UNIVERSITY
OF PARDUBICE
FACULTY
OF TRANSPORT
ENGINEERING

Vehicle and crew scheduling optimization for public transport

Optimalizace oběhů vozidel a osádek v městské hromadné dopravě

Doctoral thesis precis

2021

Kateřina Šulcová

Student:

Kateřina Šulcová

Study Programme:

P3710 Technique and Technology in Transport and Communications

Field of study:

3708V024 Technology and Management in Transport and Telecommunications: Technology

Supervisor:

doc. Ing. Radovan Soušek, Ph.D.

Department:

Department of Transport Management, Marketing and Logistics

Contents

1 Current knowledge of vehicle and crew scheduling problem

Vehicle scheduling is the task of assigning vehicles to efficiently cover all trips within a timetable. Similarly, crew scheduling assigns the crew to efficiently cover all the operating vehicles. Vehicle and crew scheduling are widely studied problems having many subsequent solutions found. The problem of minimizing the number of vehicles needed to satisfy the timetable schedule can be solved for example by set covering or graph coloring, maximum flow and many more.

The problem of minimizing the crew is generally more algorithmically demanding than basic vehicle scheduling. The reasons behind higher computational complexity of crew scheduling are further constraints posed on the crew schedules. These constraints are typically based on legal regulations, collective bargaining agreements, and on vacation schedules of the crew. Higher computational complexity of crew scheduling creates the necessity to apply heuristics to obtain the solution in reasonable time for large sized problems.

Public transportation in the Czech Republic at the time of creation of crucial parts of this work faced lack of vehicles, and especially lack of crew members, see SDP ČR (2018). Moreover, crew salary and costs of vehicle usage are the main cost items of a public transport company, as per Pels and Rietveld (2000). Even the transport companies that already use software for solving vehicle and crew scheduling tasks call for further possibilities to lower the number of vehicles and crew necessary. Therefore, we investigate novel possibilities of scheduling optimizations, focusing on lowering the amount of vehicles and crew. Even with covid 19 disease impacting public transport it is advantageous to use the scheduling optimization, not only from the perspective of stepping up the efficiency of used vehicles and crew members, but also from the perspective of risk management, especially if some of the crew members get infected with covid 19 or are quarantined.

In the following section, we define the vehicle and crew scheduling problems, and we share the relevant results. The general overview on vehicle scheduling is inspired by Daduna and Paixão (1995) and Bunte and Kliewer (2009) and overview on crew scheduling by Wren and Rousseau (1995) and Ciancio et al. (2018).

1.1 Vehicle scheduling

Based on the given timetabled trips with stop arrival and departure times defined, as well as start and end locations, the objective of vehicle scheduling problem is to assign the trips to vehicles, satisfying the following requirements:

- each trip is assigned to exactly one vehicle
- feasibility of sequence of trips that each vehicle performs has to be assured

- according to the upfront selected objective function, minimization problem needs to be solved
- further technical and company restrictions have to be respected

Generally, the objective function is a cost function. We can differentiate the costs posed on vehicle to fixed cost and operational costs. Fixed cost mostly comprise of the initial investment and maintenance, operational costs comprise of cost of fuel and attrition. For operational costs, the aim is to minimize the non-productive time and distance.

First three points in the problem description define a basic vehicle scheduling problem, while the fourth point allows the problem to be extended by additional requirements. The typical extension is for multiple depots, or multiple vehicle types, see Costa, Branco, and Paixão (1995) or Guedes and Borenstein (2018). Also, restriction on number of bus line changes in vehicle routes may be posed, see Kliewer, Gintner, and Suhl (2006). In other extensions, variable departure times of trips are allowed within a specified time window, allowing for slight changes in the originally timetabled data, see for example Daduna, Mojsilovic, and Schautze (1993), Desaulniers, Lavigne, and Soumis (1998), Schmid and Ehmke (2015), Desfontaines and Desaulniers (2018), time windows along with close trips aggregation is considered in Visentini et al. (2019), or the requirement is posed to use fixed number of vehicles, see Paixão and Branco (1988). Vehicle charging is considered within vehicle scheduling and charging optimization for a bus fleet containing also electric buses in Zhou et al. (2020).

Vehicle scheduling problem for single depot

Vehicle scheduling problem for single depot is comparatively the easiest of vehicle scheduling problems, as it can be formulated as a problem for which polynomial time algorithm is known. In the following text we briefly introduce the models used to solve the vehicle scheduling problem.

First optimal solution for the vehicle scheduling problem was provided in Saha (1972) using minimal decomposition model. Such model restrains from using dead-heading trips, therefore Bodin and Rosen (1976) solves the minimal decomposition model with dead-heading. Generally, by minimal decomposition we can solve the problem of minimum fleet size, but it is unable to take into account the vehicle operational costs.

Both fleet size and vehicle operational costs are considered within objective function in assignment model by Orloff (1976) as well as quasi-assignment model by Gavish and Shlifer (1978), where bipartite graphs are used for modeling the schedules. Later, network flow model was introduced by Bodin, Golden, et al. (1983), where minimum cost flow problem needs to be solved.

With the knowledge of vehicle load profiles, case study by Tang et al. (2018) adjusts a few trips to operate by limited stop strategy, short turn or dead-heading, in order to further lower the number of necessary vehicles. Further overview and future research paths of vehicle scheduling optimization methods which use automated data collected from intelligent transportation systems is provided in Iliopoulou and Kepaptsoglou (2019).

Another approach to vehicle scheduling optimization along with timetabling is consideration of the passenger waiting costs, see Shang et al. (2019). For periodic timetables, problem of joint optimization of the timetable and the vehicle schedule is considered in Van Lieshout (2021).

Vehicle scheduling problem for multiple depots

Within multi-depot vehicle scheduling, the vehicles are housed in several depots, and a vehicle schedule must start and end at the same depot. Unlike single depot vehicle scheduling, multiple depot vehicle scheduling is proven to be NP-hard, see Bertossi, Carraresi, and Gallo (1987). Several heuristics have been proposed, see for example Bodin, Rosenfield, and Kydes (1978), Lamatsch (1992), Mesquita and Paixão (1992).

