An Assignment Model for Scheduling Vehicles with Refueling

Viktor Árgilán

University of Szeged, Juhász Gyula Faculty of Education, Department of Applied Informatics, Boldogasszony sgt. 6, 6720 Szeged, Hungary
argilan.viktor.sandor@szte.hu

János Balogh

University of Szeged, Institute of Informatics, Department of Computational Optimization, Árpád tér 2, H-6720 Szeged, Hungary; baloghj@inf.u-szeged.hu

József Békési

University of Szeged, Institute of Informatics, Department of Computer Algorithms and Artificial Intelligence, Árpád tér 2, H-6720 Szeged, Hungary
bekesi@inf.u-szeged.hu

Balázs Dávid

InnoRenew CoE, Livade 6a, 6310 Izola, Slovenia, balazs.david@innorenew.eu
University of Primorska, FAMNIT, Glagoljaska ulica 8, 6000 Koper, Slovenia
balazs.david@famnit.upr.si

Gábor Galambos

University of Szeged, Juhász Gyula Faculty of Education, Department of Applied Informatics, Boldogasszony sgt. 6, 6720 Szeged, Hungary;
GalambosGabor@szte.hu

Miklós Krész

InnoRenew CoE, Livade 6a, 6310 Izola, Slovenia
University of Primorska, Andrej Marušič Institute, Muzejski trg 2, 6000 Koper, Slovenia

University of Szeged, Juhász Gyula Faculty of Education, Department of Applied Informatics, Boldogasszony sgt. 6, 6720 Szeged, Hungary
miklos.kresz@innorenew.eu

Attila Tóth

University of Szeged, Juhász Gyula Faculty of Education, Department of Applied Informatics, Boldogasszony sgt. 6, 6720 Szeged, Hungary
toth.attila.02@szte.hu

Abstract: The vehicle scheduling problem consists of scheduling a fleet of vehicles to cover a set of tasks at a minimum cost. The tasks are given in predetermined time intervals, and the vehicles are supplied by different depots. There are several known mathematical models that can be used to solve this problem, resulting in a valid vehicle schedule. One such approach is the multi-commodity network flow model, where the optimal schedule is computed by solving a linear integer programming problem. The main disadvantage of this model is that it can be intractable for practical scenarios that include additional vehicle constraints. These are specific restrictions that come from real-world applications, such as the refueling requirement of vehicles. When vehicles of different fuel types, including environmentally friendly ones, are considered, decisions about their refueling include many additional constraints that a valid assignment must meet. This paper presents how these vehicle-specific tasks can be included in the vehicle assignment phase. An IP-based heuristic solution is given for this specific variant of the vehicle assignment with multiple depots. Computational results on real-life and randomly generated test instances are presented where the vehicle assignment model uses an input schedule generated by the time-space network approach. The resulting integer programming problem for this assignment can be solved extremely quickly, even with a large number of variables. Computational results demonstrate that the model can effectively extend the capabilities of the standard models to be able to handle the assignment with vehicle-specific task requirements.

Keywords: vehicle scheduling; vehicle assignment; refueling constraints; fuel types; IP-based solution

1 Introduction

In the operational planning of public transport scheduling, the first phase is the vehicle scheduling phase (VSP). During this phase, the timetabled trips of the input are used to form several series of trips (called vehicle blocks) that can be executed continuously by the same vehicle. If all trips are covered with such blocks, a feasible vehicle schedule is created. To keep costs as low as possible, there may be a request to assign physical vehicles to these blocks [1]. The task of vehicle scheduling is to give a valid schedule of vehicles that perform all required tasks, which are mainly the timetabled trips. This should be done by satisfying all the arising requirements such as travel demands, vehicle properties, etc. which are given by transportation companies. Additionally, the aim is to find one of the best possible schedules according to the provided objectives.

In the case of most real-world VSPs, vehicles come from different depots (multiple depot case). Although some papers distinguish between physical depots and vehicle types, the term depot is defined in this paper to represent both physical and logical characteristics of vehicles (i.e., distinct vehicle types in multiple locations). Apart from the timetabled trips of the problem, vehicles can also cover deadhead trips between and terminal stations (departure and arrival locations of the timetabled trips) and pull-in or pull-out trips to and from depots. In the standard problem, one usually needs to minimize either the number of vehicles or the travel costs, or a linear combination of the two.

The classical definition of the VSP does not include tasks specific to the operation of the vehicles; it generally determines the depots from which the given schedule can be serviced, with the depot capacity bound on the number of vehicles in each depot. However, it does not assign a specific vehicle to each block, which does not correspond to real-world applications, where certain vehicle-specific requirements must be inserted into the schedule. From the perspective of this paper, the most important of these tasks is refueling, but others such as maintenance [2], or parking constraints between events [3] can also be considered.

The use of vehicles with alternative energy sources in public transportation has become more widespread recently and refueling decisions have become an important aspect of the schedule. Compressed natural gas (CNG), biodiesel and diesel hybrid systems have appeared, and their importance is constantly increasing [4]. To fulfill the requirements of environmental protection, public transportation companies generally replace old vehicles with new ones. Emission, which is related to environmental impact and environmental friendliness, is also an important aspect [5]. It is worth noting that long-term planning for a public transport company considers multiple cost factors such as amortization of vehicles (and decisions about their acquisition or replacement) or their ongoing maintenance over longer periods. However, this paper does not address these long-term fleet planning operations. Instead, a given fleet of vehicles is considered, and decisions are made strictly within the scope of a daily schedule.

From the perspective of daily operational management, an important aspect for fleet vehicle types is the distance they can cover without refueling (maximum operational distance) and the associated unit cost (fuel cost per km). Even if the required number of buses is available, planning their daily operations can be more complex due to the various specific parameters that have to be taken into account.

