UNIVERSITY
OF PARDUBICE
FACULTY
OF TRANSPORT
ENGINEERING

# Vehicle and crew scheduling optimization for public transport

# Optimalizace oběhů vozidel a osádek v městské hromadné dopravě

Dissertation

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

I declare:

The thesis entitled Vehicle and crew scheduling optimization for public transport is my own work. All literary sources and information that I used in the thesis are referenced in the bibliography. I have been acquainted with the fact that my work is subject to the rights and obligations arising from Act No. 121/2000 Sb., On Copyright, on Rights Related to Copyright and on Amendments to Certain Acts (Copyright Act), as amended, especially with the fact that the University of Pardubice has the right to conclude a license agreement for the use of this thesis as a school work under Section 60, Subsection 1 of the Copyright Act, and that if this thesis is used by me or a license to use it is granted to another entity, the University of Pardubice is entitled to request a reasonable fee from me to cover the costs incurred for the creation of the work, depending on the circumstances up to their actual amount.

I acknowledge that in accordance with Section 47b of Act No. 111/1998 Sb., On Higher Education Institutions and on Amendments to Other Acts (Act on Higher Education Institutions), as amended, and the Directive of the University of Pardubice No. 7/2019 Rules for Submission, Publication and Layout of Theses, as amended, the thesis will be published through the Digital Library of the University of Pardubice.

Dated in Pardubice on 10. 5. 2021

Kateřina Šulcová by own hand

Acknowledgements:

I would like to thank my supervisor Radovan Soušek for all his support throughout my studies.

I would especially like to thank my loving family for the love and support I have got over the years.

I would like to thank my husband for fruitful cooperation not only in the field of science.

I would like to thank the grandparents for taking care of the grandson, thanks to which they provided me with enough time and space for finishing this work.

Lastly, I would like to thank myself for having enough patience to finish this work.

Kateřina Šulcová

## Annotation

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.

## Keywords

vehicle scheduling, graph theory, shortest disjoint paths, case study

## Název

Optimalizace oběhů vozidel a osádek v městské hromadné dopravě

## Anotace

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.

**Klíčová slova**

oběh vozidel, teorie grafů, nejkratší cesta v grafu, case study

# Contents

# List of Figures

# List of Algorithms

# List of Tables

# Glossary

In the following text we use these symbols:

| | |
|---|---|
| $\mathbb{N}$ | Set of natural numbers, $\mathbb{N} = \{1, 2, 3, \cdots\}$ |
| $\forall$ | For each element |
| $G$ | Graph |
| $V$ | Set of vertices |
| $E$ | Set of edges |
| $v$ | Vertex |
| $e$ | Edge |
| $[t, u]$ | Ordered pair $t$ and $u$ |
| $A \times B$ | Cartesian product of sets $A$ and $B$ |
| $\binom{A}{k}$ | $k$-element subsets of mutually different elements of set $A$ |
| $\lvert A \rvert$ | Cardinality of set $A$, i. e. number of its elements |
| $m$ | Minimal number of vehicles needed to cover the timetabled trips |
| $k$ | Parameter which sets the target value for reduction of necessary number of vehicles |
| $T$ | Total amount of trips |
| $C^{(i)}$ | $i$-th set of critical trips, yielded by algorithm 1 |
| $S^{(i)}$ | $i$-th set of critical trips with alternatives added by swapping, yielded by algorithm 2 |
| $A^{(i)}$ | $i$-th set of critical trips with all alternatives, yielded by algorithm 3 |
| $U_l$ | Uniformity coefficient of line $l$ (and of each of its trip) on given weekday, see eq. 3 |
| $\delta_{ij}$ | Kronecker delta, where $\delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j \end{cases}$ |

# 1 Introduction

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 this work, we aim to design optimization algorithm in the field of vehicle and crew scheduling. We first provide overview and analysis of current knowledge within given field. Having identified a gap within current knowledge, we define goal of the dissertation, and we provide the methods of achieving the goal. We perform both basic research as well as modifications of existing algorithms to fit public transportation needs and requirements. We focus on practical usability of the theoretical result, therefore we provide the possibilities of dealing with the results within public transport company, and summarize it in a case study performed on a selected public transport company within Czech republic.

The dissertation is structured in chapters which guide the reader through introduction of the problem, current results in the field of vehicle and crew scheduling as well as current situation within Czech transport companies, the objective of stepping up the efficiency of public transport companies, which is a background objective of the dissertation, methods used to achieve the objective, discussion of the results, and a conclusion summarizing the dissertation.

In the second chapter we summarize and analyze current results available from the literature in the field of vehicle and crew scheduling. Within the analysis, we identify a gap to be covered by the dissertation, i. e. the critical trips identification and handling.

In chapter 3 we describe thoroughly the goal of the dissertation. We describe the concept of critical trips and the fundamental idea of lowering the amount of necessary vehicles, and therefore crew.

In chapter 4 we give overview of the methods which will be used for achieving the goal.

In chapter 5 we focus on critical trips identification. Here we define and analyze the problem of lowering the amount of necessary vehicles, and provide its solution by performing several transformations. We focus also on the practical usability of the results, therefore we provide multiple alternatives of critical trips for further considerations.

Within chapter 6 we propose critical trips evaluation and possibilities of their handling. We first suggest several methods for critical trips handling. Based on selected metrics, we evaluate the critical trips alternatives in order to obtain the most feasible alternative for handling.

In chapter 7 we comment both on the advancement of the theoretical result within scientific field as well as its benefits for practical usage. Also, we discuss the utilizability of the proposed optimization in multi-depot and multi-vehicle extensions of vehicle scheduling.

In chapter 8 we analyze the impact of critical trips handling within selected public transport company via a case study. We focus on evaluation of efficiency of vehicle and crew utilization.

To conclude, we briefly summarize the urge of the dissertation topic, objective of the dissertation, and the way of achieving it. We also comment on practical applicability of the results.

# 2 Current knowledge of vehicle and crew scheduling problem

Planning process in public transportation consists of several subsequent tasks. At first, we need to determine demand for transportation. Based on the demand we assemble public transport network, setup infrastructure of routes and stops for individual lines. Also, for each line we set individual trips, resulting in timetable data specifying the stop arrival and departure times of the trips. At this point, we need to cover the trips by vehicles and crew, aiming to use the resources efficiently. Two most important problems of this phase are vehicle and crew scheduling. 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 assigned vehicles. In following sections, we define the vehicle and crew scheduling problems, and we provide an overview of 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 J. M. Rousseau (1995) and Ciancio et al. (2018).

## 2.1 Vehicle scheduling

Vehicle scheduling is one of the most impactful planning problems for the public transport company. By lowering the number of necessary vehicles we usually lower the number of crew as well, and crew salaries together with costs posed on vehicles constitute the majority of public transport company costs, as per Pels and Rietveld (2000). However, the first algorithms for public transport using computer technology in the sixties were not focused on vehicle scheduling yet. First algorithms were focused on timetable data processing in order to timely produce the stop timetables and timetable booklets.

