population and population migration rate continues to increase, urban traffic travel demand is expanding, car ownership continues to increase, traffic congestion and environmental pollution and other issues are increasing. Matching urban infrastructure is far from meeting the needs of modern life. In terms of traffic, due to the irrationality of urban layout, poor street conditions in the old town, the rapid increase in motor vehicle ownership, resulting in serious congestion of urban streets, environmental pollution and increase, the actual traffic supply is far from meeting the traffic needs of residents. With the further research in the field of transportation, the importance of public transportation in urban transportation has been paid more and more attention. The operation and management mode and technical application of public transportation system determine the operating efficiency of the city's general public transport system.

The key to the urban regular bus scheduling system is still in the preparation of bus schedules, this paper is based on specific projects to discuss and realize the specific bus operation schedule preparation problems, in fact, is also to solve the core problems of urban regular bus scheduling, and finally complete the preparation of bus schedule tasks.

1. Actual schedule preparation project

1.1. Description of specific issues

The problem is based on the preparation of bus timetables under single-line scheduling. This line is the simplest ring line, that is, from the municipal government after a series of stops and then back to the city government for the next departure. Each run is a train, a car runs more than one shift a day, double shift is two drivers, one in the morning and one in the afternoon. A single shift is a driver, usually running a few cars in the morning and running a few cars in the afternoon (usually running peak). Table 1 is an excerpt of the time sheet preparation effect map.

Table 1: Pre-completed bus layout schedule effect map (excerpt).

shift	17		18		19		20	
The frequency type	Double shift		Double shift		Double shift		Single shift	
The name of the off-site station	City government departure time Return time		City government departure time Return time		City government departure time Return time		City government departure time Return time	
1	5:20	6:25	5:34	6:39	5:25	6:30	5:15	6:25
2	6:38	7:53	6:50	8:05	6:44	7:59	6:40	7:55
3	8:05	9:20	8:23	9:38	8:14	9:29	8:08	9:23
4	10:05	11:20	10:10	11:20	10:00	11:15	15:10	16:25
5	11:35	12:50	11:53	13:23	11:41	13:11	16:34	17:49
6	13:41	14:56	14:03	15:18	13:53	15:08	3 p.m. 2 cars, noon break	
7	15:10	16:25	15:34	16:49	15:18	16:32	7h hours	
8	16:43	17:58	17:07	18:22	16:52	18:07	Complete Task 5	
9	18:46	20:01	19:16	20:31	18:52	20:07	Average afternoon assignments on a single shift	
10	20:31	21:46	21:08	22:23	21:00	22:15		

Different time interval is not the same, driving plan does not involve specific cars, to the car schedule when the specific car.

Table 2: The one-way time and departure interval requirements for different time periods

The nature of the time period	**The start time of the period**	**The end time of the period**	**One-way time**	**Departure interval**	**Minimum stop time**
Peak period	4:30	5:00	70	7.0	10
Peak period	5:00	6:00	70	4.5	10
Peak hours	6:00	8:00	75	3.0	10
Peak period	8:00	16:00	75	4.5	10
Peak hours	16:00	18:00	75	3.0	10
Peak period	18:00	22:15	70	6.5	10

Driving planning is based on the departure interval discharge of a day's vehicle schedule, and in accordance with the double shift, single shift, etc. will be easy to arrange the vehicle, in the grouping to consider stop time, can return without conflict, and to consider the driver meal time, shift time (red part of the table above), refueling and other activities, and morning and afternoon driver working hours to be uniform and preferably not more than 8 hours.

Departure intervals and minimum stop times, as well as meal times, are not absolutely constant and can be broken occasionally, with the driving plan balancing between ensuring departure intervals and reducing vehicle costs.

1.2. The constraints involved

Rule constraints for enterprise operating models Implement the order:

(1) Stop time is greater than the minimum value and configure parameters such as min 3min, max 30min, and optimal 10min.

(2) The driver's continuous driving time does not exceed the upper limit such as 3h,at which time the rest time is not less than 20min;

(3) To meet the constraints of labor hours, the driver's working hours do not exceed the legal limit example such as 8h, working length of about 8h is better, should not be too long or too short, the suggested interval such as " 7h,8.5h";

(4) Peak-hour same-shift cars try not to be connected;

(5) Vehicles are collected in order of issuance, and double shifts are required for order ;(single shifts as far as possible to work late and early).

(6) The first and last shifts are assigned to different cars. (The first and last shifts should be scheduled to be scheduled).

2. The actual stage of schedule preparation

2.1. Bus departure schedule build

The overall idea is to use heuristic algorithms to compile bus timetables step by step and program in the Python language according to the actual situation and constraints.