Based on the literature (e.g., [4], [6]), it is clear that alternative fuel types such as CNG, diesel and hybrid vehicles (or recently even hydrogen fleet [7]) are also worth considering from the point of view of daily task performance. While these new types of buses play an increasingly important role in the scheduling process, they must be handled separately in daily operative planning, including the vehicle assignment phase, since: (i) they cover a shorter distance with a full tank than those that use diesel; (ii) the number of pumps with biodiesel, etc. could also be strongly restricted at fuel stations; and (iii) refueling times may be longer. When these factors are taken into account, the refueling process plays an important role in the daily routine of a public transportation company. The literature on issues similar to the ones described above is briefly reviewed in the following section.

1.1 Overview of Related Papers

Before Li's paper [6], there were no constructive solutions to handle constraints discussed in the previous section. A "constructive solution" means explicitly defining time intervals for refueling processes in vehicle scheduling, after which the same vehicle continues with the execution of the corresponding tasks of the schedule. Earlier models focused primarily on handling so-called "route-length-constraints", which are maximum distance-constraints for vehicles. Baniheshami et al. [8] dealt with this type of constraint and introduced the VSP-RC (Vehicle Scheduling Problem with Route Constraints), considering routes with given maximum operational distance. (For details, see the excellent survey of VSP-methods [9]).

Li [6] was the first to introduce the explicit usage of the intervals of these events in the schedules. They dealt with the integrated vehicle scheduling and assignment phases and gave an exact approach for the single depot case. To our knowledge, this was the first paper that integrated the phases of VSP and vehicle assignment (VA), where VA includes refueling of vehicles with alternative fuel types such as electrical, CNG and diesel. Therefore, their paper will be reviewed in more detail below due to the relevance of its constraints [10, 11, 12] and the nature of their parameters considered. In cases where only battery-electric transit bus scheduling is considered, a good review can be found in [13, 14].

In [6], Li considered various types of fuel with specific distance parameters. These are given for electric buses (120/150 km), CNG (238 km), and diesel (256 km). The number of buses required to complete the trips was estimated using a constructive heuristic, and then the minimum number of buses was obtained by slightly increasing the distances that can be covered with one refueling. The constraints are as follows: there is one reloading station with a maximum capacity of 2 in the depot. Only the trips were real-world data everything else is estimated. Two separate models for electric buses and for fuel buses were described. The battery of the electric buses can be changed or recharged at the fuel station, starting at any time t. The application of this to petrol buses is not entirely clear, but the author claims that it is essentially a special case for the electric bus model. The maximum number of solved trips is 947, up to 500 the author gets "optimal or near-optimal" solutions with CPLEX; for more trips the model does not work so effectively.

On the basis of our own observations and practical experience, the above model indeed works similarly for different fuel types with the given parameters. Therefore, in this paper, two fuel types are considered. The main parameters of the problem are the maximum operational distance without refueling, the refueling time, and the number of fuel pumps available in parallel. Additional data is also required for travel times between some geographical locations and the driving time to and from the service station, which are the same for all vehicle types. The depot is the only service station considered.

In this paper, only the scheduling of daily operational vehicle-specific tasks is considered. The questions arising from the transport traffic point of view, e.g. with arbitrary departure times / data considered in [15], are not discussed. There are efficient solutions to manage daily or even longer-term driver (shift) assignments [16], or possibly both, i.e. considering vehicle and driver tasks together [17, 18, 19].

1.2 The Problem of Refueling

The problem defined in this paper considers the case of multiple-depot vehicle assignment, where an initial vehicle schedule is already given. This can be obtained by using any existing exact or heuristic approach to solve the multiple-depot vehicle scheduling problem. The task is to assign specific vehicles to the blocks of this schedule together with their vehicle-specific restrictions, such as valid refueling events for their matching fuel types. Valid refueling means that the vehicle has to travel to the refueling station before reaching its maximum operational distance, the fuel station must have enough pumps to service the vehicle. After which, the vehicle should be able to travel to the departure station of its next trip and perform it on time.

The original goal of this research was to solve a real-world scenario, which will be introduced and analyzed in Section 5 of this paper. In this scenario, some resources are restricted; some vehicles of the public transport company (around 30% of the total fleet) use CNG as fuel type.

The developed model is capable of handling vehicles with different fuel load levels. Moreover, the starting fuel levels of vehicles at the beginning of the day are given as input parameters and can be different between specific vehicles. This results in varying initial operational distances even for vehicles that share the same type. Vehicles must be refueled before reaching their maximum operating distance, but a refueling event can take up to 20 minutes in the case of CNG. These are capacities and refueling times, as well as the initial load levels for each vehicle, are all input parameters of the problem.

An additional strong constraint of our problem is the number of available refueling pumps. For example, in our real-world case study, the number of CNG fuel pumps is 2, which means that only 6 vehicles can be refueled in one hour under the 20-minute refueling time. Taking into account the maximum operational distance of vehicles and the usual bimodal pattern of the traffic load in public transit, these refueling requirements represent a strong bottleneck. An IP model is formulated that can handle vehicles with different fuel types. The efficiency of this model is presented on both real-world and randomly generated data.

The remainder of this paper is structured as follows. In the next section, some known techniques are presented to obtain an optimal vehicle schedule, focusing on the solution of the Multi Depot Vehicle Scheduling Problem (MDVSP). One such approach is the multicommodity network flow model, where the optimal schedule is computed by solving a linear integer programming problem. The connection-based and the time-space models – the two main approaches for modeling the MDVSP - are introduced and compared, emphasizing the advantages of the time-space model. In Section 3, a new model is presented for the vehicle assignment problem introduced above, which assigns both vehicles of various fuel types and their respective refueling activities to an existing daily vehicle schedule. Finally, the efficiency of this proposed model is discussed. A short case study is given on the computational experiments of this model at the Szeged local bus company. Additionally, results on randomly generated and benchmark test instances are also reported. The preliminary results of this paper were published in [20].