Having the timetable data available in electronic form, an idea of using the data for planning problems occurred, especially for vehicle scheduling. Due to low efficiency of the existing scheduling algorithms implementations as well as high costs and low performance of computer systems, the results were at first unsatisfactory. More than a decade later, taking advantage of rapid development of computer systems, the software planning systems for vehicle scheduling started to appear and gain on importance, for example Wren (1981), J. Rousseau (1985) and Daduna and Wren (1988). Since then, there has been further 30 years of development, bringing further applicable results. In the following text we give overview of vehicle scheduling problem, focusing on the different needs of different transport companies. Several alternations of the problem are presented along with their solution methods.

### Problem definition

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).

Before we further describe the relevant results of vehicle scheduling for single and multiple depot, we should realize that the selection of the problem does not have to be dependent on the actual number of depots. If multiple depots operate independently on multiple independent areas which are free of intersections, the problem can be separated into corresponding number of independent single depot vehicle scheduling subproblems. If the transit areas naturally intersect, the problem should be handled as a multiple depot vehicle scheduling problem to obtain the most relevant solution taking into account the objective function value.

### 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. In this model we define a partially ordered set of trips, where ordering exists for a pair of trips $[t_1, t_2]$ if and only if ending station of $t_1$ is the starting station of $t_2$ and arrival time of $t_1$ is earlier than departure time of $t_2$. Obviously, 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.

## 2.2 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, where the vehicle schedule is split into two pieces of work which are satisfied by one crew member operating the morning piece of work and second crew member operating the afternoon piece of work, see Palúch and Majer (2017). This relationship originates in habits established in Czechoslovakia, which are still very persistent, especially in small and medium sized public transport companies. Vehicle and crew scheduling system Kastor by Palúch and Majer (2017) is satisfying legal obligations and collective bargaining agreements constraints typical for Czech and Slovak republic, as well as the above mentioned vehicle-crew relationship. Kastor is a multiobjective heuristic which has 4 main objectives, namely minimization of the number of vehicles and crew, minimization of dead-heading trips and minimization of line changes. In its default mode, it optimizes over following constraints:

- Vehicle schedule must start and end at the same depot (or predefined set of depots)
- Vehicle schedule can be considered either as 1 piece of work for one crew, or can be split into two pieces of work for two crew members
- All vehicle pieces of work have to satisfy the safety break condition, i. e. within each time interval of 270 minutes there has to be at least 30 minutes of break time, where the break time can be split into at most 3 pieces, each one with minimal duration of 10 minutes
- All vehicles have to visit the central depot at least once a day

- All crew shifts have to include a meal break
- There is a limit for maximal driving time of the crew
- The time lengths of all crew shifts has to be as uniform as possible

Kastor is one of the most widespread systems for vehicle and crew scheduling within Czech and Slovak republic, it is used within municipal or regional transport in following cities: Třinec, Jablunkov, Havířov, Tachov, Uherské Hradiště, Most and Litvínov, Prachatice, Strakonice, Nymburk, Lysá nad Labem, Milovice, Prievidza, Martin - Vrútky, Považská Bystrica, Trenčín and others.

## 2.3 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.

Public transport companies differ in their characteristics. Some public transport companies operate in the manner of interval transport, which is very organized and regular, see figure 1 for example of such characteristic (figure 1 is an artificially created example of interval transport characteristics). Other public transport companies operate in the manner of non-interval transport, example of its characteristic can be seen in figure 2, which is an example taken from weekday timetable of public transport company of Liberec and Jablonec nad Nisou.

We target mainly transport companies with low number of narrow steep peaks in their characteristics, a good example of which is figure 2. We aim to identify the minimal set of trips, such that omitting them from the vehicle scheduling problem lowers the amount of vehicles neces-

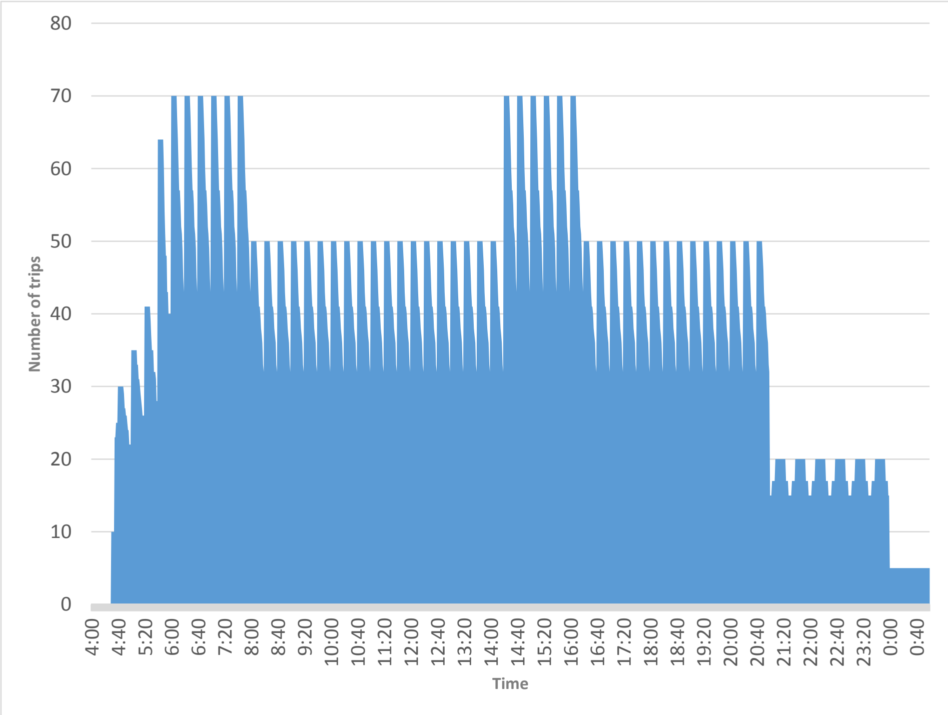


Figure 1: Number of trips for public transport company operating interval transport[1]

[1]Created by the author

sary to cover the timetable. Let us denote such trips as *critical trips*. For more rigorous problem definition using graph theory, refer to chapter 4.

Based on figure 2, we are able to identify the lower bound of the number of necessary vehicles covering all the trips to be 79, which arises during the morning peak at 7:48. Therefore it may seem, that for lowering the amount of necessary vehicles we should focus on the peak at 7:48. However, if we performed vehicle scheduling based on the timetabled trips, we can for each time moment obtain also the amount of vehicles necessary to cover the trips, see figure 3. In figure 3 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. As per figure 3, peak at 7:48 can be covered by 88 vehicles, whereas there are 90 vehicles necessary to cover the peak at 14:30.