To solve the multiple depots vehicle scheduling to optimality, branch and bound algorithm was used for computation of lower bounds by an additive procedure in Carpaneto et al. (1989). Later, integer multi-commodity flow formulation of the problem was proposed by Ribeiro and Soumis (1991), which is solved by column generation algorithm. In Oukil et al. (2007), stabilized column generation approach is proposed, which handles efficiently even highly degenerate problems.

To be able to account for diversity of traffic and driving conditions, dynamic vehicle rescheduling algorithms appear in Shen, Zeng, and Wu (2017), aiming to maximize the execution of the originally planned schedule.

1.2 Crew scheduling

Crew scheduling is constructing the crew duties to cover all the blocks within vehicle schedules in a cost effective way. The vehicle blocks can be divided into several pieces of work which start and end at predefined relief points. Crew duty consists of consecutive pieces of work which are mutually feasible. Even though a driver can change between different vehicles, unnecessary changes of vehicle blocks lead to inefficiencies.

Rules that apply to crew scheduling are specific to given country based on legal regulations. Rules are also posed within organizations by the collective bargaining agreements of the labor unions, by crew bids and vacation schedules. Sometimes we need to take into account also different qualification and licensing of the crew members. Typical restrictions are posed on the total working time as well as total spreadover, which is the duration between start and end of a duty. Also, there is a maximal length of working time without provisioning of a meal break.

Crew scheduling problem is mostly solved by the framework of set covering, set partitioning, and multiobjective models. Even satisfying the basic requirements makes crew scheduling to be NP-hard. Therefore, several heuristics were proposed, using for example genetic algorithms in Song et al. (2015) or tabu search in Cavique, Rego, and Themido (1999). Exact methods exist, solving crew scheduling to optimality. Most of them are using column generation, see Desrochers and François (1989), or branch and bound algorithms, see Barnhart et al. (1998).

Algorithms solving for both vehicle and crew scheduling appear recently in Horváth and Kis (2017), Boyer, Ibarra-Rojas, and Ríos-Solís (2018) and Ciancio et al. (2018), optimizing the collective objective function for the combined problem.

Within Czech and Slovak republic, the mutual interconnection between crew and vehicles are very tight. Usually, one vehicle is operated by one crew member or at most two crew members. This requirement together with legal obligations and collective bargaining agreements constraints typical for Czech and Slovak republic is satisfied by vehicle and crew scheduling system Kastor by Palúch and Majer (2017).

1.3 Current state analysis, focused on goal specification

Similarly as in Tang et al. (2018), we aim to adjust few trips within the timetable in order to lower the number of vehicles needed to cover the timetable. We look comprehensively at the combination of timetabling along with vehicle scheduling, aiming to identify the minimal set of trips, such that omitting them from the vehicle scheduling problem lowers the amount of vehicles necessary to cover the timetable. Let us denote such trips as *critical trips*. For more rigorous problem definition using graph theory, refer to section 3.

Figure 1 shows for each time moment both the amount of timetables trips and the amount of vehicles necessary to cover the trips. In figure 1 we can observe that due to the death-running between the end of previous trip and start of the next trip, the maximal vehicle usage occurs during the second peak at 14:30, having only 74 consecutive trips, opposed to morning peak with 79 consecutive trips. As per figure 1, peak at 7:48 can be covered by 88 vehicles, whereas there are 90 vehicles necessary to cover the peak at 14:30.

Using figure 1, we identified the peak in vehicle usage. However, based on the information from figure 1, it is very demanding to perform critical trips identification. Therefore we seek for a systematic approach for critical trips identification, using graph theory. Our approach, along with basics of graph theory, is described within next sections.

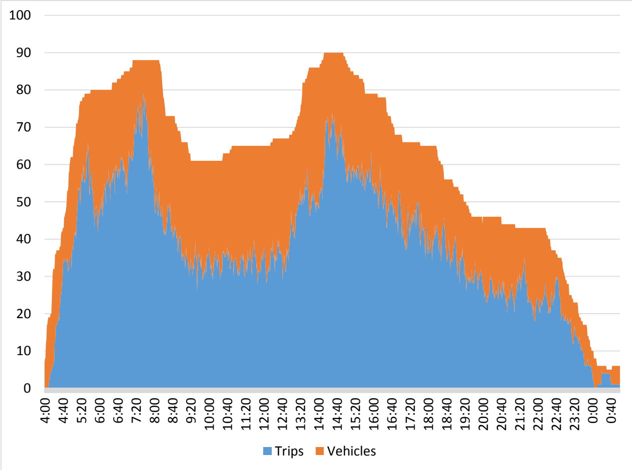

Figure 1: Number of trips and vehicles necessary to cover the timetable[1]

[1]Data source: weekday timetable of public transport company of Liberec and Jablonec nad Nisou

When the critical trips are identified, we need to remove them from the vehicle scheduling problem and handle them in a different way. Therefore, we propose several methods of critical trips handling. For selected handling method, we propose metrics to evaluate critical trips and select the most feasible ones for handling based on the metrics value, in order to fasten decision making process. We then analyze the impact of critical trips identification and handling on the efficiency of vehicle and crew utilization within public transport company via a case study conducted on a selected public transport company.

Due to the different possibilities of critical trips handling, in some cases it is more cost effective to handle critical trips with overall shortest run time, rather than minimum amount of critical trips. Therefore, we aim to provide both alternatives of critical trips.

2 Objective

The vision of usage of this work is efficiency step up of resources within public transport companies. We aim to achieve it by lowering the amount of necessary vehicles, and possibly drivers. The idea of achieving the goal is based on the fact, that reasonably small amount of trips within a timetable can be shortened, or departure times can be shifted within acceptable range. Sometimes, few trips can be even dropped out of the vehicle scheduling if it leads to lowering the number of vehicles needed to cover the updated timetable.

The main objective of this work is to design an algorithm for critical trips identification, i. e. for identification of the minimal set of trips, such that omitting them from the vehicle scheduling problem lowers the amount of vehicles necessary to cover the timetable.

We also aim to provide meaningful number of alternative critical trips, in order to be able to choose the best alternative for further handling.

To step up the usability and usage of such algorithm, we also consider subsequent objectives:

- Algorithm for identification of the set of trips with overall shortest run time, such that omitting them from the vehicle scheduling problem lowers the amount of vehicles necessary to cover the timetable.
- Proposition of methods for critical trips evaluation and handling.
- Analysis of impact of critical trips identification and handling on the efficiency of vehicle and crew utilization via a case study conducted on a selected public transport company.

