Iranian Journal of Operations Research
Vol. 9, No. 1, 2018, pp. 49-61
DOI: 10.29252/iors.9.1.49

Efficient Algorithm Based on Network Flows for Two-Depot Bus Scheduling Problem

M. Niksirat[1,*]

In this paper bus scheduling problem under the constraints that the total number of buses needed to perform all trips is known in advance and the energy level of buses is limited, is considered. Each depot has a different time processing cost. The goal of this problem is to find a minimum cost feasible schedule for buses. A mathematical formulation of the problem is developed. When there are two depots, a polynomial time algorithm is developed for the problem and theoretical results about the complexity and correctness of the algorithm is presented. Also, several examples are introduced for illustrating validity of the algorithm.

***Keywords**: Bus scheduling problem, Minimum cost network flow, Polynomial time algorithm, Fixed job scheduling.*

Manuscript was received on 01/04/2017, revised on 12/09/2017 and accepted for publication on 03/03/2018.

1. Introduction

Planning transportation network is an important task for managing transportation systems. This task consists of different steps. Step 1 is related to land use, transportation demand, etc. After assessing a transportation network and its properties, such as multiple type vehicle size, operating cost and public transportation demand, the frequency setting for each line of different transportation modes is investigated in step 2. Also, design of a timetable for servicing the related demand and frequencies is the aim of step 3. The creation of a timetable that can be implemented and that satisfies constraints, even optimizing the objective function, is considered in this step. Transportation scheduling provides powerful approach for the problem of synchronization of different modes in step 2 and step 3. This synchronization is a way to efficiency integrate different modes (Guihaire and Hao [7]).

In step 4, assignments of multiple vehicles types to trips are made, and a schedule is created for each vehicle type. Crew assignment is performed in step 5. In this step, drivers are assigned to vehicles, with the vehicle scheduling from step 4, and the number of drivers and their work times are calculated. Given the vehicles operating schedules, their maintenance and inspection requirements and the availability of maintenance resources and crews, in step 6, vehicles are scheduled for maintenance and assigned to limited existing facilities such that each vehicle is maintained in a timely manner, while the amount of time that the vehicles are pulled out of their scheduled services for maintenance is minimized (Huimin et al. [10]).

In this paper we focus on step 4, bus scheduling problem, in which the designer seeks a set of schedules for buses so that each trip is performed exactly once with minimum cost. The problem is

[1] Department of Computer Science, Birjand University of Technology, Birjand, Iran, Email: niksirat@birjandut.ac.ir.

easy if there is only one depot for buses and different efficient algorithm can be considered in this case (see e.g. Bertsekas [2] and Freling et al. [5]).

When there are two or more depots for buses, the problem is proved to be NP-hard generally. So, several researchers applied heuristic algorithms for efficiency solving the problem. The problem is well studied in the literature. Researchers focus on both heuristic and exact algorithms. See e.g. Eliiyi et al. [3], Hadjar et al. [9], Kliewer et al. [14], He et al. [8] and Fallah et al. [4] for exact algorithms and Shui et al. [18], Shafahi and Khani [17], Musavi et al. [15] and Kim and Kim [13] for heuristic solution. Also, the combination of exact and heuristic algorithms was proposed in Guedes and Borenstein [6], Wagale et al. [19], Kulkarni et al. [12] and Ibarra-Rojas et al. [11].

In this paper, bus scheduling problem is considered when there are two depots. Each depot has a different time processing cost. Also a time limitation is considered for each bus. Also the total number of buses needed to perform all trips is known in advance. A mathematical formulation of the problem is developed. Also, a polynomial time algorithm based on network flows is introduced for solving this version of the problem.

The rest of this paper is organized as follows: the bus scheduling problem is defined and is modeled in Section 2. Section 3 provides a polynomial time algorithm for solving the problem when there are two depots. The theoretical results of the proposed algorithm are discussed in Section 4. Section 5 ends the paper with a brief conclusion and future remarks.

