TRANSPORT PROBLEMS 2017 Volume 12 Issue 1
PROBLEMY TRANSPORTU DOI: 10.20858/tp.2017.12.1.10

Keywords: vehicle and crew scheduling; personal bus transport; optimization

Stanislav PALÚCH*, Tomáš MAJER
University of Žilina, Faculty of Management Science and Informatics
Univerzitná 8215/1, 010 26 Žilina, Slovakia
**Corresponding author*. E-mail: stanislav.paluch@fri.uniza.sk

KASTOR – A VEHICLE AND CREW SCHEDULING SYSTEM FOR REGULAR BUS PASSENGER TRANSPORT

Summary. The challenge in vehicle and crew scheduling is to arrange a given set of bus trips into running boards while minimizing certain objectives and complying with a given set of constraints. This scheduling was solved several tens years ago manually by a dispatcher who used his skill, experience, knowledge of history, and intuition. This attitude worked quite well in small instances but could not be applied in larger ones. Development of computers brought opportunities to build computerized vehicle and crew optimization systems. This paper describes a computer system KASTOR developed for vehicle and crew optimization, which complies with the special requirements of Czech and Slovak bus providers, and is significantly different from those in some west European countries.

1. INTRODUCTION

Slovakia, as an integral part of the European Union, invests heavily in the development of the transport system [1]. Vehicle and crew scheduling problem appeared together with implementation of first regular bus passenger transport. This problem consists in creating a daily list of trips to carry (called running boards) for every vehicle. This task was solved manually using skill, knowledge of history, and intuition. However, to create good running boards for larger instances showed to be beyond possibility using manual approach.

The widespread use of computers in the second half of the last century has offered an opportunity to use a quantitative approach for vehicle and crew scheduling of public transport. In 1975, a workshop was held in Chicago on Automated Techniques for Scheduling of Vehicle Operators for Urban Public Transportation Services. By the time of second Workshop, in Leeds in 1980, several systems were in regular use in west Europe and in North America. Subsequent workshops were held in Montreal (1983), Hamburg (1987), again in Montreal (1990), Lisbon (1993), and Cambridge (1997). A lot of literature has been issued from that time. More and more practical mathematical methods were designed, with skyrocketing computer speed and memory accompanied by falling computer cost.

Several attempts were made to design computer-aided vehicle and crew scheduling in former Czechoslovakia before 1980. The mentioned attempts were based on a simulation of dispatcher's work when creating a bus schedule. These experiments were bound to fail before they started as the mentioned attempts could not make use of human experience and intuition of dispatchers.

A serious research in vehicle and crew scheduling in Czechoslovakia started in the period 1980–1984 at Transport Research Institute in Žilina (Slovakia). This research was supported by grant "Optimization of Personal Bus Transport of enterprises ČSAD (Czechoslovak Automobile Transport)" and fully financed by government and state-owned enterprises ČSAD. Since that time, the subject of vehicle and crew scheduling is continuously developed at Transport Research Institute in Žilina and at the Faculty of Management Science and Informatics, University of Žilina.

In west European countries, vehicle and crew scheduling are considered as two consequential steps. First, an optimum bus schedule is computed. Running boards of vehicles are divided into so-called pieces of work using many ways in the second step. Thereafter, a huge number of theoretical driver shifts is created as various combinations of pieces of work. The real set of driver shifts is chosen from the set of theoretical shifts as a result of corresponding set covering problem. This attitude supposes that any driver can drive any bus. Many references to the above-mentioned approach can be found in [4, 5, 10, 13].

In former Czechoslovakia, the ties between drivers and buses were – and till now are – very stiff. There are only two possibilities: one bus – one driver or one bus two drivers. Thus, for vehicle and crew scheduling the possibilities are as follows: one bus running board–one driver shift, or one bus running board–two driver shifts (i.e., one morning shift and one evening shift).