The benefits of critical trips identification and handling is not only the straightforward lowering of the number of vehicles and crew members needed to cover the timetable, but also increasing the efficiency of the used vehicles and crew, as there is an amount of work which would have been otherwise covered by unused vehicles, which gets distributed amongst the lower number of used vehicles and crew. Also, knowledge of critical set of trips can be used as an iterative optimization loop between the processes of timetabling and vehicle scheduling, as it can give a quick feedback on the impact of a change in timetabled data.

3 Methods and solution approaches

Having analyzed current knowledge in the field of vehicle and crew scheduling and identified a gap within current knowledge, we defined goals of the dissertation, mainly critical trips identification and handling. In this section we provide the methods of achieving the goals, as well as the solution approaches. Our approach to the problem of lowering the number of vehicles and crew members is based primarily on graph theory. First, we model the problem using graphs, then we perform necessary transformations in order to solve the original problem, ideally by using analogical graph theory problem with known solution. We then propose methods for critical trips evaluation and handling. We conclude by a case study, which analyzes impact of critical trips identification and handling on the efficiency of vehicle and crew utilization within selected public transport company.

3.1 Basics of graph theory

To be able to define the fundamental problem, we first give a basic overview on specific segments of graph theory. We guide the reader through the definitions of directed and oriented graph. For a comprehensive overview of graph theory and its applications see Gross and Yellen (2005).

Directed graph is an ordered pair $G = (V, E)$, where V is a finite set of *vertices* and $E \subseteq V \times V$ is a set of *edges*. In a directed graph, the direction of the edge is essential, we say that the edge $[v_1, v_2]$ leads from vertex v_1 to v_2. *Oriented graph* is a special case of directed graph, in which each pair of vertices can be connected by at most one edge. Therefore, in oriented graphs do not exist bi-directional edges. *Path* in a graph is a sequence of edges directed in the same direction which connect a sequence of mutually distinct vertices. Several optimization methods using graph theory have been applied for vehicle scheduling optimization, for overview see section 1.

Using the above defined notion, let us define the fundamental problem for critical trips identification by modelling it using graph theory. Having a set of trips to cover, we define an oriented graph in which the vertices represent the trips. An edge of length 1 from vertex v to vertex w exists if the trip w can be serviced after the trip v by the same vehicle, i.e. if the time of death running from final station of trip v to the first station of the trip w is shorter than the break time between end of v and start of w. Figure 2 shows an example such graph, containing 8 trips.

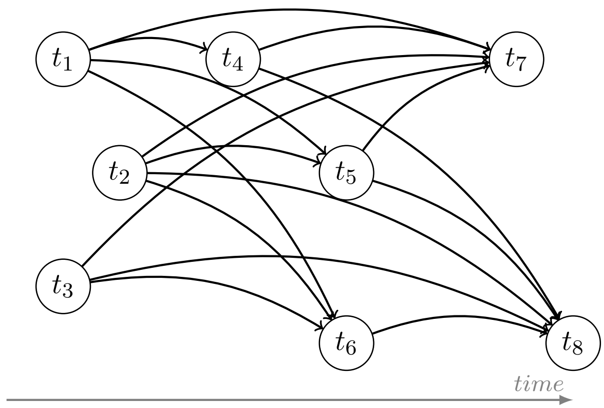

Figure 2: Graph for searching the set of longest disjoint paths[2]

[2]Created by the author

Let m denote the minimal number of vehicles necessary to cover the timetable, which can be found by algorithm from Palúch (2001). For each $k \in \mathbb{N}, k < m$, find $m - k$ disjoint paths within the above defined graph with the maximum sum of lengths. Then the trips which aren't for given k included in any of the disjoint paths are the desired critical trips to be omitted in order to cover the timetable by $m - k$ vehicles. There is no general solution for set of longest disjoint paths problem. Therefore, our main goal is to transform the problem into a problem with known solution and polynomial complexity.

3.2 Genetic algorithms

Genetic algorithms, see Deepa and Sivanandam (2010), are adaptive heuristic search algorithm inspired by evolution and the law of natural selection and genetics. The basic concept of genetic algorithms is designed to simulate processes necessary for evolution, specifically those that follow the principle of survival of the fittest. Genetic algorithms are used to generate solutions to optimization problems by relying on biologically inspired operators such as mutation, crossover and selection. For the purposes of this work, we focus specifically on crossover operator. Crossover operator creates offspring out of a combination of genetic information of at least two parents. Such offspring then combines characteristics of both of its parents. Therefore, crossover operator provides a way to generate new solutions from the existing ones.

3.3 Case study

A case study (Yin, 2017) is a detailed study of a specific subject, which is conducted in order to better understand given phenomenon. There are multiple different possibilities of conducting a case study, and its settings and parameters differ mainly based on the field in which it is used, and on its goal.

Case study starts with introduction which describes scope and purpose of the study, i. e. the goal. Next, we describe the methods used, and discuss main findings. In our case, discussion will be covering implementation of the proposed method in the real world context. We conlude by summary of our findings.

Having the results of a case study in hand, we always need to bear in mind that a single case result may be caused by random disturbance or error.

3.4 Critical trips identification

Outputs of vehicle scheduling are blocks of trips to be covered by the minimal number of vehicles m. If some of these blocks are very small, containing for instance only 1 trip, it is questionable whether it is efficient to cover them by a vehicle. Especially if the transport company is running short on vehicles, cancellation, outsourcing or some other handling possibility of the minimal block can be considered. The goal is to cover the timetable by m blocks, while minimizing the size of the last block, or minimizing the overall size of the last k blocks.

Therefore, we define an oriented graph in which the vertices represent the trips and an edge of length 1 from vertex v to vertex w exists if the trip w can be serviced after the trip v by the same vehicle, i.e. if the time of death running from final station of trip v to the first station of the trip w is shorter than the break time between end of v and start of w. Figure 2 shows an example graph containing 8 trips.

If we find $m-1$, or $m-k$ disjoint paths within this graph with the maximum sum of lengths, then the trips which aren't included in any of the disjoint paths are the desired critical trips. In the following section, we will transform this problem into the shortest disjoint path problem. Following algorithms were published in Pastirčáková and Šulc (2018).

