

Combined Vehicle and Driver Scheduling with Fuel Consumption and Parking Constraints: a Case Study

József Békési

University of Szeged, Juhász Gyula Faculty of Education, Department of Applied Informatics, Boldogasszony u. 6, 6720 Szeged, Hungary
bekesi@jgypk.szte.hu

Albert Nagy

Óbuda University, Applied Informatics, Bécsi út 96/b, 1034 Budapest, Hungary
albert.nagy@me.com

Abstract: Efficient operation is an important question for public transport companies, and that can most easily be achieved by reducing their operational costs. This can also be facilitated by the optimized scheduling of vehicles and the work of the drivers. Such an optimization task can be very complex. Due to the dramatically increased processing capabilities today, it can be performed using advanced optimization methods. Automation aims to reduce the time-consuming manual activities, thus increase efficiency and provide prompt opportunities of scenario planning for operational cost analysis purposes. In this paper, a case study is presented to solve the combined vehicle and driver scheduling problem. The applied mathematical model is discussed and the calculation results for practical examples are presented

Keywords: optimization; mathematical model; vehicle, crew and driver scheduling problem; public transport company

1 Introduction

Operational costs represent a large part of the expenses of public transport service companies. Their most important parts are vehicle fleet costs, fueling, and maintenance costs. In addition, to driver salaries. Consequently, the budget can be improved significantly by decreasing these costs. The most commonly used technique to reduce these costs is the usage of a powerful, computer-aided information system. Due to the ICT (Information and Communication

Technology) development today almost every public transportation company has its own information system. In addition to the business applications, such as accounting, these systems may also contain modules such as

- scheduling vehicles and drivers for the company-served lines,
- monitoring the work of the vehicle fleet during the day,
- notifying the dispatcher of the unusual events (malfunctions, delays, etc.),
- tracking the status of the vehicles in the fleet,

and similar other functions. The ICT environment outlined above is often the organized backbone of efficient logistics management (see, for example, [1], but there is a next step in the process, which is to find the parts of the organization and operation that can be reduced in terms of operational costs. *Strategic planning* is used to determine the route of the buses. Elements of strategic planning are described by Desaulniers and Hickman in their review paper [2]. It discusses network planning, route planning, and passenger assignment based on expected travel needs. *Tactical planning* is used when the goal is to create an optimal schedule, and the planning can include scheduling the frequency of the trips. In addition, the fulfillment of requirements such as the capacity of the lines or the types of vehicles (e.g. on which line the low-floor vehicles should run) may also be addressed here. In the case of *operational planning*, buses and drivers are scheduled to provide the service. A number of solutions have been investigated to solve operational planning issues. Examining the problems from a theoretical point of view, most of them are NP-complete, which makes it difficult to find exact or near-exact solutions to problems that occur in practice. Even in cities with hundreds of thousands of citizens, the task is complex, if legislation, individual needs, and employee interests are taken into consideration. Such restrictive conditions may include vehicle characteristics, requirements relating to working time, driving time, and breaks for drivers, but in some cases, the constraints of the stations must also be considered. One of the major directions in the development of decision support systems over the past decades has been the development of software packages that provide a comprehensive solution to the various optimization tasks. However, practice shows that there are a number of company-specific expectations and constraints due to the specific situation of businesses on the application side, that are important to transport companies and can not be handled in a uniform way by the general systems developed. In this article, a decision support system developed for the Budapest Transport Corporation is presented. The purpose of the research was to integrate a module for solving a vehicle and driver scheduling task into the company's information system. The method is based on an existing mathematical model, supplemented by special conditions required by the company. The article is structured as follows: In Section 2 a literature review on mathematical models is given for the driver and vehicle scheduling problem, then the problem and the solution method is introduced. In Section 3 the mathematical model is presented, then in Section 4.

the most important computational results are summarized. Finally, in Section 5 concluding remarks are given.

2 Materials and Methods

2.1 Literature Review

The scheduling problems of public transport are very complex. When looking at the problem from an operational research perspective, a global optimum is expected that minimizes the cost of both vehicle-related tasks and driver scheduling. A comprehensive survey of routing and scheduling problems of vehicles as well as crews is provided in [3]. It includes classification and categorization of routing and scheduling problems, a review of algorithmic techniques and solution methodologies. The solutions for the problems are, however mainly theoretical. Effective algorithms exist only for some of the tasks while for others the algorithmic capabilities described remain at low-level. Especially in case of bus public transport companies when the vehicle scheduling problem covers a given set of timetabled trips with consideration of practical requirements. The proposed modeling approaches in [3] are unable to solve real-world problem instances with thousands of scheduled trips by direct application of standard optimization software. The time-space network model is often used to reduce the number of variables in the exact optimization model. This model is discussed in [4] that uses a time-space-based network flow model instead of connection-based network model. It involves multiple depots for vehicles and different vehicle types for bus scheduling problem. This approach leads to size reduction of the corresponding mathematical models compared to connection-based network flow. The model size has been substantially reduced through the aggregation of incoming and outgoing arcs. The optimal solution could not be resulted by any of the above exact approaches. A combination of the methods should be able to solve the large amount of problems of practical interest in acceptable running times. Integer linear programming approach using combinatorial optimization can be seen in [5]. In the most widely used models today, the vehicle scheduling problem is formulated as an integer multicommodity network flow problem. In this model, optimal scheduling can be calculated as a solution to an integer linear programming problem. The methods apply branch and cut and branch and cut and price respectively. The column generation techniques seem indispensable for both approaches. Column generation is developed to make it possible to solve the huge linear programs with up to million integer variables. These rules for selecting new columns are based on Lagrangean relaxations and therefore called Lagrangean pricing. Other models also exist to solve this problem. The problem can also be formulated as a set partition problem with side constraints, whose continuous relaxation can be solved by column generation. (see