2. Bus Scheduling Problem Model

In this section bus scheduling problem is illustrated and a mathematical model is developed. The proposed model is a modified model of bus scheduling problem studied in the literatures, see e.g. Bertsekas**Error! Reference source not found. Error! Reference source not found.**[2] and **Error! Reference source not found.**Guedes and Borenstein [6]. The objective of the problem is to create efficient schedules for buses to performs all trips. A trip is the transportation of passengers from an origin station to a destination. A trip should be performed in a certain fixed time. So a trip $i(i \in \{1,2,...,N\})$ has a fixed start time s_i and a fixed finish time f_i. The start time and finish time of the trips are predetermined and deterministic and the processing time of the trip is $d_i = f_i - s_i$. A bus for performing a distinct trip should be located at start location exactly on the start time of that trip. Also the time needed to transport from destination location of trip i to the start location of trip j is $t_{i,j}$. This time can be considered as a set up time for performing trip j directly after trip i, which is sequence-dependent. Figure (1) shows an instance of bus scheduling problem with 7 trips.

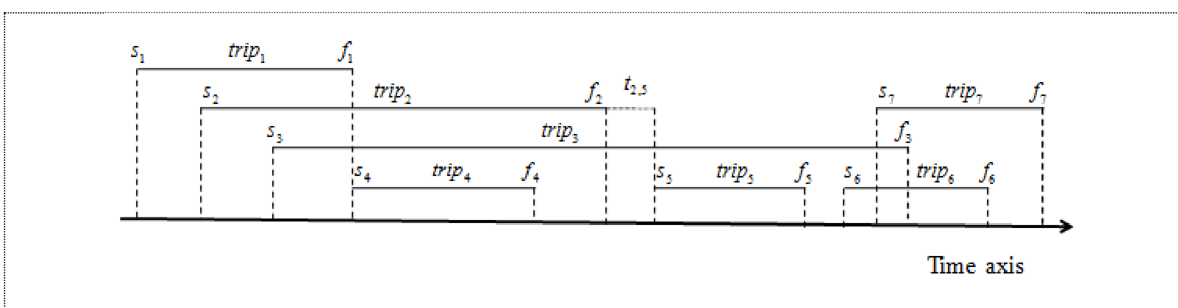

Figure 1. An instance of bus scheduling problem with 7 trips.

Assume that there are K depots with B buses $b \in \{1,2,...,l_1,l_1+1,...,l_2,...,l_{k-1}+1,...,l_k,...,l_{K-1},...,l_K\}$ in which $\{l_{k-1}+1,...,l_k\}$ is the buses of depot k. The buses in each depot has different cost. Assume C_i is the time processing cost of buses in depot k. Also, due to the fuel constraint, the travel time of each vehicle is limited to T.

Due to the time limitation of each bus and time interval of each trip, some trips cannot be performed by the same bus. More precisely, if the start time of trip j be less than $f_i + t_{i,j}$, trip j cannot be performed directly after trip i by the same bus. Also if the difference between finish time of trip j and start time of trip i is less that T, trips i and j cannot be performed by the same bus due to the time limitation of the bus. For trip i, the set of trips that cannot be performed with i by the same bus is denoted by Q_i and defined as follows:

$$Q_i = \{\{j \mid s_j < f_i + t_{i,j}\} \cup \{j \mid f_j - s_i < T\}\}.$$

The total number of buses needed for performing all trips is known in advance and is equal to B. The problem should select $B_1 < l_1$ buses of depot 1, $B_2 < l_2 - l_1$ buses of depot 2, …, and $B_K < l_K - l_{K-1}$ buses of depot K so that $B = B_1 + B_2 + ... + B_K$ and all trips are performed. For every depot k a network $G_k = (E_k, A_k)$ is considered, in which E_k contains one node for every trip and two nodes o_k and d_k for depot k i.e. $E_k = \{o_k, 1, 2, ..., N, d_k\}$ and $A_k = \{(o_k, j) \mid j = 1,...,N\} \cup \{(j, d_k) \mid j = 1,...,N\} \cup \{(i,j) \mid i = 1,...,N, j = 1,...,N, j \notin Q_i\}$. The variable $x_{i,j}^k$ is defined as follows:

$$x_{i,j}^k = \begin{cases} 1 & \text{if trip } j \text{ is performed after trip } i \text{ by a bus of depot } k \\ 0 & \text{otherwise} \end{cases}$$