3.4.1 Transformation into shortest disjoint path problem

When modelling the problem using graph theory, we need to ensure that each trip is covered by a vehicle at most once. Therefore we define graph G within which each trip is represented as a triple of input vertex t, output vertex t' and an edge from t to t'. Both vertices t, t' and the edge $[t, t']$ are uniquely identifying one specific trip, we refer to the trip itself also by t. Let us denote $time(t)$ the starting time of the trip t, and $time(t')$ the ending time of the trip t. For all trips t let us assign the cost of edge $[t, t']$ equal to $time(t') - time(t) - 1$. An edge from output vertex t' to input vertex u exists if the trip u can be serviced after the trip t by the same vehicle. The cost of such edge will be the timespan $time(u) - time(t')$, i. e. the timespan between the end of trip t and the start of trip u.

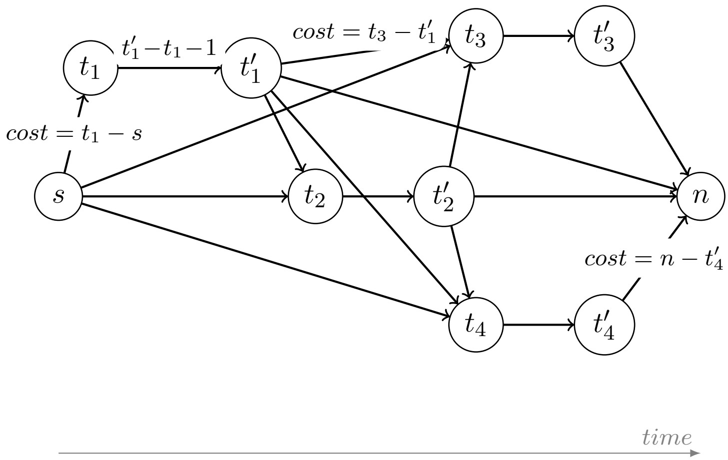

Figure 3: Graph for the shortest disjoint path problem[3]

[3] Created by the author

We define a sink n such that $time(n) > time(t')$ $\forall$ trip t. Then for each trip t we create an edge from t' to n with cost $time(n) - time(t')$. Similarly, we define a source s such that $time(s) < time(t)$ $\forall$ trip t. Then for each trip t we create an edge from s to t with cost $time(t) - time(s)$. Figure 3 shows an example of graph G containing 4 trips.

Having graph G, we solve the shortest disjoint path problem by Bhandari algorithm (Bhandari, 1999), which iteratively finds the i overall shortest disjoint paths for each $i \leq m$, i.e. the i longest blocks covering the highest possible number of trips by i vehicles. Providing such results as a feedback to the planners, they can quickly decide which number of shortest blocks to get rid of, i. e. what is the desired reduction k of the number of necessary vehicles. Within algorithm 1, we outline in pseudocode the usage of Bhandari algorithm for critical trips identification.

$G_1 := G$;
$i := 0$;
while *exists a vertex in G not included in i shortest disjoint paths* **do**
  $i := i + 1$;
  Within the graph G_i find shortest path p_i from s to n, using any algorithm allowing for the negative edge costs;
  Form graph G_{i+1} from G_i by turning all the edges of the path p_i in the opposite direction and assign them inverse costs;
  The i shortest disjoint paths $s_1^{(i)}, s_2^{(i)}, \dots s_i^{(i)}$ are formed by the paths $p_1, p_2, \dots p_i$, where all the pairs of edge and inverse edge within these paths cancel themselves out and are not included in the final i disjoint paths $s_1^{(i)}, s_2^{(i)}, \dots s_i^{(i)}$;
  The i-th set of critical trips $C^{(i)}$ is formed by all the trips t whose input vertex t is not included in the i shortest disjoint paths $s_1^{(i)}, s_2^{(i)}, \dots s_i^{(i)}$;
end

Algorithm 1: Critical trips identification

Algorithm 1 can be modified to obtain critical trips of minimal sum of lengths by changing the costs of the trips to 0 within graph G, i. e. setting for each trip t the cost of edge $[t, t']$ equal to 0.

3.4.2 Target value for reduction in the amount of necessary vehicles

Even though algorithm 1 yields specific set of critical trips, its main strength lies in quick evaluation of the target value for reduction in the amount of necessary vehicles. Let m be the minimal number of vehicles needed to cover the timetabled trips. Then the following relationship holds

$$\left|C^{(m-k)}\right| \geq k \quad \forall k \in \{0, 1, 2, \cdots, m\}, \tag{1}$$

because in order to reduce the number of necessary vehicles by k we need to remove at least k trips from vehicle scheduling problem. Observation of the pairs $\left[k, \left|C^{(m-k)}\right|\right]$ gives us quick insight into how many trips we need to handle in order to achieve reduction of k vehicles. Here, quick decision can be made on reasonable and achievable target for parameter k. Also, we need to realize that we do not have to stick only to the identified critical trips $C^{(m-k)}$, as there can exist alternative trips with similar outcome. Within next section, we perform the identification of critical trips alternatives.

3.5 Critical trips alternatives

Based on the evaluation of the size of $\left|C^{(m-k)}\right|$ we select the target value for parameter k. In this section, we provide further alternatives to the originally identified critical trips $C^{(m-k)}$, as critical trips can be interchangeable with some of the trips which are included in the $m-k$ shortest disjoint paths $s_1^{(m-k)}, s_2^{(m-k)}, \dots s_{(m-k)}^{(m-k)}$.

First, we describe heuristic approach for obtaining some alternatives to critical trips. Then, we provide an algorithm which yields all the critical trips alternatives.

Heuristic approach for obtaining critical trips alternatives

This heuristic is inspired by genetic algorithms, specifically crossover operator. It swaps critical trips with trips included in the $m-k$ shortest disjoint paths and examines whether the created schedule is feasible. In other words, for each critical trip $c \in C^{(m-k)}$ we swap it with trip t included in the $m-k$ shortest disjoint paths $s_1^{(m-k)}, s_2^{(m-k)}, \dots s_{(m-k)}^{(m-k)}$. If swapping yields a feasible schedule, then trip t is a feasible alternative to the selected critical trip c, as it can be removed instead of the critical trip c from vehicle scheduling with the same effect. Thus we form set $H^{(m-k)}$, whose elements are sets $H_c^{(m-k)}$, which for each critical trip $c \in C^{(m-k)}$ contain trip c and its identified alternatives.

