Research Article

A Model for Bus Crew Scheduling Problem with Multiple Duty Types

Mingming Chen and Huimin Niu

School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China

Correspondence should be addressed to Huimin Niu, hmniu@mail.lzjtu.cn

This paper presents an approach for solving the bus crew scheduling problem which considers early, day, and late duty modes with time shift and work intensity constraints. Furthermore, the constraint with the least crew number of a certain duty (e.g., day duty) has also been considered. An optimization model is formulated as a 0-1 integer programming problem to improve the efficiency of crew scheduling at the minimum expense of total idle time of crew for a circle bus line. Correspondingly, a heuristic algorithm utilizing the tabu search algorithm has been proposed to solve the model. Finally, the proposed model and algorithm are successfully tested by a case study.

1. Introduction

The urban bus crew scheduling problem is a fundamental and crucial problem arising in the field of transit scheduling, because there are the main resources necessary to service passengers. Given the timetable of an urban transit line during one day, the scheduling problem of a bus crew is to complete the operating arrangement of all crews with different duty types. The duty types of crew can be divided into three types: early, day, and late modes. With the influence of working hours, place of residence, living habits, and other factors, there are different requirements to the crew. For example, the crew can only choose trips with day duty when living farther away from the bus depot, the crew can select trips with early or late duty when they have temporary activities at noon, the crew can choose trips with early or day duty when they need to take care of family members at night, and some crews can be allocated to trips with any kind of duty type. Therefore, how to arrange the crew scheduling so as to utilize the manpower resources effectively and satisfy a set of constraints laid down by the labor regulations, has an important theoretical and practical significance.

The urban bus crew scheduling has received much attention in the last decade, and there have been many models and solution methods. According to the objective of model,

it can usually be divided into set covering, set partitioning, and multiobjective program. Smith and Wren [1] introduced a crew scheduling problem formulation based on set covering problem, solved by using integer linear programming. The elements covered were duty pieces, and the covering process used slack and surplus variables. Darby-Dowmanand Mitra [2] presented a natural generalization called the extended set partitioning model which formed the basis of a computer assisted bus crew scheduling system. An integrated vehicle and crew scheduling problem proposed by Mesquita and Paias [3] were described by an integer linear programming formulation combining a multicommodity network flow model with a set partitioning/covering model. Beasley and Cao [4] developed a new lower bound for the crew scheduling problem based on dynamic programming. Lourenco et al. [5] brought a mutiobjective crew scheduling model, which was tackled using a tabu search technique, metaheuristics, and genetic algorithms. Huisman et al. [6] presented an integrated approach to solve a vehicle scheduling problem and a crew scheduling problem on a single bus route.

As for the solution method of bus crew scheduling, it can be divided into mathematical programming methods including integer linear programming, column generation, branch and bound method, and heuristic methods including genetic algorithm, tabu search algorithm, colony algorithm, and so forthm.

Mitra and Darby-Dowman [7] established a computer-based bus crew scheduling model using integer programming. The approach presented by Desrochers and Soumis [8] used column generation solution procedure. Clement and Wren [9] introduced a solution for the crew scheduling problem using a genetic algorithm, and several greedy algorithms were used for assigning duties to pieces of work. Beasley and Cao [10] established a 0-1 integer programming model, and a tree-search algorithm was used to obtain the final optimum. Ceder [11] used deficit function properties to construct vehicle chains, splitted and recombined vehicle blocks into legal duties, and used a shortest-path and matching algorithm for solving the problem. Shen and Ni [12] established a crew scheduling model with minimizing the number and cost of duty and designed the multineighbourhood structure. The crew scheduling model was abstracted by Yang et al. [13] as an optimization problem for the minimum of a nonlinear constraint function, and a genetic algorithm was used to solve the problem. Niu [14] focused on how to determine the skip-stop scheduling for a congested urban transit line during the morning rush hours and formulated a nonlinear programming model with the objective of minimizing the overall waiting times and the in-vehicle crowded costs, and a bilevel genetic algorithm was proposed to solve the problem. Srdjan et al. [15] proposed architecture for a new scheduling mechanism that dynamically executes a scheduling algorithm using feedback about the current status grid nodes, and two artificial neural networks were created in order to solve the scheduling problem. A tabu search algorithm was proposed by Atli [16] to provide good solutions to resource-constrained, deterministic activity duration project scheduling problems.