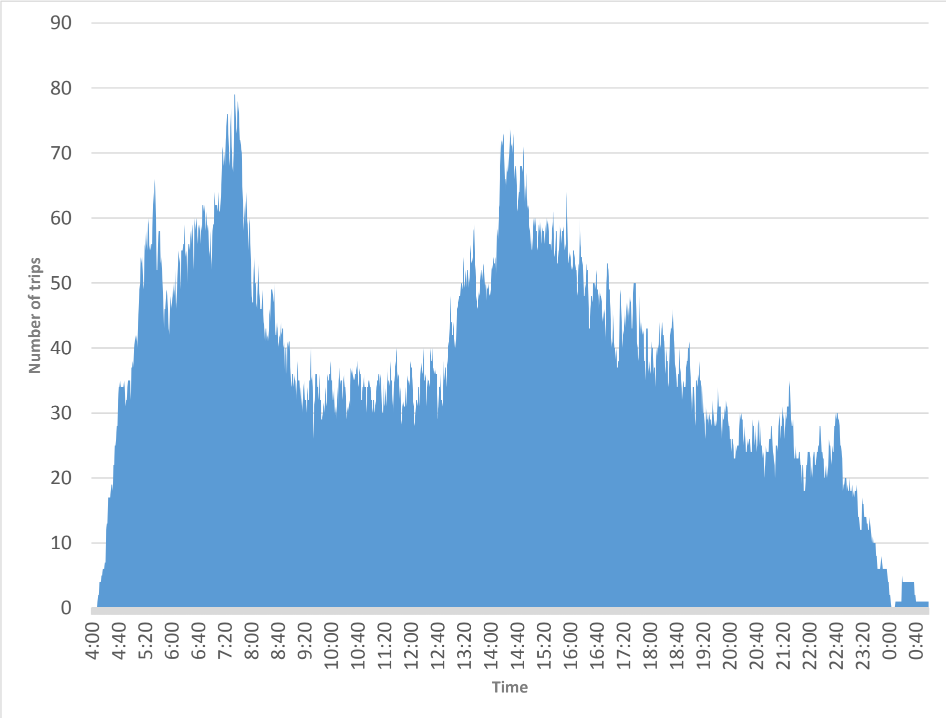


Figure 2: Number of trips for public transport company operating non-interval transport[2]

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

Using figure 3, we identified the peak in vehicle usage. However, based on the information from figure 3, 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 chapters.

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.

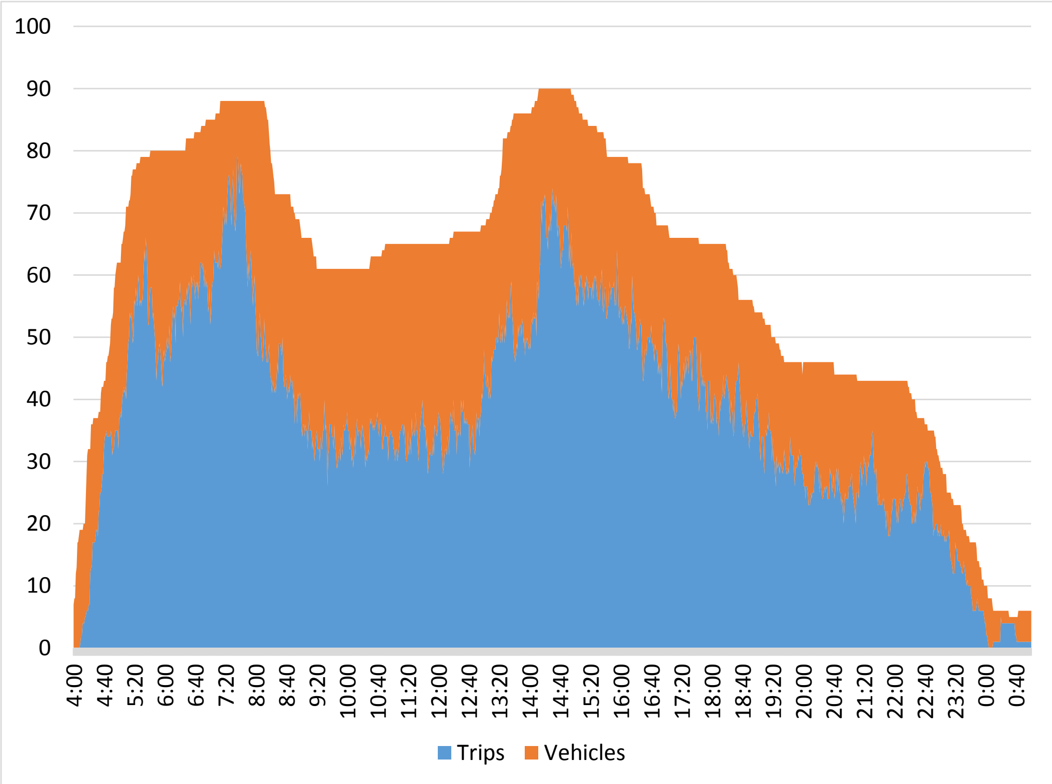


Figure 3: Number of trips and vehicles for public transport company operating non-interval transport[3]

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

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.

# 3 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.

# 4 Methods and 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 chapter we provide the methods of achieving the goals. 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.

## 4.1 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 undirected, directed and oriented graph. For a comprehensive overview of graph theory and its applications see Gross and Yellen (2005).

*Undirected graph* is an ordered pair $G = (V, E)$, where $V$ is a finite set of *vertices* and $E \subseteq \binom{V}{2}$ is a set of *edges*.

*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*.

For an undirected graph it is essential whether there exists an edge between two vertices $v_1, v_2$. In a directed graph we are adding further complexity by the direction of the edge, 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.

*Bipartite graph* is a graph whose vertices can be divided into two disjoint sets $V$ and $W$ such that every edge connects a vertex in $V$ to a vertex in $W$.

*Path* in a graph is a sequence of edges which connect a sequence of mutually distinct vertices. In a directed graph, a path has an added restriction that its edges are all directed in the same direction.

Several optimization methods using graph theory have been applied for vehicle scheduling optimization, for example assignment model by Orloff (1976) and quasi-assignment model by Gavish and Shlifer (1978), where bipartite graphs are used for modeling the schedules. Also network flow model was introduced in Bodin, Golden, et al. (1983), where minimum cost flow problem needs to be solved.

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 4 shows an example such graph, containing 8 trips.

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.