Now the model of bus scheduling problem can be stated as:

$$\begin{aligned}
&\sum_{k=1}^{K} c_k \left(\sum_{(i,j)\in A_k} t_{i,j} x_{i,j}^k + \sum_{i\in T} (f_i - s_i) \sum_{(i,j)\in A_k} x_{i,j}^k \right),\\
&s.t.\\
&\quad \sum_{k=1}^{K} \sum_{(i,j)\in A_k} x_{i,j}^k = 1, \qquad \forall i \in T,\\
&\quad \sum_{\{j \mid (o_k,j)\in A_k\}} x_{o_k j}^k = B_k, \quad \forall k=1,...,K,\\
&\quad \sum_{\{j \mid (i,j)\in A_k\}} x_{i,j}^k - \sum_{\{j \mid (j,i)\in A_k\}} x_{j,i}^k = 0, \qquad \forall i \in T,\\
&\quad B_1 + B_2 + ... + B_K = B,\\
&\quad x_{i,j}^k \in \{0,1\}, \qquad \forall (i,j) \in A_k.
\end{aligned}$$

The objective function minimizes transportation cost, which is stated as total transportation time between trips and trip times performed by each depot multiplied by depot cost. The first constraint ensures that each trip should be assigned to exactly one depot. Also the second constraint states that the number of buses used from depot k should be equal to B_k. Flow conservation constraint is stated by third constraint and the forth constraint is insured that the total number of buses should be equal to B. Finally, binary restriction of variables is stated by the fifth constraint.

[DOI: 10.29252/iors.9.1.49]

3. Polynomial Time Algorithm Based on Network Flow

In this section we propose a polynomial time algorithm for bus scheduling problem when there are two depots. So, Assume there are only two depots with cost C_1 and C_2 such that $C_1 > C_2$. The algorithm builds a flow network based on each trip start time, finish time and set up time considering the total number of buses and then obtain a minimum cost flow in this network. Before describing the algorithm, a fundamental property of a feasible schedule of the problem is stated.

Partition property: $n(n < N)$ of the total N trip can be performed by $B_1(B_1 < B)$ buses if and only if the remaining $(N-n)$ trip can be proceed by $(B - B_1)$ buses.

In the first phase of the algorithm, an acyclic network $G = (N, A)$ is built. The network has one node for every trip and two nodes s and t for the depot. For every trip we consider one arc from depot to the trip and one arc from each trip to the depot. Also for two trips i and j, (i, j) is added to the graph if and only if $j \notin Q_i$. The cost of arc is equal to the transportation time between trips or between depot and trips. A path in this network represents a sequence of trips that can be performed by the same bus. Also if any $B_1(B_1 < B)$ node-disjoint paths are removed from the network, in the remaining network there are exactly $(B - B_1)$ node-disjoint paths so that each remaining trip belongs to exactly one of this paths.

Some important properties of this network is listed in the following:

1- The network $G = (N, A)$ is a directed acyclic network.

2- Each node in this network located in at least one path from s to t.

3- There exit B node-disjoint path so that each trip belongs to exactly one of these paths. This property is followed from the feasibility of the bus scheduling problem.

4- If any $B_1(B_1 < B)$ node-disjoint paths is removed from this network the resulting network has partition property.

Theorem 3.1. The complexity of building the flow network is $O(2N + N^2)$.