For an overview on these papers, although a lot of attention has been paid to the factors associated with the required crew number, total work hours, layover and rest time, impartiality, and so forth in bus crew scheduling, only a few papers consider the duty type constraint, which should be well considered in process of urban transit operation.

The remainder of paper is organized as follows. In Section 2, we describe the urban bus crew scheduling problem and establish an optimization model with minimizing the total crew idle time. A tabu search algorithm is then designed in Section 3. In Section 4, a numerical example is provided to illustrate the application of the model and algorithm. Finally, we present the conclusions and some comments on topics requiring future research.

2. Urban Bus Crew Scheduling Model

2.1. Problem Description

Given a set of trips within a fixed planning horizon, the crew scheduling problem consists of generating a feasible set of crew schedule which includes sign-on and sign-off time, crew duty types, and a serial of trips which covers the bus schedule and satisfies a range of constraints laid down by work rules. The trips that are assigned to the same crew member define a crew duty. Together the duties constitute a crew schedule. The criteria for crew scheduling are based on an efficient use of manpower resources while maintaining the integrity of any work rule agreements. The usual crew scheduling problem uses one or more of these objectives: minimum sum of crew costs, minimum number of crew required, and maximum the efficiency of crew usage.

The crew scheduling problem in this study is to arrange the work plan for crew in a single bus line. The kind of urban bus line is a circle one, which means that the starting station and terminal are the same bus station. Each trip has fixed departure and arrival times and can be assigned to a crew member from a determined depot. All trips have been numbered in ascending departure time order. Each crew should be restricted to work in the same vehicle during the bus operating time.

In particular, each crew duty must satisfy several complicating constraints corresponding to work rule regulation. Typical examples of such constraints are maximum working time without a break, minimum break duration, maximum total working time, and maximum duration. In this paper, the considered work rules mainly involve time shift, work intensity, and duty type compatibility constraints. The time shift constraint means that the difference between departure time of latter trip and arrival time of former trip at the same station should not be less than the minimum layover time, as for the two adjacent trips carried by the same crew. The work intensity constraint can be defined that the total work time of each crew does not exceed the maximum number of work hours corresponding to the duty type of crew.

According to the different peak hours, crew duty can be divided into three types, namely, early, day, and late mode. The early duty covers morning peak hours and the range of working time is generally from 6:00 to 13:00, while the late duty covers evening peak hours, and the range of working time is generally from 15:00 to 22:00. Due to the evening peak hours forward and morning peak hours backward in weekend, the day duty covers two peak hours, and the range of working time is from 9:00 to 18:00. In this paper, we consider the three duty types mentioned before, and the sign-on and sign-off time can be chosen flexibly with work plan. The duty type compatibility constraint means that the departure time of trip should be later than the earliest sign-on time and earlier than the latest sign-off time corresponding to crew duty type.

With the different working time of duty types, crew members have special requirements. For example, the total crew number with day duty should be larger than a fixed value, which should be considered fully in the crew scheduling model. Furthermore, the crew number provided by bus company is limit, thus, the total crew number with all duty types must be no more than the crew number available.

The optimization objective of bus crew scheduling in this paper is to maximize the utilization of all crews, and consequently lower the operating cost. In order to make the crew more efficient, the objective function can be expressed by minimizing the total idle time of crew. The calculation of crew idle time can be shown in Figure 1. There are four trips assigned to a crew and the idle time section is three. The first idle time section begins with the arrival
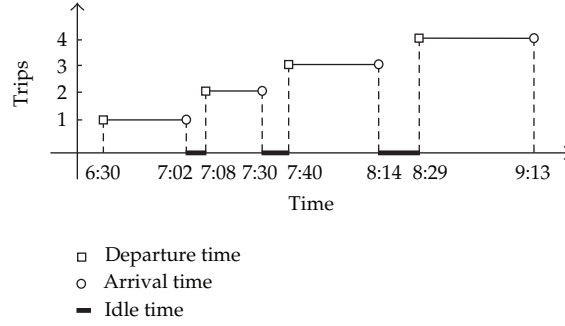
Figure 1: The crew idle time with four trips.

time of trip 1 (7:02) and ends with the departure time of trip 2 (7:08), and the value of first idle time is 6 minutes. Similarly, the second and third idle time is 10 and 15 minutes, respectively. Therefore, the total idle time of the crew is 31 minutes. The least idle time can be gained by arranging different trips with the same crew.