2 The Proposed Algorithm

Our algorithm frame consists of three steps.

• Step 1. Solve the VSP.

• Step 2. Solve the VA.

• Step3. If there is no feasible solution in Step 2, then a heuristic solution is applied for the problematic trips of Step 2. Collect the problematic trips of Step 2, and schedule them using new vehicles(s).

Some possible heuristic approaches for Step 3 can be found in [21, 22].

2.1 Solution of the VSP

Step 1 of the proposed method considers the solution of the VSP, which is one of the main daily scheduling tasks of transportation companies. The most commonly used model for vehicle scheduling is the so-called MDVSP model. It was first defined by Bodin et al. [23], and Bertossi et al. [24] proved that the problem is NP-hard (even for 2 depots). The definition of the problem in this section follows the terminology used by Löbel [25]. In this model there are given the set of *vehicles* and the set of *timetabled trips* (sometimes called *routes or journeys*) denoted by $V = \{v_1, v_2, \dots, v_{m^*}\}$ and $U = \{u_1, u_2, \dots, u_{n^*}\}$, respectively. Timetabled trips are those where vehicles carry passengers. For each $u_i \in U$ its *departure time* $dt(u_i)$, arrival time $at(u_i)$, *departure geolocation* $dg(u_i)$, and *arrival geolocation* $ag(u_i)$ are given. Two trips u_i and u_j are called compatible, if they can be scheduled by the same vehicle, i.e. $at(u_i) + \delta_{i,j} \leq dt(u_j)$, where $\delta_{i,j}$ is the traveling time between $ag(u_i)$ and $dg(u_j)$.

At the beginning of the scheduling period, vehicles park in depots, where they return after finishing their work. A vehicle may also travel to a depot during the day, for example, if it has no trips to execute for a longer period of time. (Depot can be a garage, parking place, etc.) Vehicles might have different types, for example, low-floor, articulated, etc. This gives a further classification of the set of vehicles. Combining these properties with the physical locations several subsets V_i ($i = 1, \dots, k$) of the vehicles can be derived. Usually, these classes are called *depots* in the literature.

In addition to the timetabled trips, another type of vehicle trips are distinguished, where no passengers are transported. These are called *deadhead* trips. A typical deadhead trip occurs if a vehicle moves from one station to another without performing a timetabled trip in order to increase the efficiency of its work.

Vehicle travel from or to a depot is also a type of deadhead trip. Depending on the destination, it is called either a *pull-out* or a *pull-in* trip. Usually, a timetabled trip cannot be scheduled for an arbitrary vehicle. For example, in practice, some trips can be executed only by low-floor buses. The user must specify such restrictions on the scheduling of the trips if it is needed.

A valid schedule consists of several vehicle blocks, where a block is defined as a series of trips, where every two consecutive trips are compatible. In practice, a block represents the daily work of a vehicle. A valid vehicle block always starts with a pull-out trip and ends with a pull-in trip. The block may contain additional deadhead trips between the scheduled routes. Depending on the number of depots, one can distinguish *the single-depot vehicle scheduling problem* (SDVSP) from the multiple-depot *vehicle scheduling problem* (MDVSP). For MDVSP, the basic problem is to give a schedule of the given vehicles in such a way that each trip is assigned to exactly one vehicle from its corresponding depot. Of course, not all vehicles have to be assigned in a daily schedule. Generally, the aim is to minimize the number of used vehicles, but one can also define other cost functions. In the latter case, the cost function represents the total cost of the schedule, which consists of the fixed and operational costs of the vehicles.

2.2 MDVSP Solution Techniques

Various mathematical models exist for the different types of vehicle scheduling problems. Nowadays, the most widely used techniques for the MDVSP are based on a multicommodity flow minimization model, which is solved as an IP problem. In the following, well-known flow minimization models are presented, namely the connection-based and time-space network ones. Other formulations exist as well, for example, a set covering formulation (see Hadjar et al. [26] or [9]).

The *connection-based multicommodity network flow model* has been intensively studied. The mainstreams of the research were to improve the solution methodologies for the IP problem. In addition to the exact algorithms (see Kokott and Löbel [27], Löbel [28]) some heuristic methods (see Löbel [25], [9]) were also investigated.

In the node set of a multicommodity network, each trip is represented by two nodes, according to the departure and arrival of the trip, and each depot is represented by two nodes as well, regarding the starting and ending of a schedule. The edge set consists of four types of edges. These are edges regarding trips (the node representing the departure of a trip is connected to the node representing its arrival), the deadhead edges between compatible timetabled trips, pull-out and pull-in edges from and to the depot (to the departure nodes and from the arrival nodes of trips, respectively), and the so-called depot circulation edges for each depot, from the node representing the pull-out to the node representing the pull-in, for the same depot.

Before describing the model, some notation is introduced for each of these sets mentioned in the previous list. Denote by D the set of depots and by $D_u \subseteq D$ the depot set of trip u. On the opposite side, $U_d \subseteq U$ denotes the set of those trips can be served from the depot d. The network is defined as follows. The set of nodes is as follows:

$$N = \{dt(u) \mid u \in U\} \cup \{at(u) \mid u \in U\} \cup \{dt(d) \mid d \in D\} \cup \{at(d) \mid d \in D\}, \quad (1)$$

where $dt(d)$ and $at(d)$ is the departure node and the arrival node for each $d \in D$, respectively. These nodes symbolize that a vehicle departs from and returns to the depot d.

