

UNIVERSITY OF SZEGED
DOCTORAL SCHOOL OF COMPUTER SCIENCE

**APPLICATION-ORIENTED
SCHEDULING PROBLEMS IN
PUBLIC BUS TRANSPORTATION**

Summary of the PhD Dissertation
by

Balázs Dávid

Supervisor:
Dr. Miklós Krész



Szeged, 2018

1 Introduction

Public transportation networks are huge, complex systems, and companies that operate these face a large number of difficult optimization problems every day. According to a study by Rodrigue et al. [13], customer demands in public transportation are changing, and as a result, companies have to re-evaluate transport services they provide, and rethink the structure of their transportation networks. Meanwhile, the cost of building and operating public transportation systems increases, making it harder for them to provide a relevant alternative to urban mobility, while also remaining profitable businesses.

A large portion of the expenses comes from the operating costs of the company: these mainly include standing and running costs connected to the vehicles, and salaries of the drivers they employ. Problems connected to these areas belong to the stage of *operational planning* in the optimization processes of a public transit system. Operational planning includes three interconnected subproblems: *vehicle scheduling*, *driver scheduling* and *driver rostering*. Vehicle scheduling gives a feasible daily schedule for a set of daily timetabled trips, considering the fleet of a transportation company. Driver scheduling creates the daily shifts for the employees, satisfying all regulations connected to the daily working time of a driver. These shifts are then used by

driver rostering to produce employee rosters over a longer planning period. A classical approach for an optimization framework in public transit is to solve these problems in a sequential manner, with the output of a step being the input of the next one. Because of this, efficient solution methods are needed for every step of such a system.

In his dissertation, the author studies scheduling problems connected to the vehicles of a public bus transportation company. His theses are categorized into five groups, all dealing with different subproblems of these vehicles. Each such group is studied in a separate chapter.

In the remainder of this section, we give a brief overview of the most important concepts of vehicle scheduling, and then present the results of the dissertation starting with Section 2.

Basic concepts of vehicle scheduling

The input of the vehicle scheduling problem (VSP) is the *fleet of vehicles* of a transportation company, and the set of *timetabled trips* for a single day. The goal of the problem is to assign the vehicles to trips such that every trip is serviced exactly once, and the arising costs are minimal. This assignment has to consider certain constraints: a vehicle has to be able to service all its assigned trips, and trips assigned to the same vehicle have to be compatible

with each other. This basically means that they do not overlap in time, and the vehicle has enough time to travel from the ending location of a trip to the starting location of the next trip.

Vehicles start the scheduling period in *depots*, and return there at the end of this period. A vehicle can enter these depots any time during this period, when it has no assigned trips for a longer period of time. Depots represent garages, parking places, and similar locations, where vehicles can stay until the beginning of the next scheduling period. The concept of depots can be extended to represent the types of the vehicles as well. If the problem has two or more depots, it is referred to as a *Multi-Depot Vehicle Scheduling Problem* (MDVSP), and was proven to be NP-hard by Bertossi et al. [4]. The MDVSP can also have an additional constraint for the execution of the trips called *depot-compatibility*: a subset of the depots is given for every trip t , and only vehicle belonging to this set are able to service t .

A *vehicle block* is a series of pairwise compatible traveling activities. A vehicle block usually represents the set of daily tasks of a single vehicle, which is referred to as the *vehicle duty*. A feasible vehicle block always begins with a pull-out trip from the depot, and ends with a pull-in trip. The set of these blocks is called the *vehicle schedule*.

2 Heuristics for application-oriented vehicle scheduling

We present two application-oriented heuristic approaches connected to the MDVSP. We call these methods application-oriented, as their results are not only efficient from a theoretical point of view, but can also be applied in a real-life decision support system because of their structure and quick solution time.

2.1 Heuristic size reduction with variable fixing

The first heuristic approach we propose uses the concept of variable fixing to solve the VSP: it tries to reduce its problem size by finding trips that are likely to be in the same sequence in the final solution, and combines such groups into single, long trips, called 'stable chains'. The resulting smaller problem is solved using the classical IP modeling approach, and solutions are obtained with a greatly reduced running time. This method is studied in our following publications: [6, 8].

We show three different approaches to find such sequences: the first one is based on a cost function, basically fixing 'cheap' trips into these chains. The second one builds more flexible chains by considering depot-compatibility:

only those trips are fixed together that are compatible with the same set of depots. The third approach utilizes a real-life characteristic of the problem: it creates chains using trips that belong to the same bus-line. All three methods result in good quality solutions with short running times. Through extensive testing on real-life and randomly generated input, we show that while generally good methods might be more appealing from a theoretical point of view, using approaches based on structural properties of real-life instances result in solutions that have both a better performance and structure considering practical applications.