Proof. For each trip j, two arcs (s, j) and (j, t) is added to the network. So the complexity of this step is $O(2N)$. Also for every two trips i and j the algorithm checks whether $j \in Q_i$ and adds arc (i, j) if it is necessary. Obviously $O(N^2)$ is the time to check all the trips. Now it is trivial that the complexity of constructing the network is $O(2N + N^2)$.

After constructing the network, the algorithm uses a minimum cost network flow algorithm to find B_2 node-disjoint paths with minimum transportation time Γ_1. Then the algorithm removes the trips that are located on the B_2 paths and all their incident arcs from network G. In the next step, a minimum cost network flow algorithm is used to find B_1 node-disjoint paths with the minimum

[DOI: 10.29252/iors.9.1.49]

transportation time Γ_2 in the resulting network. The flowchart of the algorithm is demonstrated in Figure (2).

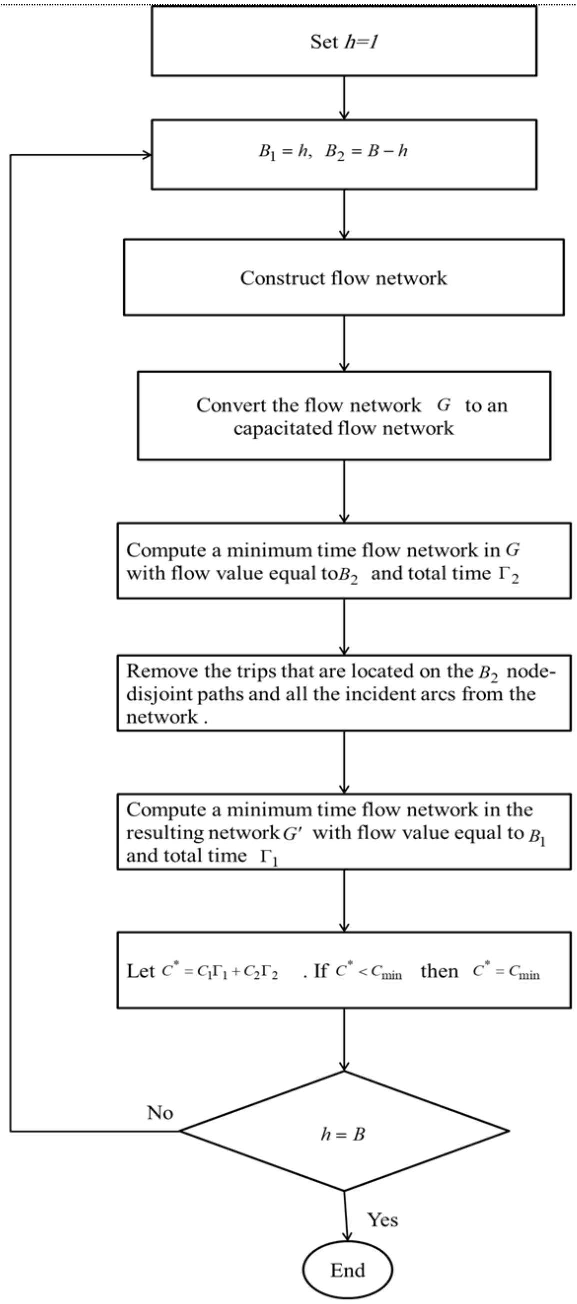

Figure 1. The flowchart of bus scheduling algorithm.

[DOI: 10.29252/iors.9.1.49]

Also the pseudocode of the algorithm is in the following:

$h = 1$;

while $h < B$

$B_1 = h,\ B_2 = B - h$

Construct flow network $G = (E, A)$.

Convert the flow network $G = (E, A)$ to an capacitated flow network: split each node $j \in E - \{s, t\}$ into two nodes j' and j''. Add arc (j', j'') with capacity equal to one and weight equal to $(f_j - s_j)$. Set the capacity of all other arcs equal to 1. Compute a minimum time flow network in $G = (E, A)$ with flow value equal to B_2 and total time Γ_2.

