

VEHICLE SCHEDULING WITH HETEROGENEOUS BUS FLEET

Stanislav Palúch¹, Štefan Peško²

The paper studies vehicle scheduling problem formulated as to minimize the number of vehicles with several types of buses. A general mathematical model is presented using graph coloring and bivalent linear programming formulation. A suboptimal algorithm is designed and the way is proposed how to exploit its result to reduce the corresponding bivalent linear programming model.

Key words: vehicle scheduling, heterogeneous bus fleet

1 Introduction

The essential element of regional and/or municipal bus transport is a **bus trip** (sometimes called a **journey**). Bus trip is one move of a bus from a starting bus stop to a finishing bus stop. Several additional bus stops can occur during this travel however these stops are not important for bus scheduling purposes. A trip is defined by four data:

$dp(t)$ – departure place of the trip t

$ap(t)$ – arrival place of the trip t

$dt(t)$ – departure time of the trip t

$at(t)$ – arrival time of the trip t

Let's have two trips:

$$t_1 = (dp(1), ap(1), dt(1), at(1)), \quad t_2 = (dp(2), ap(2), dt(2), at(2)).$$

We will say that the trip t_2 is **linkable** after trip t_1 , or the trip t_1 **precedes** the trip t_2 , and we will write $t_1 \prec t_2$ if

$$dt(2) - at(1) \geq \text{travel_time}[ap(1), dp(2)],$$

i.e. if there exists enough time for a bus to transfer from arrival place of the trip t_1 to the departure place of the trip t_2 so that it arrives to $dp(2)$ sufficiently early so that it can make the trip t_2 .

¹ Doc. RNDr. Stanislav Palúch, CSc, University of Žilina, Faculty of Management Science and Informatics, Department of Mathematical Methods, Univerzitná 8215/1, 01026 Žilina, Slovak Republic, tel.: +421 41 5134 250, E-mail: stanislav.paluch@fri.uniza.sk

² Doc. RNDr. Štefan Peško, CSc. University of Žilina, Faculty of Management Science and Informatics, Department of Mathematical Methods, Univerzitná 8215/1, 01026 Žilina, Slovak Republic, tel.: +421 41 5134 250, E-mail: stefan.pesko@fri.uniza.sk

A **running board** of a bus is a sequence of trips t_1, t_2, \dots, t_k such that $t_1 \prec t_2 \prec \dots \prec t_k$. The linkage $t_i \prec t_j$ is penalized by a cost $c(i,j)$ which can express dead mileage expenses, line change penalty, waiting time penalty etc.

A **bus schedule** is a set of running boards.

Given a set of trips T , we can formulate two fundamental vehicle scheduling problems:

VSP1: To arrange all trips from T into minimum number of running boards.

VSP1: To arrange all trips from T into minimum number of running boards with minimum total cost of all linkages.

There are a lot of additional constraints imposed on running boards. These constraints depend on legislation of corresponding country, on the way of driver duties scheduling, regional traditions and can even vary from bus provider to bus provider.

Standard vehicle scheduling assumes that all buses are the same. Practical experiences show that bus providers use several types of buses with different size and capacity. In this case the set of trips is divided into several subsets according to traffic demand (number of passengers requiring this trip) and according to possibility and/or necessity to provide trips with certain bus type. Therefore for every trip t_i and every bus type b we have one of the following additional constraints: “Trip t_i must be assigned to a bus of the type b .“, “Trip t_i must not be assigned to a bus of the type b .“ and “Trip t_i can be assigned to a bus of the type b .“ Scheduling taking into account these additional constraints will be called a Vehicle Scheduling with Heterogeneous Bus Fleet – **VSHBF** problem.

Standard vehicle scheduling problem can be transformed to an assigning problem and therefore we have a polynomial complexity algorithm for it. However, additional constraints make VSHBF problem hard.

2 Graph formulation and algorithm for VSP1

Let T be a set of trips. Trip digraph of T is a digraph $G_T = (V, E)$ with the vertex set $V = T$ and with the edge set $E = \{(i, j) ; i, j \in V, i \prec j\}$. The set E contains all ordered pairs (i, j) of trips such that trip j is linkable after trip i . Digraph G_T is a transitive acyclic digraph. Every path in G_T is a feasible running board. Hence the problem VSP1 – to arrange all trips from T into minimum number of running boards – can be solved in corresponding trip digraph as to cover all vertices of G_T with minimum number of disjoint paths.

The following auxiliary edge weighted digraph is useful for solving just formulated graph problem: $G_T^A = (V^A, E^A, d)$, where

$$V^A = \{i_1 ; i \in V\} \cup \{i_2 ; i \in V\} \cup \{s, f\}$$

$$E^A = E_1 \cup E_2 \cup E_3 \cup E_4$$

where

$$E_1 = \{(i_1, i_2) ; i \in V\},$$

$$E_2 = \{ (i_2, j_1) ; i, j \in V, i < j \},$$

$$E_3 = \{ (s, i_1) ; i \in V \},$$

$$E_4 = \{ (j_2, f) ; j \in V \}.$$

Let L be a large number. Let's define:

$$d(e) = L \text{ if } e \in E_1, \quad d(e) = 0 \text{ otherwise.}$$

The auxiliary digraph $G_T^A = (V^A, E^A, d)$ is still acyclic. Every path in G_T^A uniquely defines a path in G_T and vice versa.