for example [6]). A relationship is established between the bounds obtained by the assignment relaxation, the shortest path relaxation, the additive technique, Lagrangian decomposition, and column generation. It is shown that the additive bound technique cannot provide tighter bounds than those obtained by Lagrangian decomposition and not better than the linear programming bound. As in [7] the introduction of variable fixing, cutting planes, and the mixed branch-and-bound algorithm and the best-then-depth strategy leads to substantial improvements in the performance of a column generation algorithm to solve the scheduling problem. Dávid and Krész proposed a heuristic method [8]. The disadvantage of the models discussed and used in the literature in a specific, practical situation is that it only takes into account the rules relating to timetabled and overhead trips, bus types, and capacities required for them. However, it is not possible to include specific conditions that come from a real application environment. When planning the operational tasks of public transport, such typical vehicle-specific conditions are fuel consumption rules, various maintenance requirements (weekly, monthly, etc.), and parking rules. Parking rules may apply to both daytime and nighttime parking: where to park, what capacity the parking places have, etc. The length of the parking period may influence where in which geographical location it may be performed. The literature discusses several different versions of the vehicle scheduling problem, usually by the number of device types and the number of depots. The simplest version is the Single Depot Vehicle Scheduling Problem (SDVSP), where the vehicles belong to a single-vehicle type and are located in the same physical location. The first solution to the SDVSP problem was published by Saha [9]. The most commonly used model for solving a vehicle scheduling problem is the so-called Multiple Depot Vehicle Scheduling Problem (MDVSP). This case, which is more general than the single depot one, reflects the fact that in real life, different scheduled trips (their vehicles) may have different special needs. The vehicles are divided into different depots based on different vehicle types and the location of the vehicles. The MDVSP was defined by Bodin *et al.* and Bertossi *et al.* They showed that it is an NP-hard problem [11]. Tasks for vehicle-specific activities should also be taken into account, which requires a general framework for the integrated vehicle scheduling and assignment. A set partitioning-based mathematical model, where most vehicle-specific activities can be integrated based on the desired constraints is presented in [10]. This model is then solved using a column generation approach. The solution time can be reduced by the parallelization of the column generation process. If the transport company also uses alternative fuel vehicles in its fleet, the scheduling of these should take into account the number of kilometers per refueling, which may be much less than the mileage of conventional fuel vehicles. Alternative-fuel vehicles are getting more popular, and research is being done on how current infrastructure can serve them. The problem of vehicle scheduling consists of assigning a fleet of vehicles to service a given set of trips with start and end times. [12] presents the alternative-fuel multiple depot vehicle scheduling problem, a modification of the standard multiple depot vehicle scheduling problem where there is a given set of

fueling stations, and a fuel capacity for the vehicles. The problem is formulated as a binary integer program, and exact column generation algorithm and a heuristic algorithm to solve the problem. [13] proposes a model for electric transit buses with either battery swapping or fast charging at a battery station, and a vehicle-scheduling model with the maximum route distance constraint for compressed natural gas, diesel, or hybrid-diesel buses. Both of these scheduling models are NP-hard. An important topic is to study the location problem of battery service stations. In [14] the developments of battery-electric buses are reviewed. A qualitative analysis on the strengths and weaknesses of each range method is conducted as well as costs and emissions of transit buses powered by different sources. Buses using alternative energy sources to reduce emissions, including some toxic air pollutants and carbon dioxide are studied in [15]. Life cycle comparison between buses fueled by different kinds of alternative energy is discussed in [16] to serve as an input to cost-benefit analysis.

Recently some other papers discussed the practical issues of vehicle scheduling. Dávid and Krész studied the handling possibilities of parking and maintenance constraints [17], and rescheduling possibilities in case of disruptions [18]. Parking and maintenance activities are handled in [19] as well.

Another important problem discussed in the literature is the Crew Scheduling Problem (CSP), also known as Driver Scheduling Problem or Duty Scheduling Problem. There are many CSP solution methods and applications in the literature. One of the best known is the so-called Generate and Select technology. The method can be summarized as follows: in the first step, generate a large number of regular shifts, and then in the selection step, look for a subset of them that is optimal in cost and covers the trips. Phase one requires significant computation time. The amount of calculation depends greatly on the number of trips and the complexity of the rules. In addition, the computational complexity of the rules greatly influences the complexity of this phase, and thus the whole problem. The problem can be defined as a set covering or set partitioning task. The partitioning model is a constrained version of the covering problem where no overlap is possible. This corresponds exactly to the real problem, but in this case, the existence of a feasible solution is not guaranteed. Note that both tasks have been shown to be NP-hard [23]. There are many ways to solve this problem. A hybrid approach incorporating a genetic algorithm is presented in [20]. It derives a small selection of good shifts to seed a greedy schedule construction heuristic. A group of shifts called a relief chain and used by the genetic algorithm for schedule construction. [21] simulates the self-adjusting process for driver scheduling. It incorporates the idea of fuzzy evaluation into a self-adjusting process, combining the features of iterative improvement and constructive perturbation, to explore solution space effectively and obtain superior schedules. A flexible system for scheduling driver applying integer programming methods are used in [22] and heuristic solution techniques are used in [24, 25].