2.2. Definitions and Notations

The following definitions and notations are used throughout this paper.

N: number of trips provided by timetables in a day

G: number of duty types of crew

η: number of crew available in a day

θ: the least number of crew required with day duty

i: index of the trips, $i = 1, 2, \ldots, N$

k: index of the crew, $k = 1, 2, \ldots, \eta$

g: index of duty types, $g = 1$, represents early duty; $g = 2$, represents day duty; $g = 3$, represents late duty

z_g: number of crew with duty g

d_i: departure time of trip i at the starting station

a_i: arrival time of trip i at the terminal

T_0: the least layover time at the starting station for each crew

T_g: the maximum value of work hours for each crew with duty g

B_g: the earliest sign-on time of crew with duty g

E_g: the latest sign-off time of crew with duty g

$x_{g \cdot k}^i$: decision variables to indicate whether trip i is carried out by crew k with duty g, $x_{g \cdot k}^i$ takes two values: $x_{g \cdot k}^i = 1$, if trip i is carried out by crew k with duty g; $x_{g \cdot k}^i = 0$, otherwise

$y_{g,k}^{ij}$: a binary 0-1 variable which indicates the status of trip i and trip j carried out by crew k with duty g. $y_{g,k}^{ij}$ takes two values: $y_{g,k}^{ij} = 1$, if trip j is the next trip after trip i carried out by crew k with duty g; $y_{g,k}^{ij} = 0$, otherwise.

2.3. Bus Crew Scheduling Model

As defined above, the crew scheduling problem can be formulated as follows:

$$\min \sum_{g=1}^{G} \sum_{k=1}^{z_g} \sum_{i=1}^{N} \sum_{j=1}^{N} y_{g,k}^{ij} \cdot (d_j - a_i), \tag{2.1}$$

$$x_{g,k}^{j} \cdot d_j - x_{g,k}^{i} \cdot a_i \geq T_0, \quad \forall g, k, \tag{2.2}$$

$$\sum_{i=1}^{N} x_{g,k}^{i} \cdot (a_i - d_i) \leq T_g, \quad \forall g, k, \tag{2.3}$$

$$B_g \leq x_{g,k}^{i} \cdot d_i \leq E_g, \quad \forall g, k, \tag{2.4}$$

$$z_2 \geq \theta, \tag{2.5}$$

$$\sum_{g=1}^{G} z_g \leq \eta, \tag{2.6}$$

$$\sum_{g=1}^{G} \sum_{k=1}^{z_g} x_{g,k}^{i} = 1, \quad \forall i, \tag{2.7}$$

$$y_{g,k}^{ij} = x_{g,k}^{i} \cdot x_{g,k}^{j}, \quad \forall g, k. \tag{2.8}$$

The objective is to minimize the total idle time of crew. As to crew k with duty g, the idle time between two adjacent trips i and j can be calculated by $y_{g,k}^{ij} \cdot (d_j - a_i)$. Inequality (2.2) guarantees that the time shift constraint should be satisfied by each of two adjacent trips carried out by the same crew. Inequalities (2.3) and (2.4) represent the constraints of work intensity and duty type compatibility, respectively. Inequality (2.5) is the number of day duty constraint, which denotes that the number of crew with day duty should be more than or equal to the constant θ. Inequality (2.6) is the crew number constraint, which means that the total number of crew required should be no more than the value η. Constraint (2.7) assures that each trip will be carried out by exactly one crew. Constraint (2.8) shows the relationship between variable $y_{g,k}^{ij}$ and decision variable $x_{g \cdot k}^{i}$.

3. Algorithm Design

Tabu search algorithm uses a neighborhood search procedure to iteratively move from one potential solution to an improved solution, until some stopping criterion has been satisfied. The tabu search is a metaheuristic local search algorithm that can be used for solving
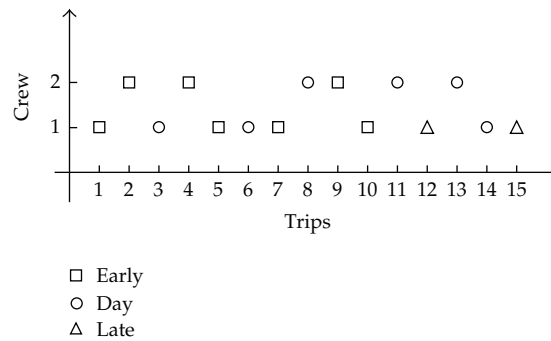
Figure 2: The expression of solution.