Generally, longest disjoint path problem would be an NP-hard problem. However, there is a clear orientation of the edges from left to right, as per the hidden time axis. In such case,

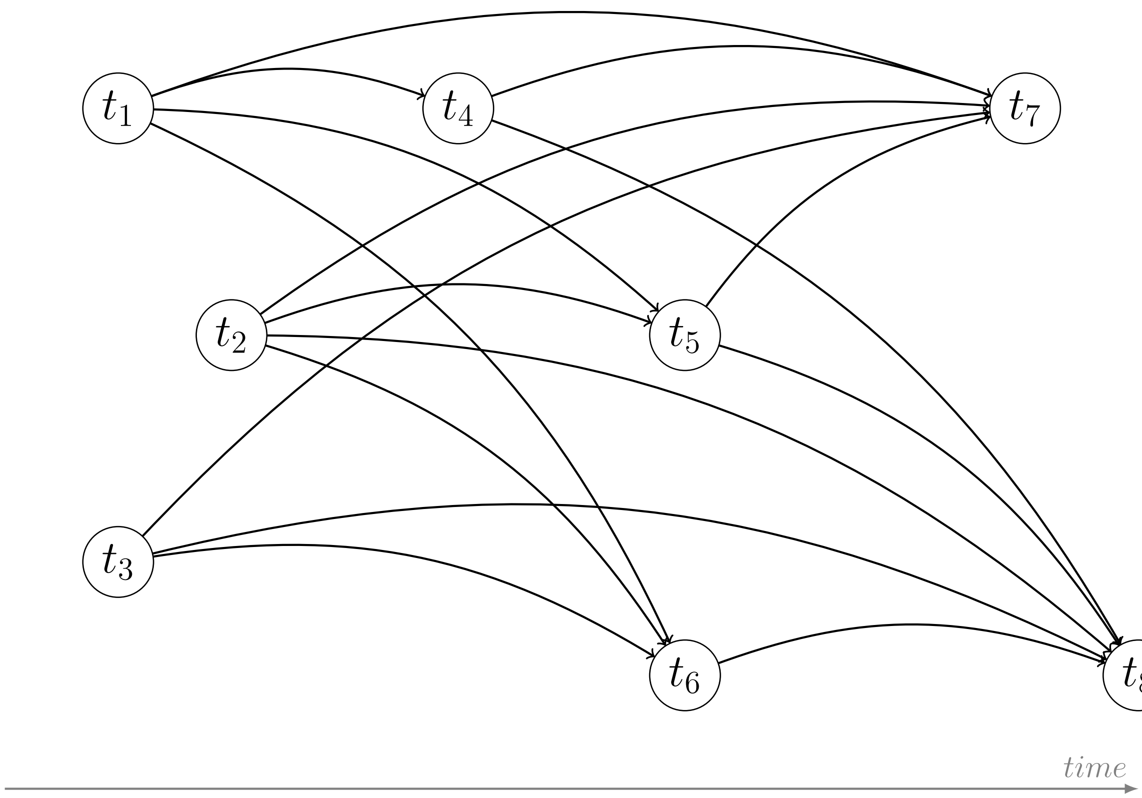


Figure 4: Graph for searching the set of longest disjoint paths[4]

[4]Created by the author

there exists a solution in polynomial time. However, there is no general solution for set of longest disjoint paths problem. Our main goal is to transform the problem into a problem with polynomial complexity.

In order to provide critical trips alternatives, we propose also heuristic algorithm inspired by genetic algorithms, which are described within next section. Having found the set of critical trips and their alternatives, we propose metrics for their evaluation to support the decision making process. We also analyze the impact of usage of such algorithm on the efficiency of vehicle and crew utilization within public transport company via a case study conducted on a selected public transport company. Details on case study are provided at the end of this section.

## 4.2 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. The quality of each generated solution is quantified by fitness function. The goal of genetic algorithm is to generate solutions with highest value of fitness function. For the purposes of this work, we focus specifically on crossover operator.

### 4.2.1 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.

## 4.3 Case study

A case study, see 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. Therefore, in this section we describe a design of case study which will be used for applying the method proposed in this dissertation within real world context.

Formally, case study consists of several sections. It starts with introduction which describes scope and purpose of the study, i. e. the goal. Next, there is a method section, in which we describe the methods used within the study. Main findings of the case study are covered within discussion section. In our case, discussion will be covering implementation of the proposed method in the real world context. Lastly, in conclusion section we summarize 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.

# 5 Critical trips identification

Vehicle scheduling is a widely studied problem having many subsequent questions and solutions found. As a vehicle is usually the most expensive asset, transportation companies tend to minimize the number of vehicles. The problem of minimizing the number of vehicles needed to satisfy the timetable schedule can be solved for example by vertex covering or graph coloring, maximum flow and many more, for overview see section 2.1.

We can optimize the costs even further within the vehicle scheduling problem, minimizing not only the costs of used vehicles, but also the costs of death trips, which can be solved by means of linear programming.

Outputs of both of these problems 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 5 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 following section, we transform this problem into the minimum cost maximum flow problem.

As we show subsequently in section 5.2, a slight modification of the following algorithm will also yield the block (or multiple blocks) with minimal sum of lengths of trips, while covering the timetable by $m$ blocks in total. Following algorithms were published in Pastirčáková and Šulc (2018).