According to the conventional approach, driver scheduling is performed after the phase of vehicle scheduling. Therefore, this is also called a sequential method. However, if the vehicle schedules are too dense, for example, there is not enough time to change drivers, then the problem can be infeasible in the driver scheduling phase. For this reason, the simultaneous optimization of the vehicle and driver scheduling may be reasonable. This problem is called Vehicle and Crew Scheduling Problem (VCSP). In 1999 Haase and Friberg published the first algorithm providing the exact solution for the single-depot case [26]. In their model, an integrated mathematical formulation was given in such a way that both sub-tasks were defined as set partition problems. The vehicle scheduling part is based on the model given by Ribeiro and Soumis [27], while the driver scheduling part uses the ideas of Desrochers and Soumis [28]. In the multi-depot case, first Gaffi *et al.* [29] discussed the integrated problem, using a heuristic method. In 2005, Huisman *et al.* [30] successfully extended the former models and algorithms of the single-depot case to the multi-depot version. This was the first general mathematical formulation of the multi-depot problem. Huisman in [31] also discusses the corresponding Lagrangian relaxations and Lagrangian heuristics. To solve the Lagrangian relaxations, column generation is applied to set partitioning type models. Haase [32] presents an exact approach for solving the simultaneous vehicle and crew scheduling problem in urban mass transit systems. This approach relies on a set partitioning formulation for the driver scheduling problem that incorporates side constraints for the bus itineraries. The proposed solution approach consists of a column generation process integrated into a branch-and-bound scheme. In 2008, Mesquita and Paías [33] also provided two mathematical formulations for this problem. In 2019 Horváth and Kis [33] proposed a novel mathematical programming formulation that combined ideas from known models and presented a solution methodology based on branch-and-price. In 2010 Steinzen *et al.* [34] gave another fully integrated VCSP approach, where the underlying vehicle scheduling model was based on the time-space network technique.

2.2 Problem Definition and Requirements

The problem is to automatically calculate optimal or approximately optimal vehicle and driver schedules for a given list of trips based on the master data and the company specific requirements and parameters in compliance with labor regulations. The optimality is measured by a given objective function and the aim of the optimization is to increase economy and efficiency. The developed model should take into consideration the following characteristics of the schedule planning for city buses:

- the problems are given by packages,
- the trips of a package can belong to a single line or a group of lines,
- the trips of a package can overlap 2 days,

- different day types are possible (e.g. working weekday, feast day, school day, etc.),
- deadhead trips from or to the depots can be possible from each end station,
- the maximum number of depots in a package is 5,
- the maximum number of vehicles that can be used in the solution of a package is 30,
- the maximum number of vehicle types in a package is 4,
- the type of vehicle can determine the fuel consumption, which should be taken into consideration in the schedule,
- breaks with a standoff or driver change are possible,
- standoff or parking is possible on more end stations or depots,
- parking place capacities are given for end stations in 5 minute intervals,
- driver change can be possible on given end stations,
- the labor regulations can be defined by several parameters.

The packages contain the following information:

- the lines and their parameters,
- the end stations, the depots, and their parameters,
- the trips,
- the parameters of the labor regulations and break rules.

The solution of the problem given by a package must satisfy the following requirements:

- each trip should be assigned to exactly one vehicle and driver schedule,
- the solution cannot use more vehicles than the number of available vehicles given in the package,
- the vehicle and driver schedules should be designed properly, adding the necessary activities (e.g. deadhead trips, breaks, maintenance, etc.)
- only such route can be set up for the vehicle (including deadhead trips to the depot) that can still be completed in terms of fuel consumption,
- between two passenger trips at least the required technological and compensatory time given for the line must be completed,
- drivers have rest periods while on duty, breaks with a standoff, vehicle change or both can be used based on the given parameters,
- driver change is possible if it is allowed for the given line on the given end station,
- in the case of a standoff, the parking space capacity should be checked,

- there is a possibility for divided working time, when the daily work of the driver is divided into 2 separated parts with a longer break between them,
- the generated driver schedules must comply with all work and rest regulations,
- the objective of the optimization process is to minimize the total net working time of the drivers, but if possible, the average of the drivers' net working times should be between 7 and 9 hours.

There are many conditions that are determined by the applicable legislation and the rules and regulations applicable to the employees of a given transport company. These include maximum working times, sufficient breaks after a given driving time, mandatory rest periods between two work periods, etc. In addition to scheduled and deadhead trips, there is a variety of technical and administrative tasks with well-defined timescales. These include passengers get on and get off times at end stations, vehicle pick-up, stopping, parking and various technological times, etc. In the following there is an overview of the most important working rules that are applicable to the drivers employed by the Budapest Transport Corporation. The default time values can be given as parameters.

- 1) The minimal length of a daily schedule (the minimal working time) is 4 hours without breaks.
- 2) The maximal length of a daily schedule (the maximal working time) is 10:30 hours with breaks.
- 3) The driving time cannot be longer than 9 hours in a schedule.
- 4) Vehicle oriented working times:
 - a) depot release time: 25 minutes,
 - b) end station release time: 20 minutes,
 - c) release time of the vehicle stopped by the same driver: 10 minutes,
 - d) vehicle change: 25 minutes,
 - e) driver change on a vehicle: 5 minutes,
 - f) stopping time at an end station: 5 minutes,
 - g) stopping time at a depot: 20 minutes.
- 5) Rules for divided work (the driver schedule is divided into two periods, usually morning and afternoon periods).
 - a) The break between the two parts cannot be shorter than 2 hours or longer than 6 hours.
 - b) The total length of the two parts cannot be longer than 10:30 hours.
 - c) The total length of the two parts with the break cannot be longer than 14 hours.
 - d) The working time of each part cannot be shorter than 2 hours.