Figure 3: Decoding diagram.

combinatorial optimization problems. Because the major advantages of this algorithm are its simplicity, speed, and flexibility, and the urban bus crew scheduling model in the paper is a complex zero-one programming problem, we can use tabu search algorithm easily. The main parameters of algorithm are designed as follows.

3.1. Expression of Solution

The two-dimensional integer array encoding method can be used for the solution of crew scheduling problem, in which rows are the crews, the first column is the duty type of crew and other columns are the trips. The trips are numbered in ascending departure time order. Assume that there are fifteen trips carried out by five crews. The trip chains of each crew can be expressed as follows: crew 1: **1**-1-5-7-10, crew 2: **1**-2-4-9, crew 3: **2**-3-6-14, crew 4: **2**-8-11-13, and crew 5: **3**-12-15; the expression of solution can be shown in Figure 2.

Based on the value of two-dimensional array, the decoding process is the inverse of encoding process. Take the data of Figure 2, as an example, there are three duty types of crew, and the crew number with early duty (duty **1**) is two. The number of trips carried out by crew 1 with early duty is 1, 5, 7, and 10, thus, the variables are $x_{1,1}^1 = 1$, $x_{1,1}^5 = 1$, $x_{1,1}^7 = 1$, and $x_{1,1}^{10} = 1$. The number of trips carried out by crew 2 with early duty is 2, 4, and 9, thus, the variables are $x_{1,2}^2 = 1$, $x_{1,2}^4 = 1$, $x_{1,2}^9 = 1$. The decoding methods of other crews are similar to the former method. The decoding results are shown in Figure 3.
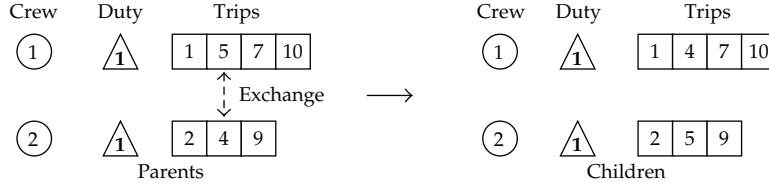
Figure 4: Trips exchange strategy.

3.2. Generation of Initial Solution

The initial solution is the starting point of algorithmic search. A superior quality of the initial solution will enable the algorithm to quickly converge to the optimal solution. In the process of generating the initial solution, the constraints with time shift, work intensity, and duty type compatibility should be satisfied. The procedure of the algorithm can be taken as follows:

Step 1 (Initialization). The trips set carried out by crew k with duty g is set to an empty set, namely, $\lambda_{g,k} = \emptyset$, for all g, k. The optional duty type set for trip i can be defined as follows: $Q_i = \{g \mid B_g \leq d_i \leq E_g, \ g \in \{1, 2, \ldots, G\}\}$, for all i. Let $i = 1$, $z_g = 0$.

Step 2. Determine the duty type g of crew carrying out trip i with the following equation: $g = \min\{s \mid s \in Q_i\}$.

Step 3. Calculate the crew number k and the largest trip number i' corresponding to duty g. If $z_g > 0$, $k = \{l' \mid \alpha_{l'} = \min\{\alpha_l \mid l \in \{1, 2, \ldots, z_g\}\}$, where $\alpha_l = \max\{s \mid s \in \lambda_{g,l}\}\}$, $i' = \alpha_k$, go to Step 4; otherwise, go to Step 6.

Step 4. Verify time shift constraint. If $d_i - a_{i'} \geq T_0$, go to Step 5; otherwise, go to Step 6.

Step 5. Verify work intensity constraint. Calculate the total work time of crew k with duty g, $H_{g,k} = \sum_s x^s_{g,k} \cdot (a_s - d_s) + a_i - d_i$. If $H_{g,k} \leq T_g$, go to Step 7; otherwise, go to Step 6.

Step 6. Let $z_g \leftarrow z_g + 1$, $k \leftarrow z_g$, go to Step 7.

Step 7. Let $\lambda_{g,k} = \lambda_{g,k} \cup \{i\}$, $i \leftarrow i + 1$, go to Step 8.

Step 8. If $i > n$, stop and output the results; otherwise, go to Step 2.