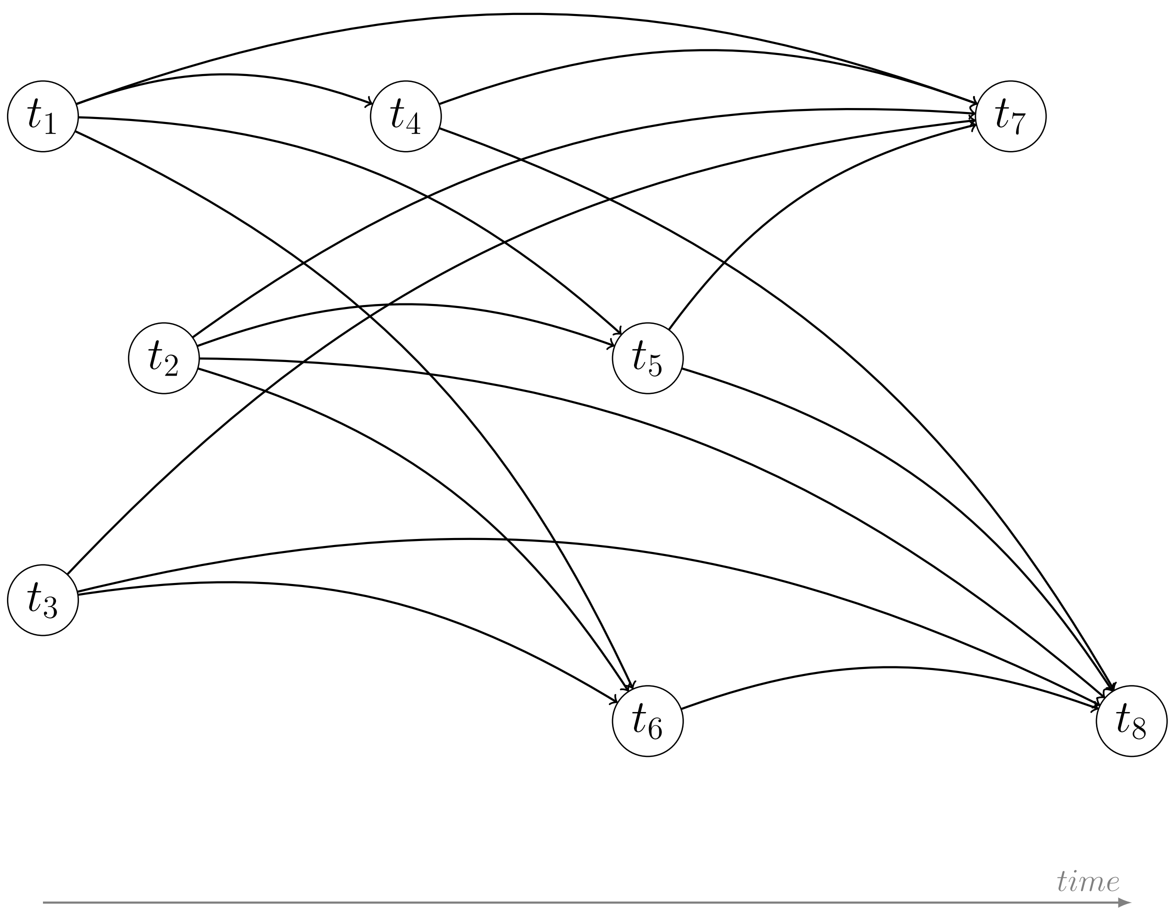


Figure 5: Graph for searching the $m - k$ longest disjoint paths[5]

[5]Created by the author

## 5.1 Transformation into minimum cost maximum flow problem

As stated in previous section, for a given timetable there exist several algorithms that can quickly find the minimal number $m$ of vehicles needed to cover all the trips within the timetable, one of them being the minimum cost maximum flow algorithm. The minimum cost maximum flow algorithm which we describe within this section constraints the maximal flow by $m - k$, i.e. the desired number of vehicles to be used.

Within the flow network, we need to ensure that each trip is covered by a vehicle at most once. Therefore each trip is represented as a triple of input vertex $t$, output vertex $t'$ and an edge from $t$ to $t'$ with capacity 1. 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 $time(t') - time(t) - 1$. Similarly to the graph in figure 5, an edge of capacity 1 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 $u$.

We define a sink $n$ such that $time(n) > time(t')$ $\forall$ trip $t$ within the timetable. Then for each trip $t$ we create an edge from $t'$ to $n$ with capacity 1 and cost $time(n) - time(t')$. Similarly, we define a source $s$ such that $time(s) < time(t)$ $\forall$ trip $t$ within the timetable. Then for each trip $t$ we create edges from $s$ to $t$ with capacity 1 and cost $time(t) - time(s)$. To be able to limit the overall capacity of such network, we further define a supersource $r$ which is connected by an edge to the source $s$, cost of the edge being zero, and the capacity $c$ being a variable dependent on the desired number of vehicles to be used by the transportation company, i.e. $c := m - k$.

After solving the minimum cost maximum flow problem for this network, see for example Goldberg and Tarjan (2018), the trips whose edges were not included in the solution are the minimum number of critical trips. Minimality of the number of critical trips can be easily proven by contradiction, as each trip included in the maximum flow minimum cost solution lowers the overall cost by one.

For illustration of the network described above we show in figure 6 the simplified version of the network containing 4 trips. To simplify and shorten the labels of edges, we denote $time(t) -$

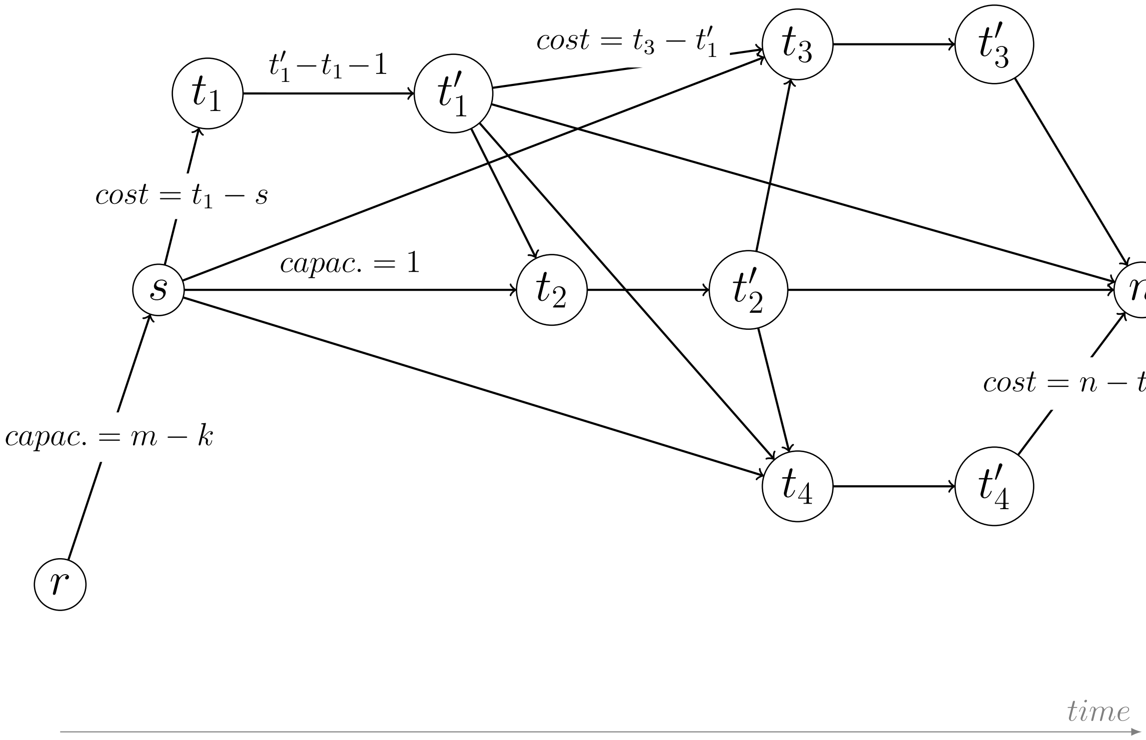


Figure 6: Network for minimum cost maximum flow problem[6]

[6]Created by the author

$time(s)$ only as $t - s$, and similarly $time(t') - time(t) - 1$ only as $t' - t - 1$.

### Conclusion of minimum-cost flow approach

The algorithm described within this section yields the set of critical trips. However, there is a necessary prerequisite of solving the standard vehicle scheduling problem and finding the minimal number $m$ of vehicles necessary. Furthermore, the planners have to decide upfront what the desired number of vehicles $m - k$ to be used within the vehicle scheduling is, or they have to run the algorithm for multiple different values of $k$. Such an approach is usable but certainly would not be a best practice. In the next section we modify the problem to obtain a more elegant solution, which gives us the overall picture for all vehicle fleet sizes at once.

## 5.2 Transformation into shortest disjoint path problem

In previous section, the capacities of all edges within the network were set to 1, except for the artificially added edge from supersource to source, which sets the maximum capacity of the network. However, by transforming the problem into the shortest disjoint paths problem, there will be no necessity for the supersource, all the capacities can also be dismissed and we keep only the costs of the edges.