All critical trips alternatives

in order to obtain comprehensive sets containing all the critical trips alternatives, we need to run the algorithm 1 multiple times with slightly different weights of trips c that are identified as critical or alternative to critical. Let us denote the total amount of trips within the timetable as T. Then the weight of trip c is reduced by $\frac{1}{T+1}$, i. e. the cost of the edge $[c, c']$ is set to $time(c') - time(c) - 1 - \frac{1}{T+1}$. Within such setup, critical trips are preferred to be chosen to the shortest disjoint paths over trips that have not been yet identified as alternatives to critical. Having the new set of weights, we iteratively run algorithm 1 as long as new critical trips alternatives are identified. Thus we form set $A^{(m-k)}$, whose elements are sets $C^{(m-k)}$ for each of the iterative runs.

3.6 Critical trips evaluation and handling

Within this section, we suggest the possible ways of critical trips evaluation and handling. For selected handling method, we propose metrics to evaluate critical trips and select the most feasible ones for handling based on the metrics value. Let us briefly outline the considered handling possibilities together with the proposed metrics for trips evaluation:

- Rescheduling of the trip to a different time frame
  - Uniformity coefficient of a line $\frac{\sum_{k=1}^{n-2} \delta_{d_k d_{k+1}}}{n-2}$, where δ_{ij} is Kronecker delta, and sequence $d_1, d_2, \ldots, d_{n-1}$ is the sequence of time differences between consecutive trips of given line on given day
- Outsourcing
  - Price of the outsourced trip
  - If price is unavailable, then the overall runtime of the trip
- Trip cancellation
  - Average trip occupancy
- Covering of the trip by a backup vehicle and a temp crew
  - Backup vehicle and temp crew availability

4 Current results

In this section, we provide the results of a case study, and then within discussion we comment both on the advancement of the theoretical result within scientific field as well as its benefits for practical usage. We also comment on the utilizability of the result in multi-depot and multi-vehicle scheduling.

4.1 Case study

Within this case study, we aim to apply the proposed method of critical trips identification, evaluation and handling on timetabled data of public transport company of Liberec and Jablonec nad Nisou. During the preparation of paper Pastirčáková and Šulc (2018), the proposed algorithms were implemented in C#, see Microsoft Corporation (2018). The same implementation was used to carry out this case study. The timetabled data are taken for Saturday schedule during winter of 2017, for buses only. There is only one bus depot. The bus fleet is heterogenous, however, Saturday schedule requires less than half of the available vehicles and can be fully covered by low-floor buses. Therefore, without limiting the generality, we treat the schedule as single-vehicle.

There are overall 997 timetabled trips within 28 routes. Figure 4 shows for each time moment within Saturday the amount of timetabled trips and the necessary amount of vehicles to cover them. The necessary amount of vehicles is computed by vehicle and crew scheduling system Kastor, see Palúch and Majer (2017). We can observe in figure 4 that the number of necessary vehicles is almost flat between 8:30 and 18:30, with a minor peak around 13:00.

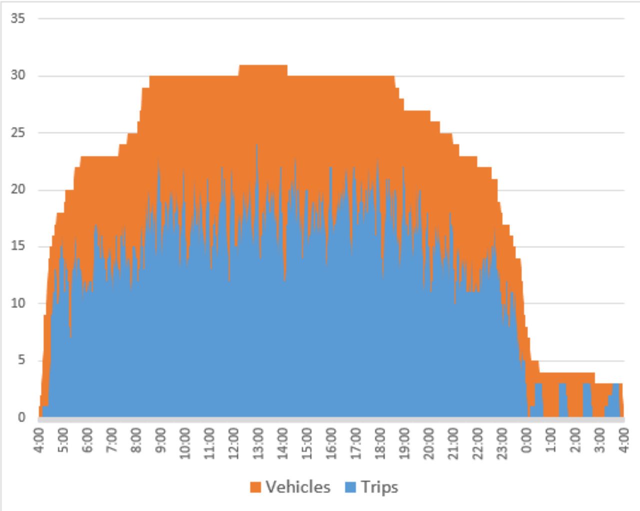

Figure 4: Number of trips and necessary vehicles for Saturday timetable[4]

[4]Data source: Saturday timetable of public transport company of Liberec and Jablonec nad Nisou

Upon the timetabled data, we run algorithm 1 to obtain the critical trips, and algorithm from section 3.5 to obtain all the alternatives. Figure 5 shows last 9 vehicle schedules after the last run of algorithm 1, i. e. the last 9 shortest disjoint paths $s_{23}^{(31)}, s_{24}^{(31)}, \ldots s_{31}^{(31)}$. Apparently, the last two critical paths $s_{30}^{(31)}$ and $s_{31}^{(31)}$ contain only one trip each. These 2 trips were identified as critical, therefore they are colored in orange. Algorithm from section 3.5 yielded 3 further alternatives of critical trips, colored in grey.

Let us denote the critical trips alternatives from figure 5 as a_1, a_2, a_3, a_4, a_5 (colored in grey and orange) from top to bottom. Having the full list of critical trips alternatives, trip a_5 was proposed by the subject matter experts to be cancelled right away. It was a trip which was not originally part of the timetable, but it was added based on a specific requirement of one board member of the transport

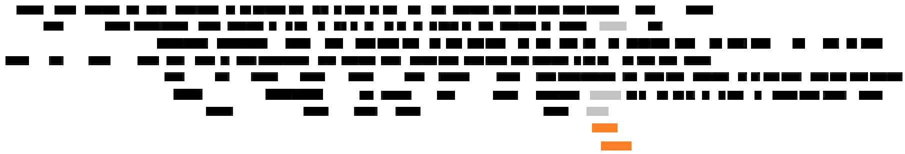

Figure 5: Segment of vehicle schedule for Saturday timetable yielded by algorithm 1 with critical trips identified in orange and critical trips alternatives in grey[5]

[5]Data source: Saturday timetable of public transport company of Liberec and Jablonec nad Nisou

company, who agreed that usage of extra vehicle and crew shift is not reasonable and cost effective for satisfying the requirement.