Labor Code and several safety rules impose several very strict constraints to worker shifts. Therefore, every running board has to comply with all safety standards and all requirements of Labor Code, such as safety break, meal break, duration of driver shift, and so on. Detailed specifications of Czech and Slovak conditions and corresponding mathematical models can be found in [3, 8, 9].

However, a simultaneous approach to vehicle and crew scheduling appeared in recent time in several papers – see [2, 7, 12] – also in west European countries. New trends appeared on single vehicle scheduling problem as well – that is, paper [11] studies vehicle scheduling with stochastic trip times.

There are a lot of vehicle and crew scheduling systems in West Europe, Canada, Israel, and other countries. However, these systems are not applicable in Slovakia and Czech Republic due to diametrically different way of managing driver shifts, different Labor Code, other requirements of bus providers, and different cultural habits. Therefore, these systems are not comparable to our attitudes, although both solve similar problem but with totally different constraints.

2. FUNDAMENTAL NOTIONS

A municipal or regional personal bus transport is performed on a transportation network. A transportation network is defined by two matrices, time distance matrix $T(p,r)$ and geographic distance matrix. The value $T(p,r)$ expresses how long it takes for a bus to move from place p to place r.

A bus timetable is composed of bus trips. Full specification of a trip comprises data on corresponding route, bus stops, and related departure and arrival times. However, it suffices to define a bus trip i by four essential data $(dt_i, dp_i, at_i, ap_i\ )$ – departure time, departure place, arrival time, and arrival place for scheduling purposes.

The trip j is linkable after the trip i if $dt_j \geq at_i + T(ap_i, dp_j)$ – that is, if a bus after finishing the trip i can pull into the departure place of the trip j. In this case we will write $i \prec j$.

A running board of a bus is a sequence of trips $i_1, i_2, \dots, i_m$ such that for every $1 \leq k < m$ the trip i_{k+1} is linkable after the trip i_k – that is, $i_1 \prec i_2 \prec \cdots \prec i_m$.

Every running board is, in fact, a sequence of trips performable by one bus – it represents the daily schedule of jobs assigned to this bus.

A bus schedule of the set of trips T is a set of running boards such that every trip is exactly in one running board.

A cost $c(i,j)$ is assigned to every possible linkage $i \prec j$. The cost $c(i,j)$ contains the length of corresponding dead kilometers connected with dead travel of a bus from the arrival place ap_i of the trip i to the departure place dp_j of the trip j. Moreover, it can contain a penalty for line changing, a penalty for too long waiting time, and so on.

A cost of a running board is the total cost of all its linkages. A cost of a bus schedule is the total cost of all its running boards.

The challenge in vehicle and crew scheduling is to assign all trips from a given set of trips S into a certain number of running boards and a certain number of drivers while minimizing certain objectives and complying with given constraints.

Main objectives are as follows:

- O1: Minimize the number of vehicles – that is, minimize the number of running boards
- O2: Minimize the number of dead kilometers
- O3: Minimize the number of drivers
- O4 Minimize the number of line changes

Main constraints are as follows:

- **C1:** All vehicles have to return after work to the places where they started in the morning. (As the starting points can be given one depot, several depots, or this condition can be replaced by the requirement that every bus returns after finishing the daily work to the starting point of corresponding running board.)
- **C2:** All running boards have to fulfill the safety break condition. We will say that running board fulfills safety break condition (SB) or is feasible if in every time interval of 270 minutes long there exists at least 30 minutes of safety break. This safety break can be in one continuous piece or two or three time intervals, with each break at least 10 minutes long.
- **C3:** The time length of every running board must be suitable for one crew or two crews. This means that it must be in time interval $\left(t_1^1, t_2^1\right)$ if it is designed for one driver or in interval $\left(t_1^2, t_2^2\right)$ for two drivers.
- **C4:** All vehicles have to visit the central depot once a day.
- **C5:** All driver shifts have to include a meal break.
- **C6:** The driving time of every driver is limited.
- **C7:** The durations of all shifts of all drivers has to be as uniform (as equal) as possible.