3.3. Neighborhood Structure

The neighborhood structure uses trips exchange and inserts a strategy between different crews. Trips exchange strategy can be described as follows: a single exchange point on both two parents' trip chains is selected. The trip number of that point is swapped between the two parent organisms. The resulting organisms are the children. For example, trip 5 of crew 1 (**1**-1-5-7-10) is exchanged with trip 4 of crew 2 (**1**-2-4-9), and new solutions can be gained. The trips chain of crew 1 has become **1**-1-4-7-10, and the trips chain of crew 2 has become **1**-2-5-9, shown in Figure 4.

Trips inserting a strategy can be described as follows: one trip from crew 1 will be inserted into the trips chain of crew 2. The insert point depends on the ascending departure
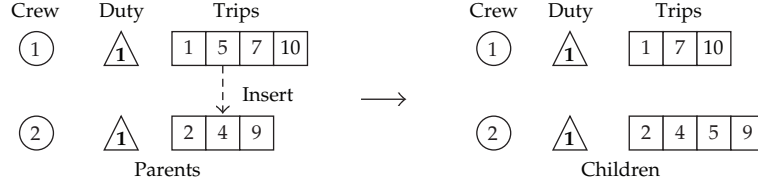
Figure 5: Trips insert strategy.

time order of the trips chain of crew 2, and two new trips chains of crew will reincarnation. For example, trip 5 of crew 1 (**1**-1-5-7-10) is inserted into the trips chain of crew 2 (**1**-2-4-9); new solutions can be gained. The trips chain of crew 1 has become **1**-1-7-10, and the trips chain of crew 2 has become **1**-2-4-5-9, shown in **Figure 5**. The trips exchange or insert operation cannot be carried out, if the trips chain of any crew cannot satisfy time shift, work intensity, or duty type compatibility constraints with any of the operations.

3.4. Evaluation of Solution

In order to search for better solutions in the algorithmic iterative process, it is necessary to evaluate the solution. It should calculate the value of objective function and considere the constraints at the same time. Because the initial solution has satisfied time shift, work intensity, and duty type compatibility constraints, and the new solution generated in neighborhood search also satisfies the former three constraints, the number of day duty and total crew constraints are the only factors which need to be considered. As for the solution which cannot satisfy the number of day duty and total crew constraints, it should be punished. Here we can take two parameters α and β as the punish factors, and both of the values take large positive numbers. If the solution can satisfy the number of day duty and total crew constraints, the value of fitness function and objective function will be equal; otherwise the value of fitness function will be much larger than the objective function, which means that the set of values for decision variables cannot create the feasible solution. The fitness function can be formulated as follows:

$$f = Z + \alpha \cdot \max\{\theta - z_2, 0\} + \beta \cdot \max\left\{\sum_{g=1}^{G} z_g - \eta, 0\right\}, \qquad (3.1)$$

where $Z = \sum_{g=1}^{G} \sum_{k=1}^{z_g} \sum_{i=1}^{N} \sum_{j=1}^{N} y_{g,k}^{ij} \cdot (d_j - a_i)$ represents the value of objective function.

3.5. Other Parameters

The record of tabu table is the transform (exchange or insert) node, and tabu length takes fixed length. Select the regulation based on the value of evaluation as aspiration criterion, that is, the solution of the objective can be free, if it is better than any of the currently known best candidate solution. The stopping criterion is based on the value of fitness function. If the best value does not change after a given number of iterations, the algorithm will stop the calculation.

Table 1: Departure time of trips.