First of all, determine the earliest departure time, according to the different time nature of the time period travel time and departure interval and other information to generate the bus departure schedule, so as to get the departure time and return time of each train.

Departure time for the firsti-1 train, departure time for the i-train, departure interval

Return time - Departure time and travel time

Create the program generated by the departure schedule using the Python programming language, as follows:

The driving interval data in the target excel table is defined and referenced first, including the time period start time and the end time of the time period, and the start time and end time are defined, and the time T is defined by the start time of the journey plus the one-way time of that time period. Time T is determined, if time T is still in that time period, then the start time of the next bus T is defined as the start time of the previous bus plus the departure interval of that time period, cycle, until the end time T is not in the time period, then start referencing the next time interval and one-way time. Start the cycle until the start and end times of all shifts for the day, i.e. departure time and return time, are scheduled.

Currently, according to the figure below, the time period is divided, according to the departure interval of different time periods, travel time, arrival time to discharge the departure time of one day.

Table 3: Generates a departure schedule

Departure time	Travel time	The time returned
1900-01-00 04:30:00	70	1900-01-00 05:40:00
1900-01-00 04:37:00	70	1900-01-00 05:47:00
1900-01-00 04:44:00	70	1900-01-00 05:54:00
1900-01-00 04:51:00	70	1900-01-00 06:01:00
1900-01-00 04:58:00	70	1900-01-00 06:08:00
1900-01-00 05:05:00	70	1900-01-00 06:15:00
1900-01-00 05:10:00	70	1900-01-00 06:20:00
1900-01-00 05:15:00	70	1900-01-00 06:25:00
1900-01-00 05:20:00	70	1900-01-00 06:30:00
1900-01-00 05:25:00	70	1900-01-00 06:35:00
1900-01-00 05:30:00	70	1900-01-00 06:40:00
1900-01-00 05:35:00	70	1900-01-00 06:45:00
1900-01-00 05:40:00	70	1900-01-00 06:50:00
1900-01-00 05:45:00	70	1900-01-00 06:55:00
1900-01-00 05:50:00	70	1900-01-00 07:00:00
1900-01-00 05:55:00	70	1900-01-00 07:05:00
1900-01-00 06:00:00	75	1900-01-00 07:15:00
1900-01-00 06:03:00	75	1900-01-00 07:18:00

2.2. Bus connections

Stage 1: Select the bus mainly by the number of cars

The connection between departure time and return time is the minimum time interval to connect, will be the departure frequency of the connection, can not connect the empty train schedule single bus. However, this convergence can only output one scheme, and cannot filter the optimal scheme.

Table 4: Actually builds a one-car junction for the stage

Departure time	Travel time	The time returned	Trips
1900-01-00 04:30:00	70	1900-01-00 05:40:00	1
1900-01-00 04:37:00	70	1900-01-00 05:47:00	2
1900-01-00 04:44:00	70	1900-01-00 05:54:00	3
1900-01-00 04:51:00	70	1900-01-00 06:01:00	4
1900-01-00 04:58:00	70	1900-01-00 06:08:00	5
1900-01-00 05:05:00	70	1900-01-00 06:15:00	6
1900-01-00 05:10:00	70	1900-01-00 06:20:00	7
1900-01-00 05:15:00	70	1900-01-00 06:25:00	8
1900-01-00 05:20:00	70	1900-01-00 06:30:00	9
1900-01-00 05:25:00	70	1900-01-00 06:35:00	10
1900-01-00 05:30:00	70	1900-01-00 06:40:00	11
1900-01-00 05:35:00	70	1900-01-00 06:45:00	12
1900-01-00 05:40:00	70	1900-01-00 06:50:00	13
1900-01-00 05:45:00	70	1900-01-00 06:55:00	14
1900-01-00 05:50:00	70	1900-01-00 07:00:00	1
1900-01-00 05:55:00	70	1900-01-00 07:05:00	2
1900-01-00 06:00:00	75	1900-01-00 07:15:00	3

Stage 2: Take the bus as the main choice

Because the minimum interval scheme is not necessarily the best solution, and may be eliminated by other constraints, and stop time is too small, the reliability of the timetable will be worse, once the vehicle can not return to the station according to the schedule, will affect the next shift departure. So given the stop time range from the minimum to the maximum stop time as a constraint, the convergence of departure time and return time can be output as long as it is within the constraint range.