To define the edges of the network, further definitions are needed. Let

$$F_d = \{(dt(u), at(u)) \mid u \in U_d\}, \ \forall d \in D \quad (2)$$

be the edges ordered to the set of timetabled trips corresponding to depot d, and let

$$B_d = \{(at(u), dt(u')) \mid \mathrm{u, u' \in U_d \text{ are compatible trips}}\}, \quad \forall d \in D \quad (3)$$

be the set of possible deadhead trips corresponding to depot d. Furthermore, let

$$R_d = \{(dt(d), dt(u)), (at(u), at(d)) \mid u \in U_d\}, \forall d \in D \quad (4)$$

be the set of pull-out and pull-in edges corresponding to the depot d. To give upper bounds on the number of vehicles in a depot, the so-called depot circulation edges must be defined.

$$K_d = \{(at(d), dt(d))\},\ \forall d \in D. \tag{5}$$

Now we are ready to define the set of edges E of the network.

$$E_d = F_d \cup B_d \cup R_d \cup K_d, \forall d \in D, \tag{6}$$

and

$$E = \cup_{d \in D} E_d. \tag{7}$$

The IP model for the MDVSP problem on the network (N, E) is as follows.

Describing the MDVSP problem as an integer programming model is a standard technique, in which it is expressed with the conditions that all trips must be completed, and the vehicles continuously perform their daily work by executing successive trips.

To do this, an integer vector on the edges of the network is defined. This vector can be considered a multi-commodity flow and will be denoted by x. To ensure the requirements of the problem, several constraints on x must be considered. If a component of a vector represents the edge e of depot d, then x_e^d will denote this component. The first type of constraint expresses that each trip should be assigned to exactly one vehicle.

$$\sum_{d \in D_u, e=(dt(u), at(u)) \in E_d} x_e^d = 1, \quad \forall u \in U. \tag{8}$$

The second type of constraints ensure that if a vehicle arrives at a station, then it should leave that.

$$\sum_{e \in n^+} x_e^d - \sum_{e \in n^-} x_e^d = 0, \quad \forall u \in U, \forall n \in N, \tag{9}$$

where for a given $n \in N$, the symbols n^+ and n^- denote the set of outgoing and incoming edges, respectively.

If there are limitations on the number of vehicles in the depots, then further constraints must be defined. If k_d is the number of buses belonging to depot d then the following conditions must be added to the model:

$$x_{at(d), dt(d)}^d \leq k_d, \forall d \in D. \tag{10}$$

Any vector x that satisfies these constraints is a feasible solution of the problem. Weights can be assigned to the edges, which represent the cost of the corresponding trip. If the aim is to minimize the number of vehicles, relatively large weights must be assigned to the pull-out edges. If c_e is the weight of edge e, then the objective function is

$$\min \sum_{e \in E} c_e x_e. \tag{11}$$

This way, based on this network, an integer programming (IP) problem is constructed, and this IP model has to be solved. Unfortunately, the number of deadhead edges can be huge in the network, and so the size of the IP will be also large even for some real problems given by transportation companies of a smaller city. To decrease the number of edges, the following model has been developed.

The time-space network model was introduced by Kliewer et al. [29] in 2006 in connection with public transportation problems. Although the model had been used before in air transport, this was its first application in the field of vehicle scheduling. The main advantage of the model is that it can solve the big problems that arise in practice. The model uses two dimensions, these are time and space. The word *space* here means geographical locations (stations), while time is represented by lines that belong to the stations. A timeline includes the departure and arrival times of the trips. Each station has a timeline, and every possible departure and arrival time defines a vertex in the network. (If more trips have the same departure or arrival time, then the corresponding vertices can be aggregated.) It is easy to see that the essential difference between the two models is that the time is connected to the locations here. Thus, the nodes of the network are given by the set of arrival and departure times of the stations. The edge set F_d is defined, i.e., the edges regarding the trips for each depot $d \in D$ like in the case of the connection-based model. Of course, timelines can be assigned to the depots as well, so the set R_d of pull-in and pull-out edges can be defined similarly. However, the main difference between the two models lies in the definition of deadhead trips. In the time-space network model, it is possible to accumulate the flow of some possible deadhead trips into one deadhead edge. Therefore, it is not necessary to represent every deadhead trip by a separate edge. The authors in [29] have used the so-called latest-first match strategy. This idea is a two-phase aggregation method:

• In the first phase, the arrival node of each trip is connected only to the first compatible trip node of the other stations. These edges are called first matches. In this phase, only these deadhead edges are considered.

• In the second phase, a further reduction in the number of deadhead edges is made. For each node, all the incoming deadhead edges are checked, and only the latest one from each timeline will be included in the network.

Therefore, the set B_d of deadhead trips will contain only a fragment of all possible overhead edges. However, in order to complete the model, the set W_d of waiting edges have to be introduced on each timeline. These edges will always follow the timeline, connecting the subsequent nodes. In this case, the set of edges belonging to $d \in D$ is:

$$E_d = F_d \cup B_d \cup R_d \cup K_d \cup W_d, \ \forall d \in D, \tag{12}$$

and the set of all edges in the graph is

$$E = \cup_{d \in D} E_d. \tag{13}$$

The corresponding IP model will be similar to that we have applied for the connection-based model.