3. INPUT DATA FOR SYSTEM KASTOR

Input data for vehicle and crew optimization contain description of transportation network and that of set of trips.

Transportation network is defined by two files. One of them contains a list of nodes of transportation network – list of all bus stops and street crossings together with their identification numbers. The system refers to nodes with their number rather than by names as numbers can be used as array indexes.

The set of trips contains a list of trips. Every trip record should contain at least the number of bus line, number of trips in line, departure and arrival times, number of departure and arrival bus stops, and the length of the trip route in kilometers. Moreover, it can contain manipulation times before and after trip, another manipulation, vehicle occupancy, vehicle capacity, and other data important for bus provider, but not necessary for optimization. These data are transferred only in order to allow bus provider to make use of them in the resulting optimized bus schedule.

Trips can input into computation in two ways – as single trips or trips can be arranged into contemporary running boards. The second way is better as it helps in revealing many errors.

Our experiences showed that input data for optimization almost every time contained several substantial errors that could completely deteriorate results of computation. For example, a wrong number of departure or arrival place can cause an increase in the number of vehicles and result in incorrect bus schedule. A similar effect has false time data of trips. Missing an edge in graph of transportation network can cause an enormous increase in time and geographical distances or even disconnectivity of transportation network. That is why the first procedure of system KASTOR – input data contains many checks for formal and pragmatic correctness of data.

If the geographical or time length of a trip is less than the geographical or time distance between departure and arrival place given by corresponding distance matrices, the system gives an error message. Moreover, if input trips are arranged in running boards, system can check their compatibility with time and geographical matrices. If time or kilometrical distance between successive trips in a running board is less than that in corresponding matrices, an error message is printed. The last type of check is comparison of time and kilometrical length of edges of transportation network. If the resulting speed (as a ratio kilometers/time) along this edge is too high or too low we can assume that an error occurred. Moreover, if triangular inequality does not hold for an edge, a warning is produced.

4. PROCEDURES OF SYSTEM KASTOR

4.1. Data input

The first procedure is input data from input files. Input files can be created manually in a text editor or transferred from the information system of relevant bus provider. This procedure computes time and geographical distance matrices and provides all possible checks for formal and pragmatic correctness of input data. The above-mentioned matrices are computed using well-known Dijkstra shortest path algorithm.

4.2. Computing the minimum number of vehicles

The second procedure of system KASTOR is used to compute the minimal number of vehicles depending on manipulation. The manipulation after a trip is a time needed for servicing vehicle and passengers immediately after arrival. The same purpose has the manipulation before a trip. Manipulations can be considered as time intervals extending the time duration of trips. As a result of increasing manipulations, two trips linkable with smaller manipulations are no longer linkable with greater manipulations. Therefore, increasing manipulations cause an increase in the minimal number of vehicles. This computation accepts the same manipulation for all trips.

User can specify the interval of manipulations for which corresponding numbers of vehicles will be calculated. Lower limit of this interval can be even a negative number. If the manipulation is a negative number we allow as feasible linkage of overlaying trips by the time equal to the manipulation. Sometimes, a small negative manipulation can cause diminishment in the number of vehicles. Operating personnel can try to let computer find a solution with negative manipulation and then decide whether it is possible to eliminate appeared overlays by small shifts of trips in time. Possible bus savings are worth this effort.

We have used the so-called bipartite trip digraph for calculating a minimum number of vehicles. A bipartite trip digraph is a digraph $G = (V_1 \cup V_2, A,)$, where V_1 is the set of all trip arrivals, V_2 is the set of all trip departures, and where the arc set A is the set of all pairs (a_i, d_j), $a_i \in V_1,\ d_j \in V_2$ such that $i \prec j$ – that is, such that the trip i after arrival a_i can catch the departure d_j of trip j (See Fig 1.a)).

