A NEW SOLUTION MODEL FOR MULTI-DEPOT MULTI-VEHICLE-TYPE VEHICLE SCHEDULING IN (SUB)URBAN PUBLIC TRANSPORT

Natalia Kliewer, Taïeb Mellouli, Leena Suhl
Decision Support & Operation Research Lab, University of Paderborn
{kliewer, mellouli, suhl}@upb.de

1 THE MULTI-DEPOT MULTI-VEHICLE-TYPE SCHEDULING PROBLEM

The Vehicle Scheduling Problem (VSP) is the task of building an optimal set of rotations (vehicle schedule), such that each trip of a given timetable is covered by exactly one rotation. The restrictions and optimization criteria may differ from one problem setting to another. For *multi-depot, multi-vehicle-type VSP* for bus transit, each one-day rotation must end at the depot of start and is to be assigned to a type of vehicle feasible for all trips of the rotation.

Within a bus tour consisting of several (productive) service trips chained with each other, the use of deadhead trips (non-productive trips between two end stations) often provides an improvement in order to serve all trips of a given timetable by a minimum number of buses. However, since deadhead trips are an additional cost factor, minimization of this cost is an important optimization goal. The basic form of the bus-scheduling problem *(basic VSP)* involves only one depot; all daily tours of vehicles must start and end at this depot. The goal is to minimize fixed costs for needed buses and costs of proceeding to and from the depot as well as deadhead trips between two end stations. This problem can be represented as a network flow model, for which efficient solution methods exist.

The *multi-depot* VSP involves several depots, so that a vehicle has to return to the same depot in the evening from which it started in the morning. Considering several types of vehicles, (for example normal bus, minibus, and kneel bus), the subset of vehicle types that can be used to carry out a trip is given for each scheduled trip. Solving the resulting *multi-depot*, *multi-vehicle-type* VSP, each rotation of the computed vehicle schedule is assigned to exactly one depot and one vehicle type. Furthermore, capacity restrictions for depots arises when a depot has only a restricted number of (overnight) parking slots for buses. Both the multi-depot and the multi-vehicle-type requirements imply that the optimization problem becomes NP-hard.

1.1 State-of-the art modeling techniques

Traditionally, mathematical optimization models for public mass transit are based on a network where trips and depots are represented by nodes (trip- and depot-nodes, respectively) and possible connections between trips given by arcs (see, e.g., [Daduna and Paixão 1995]). The rotation of a vehicle is thus generated as a flow of one unit starting in the depot and returning back to it. The trip-nodes connected represent the scheduled trips carried out by the vehicle. All flow units involved in the single-depot problem are of the same type, so that the incoming and outgoing flow of each trip-node has to be equal to one. For multi-depot case, the network is multiplied, so that there is a network layer for each depot. Additional restrictions guarantee that each scheduled trip is carried out by exactly one vehicle belonging to one of the required vehicle-types. Because of these cover constraints involving flow variables of value 0 or 1, the resulting integer optimization model is difficult to solve.

In this modeling approach, the number of possible connections between scheduled trips, corresponding to the number of integer variables, grows quadratically in dependende of the number of scheduled trips. Therefore, models with several thousand scheduled trips cannot be solved directly with state-of-the-art optimization software. However, special modeling techniques, such as column generation or branch&price with Lagrangean relaxation, have been introduced in order to solve problems of practical size. (see, e.g., [Löbel 1998]).

2 A NEW AGGREGATED NETWORK FLOW MODEL

The number of potential connections between scheduled trips determines to a great extend the number of variables in the above model. Basically, there are three types of potential connections:

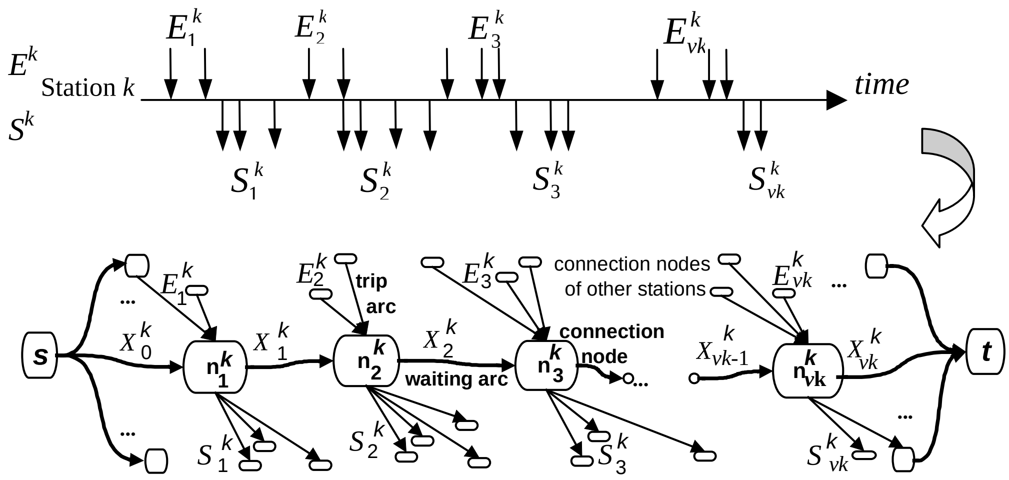
- Connections within one station
- Connections over a depot with pull in/pull out trips to/from the depot
- Connections through a directly connecting trip (deadhead)