3 New Vehicle Assignment Model

The main problem with the traditional vehicle scheduling models is that they simply classify the timetabled trips into several sets in an optimal way using the given cost function. However, in practice, transportation companies usually have additional requirements. They need to assign real vehicles to the set of trips, and these vehicles have some extra characteristics. For example, they should be refueled after some covered distance. Most of the existing MDVSP models do not handle these kinds of conditions. These models support only the composition of the trips in the vehicle schedules but are unable to take other restrictions into consideration. Even the length of the schedules cannot be easily restricted, and adding more complicated conditions to these models on the trip sets seems to be impossible. As far as we know, only a few papers deal with this problem and they are based mainly on heuristics, see for example [8] or [30]. In our application, two fuel types are considered; diesel fuel and CNG. Each vehicle has a specific fuel type, so there is no vehicle that can use both of them. This assumption comes from a real-world scenario, but, of course, the proposed model can also be extended to additional fuel types.

The following vehicle assignment model is developed for the problem.

The input of the model is as follows:

- the so-called theoretical vehicle schedule, which is the output of the MDVSP model (theoretical in that sense, that usually some vehicle blocks cannot be executed in practice: for example, there is no chance to run more than 500 kms with certain vehicles without refueling),
- the set of available vehicles,
- the refueling stations with fuel types, pump capacities and opening times,
- some other parameters that come from practice.

Notations used by the model:

n: the number of theoretical vehicle blocks in the schedule given by the MDVSP algorithm,

s_i: the i-th vehicle block ($1 \leq i \leq n$),

m: the number of available vehicles, it is assumed that $m \geq n$,

v_j: the j-th vehicle ($1 \leq j \leq m$),

mrd_j: the maximum operational distance without refueling of v_j $(1 \leq j \leq m)$,

rd_j: the required minimum operational distance without refueling of v_j $(1 \leq j \leq m)$,

crd_j: the current possible maximum operational distance without refueling of v_j $(1 \leq j \leq m)$,

p: the number of fuel stations,

f_k: the k-th fuel station $(1 \leq k \leq p)$,

t_d: the necessary time unit of refueling at a diesel fuel pump,

t_g: the necessary time unit of refueling at a natural gas pump,

$T_d(k)$: the number of possible time interval indices, when refueling can be done at a diesel fuel pump of f_k $(1 \leq k \leq p)$,

$T_g(k)$: the number of possible time interval indices, when refueling can be done at a natural gas pump of f_k $(1 \leq k \leq p)$,

d_{kt}: the number of diesel fuel pumps available for refueling in f_k in the time interval $[t, t + t_d]$, $(1 \leq k \leq p, 1 \leq t \leq T_d(k))$,

g_{kt}: the number of natural gas pumps available for refueling in f_k in the time interval $\left[t, t + t_g\right]$, $(1 \leq k \leq p, 1 \leq t \leq T_g(k)$,

x_{ijkt}: a binary variable expressing if a refueling of s_i with v_j is done at f_k in the time interval $[t, t + \nabla t]$, where $\nabla t = t_d$ or $\nabla t = t_g$, depending of the characteristics of vehicle j $(1 \leq i \leq n,\ 1 \leq j \leq m,\ 1 \leq k \leq p,\ 1 \leq t \leq T_d(i)$ or $1 \leq t \leq T_g(i))$,

c_{ijkt}: if $x_{ijkt} = 1$, then c_{ijkt} expresses the cost of the corresponding refueling.

In order to solve the problem by handling the refueling time periods, first time interval indices are generated. Based on the opening hours and t_d (or t_g), the time interval is divided into equidistant subintervals. The subintervals are generated for both types of fuel. Each subinterval has a length of t_d (or t_g). A special interval is also generated, which expresses the so-called after-work refueling. It is assumed that an arbitrary number of vehicles can be refueled in this artificial time period. The starting time of this interval is the maximum of the arrival time of the last trip and the end of the last normal time interval; the finishing time is the minimum of the starting time of the first trip and the start of the first interval. This specific interval is for refueling at night. Each interval is indexed from 1 to $T_d(i)$ and $T_g(i)$, and these intervals are denoted by their indices, and are called *time intervals.*

It can be easily seen that the possibility of refueling a vehicle with a given block in a specific time interval highly depends on the characteristics of the block. If the block does not have enough idle time in the given interval, then refueling cannot be done. This means that the corresponding variable x_{ijkt} can only be 0. To simplify the model, these variables can be omitted. To do this, the blocks are analyzed, their idle periods are calculated, and the corresponding variables are generated. By this analysis, the cost of the variables can be calculated as well. This cost is composed of the deadheads necessary to carry out the refueling and some other costs of the block belonging to the given vehicle. These costs may be different for certain types of vehicles.

Now, the exact variable generating and block cost calculation methods are presented. The variable x_{ijkt} exists for a given s_i, v_j, f_k and time interval t if and only if

1. The depot of v_j corresponds to the depot of s_i,
2. Time interval t corresponds to the fuel type of v_j,
3. s_i has an idle period in the time interval t,
4. The arrival time of the last trip before the beginning of interval t plus the deadhead trip time to the geographical location of f_k is smaller than or equal to the beginning of t,
5. The departure time of the first trip after the end of interval t minus the deadhead trip time to the geographical location of this trip from f_k is greater than or equal to the end of t,
6. t is such a time interval, which follows the arrival time of the last trip of s_i plus the deadhead trip time to the geographical location of f_k,
7. crd_j is greater than or equal to the operational distance of s_i until the beginning of the time interval t,
8. The operational distance of s_i after the refueling is at most crd_j,
9. rd_j smaller than or equal to $mrd_j - crd_j$ plus the operational distance of s_i until the beginning of the time interval t,
10. If t is an after-work time interval, then the operational distance of s_i is smaller than or equal to mrd_j.

Conditions 1-10 ensure that x_{ijkt} represents a realizable and reasonable assignment of block vehicles, taking into consideration the characteristics of the blocks and vehicles and the real-world refueling rules and requirements.