2.2 'Driver-friendly' vehicle scheduling

The second proposed heuristic approach is an iterative algorithm that produces vehicle schedules with a structure that also satisfies basic driver scheduling constraints. First, a classical VSP is solved, and the result is modified through different steps. This process iterates until all trips of the input are part of a feasible a vehicle block. We introduced this method in [1].

A VSP is solved as the first step of the algorithm, and then the resulting blocks are modified by a cut-and-join algorithm so that they satisfy the rules regarding maximum driver shift length. Blocks of the vehicle schedule are classified based on their length:

1. Class C_1 contains blocks that can be executed by a single driver.
2. Blocks in class C_2 require two drivers, so these are divided into a larger part (corresponding to a full-time shift) and smaller part (considered as a half-time shift).
3. Class C_3 contains blocks that can be covered by two full-time driver shifts.

A matching algorithm is then used to join part-time shifts into so-called split-shift. Shifts that are not part of the solution can either be considered as a full shift, or their trips can be deleted, and rescheduled to form new blocks in a future step. Driver breaks are then inserted into these shifts by a transformation step. Breaks have to be issued in given time-windows; if shifts don't have free intervals in these windows, then trips have to be removed from them. Such trips are either moved to another shift, or deleted completely. If there were any deleted trips during the iteration, the process starts again, with the deleted trips as the input of the VSP.

To present the quality of the resulting schedules, we also develop a method that gives a lower bound on the total working time of a schedule based on a timetable of trips. Solutions on real-life instances show that, based on

the gap from this lower bound, a significant improvement can be achieved compared to the original schedules of the transportation company. Any method can be applied in the first step of the process for the solution of the VSP, and because of this, good quality solutions can be achieved for the real-life test instances in several minutes.

3 Integrated vehicle scheduling and assignment

We introduce the integrated vehicle scheduling and assignment problem, which aims to give a feasible vehicle schedule that also includes tasks specific to the requirements of the vehicles executing its blocks. In real-life, a vehicle block is not only a virtual set of tasks that have to be serviced in the given sequence, but it also has a vehicle assigned to it as well. The special needs of this vehicle have to be taken into account while it is in service: for example, it can run out of fuel (and has to be refueled), or has a longer idle period (and has to be sent to a parking location). Constraints such as these are not widely studied. We will refer to them as vehicle-specific activities.

We present a set partitioning model for the integrated vehicle scheduling and assignment problem. This serves as a general framework that can integrate most vehicle-

specific activities, and also provides a flexible model where many of these application-oriented constraints can be included easily. We give a column generation-based solution method for this model, and show its efficiency on randomly generated test instances. To showcase the model, refueling is considered for these instances as their vehicle-specific activity, and multiple fuel-types are also studied for the vehicles, which is also rarely examined in other papers. We introduced the integrated vehicle scheduling and assignment problem in [3].

4 Managing disruptions with vehicle re-scheduling

Creating vehicle schedules in advance is useful from a planning perspective, but the series of tasks carried out in practice by the end of a day is generally different from the pre-planned ones. The main reasons for this are unforeseen events that happen during the execution of a daily schedule. These events are called disruptions, and the field that deals with them is referred to as disruption management. Disruptions can occur for various reasons, such as lateness, vehicle breakdown, or a newly introduced task that has to be serviced. Disruptions have to be addressed as soon as possible to restore the order of transportation. If

such an event occurs, the company has to create a new feasible schedule, where all available tasks are carried out in a feasible manner once again.

4.1 Multi-depot vehicle rescheduling

We formulate a multi-depot network model for the vehicle rescheduling problem (VRSP), which deals with the solution of a single disruption scenario. As such a problem requires a fast, real-time result, this model can only be used as quality control for quick solution algorithms. We also propose two simple, but effective heuristic approaches for the fast solution of the VRSP. One is a recursive method that traverses the search tree of the problem, and distributes trips of a single disruption scenario to the available blocks either by simple insertion, or by deleting overlapping trips from these blocks. As the size of this search tree would be extremely large for efficient use, we limit its depth using a simple practical observation: the depth of this search tree corresponds to the number of modified vehicle blocks. Since a useful practical solution of the VRSP will only contain a small number of modified blocks (as completely rescheduling the itinerary of a large number of vehicles would be hard to manage in real-time), this limit will cut mainly inefficient solutions. The resulting tree can be traversed in a short time, and multiple good results

can be presented as suggestions to the planners of a transportation company. The other method is a tabu search algorithm, which starts with an infeasible vehicle block in its schedule, and uses its neighborhood transformations to find a good quality feasible solution. This method is incentivized to remove trips from this infeasible schedule first, which leads to the method trying to fix the disrupted scenario rather than wanting to 're-optimize' its already feasible blocks by moving or swapping trips between them. The tabu search can also provide multiple feasible solutions with a short running time, if needed. Because of their ability to produce multiple good quality solutions in a short time, these algorithms seem suitable for a decision support system that helps the operators of a transportation company in their rescheduling process by giving them possible solution suggestions for the arising problems. Our model and heuristics for the VRSP were published in [9].