First of all, according to the departure time order in turn to generate the number of cars, the specified stop time range of 3-30min, if the nth train departure time and the return time of the previous n times in a number of times in the time difference in the specified time frame, that is, the number of times and the nth train connection. Then the downward convergence, and finally get the convergence scheme. However, this kind of convergence data dimension is too large, the generated scheme is too many, need to optimize the algorithm, the dimension reduction. Then try to stop the time 10min as the optimal value, preferring the nearest shift to 10min, if there are two values as close as , as a shift alternative, the scheme is preferred, the initial scheme is controlled to 10000 kinds, but not implemented. In order to achieve the preferred purpose of comparison of various schemes, optimize the above methods, establish a cycle within the specified time frame, generate different stop time range upper and lower limits, for multiple shifts that meet the convergence constraints at the same time to choose the first return shift to bridge, so as to get a variety of double shuttle bus scheduling program under the non-crossing departure conditions (first return vehicles first), when all the double shuttles connect, the connection can not be added to the empty bus schedule. However, this convergence effect is not good, the result is not satisfied with other constraints.

Table 5: Generates a phase two car junction

Departure time	Travel time	The time returned	Trips	Total working hours
1900-01-00 04:30:00	70	1900-01-00 05:40:00	1	1900-01-00 01:10:00
1900-01-00 04:37:00	70	1900-01-00 05:47:00	2	1900-01-00 01:10:00
1900-01-00 04:44:00	70	1900-01-00 05:54:00	3	1900-01-00 01:10:00
1900-01-00 04:51:00	70	1900-01-00 06:01:00	4	1900-01-00 01:10:00
1900-01-00 04:58:00	70	1900-01-00 06:08:00	5	1900-01-00 01:10:00
1900-01-00 05:05:00	70	1900-01-00 06:15:00	6	1900-01-00 01:10:00
1900-01-00 05:10:00	70	1900-01-00 06:20:00	7	1900-01-00 01:10:00
1900-01-00 05:15:00	70	1900-01-00 06:25:00	8	1900-01-00 01:10:00
1900-01-00 05:20:00	70	1900-01-00 06:30:00	9	1900-01-00 01:10:00
1900-01-00 05:25:00	70	1900-01-00 06:35:00	10	1900-01-00 01:10:00
1900-01-00 05:30:00	70	1900-01-00 06:40:00	11	1900-01-00 01:10:00
1900-01-00 05:35:00	70	1900-01-00 06:45:00	12	1900-01-00 01:10:00
1900-01-00 05:40:00	70	1900-01-00 06:50:00	13	1900-01-00 01:10:00
1900-01-00 05:45:00	70	1900-01-00 06:55:00	14	1900-01-00 01:10:00
1900-01-00 05:50:00	70	1900-01-00 07:00:00	1	1900-01-00 02:30:00
1900-01-00 05:55:00	70	1900-01-00 07:05:00	2	1900-01-00 02:28:00
1900-01-00 06:00:00	75	1900-01-00 07:15:00	3	1900-01-00 02:31:00

Here's a description of the Python programming language:

First read the departure time, return time data in the driving segment excel output of the first step, and define the time format. Search all vehicles with a difference between the return time and the start time inthe range of 3min to3 0min to arrive at a timetable that is not fully connected in phase TWO. Then add some constraints, completely connect the cars, get the car bridge table of Table 6.

Table 6: Does not add additional constraints to the car bridge table

Departure time	Travel time	The time returned	Trips
1900-01-00 04:30:00	70	1900-01-00 05:40:00	1
1900-01-00 04:37:00	70	1900-01-00 05:47:00	2
1900-01-00 04:44:00	70	1900-01-00 05:54:00	3
1900-01-00 04:51:00	70	1900-01-00 06:01:00	4
1900-01-00 04:58:00	70	1900-01-00 06:08:00	5
1900-01-00 05:05:00	70	1900-01-00 06:15:00	6
1900-01-00 05:10:00	70	1900-01-00 06:20:00	7
1900-01-00 05:15:00	70	1900-01-00 06:25:00	8
1900-01-00 05:20:00	70	1900-01-00 06:30:00	9
1900-01-00 05:25:00	70	1900-01-00 06:35:00	10
1900-01-00 05:30:00	70	1900-01-00 06:40:00	11
1900-01-00 05:35:00	70	1900-01-00 06:45:00	12
1900-01-00 05:40:00	70	1900-01-00 06:50:00	1
1900-01-00 05:45:00	70	1900-01-00 06:55:00	13
1900-01-00 05:50:00	70	1900-01-00 07:00:00	2
1900-01-00 05:55:00	70	1900-01-00 07:05:00	3
1900-01-00 06:00:00	75	1900-01-00 07:15:00	14

Stage 3: Take the bus as the main selection

Add a total working time constraint, and add the variable Total Working Time at the same time as the bridge. Assuming that the jth train can be connected to theith and condition i<j is met, the total working time of the jth train is the return time of the jth train - the return time of the ith train and the total working time of the ith train. The jth and ith cars will only be connected if the total working time is less than 16.5h.