The cost c_{ijkt} of a given s_i, v_j, f_k and time interval t consists of the following components:

1. The operational cost of v_j, denoted by $oc(v_j)$.
2. The deadhead travel cost of v_j per km, denoted by $dc(v_j)$.
3. The useful travel cost of v_j per km, denoted by $uc(v_j)$.
4. The total distance of the trips of s_i, denoted by $td(s_i)$.

5. The total distance of the deadhead trips of s_i, denoted by $dd(s_i)$, including the deadhead trips to f_k.

6. An optional extra cost of refueling at f_k in the time interval t, if we want to make difference between the refueling in time, denoted by $ec(f_k, t)$.

Using the above parameters, the cost c_{ijkt} can be calculated by the formula

$$c_{ijkt} = oc(v_j) + dc(v_j) \cdot dd(s_i) + uc(v_j) \cdot td(s_i) + c(f_k, t). \tag{14}$$

Now we are ready to give the exact definition of the algorithm for generating variables x_{ijkt} and calculating c_{ijkt}.

Algorithm GenVar;

begin

for each $s \in S$ **do**

    for each $v \in V$ **do**

        if $Depot(s)$ corresponds to $Depot(v)$ **then**

            for each $f \in F$ **do**

            begin

                Let $u = FuelType(v)$

                for each $t := 1$ to $T_u(s)$ **do**

                if Conditions 1-10 hold for s, v, f, t. **then**

                    Store variable x_{ijkt} and calculate cost c_{ijkt}

            end;

end;

After generating the variables and the costs, an integer programming model can be introduced for solving the vehicle assignment problem with refueling.

$$\min \sum_{i,j,k,t} c_{ijkt} x_{ijkt}, \tag{15}$$

s.t.

$$\sum_{j,k,t} x_{ijkt} = 1, \text{ for all } i \ (1 \leq i \leq n), \tag{16}$$

$$\sum_{i,k,t} x_{ijkt} \leq 1, \text{ for all } j \ (1 \leq j \leq m), \tag{17}$$

$$\sum_{i,j} x_{ijkt} \leq d_{kt}, \text{ for all } k, t \ (1 \leq k \leq p, 1 \leq t \leq T_d(k)), \tag{18}$$

$$\sum_{i,j} x_{ijkt} \leq g_{kt}, \text{ for all } k, t \ (1 \leq k \leq p, 1 \leq t \leq T_g(k)). \tag{19}$$

In the above model, constraint (16) ensures that each block will be assigned to exactly one vehicle. Constraint (17) says that each vehicle will be assigned to at most one block. Constraints (18) and (19) mean that in a given time interval the number of refueled blocks/vehicles cannot be greater than the capacity of the fuel station. These constraints are generated for both types of fuels. From the above-presented information it can be concluded that the suggested procedure solves the problem.

In the next sections the computational results are given in detail to see the efficiency of the method in practice.

4 Methodology and Environment

Our implemented method was tested on both real-world and randomly generated data. Real-world data was used from the Szeged local bus company, while random input was generated using a modification of the state-of-the-art method given in [31]. This modified algorithm produces a multiple depot and multiple vehicle type problem and generates similar instances to those of Hungarian transportation companies. A description of the method can be found in [32]. The above algorithm did not consider different fuel types but was extended with the generation of fuel types for the vehicles and the corresponding refueling stations. To obtain the optimal theoretical schedules, the time-space network model -- which was introduced in Section 2 -- has been applied. Therefore, the following modules are implemented:

- input-output module for reading and writing data,
- a time-space network-based MDVSP optimizer,
- a vehicle assignment module, based on our algorithm.

The database of the largest real-life timetable contains: 2700 trips, 4 vehicle types, and one physical depot with 120 vehicles. This problem produces an IP model with more than 18000 rows and 180000 columns. For solving the models, the MIP solver SYMPHONY 5.1.7 is included in the modules. The model was tested for the following sets of instances:

a) a database of real-world problems that came from the daily routine of the transportation company. A detailed part of this database can be found in [33],
b) a collection of randomly generated instances with 1000, 500 and 250 trips respectively, based on the extension of the generation method found in [32],
c) the set of 200 and 400 random trip instances used by Huisman et al. in [34], for which different fuel types and refueling stations are also generated.

5 Computational Results

Table 1 contains the most important characteristics and their computational results of the selected problems of b) and c). All test cases considered 2 different fuel types. The instances of Huisman et al. used 2 depots, while our randomly generated instances used 4.

Table 2 contains results for real-life instances. For illustrative purposes, four instances are selected from the database mentioned in a), which include two different types of weekdays, a Saturday and a Sunday. All instances use 4 depots and 2 fuel types.

Table 1
Random instance problem sizes, running times of the scheduling (RT-S) and assignment (RT-A) phases in seconds, and the number of buses generated by the model (MB).