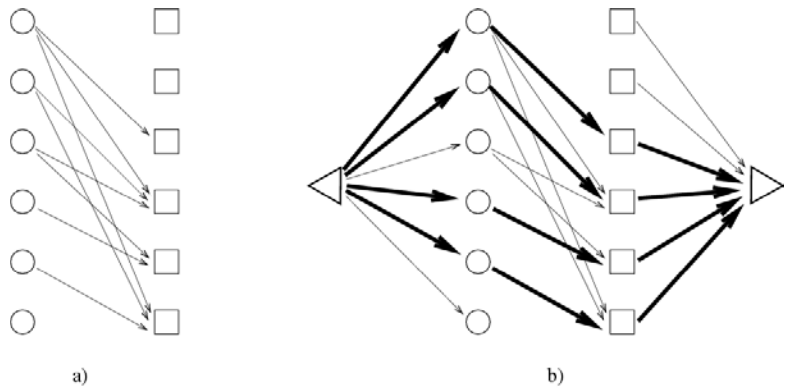

Fig. 1. a) Bipartite trip digraph b) Corresponding capacitated network with maximum flow

Adding fictive source and sink (triangles in Fig. 1.b), for all arcs of the type (source, arrival) and (departure sing) and on setting the capacity all arcs equal to 1, we obtain a capacitated network. Maximum flow in this network determines a bus schedule with minimum number of vehicles. Running boards are constructed using the following rule: if the flow is along arc $\left(a_i, d_j\right)$, the trip j is linked immediately after trip i in the same running board. The well-known Ford-Fulkerson maximum flow algorithm for this problem was used.

4.3. Single-depot and multiple-depot optimization procedures

The third procedure is computation of bus schedule with one depot. This type of optimization is intended for city transport, in which all buses pull out from the depot in the morning and pull in to the depot in the evening. Optimization is made with respect to all dead trips including pull-out and pull-in trips.

The fourth procedure is computation of bus schedule for regional bus transport, in which vehicles return after day shift to their starting bus stop. Optimization is made with respect to all dead trips including return travel from arrival bus stop of the last trip into departure bus stop of the first trip of its running board. This procedure can be used for multiple depot problems too.

We have used the so-called trip digraph for calculating the bus schedule with minimum number of vehicles and with minimum total cost. A trip digraph is a digraph $G = (V, A, c)$, where V is the set of all trips; the arc set A is the set of all pairs (i, j) of trips $i \in V, j \in V$ such that $i \prec j$ and $c: A \to \boldsymbol{R}$ is real function with domain A – $c(i, j)$ represents the cost of linkage of the trip j immediately after trip i. A running board can be modeled in trip digraph as a path in G; the length (cost) of such a path is the cost of corresponding running board. Our problem can be formulated as a problem to cover all vertices of trip digraph G with a minimum number of paths with minimum total cost. We have developed original exact graph algorithms for this problem.

There exists also an integer linear programming model for this problem:
Denote by x_{ij}, a binary decision variable: $x_{ij} = 1$ if the trip j will be linked immediately after trip i, otherwise $x_{ij} = 0$. The optimum bus schedule can be obtained by solving the following problem:

Maximize $\sum_{(i,j)\epsilon A}\left(K - c(i, j)\right) x_{ij}$ subject to

$$\sum_{(i,j)\epsilon A} x_{ij} \leq 1 \text{ for all } i \in V, \sum_{(i,j)\epsilon A} x_{ij} \leq 1 \text{ for all } i \in V, x_{ij} \in \{0,1\}$$

We preferred to use our graph theory approach because we supposed to use our system KASTOR in commercial sphere, and hence we wanted to avoid purchasing an expensive solver. Moreover, our graph theory approach can be modified for two bus types and for computation without returning trips.