6) Break rules

- a) No break is necessary for a schedule shorter than 6 hours.
- b) 30 minutes break is necessary for a schedule longer than 6 hours and shorter than 9 hours.
- c) For a schedule longer than 9 hours at least 40 minutes break is necessary, which can be divided into 2 parts (20+20 minutes).
- d) A break must not be given in the first or last hour of a schedule.
- e) The continuous working time cannot be longer than 6 hours in a schedule.
- f) A break is not part of the working time.

3 The Mathematical Model

A combined vehicle and driver scheduling optimization model is used to solve the problem described in the previous chapter. Several versions of this are known in the literature. The mathematical formalization is a modified version of the model described in the paper published by Huisman et al. [30].

The most important components of the model are the set of timetabled trips and the available vehicles. Timetabled trips are all trips where vehicles carry passengers. Each such trip is determined by the departure and arrival times, the departure and arrival stations, and the distance between them. The vehicles are divided into (physical) depots based on their location. This may be the garage, parking space or site where the vehicle is parked. Vehicles can also have various important features that allow us to group them further. Based on physical (geographical) locations and features, vehicles are classified into disjoint subsets. The subsets thus formed are called depots. In addition to timetabled trips, vehicles must also carry out other types of drivings. These are called overhead trips. For example, for the first trip of the day the vehicle must leave the night parking lot and return after the last trip of the day. Such deadhead trips can occur during the day, e.g. for longer breaks. Typical additional overhead trips are when a vehicle goes to another station after completing a trip to make another trip from there. It also needs to allow these in order to get an effective schedule. For each timetabled trip, the user can specify the depots from which the trip can be served. In practice, this might mean, for example, that certain lines may be served by given type of buses or from given locations. These requirements may be determined by the location of the station and the characteristics of the traffic. It can define certain relationships between timetabled trips. Two trips are said to be compatible if after finishing the first trip the vehicle is able to arrive at the place of departure of the second trip in time. If the first trip arrives at the destination location of the second trip, the only condition is that the first trip arrives earlier as the second trip. If the

arrival station of the first trip is not the same as the departure station of the second trip, the overhead time between the stations must be taken into account. There may be rules that include mandatory technological time between two trips, which should be taken into account when examining compatibility. The network is described by a directed graph. The nodes represent the trips, to which the departure and arrival depot nodes are added. Because multiple depots problem is handled, more depot nodes are possible. This corresponds to the usual technique. Two nodes of the graph are connected by a directed arc if the trips representing them are compatible. The arc length always represents the net working time of the driver, which corresponds to the objective function. It connects the depot nodes to the appropriate trip nodes, which can be accessed from that depot. Huisman *et al.* classifies the arcs into two main groups, namely short and long ones. Short arcs always represent the shorter events when the driver remains with the vehicle, while the long arcs represent the events when the driver stops the vehicle in the parking space and it remains unattended. Such a graph is built, which better represents the real situation, see Figure 1.

The following node types are introduced:

- source_depot: nodes representing the source depots,
- sink_depot: nodes representing the sink depots,
- trip: nodes representing timetabled trips.

There are 11 types of arcs in the model:

- start_of_schedule: arc representing the sign-on event with the first deadhead trip to the location of the first timetabled trip of a schedule,
- end_of_schedule: arc representing the last deadhead trip of a schedule to the location of the depot with the sign-off event,
- short_wait: arc representing a short wait after completing a timetabled trip, when the driver remains with the vehicle,
- short_break_endstation: arc representing a break of the driver spent at an end station after completing a timetabled trip,
- short_break_depot: arc representing a break of the driver spent in a depot after completing a timetabled trip,
- short_driverchange_endstation: arc representing a driver change event at an end station after completing a timetabled trip,
- short_driverchange_depot: arc representing a driver change event in a depot after completing a timetabled trip,
- long_stop_endstation: arc representing a driver change with a long parking of the vehicle at an end station without attendance,
- long_stop_depot: arc representing a driver change with a long parking of the vehicle in a depot without attendance,

- `long_dividedstop_endstation`: arc representing a long parking of the vehicle at the end station while the driver works in a divided schedule,
- `long_dividedstop_depot`: arc representing a long parking of the vehicle in the depot while the driver works in a divided schedule.