TN	DT	TN	DT	TN	DT	TN	DT	TN	DT	TN	DT
01	6:30	29	9:22	57	11:51	85	14:29	113	17:07	141	19:13
02	6:45	30	9:26	58	11:56	86	14:34	114	17:14	142	19:17
03	6:58	31	9:30	59	12:02	87	14:39	115	17:20	143	19:21
04	7:10	32	9:34	60	12:06	88	14:43	116	17:25	144	19:25
05	7:20	33	9:38	61	12:11	89	14:50	117	17:30	145	19:29
06	7:25	34	9:41	62	12:16	90	14:56	118	17:35	146	19:35
07	7:31	35	9:48	63	12:21	91	15:01	119	17:39	147	19:39
08	7:38	36	9:53	64	12:27	92	15:08	120	17:43	148	19:44
09	7:44	37	10:00	65	12:33	93	15:15	121	17:48	149	19:48
10	7:48	38	10:07	66	12:38	94	15:21	122	17:51	150	19:53
11	7:53	39	10:13	67	12:42	95	15:27	123	17:56	151	20:00
12	7:58	40	10:18	68	12:46	96	15:32	124	18:00	152	20:05
13	8:03	41	10:23	69	12:51	97	15:38	125	18:05	153	20:11
14	8:08	42	10:29	70	12:56	98	15:42	126	18:09	154	20:17
15	8:12	43	10:34	71	13:02	99	15:49	127	18:13	155	20:23
16	8:17	44	10:40	72	13:08	100	15:53	128	18:17	156	20:30
17	8:22	45	10:44	73	13:13	101	15:59	129	18:21	157	20:37
18	8:26	46	10:49	74	13:19	102	16:05	130	18:25	158	20:43
19	8:30	47	10:53	75	13:26	103	16:10	131	18:29	159	20:50
20	8:34	48	10:58	76	13:33	104	16:16	132	18:34	160	20:57
21	8:39	49	11:03	77	13:40	105	16:21	133	18:39	161	21:05
22	8:45	50	11:09	78	13:48	106	16:27	134	18:43	162	21:13
23	8:51	51	11:15	79	13:55	107	16:32	135	18:47	163	21:20
24	8:57	52	11:22	80	14:02	108	16:38	136	18:51	164	21:27
25	9:03	53	11:29	81	14:08	109	16:42	137	18:55	165	21:35
26	9:09	54	11:35	82	14:13	110	16:48	138	19:00	166	21:43
27	9:14	55	11:41	83	14:18	111	16:53	139	19:04	167	21:50
28	9:18	56	11:46	84	14:23	112	16:59	140	19:08	168	22:00

TN: trip number; DT: departure time.

4. Numerical Example

In a certain urban bus line, there are 168 trips in a day. The departure time of each trip can be obtained from Table 1. The running time from the starting station to the terminal is 30 minutes. The least layover time at starting station for each crew is 3 minutes. The maximum number of crew provided by bus company is 26, and the least crew number with day duty is 9.

The parameters of tabu search algorithm are taken as follows: the tabu length is 6, both the punish factors α and β are 10000, and the given number of iterations without improving the solution is 100. The other parameter is given in Table 2. The optimal solution can be calculated using VC++ program, shown in Table 3. The objective value of optimal solution is 1198 minutes.

Table 3 shows that the total number of crew is 26, where the crew number with early, day, and late duty is 8, 10, and 8, respectively. The earliest and latest departure times of each crew satisfy duty type compatibility constraint. Due to the same running time of trips, the crew with longest working time is the one who carries out with the largest number of trips,

Table 2: Parameter of different duty types.

Duty types	Earliest sign-on time	Latest sign-off time	Maximum working time (min)
Early	6:00	13:00	300
Day	9:00	18:00	300
Late	15:00	22:00	300

Table 3: The optimal results of crew scheduling.

Crew number	Duty type	Trips	Earliest departure time	Latest departure time
01	Early	1-4-9-16-24-31	6:30	9:30
02	Early	2-5-11-18-26-35-47-53-59	6:45	12:02
03	Early	3-7-14-22-30-39-46	6:58	10:49
04	Early	6-12-20-28-36	7:25	9:53
05	Early	8-15-23-32-40-48-54	7:38	11:35
06	Early	10-17-27-37-44	7:48	10:40
07	Early	13-21-29-38-45	8:03	10:44
08	Early	19-25-33-42-51-60-67	8:30	12:42
09	Day	34-41-52-61-68-74	9:41	13:19
10	Day	43-50-56-63-70	10:34	12:56
11	Day	49-55-62-69-75-80	11:03	14:02
12	Day	57-64-71-77-83-91-97-104-111	11:51	16:53
13	Day	58-65-72-78-84-90	11:56	14:56
14	Day	66-73-79-85-92-99-106	12:38	16:27
15	Day	76-81-88-95-102-109	13:33	16:42
16	Day	82-89-96-103-110	14:13	16:48
17	Day	86-93-100-108-114-121	14:34	17:48
18	Day	87-94-101-107-113	14:39	17:07
19	Late	98-105-112-118-127-135-143-151-157	15:42	20:37
20	Late	115-123-131-139-147-155-162-167	17:20	21:50
21	Late	116-124-132-140-148-156-164-168	17:25	22:00
22	Late	117-125-133-141-149	17:30	19:48
23	Late	119-129-137-145-153-161-166	17:39	21:43
24	Late	120-128-136-144-152-160-165	17:43	21:35
25	Late	122-130-138-146-154-159	17:51	20:50
26	Late	126-134-142-150-158-163	18:09	21:20

namely, the numbers 2, 12, and 19. The actual working time of the former three crews has the same value of 270 minutes, which is smaller than the maximum working time. The average crew working time with day duty is relatively small with the influence of time shift and the number of day duty constraints.