According to the currently generated timetable, when 12 cars start, the first car returns to be able to connect the frequency, so first define 12 buses, arrange them all as double buses, and then from the first bus departure time 4:30, to the last bus departure time 22:12, the 12 cars connected, free of time to add a single bus and other double shuttle. The problem is that some of the first 12 cars may not be able to form a double bus; the total working time < the 16.5-hour constraint, resulting in the next time not being covered by the double bus if it leaves in sequence. In practice, we try to make double shuttles cover morning and evening shifts, with single buses concentrated on peak shifts.

Table 7: Generates a three-car interface for stages

Departure time	Travel time	The time returned	Trips
1900-01-00 04:30:00	70	1900-01-00 05:40:00	1
1900-01-00 04:37:00	70	1900-01-00 05:47:00	2
1900-01-00 04:44:00	70	1900-01-00 05:54:00	3
1900-01-00 04:51:00	70	1900-01-00 06:01:00	4
1900-01-00 04:58:00	70	1900-01-00 06:08:00	5
1900-01-00 05:05:00	70	1900-01-00 06:15:00	6
1900-01-00 05:10:00	70	1900-01-00 06:20:00	7
1900-01-00 05:15:00	70	1900-01-00 06:25:00	8
1900-01-00 05:20:00	70	1900-01-00 06:30:00	9
1900-01-00 05:25:00	70	1900-01-00 06:35:00	10
1900-01-00 05:30:00	70	1900-01-00 06:40:00	11
1900-01-00 05:35:00	70	1900-01-00 06:45:00	12
1900-01-00 05:40:00	70	1900-01-00 06:50:00	
1900-01-00 05:45:00	70	1900-01-00 06:55:00	1
1900-01-00 05:50:00	70	1900-01-00 07:00:00	
1900-01-00 05:55:00	70	1900-01-00 07:05:00	2
1900-01-00 06:00:00	75	1900-01-00 07:15:00	3
1900-01-00 06:03:00	75	1900-01-00 07:18:00	
1900-01-00 06:06:00	75	1900-01-00 07:21:00	4
1900-01-00 06:09:00	75	1900-01-00 07:24:00	
1900-01-00 06:12:00	75	1900-01-00 07:27:00	5
1900-01-00 06:15:00	75	1900-01-00 07:30:00	
1900-01-00 06:18:00	75	1900-01-00 07:33:00	
1900-01-00 06:21:00	75	1900-01-00 07:36:00	6

Stage 4: Bus-based selection

In the actual scheme, the number of double shifts is a certain value, and the number of connections for a single shift is less than or equal to half of the number of double shifts. The number of scheme convergences generated by the appeal method is more dispersed from 12 to 1, and there are big problems when it comes to single-shift shifts.

Next, the number of double shuttles will be determined based on the number of vehicles required during the peak period. Morning peak time is short, first determine part of the double shuttle bus, the part of the double shuttle bus in the day's departure frequency to connect, the morning rush hour began to send a single shuttle and add a new double shuttle, priority to add a new double shuttle bus so that the number of double shuttles can cover the afternoon peak and evening peak, morning peak, evening peak double bus can not connect the frequency of the single bus.

According to the resulting timetable, the total time span is 18 hours and 52 minutes, the total length of the two-bus constraints of 16 hours and 30 minutes, then the first batch of double-bus after the departure, just in time for the morning peak, began to clip a single bus, in order to meet the double bus work The total length of time constraints, bear the last bus of the day the start time of the double bus can not be earlier than 6:50, so until

23:20-16:30 - 6:50 can start adding a new double bus , The number of double shuttles is determined by the number of vehicles required during the peak period.