Problem	Trips	Depots	RT-S	RT-A	MB
$Random1$	1000	4	77	43	150
$Random2$	500	4	5	1	77
$Random3$	250	4	1	1	51
$Huisman1$	200	2	1	1	24
$Huisman2$	200	2	1	1	20
$Huisman3$	400	2	1	2	40
$Huisman4$	400	2	1	3	38

Table 2
Real-life problem sizes, running times of the scheduling (RT-S) and assignment (RT-A) phases in seconds, the number of buses used by the company (CB) and the ones generated by the model (MB)

Problem	Trips	Depots	RT-S	RT-A	CB	MB
$Szeged1$	2724	4	1179	14	107	96
$Szeged2$	2690	4	872	8	107	96
$Szeged3$	1981	4	431	5	65	54
$Szeged4$	1768	4	250	1	54	44

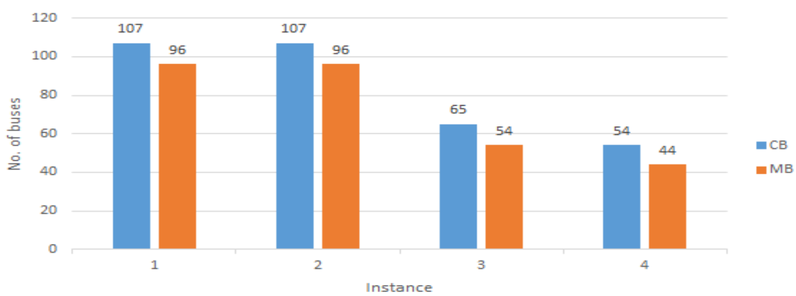

Figure 1
Comparison of the bus numbers used by the company and the model (CB and MB, respectively), on instances 1 to 4 (Szeged-1,2,3,4, respectively)

The tables above show that the assignment model is very effective. The IP problem can be solved extremely quickly, even in the case of a large number of variables. The results of the algorithms on real-world instances are promising: 10% less buses compared to the current number of buses used by the company (see Figure 1). It can be seen that the model is capable to successfully and efficiently handle the vehicle-specific requirements that are required for vehicles with different fuel types, e.g. refueling, including their journey to the depot or refueling point, the refueling time and the range of vehicles per refueling. Overall, this shows that the model can effectively extend the capabilities of existing MDVSP models to handle the assignment of real vehicles with fuel consumption restrictions, which is an important constraint for a practical problem.

Regarding the scalability of the model considered, we could only verify the applicability for our existing real-world instances. Szeged is a typical medium-sized city, and as discussed in [17], for a larger city the service area is usually divided into districts, which would mean several smaller sub-instances to solve independently. For random inputs, the literature standard has been used to generate these examples. The computational results showed that above a critical level, the not very realistic examples caused problems. A method for this was designed in [32], but even with this, it was not possible to generate larger examples in a sophisticated way.

Application. The proposed method was successfully integrated into a system that solves vehicle and driver scheduling problems as two phases of operational planning in public transportation. This system was discussed in [33]. The vehicle assignment phase of this system is based on the method presented in this paper. Another application of the method, which provides a common driver-friendly approach to vehicle and driver scheduling, is given in [35].

Conclusions

In this paper, a new vehicle assignment model combined with refueling is presented. To our knowledge, the proposed model is novel in the literature. The problem originates from a real-world application and uses the results of the classic MDVSP method. The algorithm and the model are tested on both randomly generated instances and real-world data supplied by a transportation company. A new software module was also developed, which can be used in practice. Unfortunately, if the vehicle resources of the company do not fit in the structure of the output coming from the MDVSP solver, the model may lead to an infeasible problem. This can happen, for example, if the MDVSP schedules do not have enough free time for refueling. In this case, the original schedules should be modified. It is an open question whether there are such models for vehicle scheduling that are able to handle schedule restrictions better. The computational result shows that our model is effective in practice. A further question may be to prove theoretically why one gets a good performance of the IP model.

Acknowledgement

This work of J. Balogh was supported by the grant TKP2021-NVA-09 of the Ministry for Innovation and Technology. The research was supported by the BioLOG project: Balázs Dávid and Miklós Krész are grateful for the support of National Center of Science (NCN) through grant DEC-2020/39/I/HS4/03533, the Slovenian Research and Innovation Agency (ARIS) through grant N1-0223 and the Austrian Science Fund (FWF) through grant I 5443-N. They have been also supported by the research program CogniCom (0013103) at the University of Primorska. Balázs Dávid is also grateful for the support of the Slovenian Research and Innovation Agency (ARIS) through grant J1-50000. The computational tests of this paper were supported by the former Szeged City Bus Company (Tisza Volán, Urban Transport Division). They now operate under the new name Volánbusz.

References

[1] A. Nagy and J. Tick, Modeling of bus transport operative planning tasks. In: IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI 2020), 89-94

[2] R. Adonyi, I. Heckl, and F. Olti, Scheduling of bus maintenance by the P-graph methodology. Optimization and Engineering, 14, 565-574, 2013

[3] B. Dávid and M. Krész, Multi-depot bus schedule assignment with parking and maintenance constraints for intercity transportation over a planning period. Transportation Letters, 12(1), 66-75, 2020

[4] J-Q. Li and K.L. Head, Sustainability provisions in the bus-scheduling problem. Transportation Research Part D, 14, 50-60, 2009

[5] M. Barany, B. Bertok, Z. Kovacs, F. Friedler, and L.T. Fan, Solving vehicle assignment problems by process-network synthesis to minimize cost and environmental impact of transportation. Clean Technologies and Environmental Policy, 13, 637-642, 2011

[6] J-Q. Li, Transit bus scheduling with limited energy. Transportation Science, 48(4), 521-539, 2013

[7] A. Golla, F. vom Scheidt, N. Röhrig, P. Staudt, and C. Weinhardt, Vehicle Scheduling and refueling of Hydrogen Buses with On-site Electrolysis. INFORMATIK 2020. DOI: 10.18420/inf2020_70. Gesellschaft für Informatik, Bonn. ISBN: 978-3-88579-701-2, pp. 795-806, 2021

[8] M. Banihashemi and A. Haghani, Optimization Model for Large-Scale Bus Transit Scheduling Problems. Transportation Research Record, 1733(1), 23-30, 2000. DOI: https://doi.org/10.3141/1733-04