5. Conclusions

In this paper, a 0-1 integer programming model with the objective of minimizing the total idle time of crew for a circle bus line in consideration of mainly various working rules and duties

of different crews is developed. The considered working rules mainly involve time shift, work intensity, and duty type compatibility constraints, and the duties of crew include early, day, and late modes. A tabu search algorithm is then proposed to solve the model, and a numerical example has been given to show the effectiveness of the obtained results. The result shows that the model can effectively solve the scheduling problem with multiple duty types, and this method can thus serve as a useful tool for urban transit management department with more reasonable and pertinent assistant decision support. Furthermore, considering the crew scheduling problem with multiple transit lines and multiple depot constraints is an important topic for further research.

Acknowledgments

References

[1] B. M. Smith and A. Wren, "A bus crew scheduling system using a set covering formulation," *Transportation Research Part A*, vol. 22, no. 2, pp. 97–108, 1988.

[2] K. Darby-Dowman and G. Mitra, "An extension of set partitioning with application to scheduling problems," *European Journal of Operational Research*, vol. 21, no. 2, pp. 200–205, 1985.

[3] M. Mesquita and A. Paias, "Set partitioning/covering-based approaches for the integrated vehicle and crew scheduling problem," *Computers and Operations Research*, vol. 35, no. 5, pp. 1562–1575, 2008.

[4] J. E. Beasley and B. Cao, "A dynamic programming based algorithm for the crew scheduling problem," *Computers and Operations Research*, vol. 25, no. 7-8, pp. 567–582, 1998.

[5] H. R. Lourenço, J. P. Paixão, and R. Portugal, "Multiobjective metaheuristics for the bus-driver scheduling problem," *Transportation Science*, vol. 35, no. 3, pp. 331–343, 2001.

[6] D. Huisman, R. Freling, and A. P. M. Wagelmans, "Multiple-depot integrated vehicle and crew scheduling," *Transportation Science*, vol. 39, no. 4, pp. 491–502, 2005.

[7] G. Mitra and K. Darby-Dowman, "A computer-based bus crew scheduling system using integer programming," *Computer Scheduling of Public Transport*, vol. 35, no. 5, pp. 223–232, 1985.

[8] M. Desrochers and F. Soumis, "A column generation approach to the urban transit crew scheduling problem," *Transprotation Science*, vol. 23, no. 1, pp. 1–13, 1989.

[9] R. Clement and A. Wren, "Greedy genetic algorithms, optimizing mutations and bus driver scheduling," *Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, vol. 430, pp. 213–235, 1995.

[10] J. E. Beasley and B. Cao, "A tree search algorithm for the crew scheduling problem," *European Journal of Operational Research*, vol. 94, no. 3, pp. 517–526, 1996.

[11] A. Ceder, *Public Transit Planning and Operation-Theory, Modelling and Practice*, chapter 10, Elsevier, Oxford, UK, 2007.

[12] Y. Shen and Y. Ni, "Model for crew scheduling with time windows and multi-neighborhood structures," *Journal of Huazhong University of Science and Technology*, vol. 36, no. 12, pp. 31–34, 2008.

[13] Y. J. Yang, Y. P. Wang, and X. M. Zhao, "Research on staff scheduling of urban passenger taxi dispatching center based on genetic algorithm," *Journal of Highway and Transportation Research and Development*, vol. 27, no. 7, pp. 142–146, 2010.

[14] H. M. Niu, "Determination of the skip-stop scheduling for a congested transit line by bilevel genetic algorithm," *International Journal of Computational Intelligence Systems*, vol. 6, no. 4, pp. 1158–1167, 2011.

[15] V. Srdjan, E. Aleksandar, and L. Imre, "Optimization of workflow scheduling in utility management system with hierarchical neural network," *International Journal of Computational Intelligence Systems*, vol. 4, no. 4, pp. 672–679, 2011.

[16] O. Atli, "Tabu search and an exact algorithm for the solutions of resource-constrained project scheduling problems," *International Journal of Computational Intelligence Systems*, vol. 4, no. 2, pp. 255–267, 2011.