Therefore, there were 4 alternatives a_1, a_2, a_3, a_4 left for handling of 1 critical trip. The chosen handling method was trip rescheduling. Based on the value of uniformity coefficient, alternatives a_3 and a_4 were selected as candidates for rescheduling, with time window of 15 minutes. The trip shifting algorithm as per Schmid and Ehmke (2015) found a feasible schedule for alternative a_4 with time shift of -8 minutes, which was considered reasonable for given trip and route. Therefore, trip a_4 was rescheduled by shifting it to an 8 minutes earlier timeframe.

Comparison of vehicle and crew scheduling results before and after critical trips handling is provided in table 1. By handling 0.2% of trips, we lowered the amount of necessary vehicles by 6.5% and the amount of crew shifts by 3.4%. Solutions provided by Kastor have as uniform shifts as possible. Within the original solution, Kastor distributed 997 trips between 31 vehicles and 59 crew shifts, where the lengths of all crew shifts should be as uniform as possible. By handling 2 trips, it was possible to cover 996 trips by only 29 vehicles and 57 crew shifts, which stepped up vehicle utilization as well as the amount of productive time of the crew. After critical trips handling, one vehicle covered on average 2.1 more trips than before, and one crew shift covered on average 0.6 more trips than before.

Measure	**Original count**	**Count after handling**
Number of trips	997	996
Number of vehicles	31	29
Number of crew shifts	59	57

Table 1: Results of vehicle and crew scheduling before and after critical trips handling[6]

[6]Data source: Saturday timetable of public transport company of Liberec and Jablonec nad Nisou

4.2 Discussion

In this section we comment both on the advancement of the theoretical result within scientific field as well as its benefits for practical usage. We also comment on the utilizability of the result in multi-depot and multi-vehicle scheduling extensions of vehicle scheduling problem.

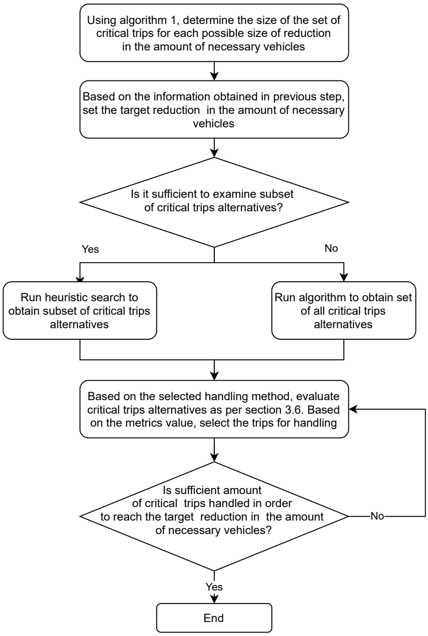

Figure 6: Flowchart summarizing the process behind the proposed optimization[7]

[7]Created by the author

Within figure 6, we summarize the process behind the proposed optimization. Based on a thorough literature review, we can conclude that the proposed optimization is novel and original. In literature, there is no evidence of setting the target reduction in the amount of necessary vehicles based on the size of sets of critical trips, together with provision of critical trips alternatives and methods for their further evaluation and handling.

The proposed optimization as well as all handling methods and their evaluation are designed in the way to pose the least negative effect on the passenger, while optimizing for the efficiency of resources utilization. When applied on an existing timetable, proposed optimization may lower the passenger's comfort. Even though it is designed in order to lower the passenger's comfort the least possible, and to the least possible amount of passengers, indisputably, it still can affect passenger's comfort. However, recently due to covid 19 disease many transportation companies implement reduced timetables.

These reduced timetables can be quickly evaluated by proposed algorithms and optimized to reduce the necessary amount of vehicles and possibly crew members to cover it. Similarly, when the standard timetable should be reinstated, the optimized timetable can be published instead.

4.2.1 Discussion on the multi-depot and multi-vehicle extensions

Both multi-depot and multi-vehicle extensions of vehicle scheduling are NP-hard, opposed to the polynomial complexity of single-depot vehicle scheduling, which we were focusing on within this work. We keep the problems of multi-depot and multi-vehicle scheduling extensions as open problems for future research. If the multi-depot or multi-vehicle problem is weakly conditioned, i. e. there are only few fixed requirements, then if we apply the proposed optimization and run the standard vehicle scheduling above the resulting timetable, there is high probability that the resulting savings in number of vehicles would be the same. In general, the probability lowers with more fixed requirements on the depots.

Furthermore, the proposed algorithm could be used as a part of genetical heuristic for the multi-depot or multi-vehicle scheduling task. When crossing the vehicle schedules in between the depots or vehicle types, the proposed algorithm can for each depot give suggestions on which of the tasks to be accomplished by another depot or vehicle type, so that within given depot or vehicle type, the tasks are executable by lower amount of vehicles. However, such an algorithm is beyond the scope of this dissertation and is kept as an open problem for future research.

5 Benefits of the dissertation

During the development of crucial parts of this work, public transport in the Czech Republic was afflicted by lack of vehicles and crew members, see SDP ČR (2018). Furthermore, the costs posed on vehicles and crew are generally the two highest costs of a public transport company. Therefore, the goal of this dissertation is to design an algorithm that enables to lower the amount of necessary vehicles, resp. crew members.

In this work, we provide overview and analysis of current knowledge in the field of vehicle and crew scheduling. Having identified a gap within current knowledge, we define goal of the dissertation, i. e. critical trips identification, evaluation and handling, and we provide the methods of achieving the goal.

We designed a comprehensive algorithm to identify critical trips to be removed from vehicle scheduling for lowering the number of vehicles necessary, see Pastirčáková and Šulc (2018), and presented it on a conference WMSCI 2018, where it won prize for the best paper of the session. As a part of this algorithm, we evaluate the size of the sets of critical trips, in order to set the target reduction in the amount of necessary vehicles, and we provide multiple critical trips alternatives.

Taking into account the applicability of the theoretical result in practice, we suggest further steps for handling the critical trips. We propose measures for critical trips evaluation based on which the transportation company can easily and quickly compare feasibility of critical trips handling and select the best alternative for handling, therefore enabling quick decision upon critical trips handling. We as well provide the evaluation of the impact of critical trips handling on the efficiency of used resources of a selected public transport company via a case study. We also make suggestions on the future research possibilities and share the open questions.

Based on a thorough literature review, we can conclude that the proposed optimization is novel and original. In literature, there is no evidence of setting the target reduction in the amount of necessary vehicles based on the size of sets of critical trips, together with provision of critical trips alternatives and methods for their further evaluation and handling.