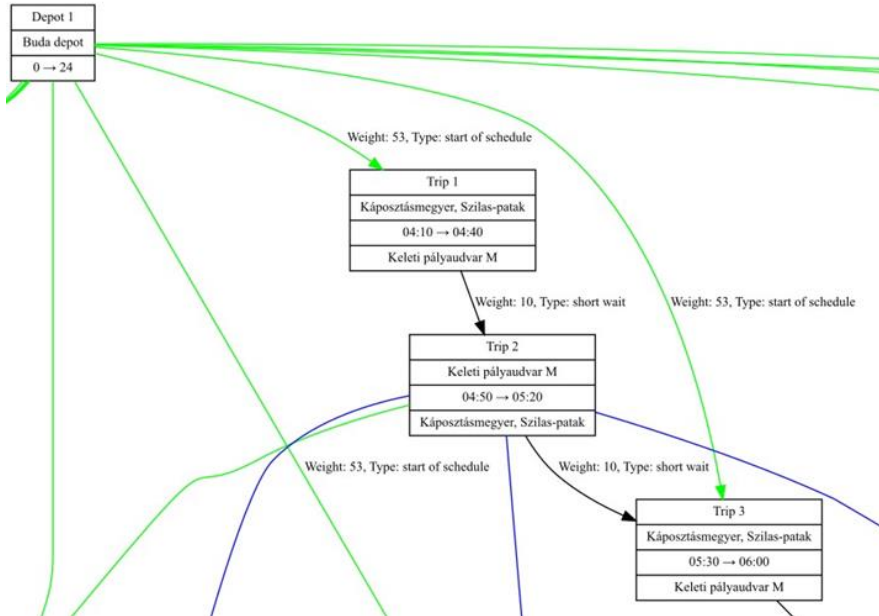


Figure 1
Part of the network

3.1 The Main Steps of the Calculation Process

- 1) In the first phase of the calculation, the appropriate input data and parameters are read and the graph is created. Here, the technological and compensatory times are taken into account, as well as certain rules regarding the duration of the waits or breaks. These can also be controlled by parameters. This means, for example, that if two trips are too close in time, those will not be connected, which means that those will not be executed one after the other by the same vehicle or driver. The first step in generating a graph is to create nodes based on the timetabled trips, location information, and vehicle types received. Then, add the above-listed arcs, also taking into account the parameters and labor rules. There may be different types of parallel arcs that have different weight values. For example, the driver may take a break at the end station or in a depot, depending on parking possibilities and station capabilities. However, these mean different working times, so if a model is to reflect the real situation fairly well, it needs to include both options. This

- increases the number of arcs, which itself would not necessarily be a problem in the size of the optimization model, but practical experience has shown that the number of possible driver schedules can be critically large in some cases and this must be handled in some way.
- 2) In Phase 2, if possible, all regular driver schedules are generated. There is already a check that takes into account all the rules specified in the specification. The generating process is done systematically by traversing the base graph with depth-first search. However, every vertex is checked on the fly, and if the subschedule already formed does not conform to any rule, then this branch is cut off. When a complete schedule is done during the generation, a final check runs, which only accepts the schedule if it is found regular. The generated driver schedules are stored. If too many schedules are generated, the process is stopped and a heuristic method is executed, to decrease the size of the problem. This method will be described later. Parameters can also be used to limit the number of schedules.
 - 3) In Phase 3, the mathematical model is constructed, which is essentially an extended version of the already mentioned VCSP model described in detail in [18]. The model includes constraints on fuel consumption, parking places and manages different vehicle types and locations.

3.2 The Formal Description of the Model

The set of timetabled trips are denoted by $U = \{u_1, u_2, \dots, u_n\}$. Let D be the set of depots, and $D_u \subseteq D$ the depot set of trip u : this includes those depots, from which u can be served. Note that in this case, a depot can represent a combination of physical locations and vehicle types. Denote $U_d \subseteq U$ the set of those trips that can be served from depot d . For every $d \in D$ two nodes are defined $dt(d)$ and $at(d)$ representing that a vehicle starts at depot d and goes back there. The set of N is then defined as follows

$$N = \{u \in U\} \cup \{dt(d) | d \in D\} \cup \{at(d) | d \in D\}.$$

To give the arcs of a network, the following notations are introduced.

$$B_d = \{(u, u') | u, u' \in U_d \text{ are compatible trips}\}, \forall d \in D.$$

Other deadhead trips corresponding to depot d are the first and the last trips of the vehicle from the depot or to the depot.

$$R_d = \{(dt(d), u), (u, at(d)) | u \in U_d\}, \forall d \in D.$$

This way the set of arcs can be defined belonging to depot d in the network:

$$A_d = B_d \cup R_d, \forall d \in D,$$

and the set of all arcs of the graph is

$$A = \bigcup_{d \in D} A_d.$$

After these preparations, it is ready to define the VCSP problem on the network $G = (N, A)$. For this it is necessary to define an integer vector x , which can be considered as a multicommodity flow. The dimension of the vector is equal to the number of arcs in the network. If the arc e belongs to the depot d ($e \in A_d$), then the component of the vector corresponding to the arc $e \in E$ is denoted by x_e^d . The value of x_e^d will be 1 if the given arc is included in the schedule, otherwise, it will be 0. The first condition ensures that every timetabled trip is scheduled exactly once.

$$\sum_{d \in D, e \in u_d^+} x_e^d = 1, \quad \forall u \in U. \quad (1)$$

where u_d^+ denotes the set of those outgoing arcs of the vertex u which belong to A_d . It ensures that all vehicles return to a depot by the end of the scheduling period. In other words, if a vehicle belonging to a particular depot arrives at an end station, it must leave it.

$$\sum_{e \in u_d^+} x_e^d - \sum_{e \in u_d^-} x_e^d = 0, \quad \forall u \in U, \forall d \in D, \quad (2)$$

where u_d^- denotes the set of those incoming arcs of the vertex u which belong to A_d . When defining fuel conditions, it considers the distance in km that can be covered by a single refueling. This requires to sum up the distance traveled by the vehicles and then limit it by the values given in the parameters. First, assign new variables to each vertex of N , excluding the set $\{at(d) | d \in D\}$. It is denoted this vector by t . The component of t belonging to vertex v will be denoted by t_v . The inequality that calculates the running distance up to a given node can be given as follows:

$$t_{v'} \geq x_e^d (t_v + \delta_e), \quad \forall v \in N \setminus \{at(d) | d \in D\}, \forall e \in v^+,$$

where δ_e is the distance represented by arc e , v' is its tail vertex and v^+ is the set of all outgoing arcs of vertex v . Note that these constraints are nonlinear, but those can be easily linearized using the following form:

$$t_{v'} \geq t_v + \delta_e - (1 - x_e^d)L, \quad \forall v \in N \setminus \{at(d) | d \in D\}, \forall e \in v^+, \quad (3)$$

where L is a constant larger than the longest possible distance in the system. Further constraints are necessary to check if a running distance of a vehicle remains under its maximal possible value.