A **semipath** in a digraph is an alternating sequence of vertices and edges of the form $v_1, e_1, v_2, e_2, \dots, v_{k-1}, e_{k-1}, v_k$, where $e_i = (v_i, v_{i+1})$ or $e_i = (v_{i+1}, v_i)$ and where every vertex occurs at most once. (Roughly speaking – a semipath in a digraph is a path in which edges can be used in reverse direction.) The length of a semipath is the sum of costs of edges used in direction minus the sum of costs of edges used in reverse direction.

Algorithm 1:

Step1: Find a shortest (s, f) – path in G_T^A . Mark the edges of that pas as used, all other edges as unused

Step2: While the set E_1 contains an unused edge do:

Find a shortest (s, f) – semipath in G_T^A .

Mark edges with right direction of that pas as used.

Mark edges with reverse direction of that pas as unused.

Step3: Edges from E_2 define trip linkages from what corresponding bus schedules can be constructed.

The resulting bus schedule doesn't optimize the total cost of bus schedule. In the case that one wants to optimize total linkage cost it suffices to set

$$d(e) = \text{linkage cost of corresponding trips for all } e \in E_2 .$$

However, practical experiences show that it doesn't suffice to minimize linkage cost since legislation, regional tradition and bus operators demand additional requirements. Unfortunately the resulting scheduling problem is no longer polynomial after implementation of consequential objectives and constraints. That's why we have developed a sophisticated neighborhood search procedure based on multiple application of assignment problem which can find a suboptimal solution for very complex objective function. This procedure requires an optimum starting solution from the point of view of VSP1 – i.e. with optimum number of buses.

3 Graph coloring formulation for VSP1

Let T be a set of trips. Collision graph of T is a graph $G_C = (V, H)$, with the vertex set $V = T$ and with the edge set

$$H = \{ (i, j) ; i, j \in V, (i, j) \notin E \text{ and } (j, i) \notin E \}.$$

The set H is the set of such pairs of trips which cannot be serviced by one bus – **incompatible trips**. Every independent set in G_C represents a feasible running board and vice versa. Hence to find an optimum solution of VSP1 means to find an optimum coloring of G_C .

Suppose that $V = \{1, 2, \dots, n\}$. Let x_{ij} be a decision variable, $x_{ij} = 1$ if and only if the trip i is in the running board j , otherwise $x_{ij} = 0$. Denote by n the number of trips in T, let m be an upper bound of number of buses. Then VSP1 can be formulated as follows:

Mathematical model 1:

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^m jx_{ij}$$

$$\text{Subject to: } \sum_{j=1}^m x_{ij} = 1 \quad \text{for all } i \in V$$

$$x_{ik} + x_{jk} \leq 1 \quad \text{for } k \in \{1, 2, \dots, m\}$$

$$\text{and for all } i, j \in V \text{ such that } (i, j) \in H$$

$$x_{ij} \in \{0, 1\} \quad \text{for } i \in V, j \in V$$

Constraints in the first row say that every trip is exactly in one bus schedule. Constraints in the second row say that incompatible trips cannot be in the same bus schedule.

Denote by $V(i)$ the set of all neighbors of the vertex $i \in V$. The set $V(i)$ is in fact the set of all trips incompatible with the trip j . The large number of constraints in the second row can be reduced and we obtain the following mathematical model:

Mathematical model 2:

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^m jx_{ij}$$

Subject to:

$$\sum_{j=1}^m x_{ij} = 1 \quad \text{for } i = 1, 2, \dots, n$$

$$nx_{ik} + \sum_{j \in V(i)} x_{jk} \leq n \quad \text{for } i = 1, 2, \dots, n; k = 1, 2, \dots, m$$

$$x_{ij} \in \{0, 1\} \quad \text{for } i = 1, 2, \dots, n; j = 1, 2, \dots, m$$

This formulation converts polynomial problem VSP1 into NP-hard graph coloring problem therefore it has no practical nor important theoretical meaning. We introduce it as a first step for Vehicle Scheduling with Heterogeneous Bus Fleet – **VSHBF** problem, which is by our conjecture a hard problem.

4 Graph coloring formulation for heterogeneous fleet case

Denote by $F(i)$ the set of vehicles which can provide the trip i . Mathematical model is different from the last one only in the first constraint which says that the trip i has to be serviced by the desired bus type.

Mathematical model 3:

$$\text{Minimize } \sum_{i=1}^n \sum_{j=1}^m jx_{ij}$$

Subject to:

$$\sum_{j \in F(i)} x_{ij} = 1 \quad \text{for } i \in V$$

$$nx_{ik} + \sum_{j \in V(i)} x_{jk} \leq n \quad \text{for } i \in V; \quad k \in \{1, 2, \dots, m\}$$

$$x_{ij} \in \{0, 1\} \quad \text{for } i \in V; \quad j \in V$$

5 Two bus type problem

In the two bus type problem we have two types of trips and two types of buses. The trips of the first type are crowded trips requiring service by high capacity buses of the first type like hinged buses (we will call them maxibuses). The rest of trips are ordinary trips of the second type requiring ordinary buses. Ordinary trip can be serviced by maxibus too, but this is not a desirable instance and should occur only if necessary.