Bus schedule obtained using the above-mentioned procedures is optimal, but does not comply with conditions imposed on crews. These can be achieved with the following procedure. We define a complex objective function of a running board, which includes dead mileage, pull-outs and pull-ins, penalties for violation of Labor code, and other penalties dependant on special requirements of transport provider. Thereafter, we minimize the total cost of bus schedule calculated using the defined objectives of running boards. The resulting optimization problem is no longer linear.

We start with bus schedule obtained using previous optimization procedure. All running boards are cut into heads and tails. Thereafter, we try to find an optimum combination of heads and tails of various running boards that leads to an assignment problem. If such assignment brings an improvement we accept it and try another cutting place until no improvement can be found. A substantial enhancement of the above-mentioned attitude can be obtained by dividing running boards into heads, mids, and tails and by assigning mids to various fixed heads and tails. The described optimization procedure proved to be very successful. Approximately 4 million assignment procedure calls occurred in computation for 1000 trips of a regional bus transport.

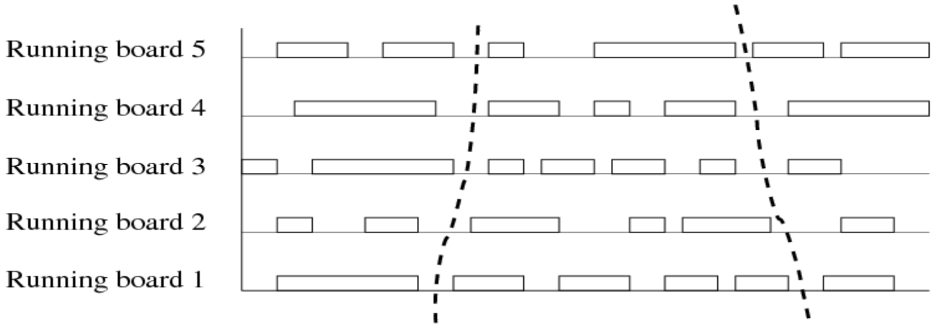

Fig. 2. Splitting running boards into heads, mids, and tails

4.4. Output procedures

The remaining two procedures are devoted to short and complete output of optimization results.

4.5. Menu of optimization procedures

Both procedures – “Optimization for one depot” and “Optimization for regional bus transport” – can be computed in following ways:

- **Optimization with all trips**

- **Optimization without the so-called returning trips.** Returning trips are trips that are inserted into timetable because there is a bus travelling along a route but without passenger demand. This type of optimization has three possibilities:
  - without all returning trips
  - without returning trips, which increases the number of buses
  - without returning trips, which increases the amount of dead kilometers

- **Optimization with two types of vehicles.** In this type of optimization, 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, such as hinged buses (we will call them maxibuses). The rest of the 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 result is a bus schedule with minimum number of maxibuses and minimum number of ordinary buses, wherein maxibuses carry ordinary trips only if it helps in diminishing the number of ordinary buses.

All mentioned possibilities can be combined with choice of zero manipulations, given manipulations, and uniform manipulations. All combinations of mentioned parameters are computed in such a way that the resulting running boards for vehicles comply with all conditions laid to drivers shifts – namely, safety break and time length of shifts.

5. APPLICATION OF STSTEM KASTOR IN PRACTICE

System KASTOR was successfully applied in municipal and regional bus transport of the following Czech and Slovak towns: Trinec, Jablunkov, Havířov, Tachov, Uherské Hradiště, Most and Litvínov, Prachatice, Strakonice, Nymburk, Lysá n/L., Milovice, Prievidza, Martin - Vrútky, Piešťany, Považská Bystrica, Trenčín and others.

Different operators demanded very often various requirements and unlike constraints. Therefore, the objective function had to be adapted to the individual case and sometimes we had to change some optimization algorithms.