Table 8: Stage Four-Car Convergence Solution Idea

1900-01-00 06:18:00	75	1900-01-00 07:33:00	Single shuttle
1900-01-00 06:21:00	75	1900-01-00 07:36:00	6
1900-01-00 06:24:00	75	1900-01-00 07:39:00	7
1900-01-00 06:27:00	75	1900-01-00 07:42:00	Single shuttle
1900-01-00 06:30:00	75	1900-01-00 07:45:00	8
1900-01-00 06:33:00	75	1900-01-00 07:48:00	Single shuttle
1900-01-00 06:36:00	75	1900-01-00 07:51:00	9
1900-01-00 06:39:00	75	1900-01-00 07:54:00	10
1900-01-00 06:42:00	75	1900-01-00 07:57:00	Single shuttle
1900-01-00 06:45:00	75	1900-01-00 08:00:00	11
1900-01-00 06:48:00	75	1900-01-00 08:03:00	12
1900-01-00 06:51:00	75	1900-01-00 08:06:00	Single shuttle
1900-01-00 06:54:00	75	1900-01-00 08:09:00	Double shuttle
1900-01-00 06:57:00	75	1900-01-00 08:12:00	1
1900-01-00 07:00:00	75	1900-01-00 08:15:00	Single shuttle
1900-01-00 07:03:00	75	1900-01-00 08:18:00	Double shuttle
1900-01-00 07:06:00	75	1900-01-00 08:21:00	2
1900-01-00 07:09:00	75	1900-01-00 08:24:00	Single shuttle
1900-01-00 07:12:00	75	1900-01-00 08:27:00	Double shuttle
1900-01-00 07:15:00	75	1900-01-00 08:30:00	Single shuttle

Stage 34 did not carry out program compilation, because the program compilation level is limited, in the search for algorithms and programming software to continue to learn.

Stage 5: Priority convergence ideas (looking for possible algorithms)

Consider how long the front of the vehicle ran, the more shifts in front of the car, in the re-connection of the priority more backward, priority to connect the front running fewer cars.

This is a sort of search for Python's selection for this problem. Select sorting is a simple and intuitive sorting algorithm. First, find the smallest (large) element in the unsorted sequence and save it at the beginning of the sort sequence. Then, continue to find the smallest (large) element from the remaining unsorted elements and place it at the end of the sort sequence. According to this analogy, until all elements are sorted. Define a new variable, "Vehicle Junctions", that will be able to be output before connecting all vehicles in the current shift, then sorted by Python's selection sort, and selected the vehicles with the fewest previous bridging shifts.

Currently continuing to learn about algorithms and programming languages and methods.

2.3. Overview of the problems faced

At this stage of the results, the completion of the project progress and problems faced by a summary, and thinking about the actual constraints may cause the layout system is not coordinated or does not meet the actual situation, specifically included in the following issues:

First arrange a double shuttle bus, spare time with a single shuttle to supplement, may lead to a single bus adjacent to the two times between the time interval is too large or a single bus only bear one or two shifts of the situation.

Depending on the number of double buses determined by the demand for the peak period, can be used as the minimum value but not as a fixed value, the more double buses the better.

Double shuttles line from both ends to the middle, one in the morning and one in the evening, with time squeezed in the middle.

If all the shifts are covered, there may be cars that only run once or twice a day.

You can first estimate how many cars are needed in total, and the total number of vehicles in demand can also have a lower limit.

Five cars together to occupy the task, coupled with the optimal stop time of ten minutes this constraint, there may be no row, may lead to the late no car put.

Try not to always limit, you can have priority principles.

Consider how long the front of the vehicle ran, the more shifts in front of the car, in the re-connection of the priority more backward, priority to connect the front running fewer cars.

Double shuttle shift can not require stop time, shift stop time does not limit the maximum value, limit the minimum value.

3. Conclusion

First of all, the article describes the specific problems that will need to be solved, lists the constraints that need to be solved, and solves the actual specific problems.

The article describes the problem solving process in detail, discusses the initial establishment of the departure schedule and the way of connecting the trains, programs the existing results to achieve automated solution, and writes out the problems and needs to be solved at this stage. The establishment of an initial schedule that did not meet all constraints was finalized.

4. Prospects for the future

In this study, I focused on the preparation method of Python program and the basic neural algorithm part, according to known research, the accuracy and efficiency of the method in the existing multi-planning model is not high. Other, more efficient neural networks and algorithms, such as mixed neural networks, could be tried in future studies.

Secondly, for the preparation of bus timetables has not yet met more practical constraints, I believe that in the future work will be more suitable for practical problem solving.

References

[1]. Desaulniers, G., Hickman, M., 2007. Public transit. In: Barnhart, C., Laporte, G. (Eds.), Handbooks in operations research and management science, Transportation, vol. 14. North-Holland, pp.69- -128.

[2]. Palma, A.D., Lindsey, P., 2001. Optimal timetables for public transportation. Transportation Research Part B 35, 789-813.

[3]. WeiJinli, Guo Yajun, Zhang Mengmeng. The optimization method of bus line driver scheduling based on the collection coverage theory. 2016.

[4]. Chen Xi, NiuXueqin, Chen Xuewu, Wang Wei. Bus line frequency optimization model. Road Traffic Technology, 2004, 21 (2), 103-108.

[5]. Huang Splash, Zhang Guowu. A study on the method of public transportation implementation. Northern Jiaotong University, 2001, 100044.