The simplest attitude to this problem is to decompose it into two independent scheduling problem - one for crowded trips and maxibuses and one for ordinary trips and ordinary buses. However, this attitude needn't be optimal since maxibuses can service several ordinary trips what can decrease the number of ordinary buses. Nevertheless just mentioned decomposition gives us the exact number of necessary maxibuses and a upper bound of ordinary buses.

Let us partition the set of trip T into two subset - the first the set of must-trips and the set of may-trip. Algorithm 1 can be modified in order to give a bus schedule with minimum number of vehicles containing all must-trips and maximum possible number of may-trips. Here is the following modification:

1. For all $e \in E_1$ set $d(e) = L^2$, if $e = (i_1, i_2)$ where i is a must-trip,
 $d(e) = L$, if $e = (i_1, i_2)$ where i is a may-trip

2. Modify Step2 of Algorithm 1 as follows:

Step2: While the set E_1 contains an unused edge $e = (i_1, i_2)$ where i is a must-trip do:

We will refer to such modified algorithm as **Algorithm 2**.

Several may-trips remain not scheduled after finishing Algorithm 2.

Now we are prepared to formulate an algorithm for exact minimization of maxibuses and suboptimal minimization of ordinary buses.

Algorithm 3:

Step1: Declare all crowded trips as must-trips and all other trips as may-trips.

Run **Algorithm 2.**

Step2: Declare all unscheduled trips from the Step1 as must-trips and all ordinary trips as may-trips.

Run **Algorithm 2.**

The result is the set of running boards for all ordinary buses with several unscheduled ordinary trips which do not increase the number of maxibuses.

Step3: Run **Algorithm 1** for all crowded trips and all unscheduled ordinary trips from the Step2.

The result is the set of running boards for maxibuses containing all must-trips and all till now unscheduled may-trips.

This algorithm was used for many real world computation with great success. Unfortunately, several cases occurred when Algorithm 3 gave more ordinary buses than the exact minimum.

Therefore we proposed the following procedure:

- Run **Algorithm 1** for all crowded trips from T .
The result is the exact minimum number of maxibuses n_M .
- Run **Algorithm 1** for all trips from T regardless of the trip and bus type.
The result is a lower bound $LB_{-n_{ALL}}$ of all buses.
Since n_M is exact minimum of maxibuses, we have a lower bound of ordinary buses
 $LB_{-n_O} = LB_{-n_{ALL}} - n_M$.
- Run **Algorithm 1** for all ordinary trips.
The result is a lower bound UB_{-n_O} of ordinary buses in two bus type schedule. Upper bound of all buses is $UB_{-n_{ALL}} = UB_{-n_O} + n_M$.

The values n_M , UB_{-n_O} can be used to reduce the size of Model 3 by reducing the sizes of sets $F(i)$.

The lower bound $LB_{-n_{ALL}}$ can be used in the following way. If the degree of a ordinary trip i (i.e. the number of incompatible trips with the trip i) is less than $LB_{-n_{ALL}}$ it can be colored by one of colors from the set $\{1, 2, \dots, LB_{-n_{ALL}}\}$ regardless of coloring of its neighbors. Therefore such trip can be removed from the graph G_C . A sequence of such removal can reduce the problem size significantly.

Similarly a crowded trip can be removed from the graph G_C if its degree is less than n_M . In practice $n_M \ll LB - n_{ALL}$ that's why such reduction will be probably negligible.

Acknowledgement. This research was supported by grant VEGA 1/0135/08.

Literature References

1. CZIMMERMANN, Peter, PEŠKO, Štefan: Autobusové rozvrhy s dvoma typmi vozidiel- Bus Schedules with Two Types of Vehicles, 3-rd International Conference APLIMAT 2004, pp. 321-326
2. PALÚCH, Stanislav: Bus Scheduling as a Graph Coloring Problem, *Komunikácie/Communications*, vol. 4, 2003
3. PALÚCH, Stanislav: Graph Theory Approach to Bus Scheduling Problem, *Studies of the faculty of management science and informatics*, Vol. 9, October, 2001, pp. 53-57
4. PALÚCH, Stanislav: Simultaneous Bus and Crew Scheduling. International Scientific Conference on Mathematics, Žilina, June 30. – July 3., 1998
5. PEŠKO, Štefan: Flexible Bus Scheduling with Overlapping Trips, In: *Quantitative Methods in Economics (Multiple Criteria Decision Making)*, June 5-7, 2008
6. PEŠKO, Štefan: The Minimum Fleet Size Problem for Flexible Bus Scheduling, *Studies of the faculty of management science and informatics*, Vol. 9, October, 2001, pp. 59-65
7. PLESNÍK, Ján: *Grafové algoritmy – Graph Algorithms*, VEDA, Bratislav, 1983