$$t_{v'} + \delta_e + x_e^d L \leq r_d + L \quad \forall v \in \{at(d) | d \in D\}, \forall e \in v^-, \quad (4)$$

where L is a constant larger than the longest possible distance in the system, δ_e is the distance represented by arc e , v' is its head vertex, r_d is the maximal possible running distance allowed for the vehicles in depot d and v^- is the set of all incoming arcs of vertex v . The next constraints ensure that the capacity of the depots is not violated.

$$\sum_{e \in at(d)^-} x_e^d \leq k_d, \quad \forall d \in D, \quad (5)$$

where k_d is the number of vehicles available in depot d . In the following, it presents the conditions for the driver schedules. The set of valid driver schedules generated in Phase 2 for depot d is denoted by S_d . It assigns a vector of variables y to the schedules, where the component $y_s^d = 1$ if $s \in S_d$ is included in the solution and $y_s^d = 0$ otherwise. $S_d(u) \subseteq S_d$ denotes the set of those driver schedules which contain the vertex $u \in U_d$. Similarly $S_d(e) \subseteq S_d$ denote the set of those driver schedules that contain the arc $e \in A_d$. Similarly to the vehicle part, the first equality ensures that each trip is included in exactly one driver schedule.

$$\sum_{s \in S_d(u)} y_s^d - \sum_{e \in u_d^+} x_e^d = 0, \quad \forall u \in U_d, \forall d \in D. \quad (6)$$

During the daily work, certain events require a driver to attend the vehicle, while others do not require this. This can be modeled in such a way that certain types of arcs should be covered by both vehicle and driver schedules, while others are covered only by vehicle schedule.

It is denoted by A_q^d the set of arcs of type short_wait, short_break_endstation, short_break_depot, long_dividedstop_endstation, and long_dividedstop_depot.

The arcs of A_q^d should be covered by both vehicle and driver schedules. This can be expressed by the following equality:

$$\sum_{s \in S_d(e)} y_s^d - x_e^d = 0, \quad \forall e \in A_q^d, \forall d \in D. \quad (7)$$

Note that if short breaks with vehicle changes are allowed, then these conditions can be changed to $0 \leq \sum_{s \in S_d(e)} y_s^d - x_e^d \leq 1$ for short break type arcs e . Similar links can be given between the deadhead trips from or to the depots and the driver schedules containing them.

$$0 \leq \sum_{s \in S_d(e)} y_s^d - x_e^d \leq 1, \quad \forall e \in dt(d)^+, \forall d \in D, \quad (8)$$

$$0 \leq \sum_{s \in S_d(e)} y_s^d - x_e^d \leq 1, \quad \forall e \in at(d)^-, \forall d \in D. \quad (9)$$

Let $T_1, \dots, T_m$ be the possible time slots for which parking capacities should be checked. Furthermore, denote P the set of those stations and parking locations that can be used by the vehicles and $A_l^d(T)$ the set of those arcs of A_d that covers time slot T at location l . Denote the parking capacity of location l in time slot T for vehicles of depot d by $c_l^d(T)$. The parking constraints look like this

$$\sum_{e \in A_l^d(T)} x_e^d \leq c_l^d(T), \quad T = T_1, \dots, T_m, \forall l \in P, \forall d \in D. \quad (10)$$

The constraints for the average working time can be given as follows:

$$\sum_{d \in D} \sum_{s \in S_d} (w_s^d - w_{max}) y_s^d \leq 0, \quad (11)$$

$$\sum_{d \in D} \sum_{s \in S_d} (w_s^d - w_{min}) y_s^d \geq 0, \quad (12)$$

where w_s^d is the net working time of schedule $s \in S_d$ and w_{max} and w_{min} are the given maximal and minimal average net working times resp.

Furthermore, assume the followings:

$$x_e^d, y_s^d \in \{0,1\}, t_i \geq 0, \quad \forall e \in A_d, \forall s \in S_d, \forall d \in D, \forall i \in N \setminus \{at(d) | d \in D\}. \quad (13)$$

After these preparations, it is are ready to give the mathematical programming formulation of the problem.

$$\begin{aligned} & \text{minimize} && \sum_{d \in D} \sum_{s \in S_d} w_s^d y_s^d \\ & \text{subject to} && (1) \dots (13) \end{aligned}$$

As mentioned in the description of Phase 2 the main difficulty of the model is that in some cases the number of possible driver schedules can be extremely large. Usually, the theoretical models in the literature use column generation to find the exact optimum. However the company's main aim was not getting an optimal solution, rather a feasible solution that can be calculated in a relatively short time and which is good enough to be used in practice. This means that it should satisfy all the constraints and it should not be far from the optimum. The experiments showed that in the case of practical instances it is not always easy to find a feasible initial solution by a simple heuristic. A trip contraction procedure is applied to handle this situation. This method decreases the number of vertices and arcs of the graph, so the number of possible driver schedules is also decreased. The following greedy trip grouper algorithm is developed, see Figure 2. The number of trips that will be collected in a group is given by a parameter n . The procedure will create a new set containing the trip groups.