Remove the nodes $j \in E - \{s, t\}$ that are located on the B_2 node-disjoint paths and all the incident arc from the network G. Set the lower bound of each trip node in the network equal to 1.

Compute a minimum time flow network in the resulting network $G' = (E', A')$ with flow value equal to B_1 and total time Γ_1.

Let $C^* = C_1\Gamma_1 + C_2\Gamma_2$. If $C^* < C_{\min}$ then $C_{\min} = C^*$.

$h = h + 1$;

End

Return $C_{\min}$

Figure 3. The pseudocode of Bus scheduling algorithm.

Theorem 3.2. The bus scheduling algorithm correctly computes the minimum cost feasible schedule when there are two depots.

Proof. let Γ be the minimum total time to performing all trips. If Γ_1 be the total time of buses of depot 1 and Γ_2 be the total time of buses in depot 2, then $\Gamma = \Gamma_1 + \Gamma_2$. Science $C_{\min} = C_1\Gamma_1 + C_2\Gamma_2 = C_1\Gamma + (C_2 - C_1)\Gamma_2$ and $C_2 - C_1 > 0$, $C_{\min}$ is minimized if Γ_2 is minimized. The

[DOI: 10.29252/iors.9.1.49]

algorithm obtain the minimum value of Γ_2 and then tries to perform the remaining trips by depot 1 by minimum total time. The feasibility of the schedule is trivial from the way of constructing the flow network. So the algorithm correctly computes the minimum cost.

Theorem 3.3. The complexity of bus scheduling algorithm is $O\left(B\left(2N+N^2+S\left(|E|,|A|\right)+S\left(|E'|,|A'|\right)\right)\right)$, in which $S\left(|E|,|A|\right)$ is the complicity of a minimum cost flow algorithm for network $G=(E,A)$.

Proof. In every iteration of bus scheduling algorithm, according to proposition 1 the algorithm constructs flow network $G=(E,A)$ in $O\left(2N+N^2\right)$then the algorithm computes a minimum time flow in networks $G=(E,A)$ and $G=(E,A)$ in $S\left(|E|,|A|\right)$ and $S\left(|E'|,|A'|\right)$. So the complexity of each iteration is $O\left(2N+N^2+S\left(|E|,|A|\right)+S\left(|E'|,|A'|\right)\right)$. Nothing that the algorithm has $B-1$ iterations the complexity of the algorithm is $O\left(B\left(2N+N^2+S\left(|E|,|A|\right)+S\left(|E'|,|A'|\right)\right)\right)$. If we use cost scaling algorithm, demonstrated by Ahuja et al. [1], for computing minimum time flow the complexity of the algorithm is $O\left(B\left(2N+N^2+|E|^2|A|\log\left(|E|\tau^*\right)+|E'|^2|A'|\log\left(|E'|\tau^{**}\right)\right)\right)$, in which $\tau^*=\min\{t_{i,j}\mid(i,j)\in A\}$ and $\tau^{**}=\min\{t_{i,j}\mid(i,j)\in A'\}$. In this case, bus scheduling algorithm can solve the problem in polynomial time.

4. Numerical Experimentation

In this section, three examples are proposed to illustrate the main concepts and results of the proposed algorithm.

Example 4.1. In this section we consider a simple example with two depots and 7 trips and solve it by the bus scheduling algorithm to show the correctness of the proposed method. The data about the trips is presented in table (1). Also the transportation time between trips and between trips and depots is given in Table (2). Assume $B=3$, $C_1=9$ and $C_2=2$. In the iteration of the algorithm which $B_1=2$ and $B_2=1$, the algorithm first builds a flow network depicted in Figure (4). The flow network is converted to a capacitated network depicted in Figure (5).

Table 1: data of trips for example 4.1.