The proposed optimization as well as all handling methods and their evaluation are designed in the way to pose the least negative effect on the passenger, while optimizing for the efficiency of resources utilization. When applied on an existing timetable, proposed optimization may lower the passenger's comfort. Even though it is designed in order to lower the passenger's comfort the least possible, and to the least possible amount of passengers, indisputably, it still can affect passenger's comfort.

Currently, due to covid 19 disease and the implemented restrictions which impact public transport especially by lowering the demand, many transportation companies implement reduced timetables. These reduced timetables can be quickly evaluated by the proposed method and optimized to reduce the necessary amount of vehicles and possibly crew members to cover it. Such approach to reduced timetables is advantageous not only for stepping up the efficiency of used vehicles and crew members, but also from the perspective of risk management, especially if some of the crew members get infected with covid 19 or are quarantined. And the fact that the creation of reduced timetable affects the passengers regardless any further optimization, it is the right time apply the optimization without passenger noticing any further change.

To summarize the benefits, the main advantages of finding and handling the critical trips are:

- Lowering the number of vehicles and crew members needed to cover the timetable.
- Increasing the efficiency of the used vehicles and crew.
- Decreasing the amount of drivers necessary for duty rostering.
- Utilization of knowledge of critical set of trips for optimizing the iterative process of timetabling $\rightarrow$ vehicle scheduling $\rightarrow$ timetabling for higher efficiency and lower operating costs.

References

Barnhart, C. et al. (1998). Branch-and-Price: Column Generation for Solving Huge Integer Programs. *Operations Research* 46, pp. 293–432.

Bertossi, A., P. Carraresi, and G. Gallo (1987). On some matching problems arising in vehicle scheduling models. *Networks* 17, pp. 271–281.

Bhandari, R. (1999). Survivable networks: algorithms for diverse routing. *Springer* 477, p. 46.

Bodin, L., B. Golden, et al. (1983). Routing and scheduling of vehicles and crews: the state of the art. *Computers and Operations Research* 2, pp. 63–211.

Bodin, L. and D. Rosen (1976). Estimation of the operating cost of mass transit systems. *Technical report, State University of New York*.

Bodin, L., I. Rosenfield, and D. Kydes (1978). UCOST: A micro approach to a transit planning problem. *Journal of Urban Analysis 5*.

Boyer, V., O. J. Ibarra-Rojas, and Y. Á. Ríos-Solís (2018). Vehicle and Crew Scheduling for Flexible Bus Transportation Systems. *Transportation Research Part B: Methodological* 112, pp. 216–229. DOI: 10.1016/j.trb.2018.03.016.

Bunte, S. and N. Kliewer (2009). An overview on vehicle scheduling models. *Public Transport* 1.4, pp. 299–317. ISSN: 1613-7159. DOI: 10.1007/s12469-010-0018-5.

Carpaneto, G. et al. (1989). A branch and bound algorithm for the multiple depot vehicle scheduling problem. *Networks* 19, pp. 531–548.

Cavique, L., C. Rego, and I. Themido (1999). Subgraph ejection chains and tabu search for the crew scheduling problem. *Journal of the Operational Research Society* 50.6, pp. 608–616. ISSN: 1476-9360. DOI: 10.1057/palgrave.jors.2600728.

Ciancio, C. et al. (2018). An integrated algorithm for shift scheduling problems for local public transport companies. *Omega* 75, pp. 139–153. ISSN: 0305-0483. DOI: https://doi.org/10.1016/j.omega.2017.02.007.

Costa, A., I. Branco, and J. Paixão (1995). Vehicle scheduling problem with multiple type of vehicles and a single depot. *Proceedings of the Sixth International Workshop on Computer-Aided Scheduling pf Public Transport* 34, pp. 115–129.

Daduna, J., M. Mojsilovic, and P. Schautze (1993). Practical experiences using an interactive optimization procedure for vehicle scheduling, pp. 37–52.

Daduna, J. and J. Paixão (1995). Vehicle Scheduling for Public Mass Transit — An Overview. 430.

Deepa, S. N. and S. N. Sivanandam (2010). *Introduction to genetic algorithms*. Springer.

Desaulniers, G., J. Lavigne, and F. Soumis (1998). Multi-depot vehicle scheduling problems with time windows and waiting costs. *European Journal of Operational Research* 494.

Desfontaines, L. and G. Desaulniers (2018). Multiple depot vehicle scheduling with controlled trip shifting. *Transportation Research Part B: Methodological* 113, pp. 34–53. ISSN: 0191-2615. DOI: `https://doi.org/10.1016/j.trb.2018.05.011`.

Desrochers, M. and S. François (1989). A Column Generation Approach to the Urban Transit Crew Scheduling Problem. *Transportation Science* 23, pp. 1–65.

Gavish, B. and E. Shlifer (1978). An approach for solving a class of transportation scheduling problems. *European Journal of Operational Research* 3, pp. 122–134.

Gross, L. and J. Yellen (2005). Graph Theory and Its Applications. *CRC Press*.

Guedes, P. C. and D. Borenstein (2018). Real-time multi-depot vehicle type rescheduling problem. *Transportation Research Part B: Methodological* 108, pp. 217–234. ISSN: 0191-2615. DOI: `https://doi.org/10.1016/j.trb.2017.12.012`.

Horváth, M. and T. Kis (2017). 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*. ISSN: 1613-9178. DOI: `10.1007/s10100-017-0489-4`.

Iliopoulou, Ch. and K. Kepaptsoglou (2019). Combining ITS and optimization in public transportation planning: state of the art and future research paths. *European Transport Research Review* 11, pp. 1–16.

Kliewer, N., V. Gintner, and L. Suhl (2006). Line change considerations within a time-space network based multi-depot bus scheduling model. *Proceedings of the Ninth International Workshop on Computer-Aided Scheduling of Public Transport*.

Lamatsch, A. (1992). An approach to vehicle scheduling with depot capacity constraints. *Derochers, M.I Rousseau, J.-M.* Pp. 181–195.

Mesquita, M. and J. Paixão (1992). Multiple depot vehicle scheduling problem: A new heuristic based on quasi-assignment algorithms. *Computer-Aided Transit Scheduling*. Ed. by Daduna J and Wren A, pp. 167–180.