Algorithm 1 Greedy Trip Grouper

```

1: procedure groupTrips( $n$ :Integer,  $S$  : Set of Trips)
2:   for all  $T \in S$  do
3:     Next( $T$ )  $\leftarrow$  The closest compatible trip to  $T$ 
4:   for all  $T \in S$  do
5:     if Merged( $T$ ) = false then
6:       actTrip  $\leftarrow T$ 
7:       MList  $\leftarrow \emptyset$ 
8:       for  $i \leftarrow 1, n$  do
9:         Merged(actTrip)  $\leftarrow$  true
10:        MList  $\leftarrow$  MList  $\cup$  actTrip
11:        if Next(actTrip) = null then
12:          actTrip  $\leftarrow$  Next(actTrip)
13:        else
14:          Exit for
15:      Add MList to the output

```

Figure 2
Greedy Trip Grouper

4 Discussion

Several real problems were solved by the method. There was parts of the tasks that arose during the daily work of the company. The problems were selected by experts to represent the various cases that can happen in practice. The mathematical models were generated by the system and these were solved by an optimization solver. The following tables present the most important characteristics of the inputs and the results of the computations. Table 1 shows the most important characteristics of the problems, the number of trips, vehicles, vehicle types, and depots. Table 2 presents information about the models, the sizes of the graphs, and IP models and the number of valid driver schedules. Finally, Table 3 gives the details of the solutions. In some cases, the trip grouper heuristic was used to get a solution. The number of trips in a group is given in the table. There were two possibilities to stop the optimization process. If optimal solution had been received, then the solver finished normally. If the solution was not improving for a longer duration, then the solution process was stopped. The running times are also displayed in the table. These change on a large scale, from a couple of seconds to hours depending on the problems.

Based on the results, this method can automate manual vehicle and driver scheduling in large part. The size of the problem highly depends on the characteristics of the input, such as the number of trips, vehicle types and depots, and the average length of trips. In some cases; the number of valid driver schedules was too large to solve the original problem. The trip grouper heuristic was able to handle this situation in most of the cases. The running times are very diverse, but 14 of the 20 problems were solved in 30 minutes and 5 of them in 1 minute. These times include all phases, i.e. the graph and the driver schedule generation and the solution of the mathematical program. 12 problems were solved to optimality, without using the heuristic method, or optimization process stopping.

4.1 Computational Results

Table 1
Properties of the problems solved

Number	Number of trips	Number of vehicles	Vehicle types	Depots
1	162	10	1	1
2	193	10	2	2
3	87	6	2	1
4	444	11	1	1
5	95	3	1	1
6	201	7	2	2
7	329	21	5	2

8	134	6	2	2
9	229	6	2	1
10	116	5	1	1
11	134	6	2	2
12	167	9	5	1
13	196	15	2	4
14	98	7	1	2
15	245	9	1	1
16	811	20	2	2
17	123	5	1	1
18	811	20	1	1
19	162	10	1	1
20	149	7	1	1

Table 2
Properties of the models (* = Trip grouper is used)

Problem	Graph			IP Model	
	Vertices	Arcs	Driver schedules	Columns	Rows
1	164	3698	6448	10394	5720
2	108*	11023	127432	138947	27977
3	91	4164	46608	51038	9595
4	157*	4427	1053086	1057822	10600
5	99	1604	46947	48701	3978
6	209	22850	510926	534540	57263
7	349	10459	85869	97909	26709
8	142	5808	11495	17712	14249
9	233	8330	502702	511724	20834
10	118	781	91756	92720	1997
11	142	5808	11495	17712	14249
12	177	4309	57875	62791	10771
13	212	22612	19337	42940	48478
14	102	9836	119306	129443	24910
15	247	4374	160352	165221	9488
16	216*	12734	264298	278084	29791
17	133	9032	279244	288527	21386
18	216*	12499	369327	382253	31219
19	166	3763	27973	32065	9869
20	153	6440	91561	98304	16010

Table 3
Properties of the solutions

Problem	Solution status	Running time (sec)	Trip grouper
1	Optimal	8	No
2	Optimal	998	Yes (2)
3	Optimal	138	No
4	Optimal	11364	Yes (3)
5	Optimal	543	No
6	Optimal	5199	No
7	Optimal	49	No
8	Optimal	134	No
9	Optimal	4467	No
10	Optimal	42	No
11	Optimal	27	No
12	Optimal	35	No
13	Optimal	679	No
14	Optimal	1212	No
15	Stopped	6432	No
16	Optimal	17570	Yes (4)
17	Stopped	1506	No
18	Stopped	25170	Yes (4)
19	Stopped	1512	No
20	Stopped	1531	No

Conclusion

In this paper, the use of a combined vehicle and driver scheduling model is studied for practical problems. First, a literature review on mathematical models for the vehicle and driver scheduling problems are briefly reviewed, then the scheduling problems and the solution methodology are discussed. A real problem is presented in a case study. The mathematical model is described and the most important calculation results are summarized. Based on the experience during the case study, these kinds of methods can help the planning process of transport companies with the existing constraints. Research on advanced scheduling models for public transport management systems is proven to be a relevant area of further examinations. The expected results hold out a promise to improve the operative planning activities of public transport.

Acknowledgement

József Békési was supported by the EU-funded Hungarian grant EFOP-3.6.2-16-2017-00015 and by the scientific grant of Juhász Gyula Faculty of Education, University of Szeged.