Trips	Start time	Finish time
1	5	6
2	14	20
3	30	37
4	31	38
5	63	69

[DOI: 10.29252/iors.9.1.49]

6	48	52
7	75	80

[DOI: 10.29252/iors.9.1.49]

Table 2: Data on transportation time for example 4.1.

		Trips							Depots	
		1	2	3	4	5	6	7	1	2
Trips	1	-	5	21	20	57	36	-	70	40
	2	5	-	9	11	38	-	53	60	75
	3	21	9	-	10	-	10	-	45	20
	4	20	11	10	-	26	9	34	30	60
	5	57	38	-	26	-	70	6	40	50
	6	36	-	10	9	10	-	24	90	30
	7	-	53	-	34	6	24	-	55	55
Depots	1	5	13	29	30	60	40	70	-	-
	2	4	14	25	30	45	48	75	-	-

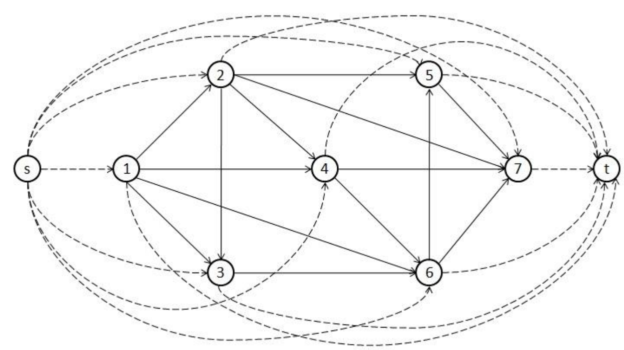

Figure 4. The flow network.

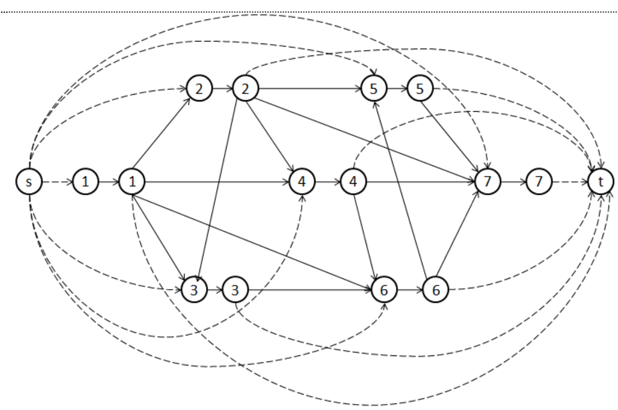

Figure 5. The converted capacitated network.

The minimum cost network flow algorithm is used to construct a flow of value 3 in the capacitated network. The result of this algorithm is as follows:

$s \to 1 \to 4 \to t$

$s \to 2 \to t$

The algorithm removes trips 1, 2 and 4 and all their incident arcs from the network. the resulting network is shown in Figure (6).

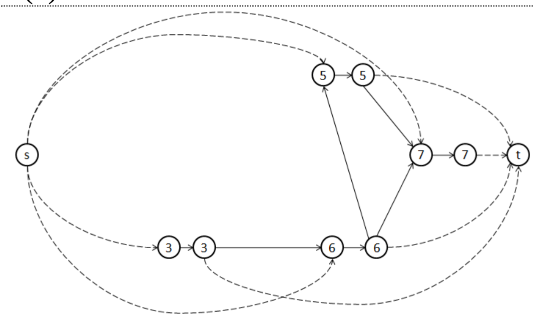

Figure 6. The resulting network by removing trips 1, 2 and 4.

The result of minimum cost flow algorithm in the network of figure (6) is: $s \to 3 \to 6 \to 5 \to 7 \to t$. Also the cost of this iteration is 1534. The minimum cost of the algorithm is 947 which is obtained for the values $B_1 = 1$ and $B_2 = 2$. This problem is modeled as an integer programming in AMPL software and solved with this software. The results of AMPL software are as the same as the proposed method.