We use a special modeling technique for each connection type, as described in the following:

2.1 Connections within one station

A connection within one station means the (standard) case if the successor trip starts at the same station where the predecessor trip ends. If we only permit this type of connection, we have the *simple vehicle scheduling problem* without deadheading and without depots. This corresponds to a basic vehicle scheduling problem in rail and air traffic (see [Mellouli 2001]).

Our approach is based on an aggregated time-space network being generated after pre-sorting the scheduled arrivals E^k and departures S^k at each station k by arrival and departure times respectively. (Arrival events are considered before departure events with same point in time). Introducing arcs for scheduled trips, connections at the same station are modeled by going through a connection node and eventually through introduced waiting arcs (along the resulting *connection line* of the station). Each vehicle corresponds to a unit flow



through the network. The flow over each scheduled trip arc must be equal to one.

Figure 1: Construction of an aggregated time-space network

Standing vehicles at same station and time are aggregated by flows on waiting arcs. The model restrictions conserve the balance of incoming and outgoing flow through each node, and thus of incoming, standing and outgoing vehicles at each station.

2.2 Connections over a depot with pull in/pull out trips to/from the depot

In analogy to stations (see Fig. 1), we build a connection line for each depot, although there may not be scheduled trips starting or ending directly in a depot. To each scheduled trip *i* we introduce, if necessary, arcs for potential pull-out and pull-in trips from/to each depot directly before/after carrying out *i* (with associated deadhead costs). Because it is more favorable for buses to stand at a depot than at other stations, we place a higher cost for waiting arcs outside depots, therefore avoiding long waiting times outside the depots. The capacity restrictions involved are modeled by flow bounds on appropriate arcs.

2.3 Connections through a directly connecting trip (deadhead)

Basically, each pair of compatible trips can be connected through deadheading. However, as the number of possible deadhead trips is prohibitively high, a direct modeling of all of them in large networks implies a problem size which cannot be handled by state-of-the-art solvers.

Thus, a crucial modeling technique is aggregation of possible matches - directly connecting trips between scheduled trips. Our aggregation is carried out in two steps, as explained below:

2.3.1 First Stage Aggregation

For each arriving scheduled trip *i* at station *k'* we determine the first trip compatible with *i* at each other station *k (k≠k', k* and *k'* are non-depot stations*)*. We call this trip *first-match(i,k)***.** We introduce only those deadheading arcs into the model which correspond to the first matches. Thus, the number of arcs is reduced significantly compared to the original situation (see Table 1).Nevertheless, all possible connections remain feasible. Each scheduled trip *j* compatible to a scheduled trip *i* can now be reached in the model network over *first-match(i, start-station(j)*) - possibly via waiting arcs in *start-station(j).*