Similarly as in previous section, 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 $t'$ with cost 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 $t$, having the cost of $time(u) - time(t')$.

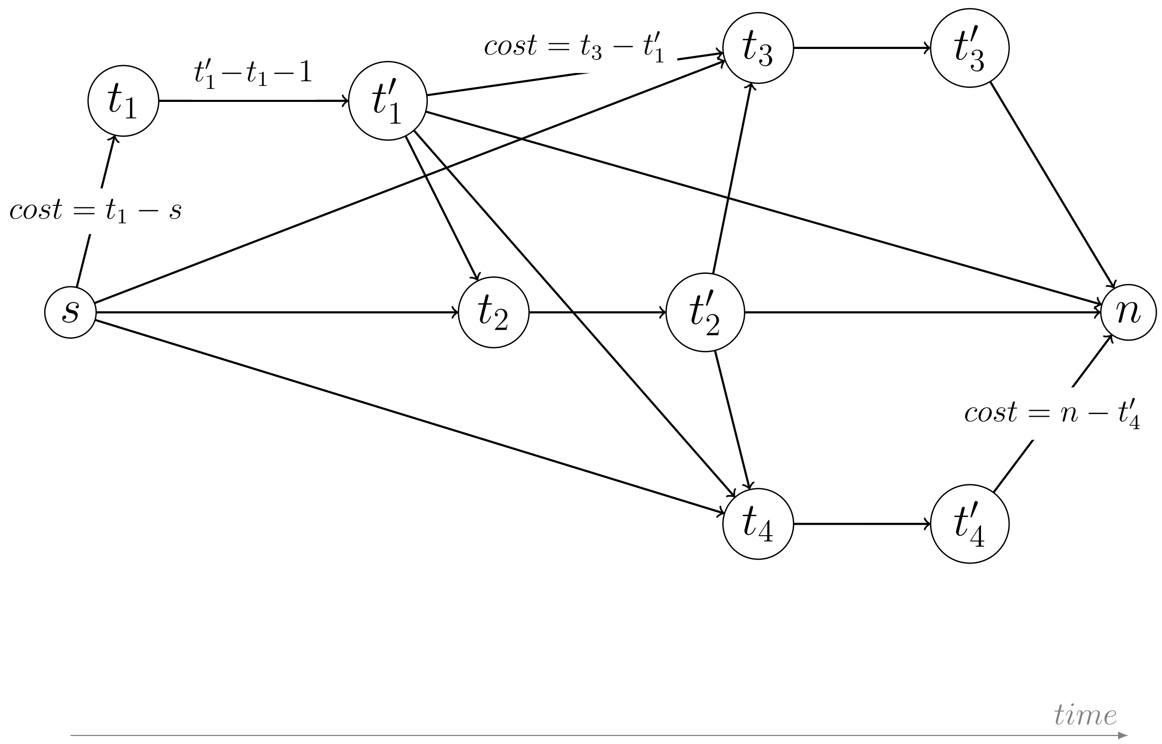


Figure 7: Graph for the shortest disjoint path problem[7]

[7] 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 7 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 $s_1^{(i)}, s_2^{(i)}, \ldots s_i^{(i)}$ are formed by the paths $p_1, p_2, \ldots p_i$, 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 $s_1^{(i)}, s_2^{(i)}, \ldots s_i^{(i)}$;
>
> 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 $s_1^{(i)}, s_2^{(i)}, \ldots s_i^{(i)}$;

**end**

**Algorithm 1:** Critical trips identification

Even though the algorithm 1 contains a $while$ cycle, from the vehicle scheduling problem we know that there will be exactly $m$ iterations, because with $m$ disjoint paths we can cover all the trips. Proof of correctness of the algorithm is similar to the proof from section 5.1.

## 5.3 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. Obviously, one of the most favourable situations for critical trips handling occurs for maximal $k$ such that $\left|C^{(m-k)}\right| = k$, because in this situation we need to remove exactly $k$ trips from vehicle scheduling to achieve the reduction in the number of necessary vehicles by $k$.

After removal of critical trips from vehicle scheduling problem, the $m-k$ shortest disjoint paths $s_1^{(m-k)}, s_2^{(m-k)}, \ldots s_{(m-k)}^{(m-k)}$ yielded by algorithm 1 form a feasible schedule for $m-k$ vehicles. However, it is advisable for public transport companies to run their proprietary vehicle scheduling algorithm over the timetabled data with critical trips removed, as the $m-k$ shortest disjoint paths tend to be highly non-uniform.

Even though algorithm 1 yields specific set of critical trips, its main strength lies in quick evaluation of the amount of critical trips for different $k$. 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.

## 5.4 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 $s_1^{(m-k)}, s_2^{(m-k)}, \ldots s_{(m-k)}^{(m-k)}$.

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

### 5.4.1 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 $s_1^{(m-k)}, s_2^{(m-k)}, \ldots s_{(m-k)}^{(m-k)}$. 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. Previous description is summarized in algorithm 2.

**Input**:

Output of algorithm 1;

Value of parameter $k$, i. e. the desired reduction in the number of necessary vehicles;

**Steps**:

**for *each*** $c \in C^{(m-k)}$ **do**

    $H_c^{(m-k)} := \{c\}$;

    $H^{(m-k)} := \emptyset$;

    **for** $i = 1, \dots, m-k$ **do**

        **if** *exists exactly one trip t within path* $s_i^{(m-k)}$ *which overlaps with trip c* **then**

            **if** *swapping trip t and c creates feasible vehicle schedule* **then**

                insert $t$ to $H_c^{(m-k)}$;

            **end**

        **end**

        **if** *none trip within path* $s_i^{(m-k)}$ *overlaps with trip c* **then**

            denote $u$ the nearest previous trip to $c$ within $s_i^{(m-k)}$

            **if** *swapping trip u and c creates feasible vehicle schedule* **then**

                insert $u$ to $H_c^{(m-k)}$;

            **end**

            denote $v$ the nearest later trip to $c$ within $s_i^{(m-k)}$

            **if** *swapping trip v and c creates feasible vehicle schedule* **then**

                insert $v$ to $H_c^{(m-k)}$;

            **end**

        **end**

    **end**

    insert $H_c^{(m-k)}$ to $H^{(m-k)}$;

**end**

**Algorithm 2:** Swapping procedure for obtaining critical trips alternatives

### 5.4.2 All critical trips alternatives

Previously described heuristic approach may not yield all interchangeable critical trips. To obtain comprehensive sets containing all the 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. Previous description is summarized in algorithm 3.

The number of substractions of $\frac{1}{T+1}$ from the cost of a trip is limited by the total amount of trips $T$. Therefore, the sum of all substracted weights is in absolute value strictly lower than 1, and thus it does not structurally change the behavior of algorithm 1, and the proof of correctness would be similar to algorithm 1.