[9] S. Bunte and N. Kliewer, An overview on vehicle scheduling models. Journal of Public Transport, 1(4), 299-317, 2009

[10] J. D. Adler and P.B. Mirchandani, The vehicle scheduling problem for fleets with alternative-fuel vehicles. Transportation Science, 51(2), 441-456, 2016

[11] L. Lu, H.K. Lo, and F. Xiao, Mixed bus fleet scheduling under range and refueling constraints. Transportation Research Part C: Emerging Technologies, 104, 443-462, 2019

[12] H. Li, A framework for optimizing public transport bus fleet conversion to alternative fuels. PhD Thesis, Georgia Institute of Technology, 2019

[13] J-Q. Li, Battery-electric transit bus developments and operations: A review. International Journal of Sustainable Transportation, 10(3), 157-169, 2016

[14] J. Pasha, B. Li, Z. Elmi, A.M. Fathollahi-Fard, Y. Lau, A. Roshani, T. Kawasaki, and M. A. Dulebenets, Electric vehicle scheduling: State of the art, critical challenges, and future research opportunities, Journal of Industrial Information Integration, 38, 100561, 2024

[15] Zs. Ercsey, A. Nagy, J. Tick, and Z. Kovács, Bus Transport Process Networks with Arbitrary Launching Times. Acta Polytechnica Hungarica, 18(4), 125-141, 2021

[16] Zs. Ercsey and Z. Kovács, Multicommodity network flow model of a human resource allocation problem considering time periods. Central European Journal of Operations Research, 32(4), 1041-1059, 2024

[17] J. Békési, B. Dávid, and M. Krész, Integrated Vehicle Scheduling and Vehicle Assignment. Acta Cybernetica, 23(3), 783-800, 2018

[18] J. Békési and A. Nagy, Combined Vehicle and Driver Scheduling with Fuel Consumption and Parking Constraints: a Case Study. Acta Polytechnica Hungarica, 17(7), 45-65, 2020

[19] M. Horváth and T. Kis, Computing strong lower and upper bounds for the integrated multiple-depot vehicle and crew scheduling problem with branch-and-price. Central European Journal of Operations Research, 27(1), 39-67, 2019

[20] J. Balogh, J. Békési, G. Galambos, and M. Krész, Model and Algorithm for a Vehicle Scheduling Problem with Refueling. Proceedings of the 9th Workshop on Models and Algorithms for Planning and Scheduling Problems, Abbey Rolduc, The Netherlands, June 29-July 3, 2009, 229-231

[21] B. Dávid and J. Balogh, An Algorithmic Framework for Real-Time Rescheduling in Public Bus Transportation. In: MATCOS-13 Proceedings of the 2013 Mini-Conference on Applied Theoretical Computer Science, pp. 29-33, University of Primorska Press, Koper, 2016. ISBN 978-961-6984-20-1 URL: http://www.hippocampus.si/ISBN/978-961-6984-20-1.pdf

[22] B. Dávid and M. Krész, The dynamic vehicle rescheduling problem. Central European Journal of Operations Research, 25(4), 809-830, 2017

[23] L. Bodin, B. Golden, A. Assad, and M. Ball, Routing and Scheduling of Vehicles and Crews: The State of the Art. Computers and Operations Research, 10, 63-211, 1983

[24] A. A. Bertossi, P. Carraresi, and G. Gallo, On Some Matching Problems Arising in Vehicle Scheduling Models. Networks, 17, 271-281, 1987

[25] A. Löbel, Optimal Vehicle Scheduling in Public Transit. Ph.D. thesis, Technische Universitaet at Berlin, 1997

[26] A. Hadjar, O. Marcotte, and F. Soumis, A Branch-and-Cut Algorithm for the Multiple Depot Vehicle Scheduling Problem. Tech. Rept. G-2001-25. Les Cahiers du Gerad, Montreal, 2001

[27] A. Kokott and A. Löbel, Lagrangian Relaxations and Subgradient Methods for Multiple-Depot Vehicle Scheduling Problems. ZIB-Report 96-22, Konrad-Zuse-Zentrum für Informationstchnik, Berlin, Germany, 1996

[28] A. Löbel, Vehicle Scheduling in Public Transit and Lagrangian Pricing. Management Science, 44, 1637-1649, 1998

[29] N. Kliewer, T. Mellouli, and L. Suhl, A Time-Space Network Based Exact Optimization Model for Multi-Depot Bus Scheduling. European Journal of Operational Research, 175, 1616-1627, 2006

[30] H. Wang and J. Shen, Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints. Applied Mathematics and Computation, 190, 1237-1249, 2007

[31] G. Carpaneto, M. Dell'Amico, M. Fischetti, and P. Toth, A branch and bound algorithm for the multiple depot vehicle scheduling problem. Networks, 19, 531-548, 1989

[32] B. Dávid, Application-oriented scheduling problems in public bus transportation, PhD Thesis, University of Szeged, 2018. URL: https://doktori.bibl.u-szeged.hu/id/eprint/9745

[33] J. Békési, A. Brodnik, M. Krész, M., and D. Pas, An Integrated Framework for Bus Logistics Management: Case Studies. In: S. Voss, J. Pahl and S. Schwarze (eds.), Logistik Management: Systeme, Methoden, Integration, Springer, 389-411, 2009

[34] D. Huisman, R. Freling, and A. P. M. Wagelmans, Multiple-depot integrated vehicle and crew scheduling. Transportation Science, 39, 491-502, 2005

[35] V. Árgilán, J. Balogh, J. Békési, B. Dávid, M. Krész, and A. Tóth, Driver scheduling based on driver-friendly vehicle schedules. Proceedings of OR 2011, International Conference on Operations Research, Springer-Verlag, 323-328, 2011