Microsoft Corporation (2018). *C# language specification*. Accessed: 17. 4. 2021. URL: `https://docs.microsoft.com/dotnet/csharp/language-reference/language-specification`.

Orloff, C. (1976). Route constrained fleet scheduling. *Transportation Science* 2, pp. 149–168.

Oukil, A. et al. (2007). Stabilized column generation for highly degenerate multiple-depot vehicle scheduling problems. *Computers and Operations Research* 34, pp. 817–834.

Paixão, J. and I. Branco (1988). Bus Scheduling with a Fixed Number of Vehicles. *Computer-Aided Transit Scheduling*. Ed. by Daduna J and Wren A, pp. 28–40.

Palúch, S. (2001). Graph Theory Approach to Bus Scheduling Problem. *Studies of the faculty of management science and informatics* 9, pp. 53–57.

Palúch, S. and T. Majer (2017). KASTOR – a vehicle and crew scheduling system for regular bus passenger transport. *Transport Problems* T. 12, z. 1, pp. 103–110. ISSN: 1896-0596.

Pastirčáková, K. and J. Šulc (2018). Detection of minimal set of trips causing the necessity to use extra vehicle for vehicle scheduling problem. *Journal on Systemics, Cybernetics and Informatics: JSCI* 36.3, pp. 873–886.

Pels, E. and P. Rietveld (2000). Cost functions in transport. *Handbook of transport modelling, Elsevier*.

Ribeiro, C. and F. Soumis (1991). A column generation approach to the multiple-depot Vehicle scheduling problem. *Rep G-91-32, GERAD, Montreal*.

Saha, J. (1972). An algorithm for bus scheduling problems. *Operational Research Quarterly* 21, pp. 37–52.

Schmid, V. and J. Ehmke (2015). Integrated timetabling and vehicle scheduling with balanced departure times. *OR Spectrum* 37.4, pp. 903–928. ISSN: 1436-6304. DOI: 10.1007/s00291-015-0398-7.

SDP ČR (2018). Dopravní podniky představují další společnou kampaň. *Československý dopravák*. Accessed: 1. 9. 2019. URL: http://www.cs-dopravak.cz/zpravy/2018/9/17/dopravn-podniky-pedstavuj-dal-spolenou-kampa.

Shang, H. et al. (2019). Vehicle Scheduling Optimization considering the Passenger Waiting Cost. *Journal of Advanced Transportation* 2019, pp. 1–13.

Shen, Y., Z. Zeng, and Z. Wu (2017). Dynamic vehicle scheduling based on HTN. *2017 36th Chinese Control Conference (CCC)*, pp. 3066–3071. DOI: 10.23919/ChiCC.2017.8027828.

Song, C. et al. (2015). Improved Genetic Algorithm with Gene Recombination for Bus Crew-Scheduling Problem. *Mathematical Problems in Engineering*.

Tang, C. et al. (2018). Vehicle Scheduling of Single-Line Bus Service Using Operational Strategies. *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11. ISSN: 1524-9050. DOI: 10.1109/TITS.2018.2841061.

Van Lieshout, R. N. (2021). Integrated Periodic Timetabling and Vehicle Circulation Scheduling. *Transportation Science*. DOI: 10.1287/trsc.2020.1024.

Visentini, M. S. et al. (2019). Exploiting the timetabling flexibility in the context of the vehicle scheduling problem with heterogeneous fleet. *Pesquisa Operacional* 39, pp. 85–108. ISSN: 0101-7438.

Wren, A. and J. M. Rousseau (1995). Bus Driver Scheduling — An Overview. Ed. by Joachim R. Daduna, Isabel Branco, and José M. Pinto Paixão, pp. 173–187.

Yin, R. K. (2017). *Case Study Research and Applications: Design and Methods*. SAGE Publications. ISBN: 9781506336183.

Zhou, G. et al. (2020). Collaborative Optimization of Vehicle and Charging Scheduling for a Bus Fleet Mixed With Electric and Traditional Buses. *IEEE Access* 8, pp. 8056–8072. DOI: 10.1109/ACCESS.2020.2964391.

List of publications

[I] Pastirčáková K. and J. Šulc. Detection of minimal set of trips causing the necessity to use extra vehicle for vehicle scheduling problem. *Journal on Systemics, Cybernetics and Informatics: JSCI*, 16 (3), pp. 1–4, 2018.

[II] Šulc J. and K. Šulcová. Algorithm for on-line calendar text generation in public transport. *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Herl'any, Slovakia, pp 425–430, 2021.

[III] Šulc J. and K. Šulcová. Usage of RAPTOR for travel time minimizing journey planner. *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, Herl'any, Slovakia, pp 419–424, 2021.

[IV] Šulcová K. and J. Šulc. Detection of minimal set of trips causing the necessity to use extra vehicle for vehicle scheduling problem – a case study. *10th International Scientific Conference for Doctoral Students*, Košice, Slovakia, 2021.

Abstract

Vehicle scheduling problem addresses the task of assigning vehicles to cover all trips in a timetable. Minimum number of vehicles is determined by the number of trips in the peak hours of demand. In this work, we propose an approach to detect the minimal set of trips (critical trips), such that omitting them allows to lower the amount of necessary vehicles. We give overview of the size of the set of critical trips depending on the value of the target reduction in number of vehicles, in order to select appropriate target value. We provide methods for critical trips evaluation and handling. In a case study we show the usage of this algorithm on timetabled data of selected public transport company, where modification of 2 trips lead to reduction of both necessary vehicles and crew by 2.

Souhrn

Při zoběhování vozidel pokrýváme množinu všech spojů z jízdního řádu vozidly. Minimální počet potřebných vozidel je dán hlavně počtem spojů ve špičce, kdy je hustota spojů nejvyšší. V této práci navrhujeme způsob detekce minimálního počtu spojů (tzv. kritických spojů) takových, že jejich odebráním z úlohy zoběhování vozidel snížíme počet potřebných vozidel. Určíme velikost množiny kritických spojů pro každou hodnotu targetu redukce počtu vozů, vyšetřením čehož získáme výslednou velikost targetu redukce počtu vozů. Dále poskytujeme metody pro evaluaci a modifikaci kritických spojů. V case study aplikujeme navržený algoritmus na jízdní řád vybraného dopravního podniku, kde modifikací 2 spojů došlo ke snížení potřebných vozidel i osádek o 2.