If the number of repetitive runs of algorithm 1 within algorithm 3 is high, it significantly increases the computational complexity. Here it is appropriate to note that one of the advantages of algorithm 1 for critical trips identification is its GPU parallelization possibility. The essential part of the algorithm, i. e. Bellman-Ford shortest path algorithm has been parallelized for GPU by Sengo and Garland (2020). Therefore, there is a need to finalize the GPU implementation of further steps of the algorithm, which should be feasible. However, it is out of scope of this dissertation and is kept as a topic for future research.

**Input**:

Output of algorithm 1;

Value of parameter $k$, i. e. the desired reduction in the number of necessary vehicles;

**Steps**:

$i := 0$;

$C_0^{(m-k)} := C^{(m-k)}$ ;

$A^{(m-k)} = \emptyset$;

Set directions of all edges to the original ones, i. e. from $s$ to $n$;

Set costs of all edges to their absolute values;

**while** *exists trip* $c \in C_i^{(m-k)}$ *with cost equal to* $time(c') - time(c) - 1$ **do**

  insert $C_i^{(m-k)}$ to $A^{(m-k)}$;

  **for** ***each*** $c \in C_i^{(m-k)}$ **do**

    set the cost of trip $c$ to $time(c') - time(c) - 1 - \frac{1}{T+1}$;

  **end**

  $i := i + 1$;

  run algorithm 1 upon the graph with updated weights, and denote $C^{(m-k)}$ as $C_i^{(m-k)}$ ;

**end**

**Algorithm 3:** Algorithm for obtaining all alternatives to critical trips

**Conclusion of shortest disjoint path finding**

In this section we provided quick and elegant solution to the original problem. By providing the solutions for each possible size of vehicle fleet we give the planners full information, which is necessary base for the decision on how many vehicles it is reasonable and cost effective to use. We also described algorithm for yielding comprehensive set of all critical trips alternatives.

## 5.5 Modification to obtain critical trips of minimal sum of lengths

A public transportation company which operates both high frequency short trips within a city and lower frequency suburban transport, where much longer trips occur, might question whether all the trips are interchangeable. In algorithm 1 we consider all trips equal, while it may be useful to be able to interchange multiple shorter trips instead of one long trip as critical ones to be excluded from vehicle scheduling, or vice versa.

Therefore, we slightly modify the algorithm to obtain critical trips of minimal sum of lengths. To achieve it, we need to modify the costs of trips in the following way. Currently both within the minimum cost maximum flow problem and shortest disjoint path problem, for each trip $t$ the cost of edge $[t, t']$ equals to $time(t') - time(t) - 1$. If we set the costs of edges $[t, t']$ to $0$, then the identified critical trips will be of minimal overall length, as each minute of a trip is discounted. Then, both within the minimum cost maximum flow problem and shortest disjoint path problem the solution will yield the blocks with longest trips, while leaving out the critical trips with lowest overall sum of running times. Proof would be similar to the one already shown, by dispute.

Further modifications of the graph are possible, we can modify the costs of edges $[t, t']$ to any interpolated value between 0 and $time(t') - time(t) - 1$, which then optimizes for certain combination of minimal number of trips and shortest time of the critical trips.

# 6 Critical trips evaluation and handling

Within this chapter, 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.

For the purpose of this chapter we do not consider the modification to obtain critical trips of minimal sum of lengths from section 5.5, as within the modification, we already used a specific metric – length of a trip. Therefore, as critical trips we consider the minimal number of trips such that when they are removed from schedule, the minimal number of vehicles needed to satisfy the schedule lowers by $k$. Within section 5.3, we gave an algorithm which can quickly help us to decide, what is the desired reduction in number of vehicles. Mutually interchangeable alternatives for critical trips were provided within section 5.4. Here, we are evaluating the alternatives in order to select the ones that are the most feasible for handling.

## 6.1 Critical trips handling possibilities

As we are evaluating the alternatives in order to select the ones that will have least complicated handling, let us outline the considered handling possibilities:

- Rescheduling of the trip to a different time frame
- Outsourcing
- Trip cancellation
- Covering of the trip by a backup vehicle and a temp crew

Trip cancellation or rescheduling is not usually in the core competency of the transportation company, viability of this option has to be negotiated with the transportation authority, which orders the transportation.

However, there are certain factors that we should consider before starting the negotiation with authorities. For example, a school or factory buses are almost impossible to cancel, and for rescheduling, there would be probably a very tight time window. Therefore, the transportation company needs to perform strategic preprocessing of different alternatives of critical trips and choose the most viable alternative to be presented to the authorities.

## 6.2 Metrics for evaluation of selected handling possibilities

We can support the decision making by metrics presented within this section, where we propose for each handling possibility a metric for trip evaluation. Note that the list of handling possibilities as well as the list of metrics may not be complete. Having a set of requirements within specific environment, it can be extended and combined to satisfy the requirements.

### 6.2.1 Rescheduling of the trip to a different time frame

When considering trip shifting to a different time frame, it is preferable to select trips from the lines with non-uniform differences between consecutive trips. For example trips within lines which operate in the manner of interval transport are highly unsuitable for rescheduling. Therefore, we define the uniformity coefficient, which evaluates uniformity of the sequence of time spans between consecutive trips of a selected line.

Let us define Kronecker delta as

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases} \tag{2}$$

Let $t_1, t_2, \ldots, t_n$ be the ordered sequence of start times of trips within selected line $l$ on given weekday. Then the sequence $d_1 = t_2 - t_1,\ d_2 = t_3 - t_2,\ \ldots,\ d_{n-1} = t_n - t_{n-1}$ is the sequence of time differences between consecutive trips. Then the uniformity coefficient of line $l$ (as well as each of its trip) on given weekday is defined as

$$U_l = \frac{\sum_{k=1}^{n-2} \delta_{d_k d_{k+1}}}{n-2}, \tag{3}$$

i. e. it counts the number of value changes in the sequence $d_1,\ d_2,\ \ldots,\ d_{n-1}$, and normalizes the result by dividing it by the amount of possible changes. The value of $U_l$ ranges from 0 to 1, where $U_l = 1$ for a line where $d_1 = d_2 = \cdots = d_{n-1}$ and $U_l = 0$ for a line where $d_1 \neq d_2, d_2 \neq d_3, \ldots, d_{n-2} \neq d_{n-1}$.

We evaluate the uniformity coefficient of each critical trip, and select those with low enough value of uniformity coefficient as candidates for rescheduling. Then, the maximal time shift window needs to be set for each candidate.

It is advisable not to use the uniformity coefficient for lines with number of trips less than or equal to 3 on the given day. If a critical trip is part of such line, then we recommend to evaluate manually the possibility of trip rescheduling, and its maximal time window. Also note that we need to be conscious about the special cases of time difference sequence like $7, 8, 7, 8, 7, \ldots$, for which the desired uniformity coefficient value is 1. Therefore, if these special cases exist within the timetable, we need to redefine the Kronecker delta so that it considers not only equality, but also equality with small difference.