4.2 Dynamic vehicle rescheduling

We also introduce the dynamic vehicle rescheduling problem (DVRSP). While papers dealing with disruption management consider the solution of a single disruption scenario, our aim was to provide an alternative evaluation for rescheduling methods. As multiple disruptions happen during a day, the solutions for these problems should not

be evaluated independently. We introduce the concept of DVRSP for handling multiple independent disruptions of vehicle schedules over a daily horizon. While the classical method of managing disruption with the VRSP focuses on solving single disruptions to optimality, the DVRSP aims for a good quality solution at the end of the day after managing a series of disruptions. Because the problem itself is dynamic, and the input (the list of disruptions) arrives in an online manner, we also presented the concept of the quasi-static DVRSP, which gives us an 'offline' version of the problem where all the disruptions are known in advance. The quality of a solution for the DVRSP can be measured using this model.

We apply both heuristic methods presented for the VRSP to solve a series of disruptions for pre-planned daily schedules, and show that they give good quality solutions in a short time for the DVRSP. Each input instance is tested independently with both heuristics. Their combined results are also presented, where every disruption scenario is solved by both methods, and the better quality result is chosen as the solution of the current step. This somewhat simulates the real-life decision making process of transportation planners, where they have to choose one of multiple possible solutions to solve each disruption. The DVRSP was proposed in [10].

5 Vehicle assignment over a planning horizon

While all previous sections study problems connected to a single daily vehicle schedule, it is important to remember that the long-term plans of transportation companies are created in advance for a horizon of several days or weeks. We introduce the schedule assignment problem for public bus transportation, which aims to assign the pre-planned daily vehicle blocks of a planning horizon to buses in the fleet of a transportation company. As the problem considers a long-term plan for several days or weeks, activities connected to the vehicles such as daily parking and regular preventive maintenance have to be taken into account.

We give a state-expanded multi-commodity flow network for this problem, which is then solved by a MIP solver. The efficiency of the model is presented on both real-life and randomly generated instances, and its results are promising for different number of vehicle types, varying parameters for the time limit of the preventive maintenance, and different lengths of the planning period as well. Instead of optimizing the period sequentially on a day-by-day basis, the model represents the structure of the entire planning horizon, and achieves a solution by considering all arising constraints of every day together. We presented

a simple assignment model for the problem with parking in [7], and proposed the state expanded model that also considers maintenance in [11].

6 Concepts for decision support systems

Transportation systems are hard to design, as they incorporate many different modules that have to work together in order to provide a good quality service. The parts of such a system connected to vehicles include: planning, evaluation and real-time control. While the planning phase is extensively studied through different problems in the literature, the other two steps are rarely considered as part of a decision support system. We introduce multiple concepts that are not well studied from this application-oriented aspect, but can serve as an important module of such a system. We present real-time control through a decision support framework for vehicle rescheduling, and give a timed automata-based model and a test instance generator that can be used for evaluating different modules of an optimization system.

Disruption management is an important part of the everyday operations of a transportation company. Because of this, we propose a solution framework for this problem, which is capable of handling multiple solution approaches

in parallel, analyzing their results, and providing multiple different suggestions to the operators of the company. This system has a high number of different parameters that can be modified depending on the type of disruption, or the structure of the solution. We proposed this system in [2].

We also introduce a new modeling technique for the schedule assignment problem using timed automata. This is developed as a model for transportation scheduling problems where the solution process is easy to follow, and the structure represents every important aspect of the input scenario. The resulting system can be efficiently applied to visualize the structure of the problem, and validate certain queries about the input. This idea was introduced in [12].

Finally, we present a random instance generation method that is useful to create input with a structure similar to real-life problems. Methods that are developed for transportation systems have to be tested extensively, but proper benchmark data is unfortunately not available freely. We developed this method as an alternative to the existing state-of-the-art approach by Carpaneto et al. [5], as we felt that the instances generated by their algorithm were not similar enough to the real-life instances of the city of Szeged. Our approach provides instances with multiple depots and vehicle types, and can also include data for refueling activities. This method was given in [8].

7 Summary of the theses

The author categorizes his theses into five different problem groups:

I We proposed several application-oriented heuristics for the MDVSP.