References

- [1] M. Meilton, Selecting and implementing a computer aided scheduling system for a large bus company, Algorithms: Combinatorial Analysis. In Computer-Aided Scheduling of Public Transport, (eds. S. Voss and J.R. Daduna), 203-214, Springer-Verlag, Berlin, 2001
- [2] G. Desaulniers and M. D. Hickman, Public Transit, In Handbook in OR & MS, (eds. C. Barnhart and G. Laporte), Vol. 14, Chapter 2, Elsevier B. V., 2007
- [3] 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
- [4] 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
- [5] A. Löbel, Optimal Vehicle Scheduling in Public Transit, PhD. thesis, Technische Universitaet at Berlin, 1997
- [6] C. C. Ribeiro and F. Soumis, A Column Generation Approach to the Multiple-Depot Vehicle Scheduling Problem, Operations Research, 42(1), 41-52, 1994
- [7] 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
- [8] B. Dávid and M. Krész, Application Oriented Variable Fixing Methods for the Multiple Depot Vehicle Scheduling Problem, Acta Cybernetica, 21(1), 53-73, 2013
- [9] J. L. Saha, An algorithm for bus scheduling problems, Operational Research Quarterly, 21(4), 463-474, 1972
- [10] J. Békési, B. Dávid, M. Krész, Integrated Vehicle Scheduling and Vehicle Assignment, Acta Cybernetica, 23(3), 783800, 2018
- [11] A. A. Bertossi, P. Carraresi, and G. Gallo, On Some Matching Problems Arising in Vehicle Scheduling Models, Networks, 17, 271-281, 1987
- [12] J. D. Adler and P. B. Mirchandani, The vehicle scheduling problem for fleets with alternative-fuel vehicles, Transportation Science, 51(2), 441-456, 2016
- [13] J. Li, Transit bus scheduling with limited energy, Transportation Science, 48(4), 521-539, 2013
- [14] J. Li, Battery-electric transit bus developments and operations: A review, International Journal of Sustainable Transportation, 10(3), 157-169, 2016

- [15] J-Q. Li and K. L. Head, Sustainability provisions in the bus-scheduling problem, *Transportation Research, Part D*, 49, 50-60, 2009
- [16] A. Rabl, Environmental benefits of natural gas for buses, *Transportation Research, Part D*, 7, 391-405, 2002
- [17] 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
- [18] B. Dávid and M. Krész, The dynamic vehicle rescheduling problem, *Central European Journal of Operations Research*, 25(4), 809-830, 2017
- [19] A. Haghani and Y. Shafahi, Bus maintenance systems and maintenance scheduling: model formulations and solutions. *Transportation Research Part A: Policy and Practice*, 36(5), 453-482, 2002
- [20] R. S. K. Kwan, A. S. K. Kwan and A. S. K. Wren, Evolutionary Driver Scheduling with Relief Chains, *Evolutionary Computation*, 9, 445-460, 2001
- [21] J. Li. A Self-Adjusting Algorithm for Driver Scheduling, *Journal of Heuristics*, 11, 351-367, 2005
- [22] A. Wren, S. Fores, A. S. K. Kwan, R. S. K. Kwan, M. E. Parker and L. Proll, A flexible system for scheduling drivers, *Journal of Scheduling*, 6(5), 437-455, 2003
- [23] M. R. Garey and D. S. Johnson, *Computers and Interactability: A Guide to the Theory of NP-Completeness*, Freeman, San Fransisco, 1979
- [24] A. Tóth and M. Krész, A flexible framework for driver scheduling, In *Proceedings of the 11th International Symposium on Operational Research, Slovenia, SOR'11*, 341-346, 2011
- [25] A. Tóth and M. Krész, An efficient solution approach for real-world scheduling problems in urban bus transportation, *Central European Journal of Operations Research*, 21(1), 75-94, 2013
- [26] K. Haase and C. Friberg, An exact branch and cut algorithm for the vehicle and crew scheduling problem, In *Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, 471, (ed. N.H.M. Wilson) 63-80, Springer, Berlin, 1999
- [27] 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
- [28] M. Desrochers and F. Soumis, A column generation approach to the urban transit crew scheduling problem, *Transportation Science*, 23(1), 1-13, 1989

- [29] A. Gaffi and M. Nonato, An integrated approach to the extra-urban crew and vehicle scheduling problem, In *Computer-Aided Transit Scheduling*, (ed. N. H. M. Wilson), *Lecture Notes in Economics and Mathematical Systems*, 471, 103-128, Springer, Berlin, 1999
- [30] D. Huisman, R. Freling and A. P. M. Wagelmans, Multiple-depot integrated vehicle and crew scheduling, *Transportation Science*, 39, 491-502, 2005
- [31] R. Freling, D. Huisman, and A. P. M. Wagelmans, Models and algorithms for integration of vehicle and crew scheduling. *Journal of Scheduling*, 6, 63-85, 2003
- [32] K. Haase, G. Desaulniers, and J. Desrosiers, Simultaneous vehicle and crew scheduling in urban mass transit systems, *Transportation Science*, 35(3), 286-303, 2001
- [33] M. Mesquita and A. Paias, Set partitioning/covering-based approaches for the integrated vehicle and crew scheduling problem, *Computers and Operations Research*, 35(5), 1562-1575, 2008
- [34] I. Steinzen, V. Gintner, L. Suhl and N. Kliewer, A Time-Space Network Approach for the Integrated Vehicle- and Crew-Scheduling Problem with Multiple Depots, *Transportation Science*, 4(3), 367-382, 2010