Optimization savings ranged from 5% to 20% of dead mileage and 5% to 10% of number of used vehicles. First real-world application in town Třinec reduced the number of buses from 22 to 21 and consumption of fuel by 10 thousands liters a year. The greatest savings in bus number was achieved in town Havířov where the whole organization of bus transport had to be changed due to the relocation of the bus station. Bus schedule optimized using KASTOR yielded 47 buses, whereas the previous manually prepared schedule needed 54 vehicles. However, in several cases the number of vehicles could not be reduced due to very strong morning peak. Latest applications of system KASTOR were a part of complex optimization of regional or municipal bus transport as a consequence of rapid decline in passenger number. Corresponding reduction in the number of trips and sometimes changes in line system does not make possible the comparison of the type before–after. Moreover, confidentiality agreements with bus providers prevent us from publishing details about single instances as they are considered as trade secrets.

References

1. Tkáčová, A. & Danko, J. & Zitricky, A. Verejne obstarávaním v dopravnej infraštruktúre Slovenska v rokoch 2010-2015. *Scientific Papers of the University of Pardubice. Faculty of Economics and Administration. Series D.* 2016. No. 3 (38). P. 190-201. [In Slovak: Public procurement in transport infrastructure of Slovakia between 2010-2015].
2. de Athayde Prata, B. A multiobjective metaheuristic approach for the integrated vehicle and crew scheduling. *Journal of Transport Literature*. 2016. Vol. 10. No. 2. P. 10-14.
3. Černá, A. Optimization of Regional Bus Transport. *Proceedings of International Conference "Transportation Science.* Fakulta Dopravní ČVUT Praha. 6.-7.11.2001. P. 70-75.
4. Daduna, J.R. & Wren, A. (eds.). Computer-Aided Transit Scheduling. Proceedings of the Fourth Interntional Workshop on Computer Aided Scheduling of Public Transport. Springer Verlag. 1988. ISBN 0-387-19441-X, 3-540-19441-X.
5. Daduna, J.R. & Branco, I.B. & Paixao, J.M.P.A. (eds.)*: Computer-Aided Transit Scheduling.* Proceedings. Lisbon, Portugal. July 1993. Springer Verlag. 1988. ISBN 3-540-60193-7.
6. Czimmermann, P. & Peško, Š. Bus Schedules with Two Types of Vehicles. In: *3-rd International Conference APLIMAT* 2004. P. 321-326.
7. Kliewer, N. & Amberg, B. Multiple depot vehicle and crew scheduling with time windows for scheduled trips. *Public Transport*. 2012. Vol. 3. No. 3. P. 213-244.
8. Palúch, S. Simultaneous Bus and Crew Scheduling. *International Scientific Conference on Mathematics*. Žilina. June 30 – July 3, 1998.
9. Palúch, S. Graph Theory Approach to Bus Scheduling Problem. *Studies of the faculty of Management Science and Informatics*. 2001. Vol. 9. P. 53-57.
10. Rousseau, J.M. (eds.). *Computer Scheduling of Public Transport.* Amsterdam, New York, Oxford: North Holland. 1985.
11. Shena, Y. & Xua, J. & Lic, J. A probabilistic model for vehicle scheduling based on stochastic trip times. *Transportat. Research. Part B: Method.* 2016. Vol. 85. P. 19-31.

12. Steinzen, I. & Gintner, V. & Suhl, L. & Kliever, N. A Time-Space Network Approach for the Vehicle- and Crew-Scheduling Problem with Multiple Depots. *Transportation Science.* 2010. Vol. 44. No. 3. P. 367-382.
13. Wren, A. (eds.). *Computer Scheduling of Public Transport.* Papers based on presentation at the International Workshop held at the University of Leeds. 16-18 July, 1980. North Holland Publishing company. Amsterdam - New York – Oxford. 1981.

Acknowledgements

This work was supported by the research grant APVV-14-0658 ”Optimization of Urban and Regional Public Personal Transport”.

Received 16.09.2015; accepted in revised form 17.02.2017