Example 4.2. In this example, a real network, the COST 239 European Optical Network is considered (Niksirat et al. **Error! Reference source not found.**[16]). The network is shown in Figure (7). Also the data is presented in Table (3) and Table (4).

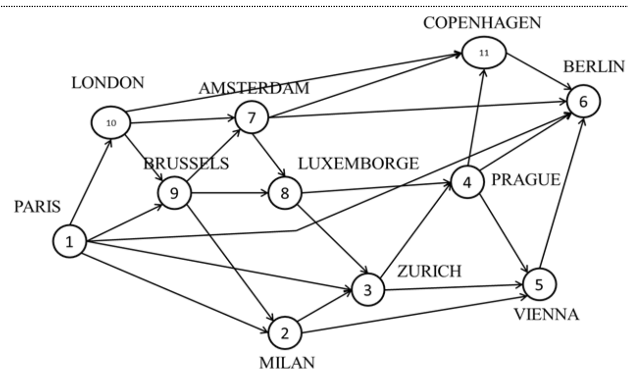

Figure 7. Network of example 4.2.

[DOI: 10.29252/iors.9.1.49]

Table 3. Data on transportation time for of example 4.3.

		Trips											Depots	
		1	2	3	4	5	6	7	8	9	10	11	1	2
Trips	1	-	210	280	-	-	490	-	-	190	970	-	20	50
	2	-	-	270	-	150	-	-	-	-	-	-	40	50
	3	-	-	-	270	330	-	-	-	-	-	-	45	60
	4	-	-	-	-	250	190	-	-	-	-	290	20	25
	5	-	-	-	-	-	230	-	-	-	-	-	30	60
	6	-	-	-	-	-	-	-	-	-	-	-	45	65
	7	-	-	-	-	-	300	-	250	-	-	260	40	60
	8	-	-	270	240	-	-	-	-	-	-	-	35	40
	9	-	180	-	-	-	-	320	250	-	-	-	35	45
	10	-	-	-	-	-	-	250	-	150	-	410	25	70
	11	-	-	-	-	-	190	-	-	-	-	-	15	20
Depots	1	12	13	25	25	25	30	50	30	20	25	30	-	-
	2	15	20	25	30	35	40	60	40	45	40	45	-	-

Table 4. Data of trips for example 4.3.

Trips	Trip time
1	150
2	280
3	280
4	340
5	230
6	220
7	190
8	350
9	280
10	310
11	310

[DOI: 10.29252/iors.9.1.49]

We use the proposed algorithm and AMPL software for this real network. The results of AMPL and the proposed method is the same and the optimal solution is equal to 15720. Also the optimal schedules for each depot are given in the following:

For depot 1:

$s \to 1 \to t.$

For depot 2:

$s \to 8 \to 4 \to 6 \to t$

$s \to 10 \to 9 \to 2 \to 5 \to t$

$s \to 3 \to t$

$s \to 7 \to 11 \to t.$

5. Conclusion

In this paper bus scheduling problem in transportation networks is investigated. We consider the case that there are two depots which every depot has a different cost. Time limitation constraint is considered for buses and the total number of buses is predetermined. The problem is defined and a mathematical integer programming model is developed. Then a polynomial time algorithm for the problem is proposed. The approach of the algorithm is to decompose the problem into two minimum cost network flow problem. Theatrical results of the correctness and complexity of the algorithm is presented. For illustrating the algorithm two numerical examples are introduced. In the future works, the authors are interested to investigate the problem with more than two depots. Also the bus scheduling problem in multimodal transportation network is interesting.

References

[1] Ahuja, R.K., Magnanti, T.L. and Orlin, J.B. (1993), Network flows: theory, algorithms, and applications, Englewood Cliffs, Prentice-Hall International, New Jersey.

[2] Bertsekas, D. P. (1991), Linear Network Optimization: Algorithms and Codes, MIT Press, Cambridge, MA.