I/1 We presented three variable fixing heuristic methods for the MDVSP. Their results showed that exploiting an important practical property (considering the bus-lines of the trips) provides good solutions with a structure that is closer to real-world schedules.

I/2 We introduced a 'driver-friendly' iterative algorithm, which creates solutions for the MDVSP with a structure satisfying the most important rules of driver scheduling.

II We presented the integrated vehicle scheduling and assignment problem, which creates vehicle schedules that also consider the vehicle-specific activities (like refueling or parking) of the buses in service. A set partitioning model was developed for this problem.

III We studied two main concepts in the area of vehicle rescheduling.

- III/1** We proposed a multi-depot network model for the VRSP. We also designed two heuristic solution methods for the problem, which are both able to provide multiple good quality solutions with a short running time.
- III/2** We introduced the dynamic vehicle rescheduling problem, which aims to evaluate solutions for a series of disruptions by studying the resulting schedule at the end of the day instead of considering every disruption as a stand-alone problem.
- IV** We introduced the schedule assignment problem, which assigns vehicles to the pre-planned daily blocks of the company over a longer horizon. We formulate a state-expanded multi-commodity network model for the problem, which considers both parking and maintenance constraints of vehicles executing blocks over the planning period.
- V** We studied three different concepts that can be integrated into the optimization system of a transportation company. These can be used for real-time control and evaluation for certain modules of the system.
- V/1** We proposed a decision support framework for vehicle rescheduling. This framework can apply multiple

Bibliography

- [1] Viktor Árgilán, János Balogh, József Békési, Balázs Dávid, Miklós Krész, and Attila Tóth. Driver scheduling based on “driver-friendly” vehicle schedules. In *Operations Research Proceedings 2011*, pages 323–328. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-29210-1.
- [2] János Balogh and Balázs Dávid. An algorithmic framework for real-time rescheduling in public bus transportation. In *Proceedings of the 2013 Mini-Conference on Applied Theoretical Computer Science (MATCOS)*, pages 29–33, 2016. ISBN 978-961-6984-21-8.
- [3] József Békési, Balázs Dávid, and Miklós Krész. Integrated vehicle scheduling and vehicle assignment. *Acta Cybernetica*, 2017. 18 pages, submitted.
- [4] Alan A Bertossi, Paolo Carraresi, and Giorgio Gallo. On some matching problems arising in vehicle scheduling models. *Networks*, 17(1): 271–281, 1987.
- [5] G. Carpaneto, M. Dell’amico, M. Fischetti, and P. Toth. A branch and bound algorithm for the multiple depot vehicle scheduling problem. *Networks*, 19(5):531–548, 1989. ISSN 1097-0037. doi: 10.1002/net.3230190505.

- [6] Balázs Dávid. Heuristics for the multiple-depot vehicle scheduling problem. In *Proceedings of the 2010 Mini-Conference on Applied Theoretical Computer Science (MATCOS)*, pages 23–28, 2011. ISBN 978-961-6832-10-6.
- [7] Balázs Dávid. Schedule assignment for vehicles in inter-city bus transportation over a planning period. In *Middle-European Conference on Applied Theoretical Computer Science (MATCOS 2016): Proceedings of the 19th International Multiconference INFORMATION SOCIETY - IS 2016*, pages 9–12, 2016. ISBN 978-961-264-104-7.
- [8] Balázs Dávid and Miklós Krész. Application oriented variable fixing methods for the multiple depot vehicle scheduling problem. *Acta Cybernetica*, 21(1):53–73, 2013. ISSN 0324-721X. doi: 10.14232/actacyb.21.1.2013.5.
- [9] Balázs Dávid and Miklós Krész. A model and fast heuristics for the multiple depot bus rescheduling problem. In *PATAT 2014: Proceedings of the 10th International Conference of the Practice and Theory of Automated Timetabling*, pages 128–141, 2014. ISBN 978-0-9929984-0-0.
- [10] Balázs Dávid and Miklós Krész. The dynamic vehicle rescheduling problem. *Central European Journal of Operations Research*, 25(4): 809–830, 2017. ISSN 1613-9178. doi: 10.1007/s10100-017-0478-7.
- [11] Balázs Dávid and Miklós Krész. Multi-depot bus schedule assignment with parking and maintenance constraints for intercity transportation over a planning period. *Transportation Letters*, 2017. 16 pages, submitted.
- [12] Balázs Dávid, Máté Hegyháti, and Miklós Krész. Linearly priced timed automata for the bus schedule assignment problem. In *Proceedings of the 4th International Conference on Logistic Operations Management - GOL'2018*, 2018. 7 pages, accepted.
- [13] Jean-Paul Rodrigue, Claude Comtois, and Brian Slack. *The geography of transport systems*. Routledge, 2009. ISBN 978-0415483247.