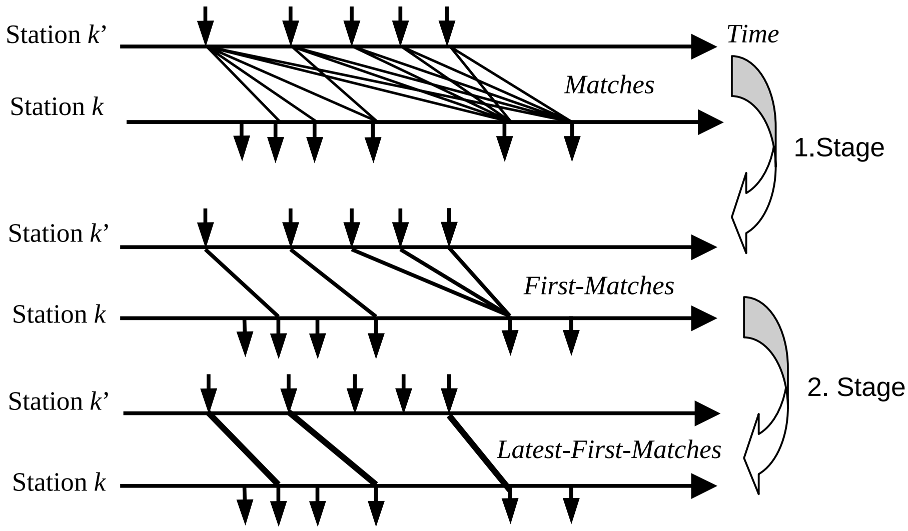


Figure 2: Two stages of model aggregation

2.3.2 Second Stage Aggregation

At the second stage, the number of arcs is further reduced. We aggregate the set of first-matches in a (smaller) set of latest-first-matches (see Fig. 2). The latest first matches can be determined in the following way: Let *S* be the set of incoming trips *i* at station *k'*, having the same first match j_S at station *k.* Let i_L be the latest incoming trip in S. Then the latest first match for each element in S is the *first-match*(i_L, *k'*). By removing all arcs corresponding to first matches but being no latest-first matches, we reduce the network significantly, but do not loose any possible connections. Every scheduled trip *j* compatible to scheduled trip *i* can now be reached over the corresponding latest first match to the start station of *j*, and possibly via waiting arcs in one or both connection lines.

Table 1 illustrates the impact of the two-stage aggregation process for timetables of three public transport enterprises. The resulting flow model contains one network layer for each pair of depot and vehicle type, where 0/1-variables on trip arcs and integral flow variables on latest first match arcs are defined. Latest first matches are computed separately for each layer. In addition to the flow balance equations it is required that for each scheduled trip, the sum of flow variables on the corresponding trip arcs of all network layers is equal to one (in order to guarantee that the trip is carried out by exactly one vehicle). The mathematical model thus basically consists of a minimum-cost-flow network optimization model for each network layer, which is of mixed-integer type because of the latter cover constraints.

	# Scheduled trips	# Bus stops	# Matches	# First matches	# Latest first matches
Timetable T1 (regional)	682	50	195,618	19,457	**6,292**
Timetable T2 (medium)	2,047	21	649,525	12,854	**4,657**
Timetable T3 (large)	3,776	58	506,776	17,641	**10,252**

Table 1: Number of first matches and latest first matches

The number of variables in the mathematical model mainly depends on the number of arcs for latest first matches, multiplied with the number of vehicle types. Even the number of first matches is limited by the number of bus stations multiplied by the number of plan trips. Since the number of stations is always smaller than the number of plan trips (see also Table 1), we notice a crucial model reduction compared to classical models (see Section 1.1).

It is an important characteristic of the aggregated network that any feasible flow, also an optimal flow, represents *a bundle* or *a class* of vehicle schedules. With the help of a suitable flow decomposition procedure, we can extract a vehicle schedule with an optimal flow and desired characteristics, e.g., first-in first-out dispatching strategy.

3 IMPLEMENTATION AND RESULTS

We designed and implemented a software component that supports public transport planners in constructing vehicle schedules. Table 2 summarizes the size of mathematical models, results, and optimization time for vehicle scheduling problems of two large German cities. The optimization library MOPS (see [Suhl 1994]) was used for the solution of the models.

Time-table	# Depots and # Vehicle types	Restrictions	Variables	Nonzeros	Optimization time (min.)	Total costs (10^3)
T2	4x3	26,637	116,546	192,914	9.19	7,260
T2	6x3	39,629	137,477	294,658	13.30	6,483
T2	8x3	53,355	186,809	399,890	28.40	6,444
T2	9x3	59,314	211,865	453,286	62.00	6,432
T3	1x1	12,253	39,442	58,154	0.24	29,624
T3	3x1	27,456	108,869	174,154	9.17	11,635

Table 2 Computational results for timetables T2 and T3

REFERENCES

Daduna, J.R. und Paixão, J.M.P. (1995). Vehicle Scheduling for Public Mass Transit – An Overview. In Daduna et al. (Eds.) *Computer-Aided Transit Scheduling*, LNEMS 430, pp. 76-90. Springer. Berlin.

Löbel, A. (1998). Optimal Vehicle Scheduling in Public Transit. PhD thesis. Shaker. Aachen.

Mellouli, T. (2001). A Network Flow Approach to Crew Scheduling based on an Analogy to a Train/Aircraft Maintenance Routing Problem. In Voß et al. (Eds.) *Computer-Aided Scheduling of Public Transport*, LNEMS 505, pp..91-120. Springer. Berlin.

Suhl, U. (1994). MOPS: A Mathematical OPtimization System. *European Journal on Operational Research 72*, pp. 312-322.