[3] Eliiyi, D.T., Ornek, A., Karaku, S.S. (2009), A vehicle scheduling problem with fixed trips and time limitations, *International Journal of Production Economics*, 117,150‑161.

[4] Fallah, M., Mohajeri, A., Jamshidi, M. (2017), A Mixed Integer Programming Approach to Optimal Feeder Routing for Tree-Based Distribution System: A Case Study**,** *Iranian Journal of Operations Research***,** 8(1), 15-43.

[5] Freling, R., Wagelmans, A.P.M., and Paixo, J.M.P. (2001), Models and Algorithms for Single-Depot Vehicle Scheduling, *Transportation Science*, 35, 165-180.

[6] Guedes, P.C. and Borenstein, D. (2015), Column generation based heuristic framework for the multiple-depot vehicle type scheduling problem, *Computers & Industrial Engineering*, 90, 361-370.

[7] Guihaire, V. and Hao, J.K. (2008), Transit network design and scheduling: A global review, *Transportation Research Part A*, 42, 1251-1273.

[8] He, F., Yang, J. and Li, M. (2018), Vehicle scheduling under stochastic trip times: An approximate dynamic programming approach, *Transportation Research Part C: Emerging*

[DOI: 10.29252/iors.9.1.49]

	Technologies, 96, 144-159.
[9]	Hadjar, A., Marcotte, O. and Soumis, F. (2006), A branch-and-cut algorithm for the multiple depot vehicle scheduling problem, *Operations Research*, 54, 130-149.
[10]	Huimin, N., Xuesong, Z. and Xiaopeng, T. (2018), Coordinating assignment and routing decisions in transit vehicle schedules: A variable-splitting Lagrangian decomposition approach for solution symmetry breaking, *Transportation Research Part B: Methodological*, 107, 70-101.
[11]	Ibarra-Rojas, O.J., Giesen, R. and Rios-Solis, Y.A. (2014), An integrated approach for timetabling and vehicle scheduling problems to analyze the tradeoff between level of service and operating costs of transit networks, *Transportation Research Part B*, 70, 35-46.
[12]	Kulkarni, S., Krishnamoorthy, M., Ranade, A., Ernst A.T. and Patil, R. (2018), A new formulation and a column generation-based heuristic for the multiple depot vehicle scheduling problem, *Transportation Research Part B: Methodological*, 118, 457-487.
[13]	Kim, J.U. and Kim, Y.D. (1999), A decomposition approach to a multi-period vehicle scheduling problem, *Omega*, 27, 421-430.
[14]	Kliewer, N., Mellouli, T. and Suhl, L. (2006), A timespace network based exact optimization model for multi-depot bus scheduling, *European Journal of Operational Research*, 175, 1616-1627.
[15]	Musavi, M.M., Tavakkoli-Moghaddam, R. and Rayat F. (2017), A Bi-Objective Green Truck Routing and Scheduling Problem in a Cross Dock with the Learning Effect**,** *Iranian Journal of Operations Research*, 8(1), 2-14.
[16]	Niksirat, M., Hashemi, S.M. and Ghatee M. (2016), Branch-and-price algorithm for fuzzy integer programming problems with block angular structure, *Fuzzy Sets and Systems*, 296, 70-96.
[17]	Shafahi, Y., and Khani, A. (2010), A practical model for transfer optimization in a transit network: Model formulations and solutions, *Transportation Research Part A*, 44, 377-389.
[18]	Shui, X., Zuo, X., Chen, C. and Smith, A.E. (2015), A clonal selection algorithm for urban bus vehicle scheduling, *Applied Soft Comput*ing, 36,36-44.
[19]	Wagale, M., Singh, A.P., Sarkar, A.K., and Apkatkar, S. (2013), Real-time optimal bus scheduling for a city using a DTR model, *Procedia - Social and Behavioral Sciences*, 104, 845-854.

[DOI: 10.29252/iors.9.1.49]