When the maximal time shift window is set for each candidate for rescheduling, we can use the trip shifting algorithm proposed in Schmid and Ehmke (2015) for trip rescheduling to a different time frame within selected time window, where the objective is to lower the amount of necessary vehicles. Even though trip shifting is NP hard, we are at least lowering the computational complexity of the task by setting the variable time windows only for selected critical trips, keeping all the other trips' times fixed.

In case that the trip gets shifted to a very tight time window with consecutive trip within same line, trip merging can be considered as well, where two consecutive trips are merged into one with sufficient capacity. Even in the case where trip rescheduling doesn't yield feasible schedule, if trip merging is considered upfront, it might still provide feasible solution, as the input for trip rescheduling algorithm is modified timetable with merged trip and specific time window.

### 6.2.2 Outsourcing

When considering outsourcing, the main parameter is its price. Usually, there are only few companies able to handle the outsourced trip. If their tariffs are known, then we compare the outsourcing prices based on the tariffs. Otherwise, the price of outsourcing is usually strongly dependent on the length of the outsourced trip. Therefore, the modification to obtain critical trips of minimal sum of lengths from section 5.5 should be considered.

Note that especially for trips which have low average occupancy we can incorporate also on-demand transport services, see Aarhaug (2016), for example carsharing or taxi, as a flexible alternative to the standard and more rigid public transport system. The usage of taxis for routes and times of low demand increases the efficiency of public transport, see Last Mile (2017). For example, it allows better and less costly operation for services that would otherwise be loss-making.

### 6.2.3 Trip cancellation

When considering trip cancellation, we need to be conscious of transport demand in order to make the least possible reduction of passenger comfort and satisfaction. The demand can be approximated by the average trip occupancy. Therefore, for trip cancellation we consider the trips with low average trip occupancy. However, especially in this case we should consider the possibilities of on-demand transport services described in section 6.2.2.

### 6.2.4 Covering of the trip by a backup vehicle and a temp crew

In case that there is a backup vehicle and temp crew available within the transport company, it can be utilized to cover the critical trips, especially in case that none of the previous options are feasible.

## 6.3 Summary of critical trips evaluation and handling

Within previous section, we suggested several possibilities for critical trips handling, and respective metrics for critical trips evaluation, in order to select the most feasible alternatives for handling. Based on specific requirements of given public transport company, further metrics can be incorporated. Also, multiple metrics can be combined into one objective function, or multiple-criteria decision analysis can be used for critical trips evaluation within specific environment.

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, 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
- 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.

# 7 Discussion

In this chapter, 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.

Within figure 8, 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, 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. 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. 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. Similarly, when the standard timetable should be reinstated, the optimized timetable can be published instead.

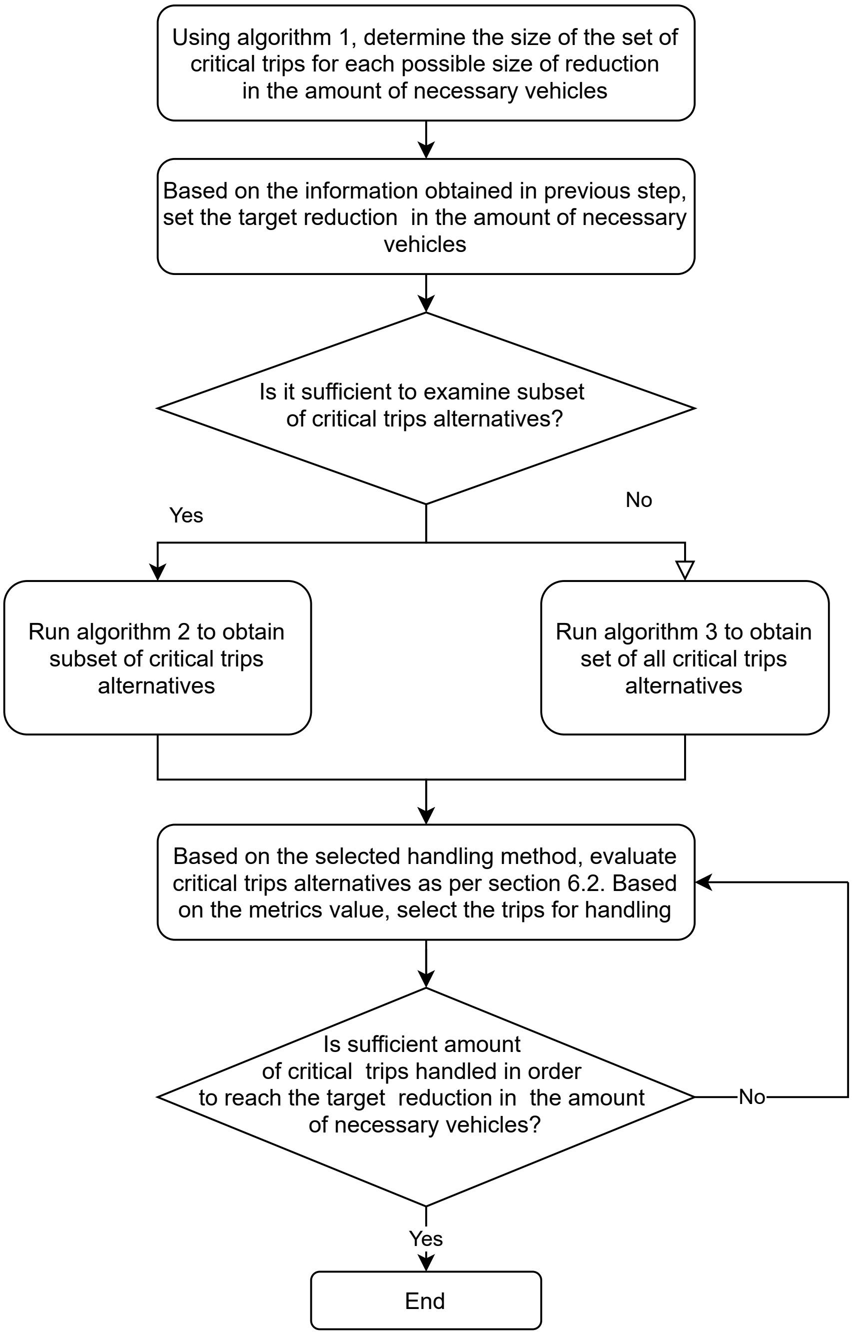


Figure 8: Flowchart summarizing the process behind the proposed optimization[8]

[8]Created by the author

## 7.1 Discussion on the multi-depot and multi-vehicle extensions

As stated in chapter 2, 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 problem of multi-depot and multi-vehicle scheduling extensions as open problems for future research.

### 7.1.1 Multi-depot extension

Let us first focus on the easiest and least constrained version of multi-depot vehicle scheduling, in which there is the extra requirement on numbers of vehicles available in each depot are not exceeded, and each vehicle schedule must start and end at the same depot. Within public transport we can assume that the depots are relatively close to each other, and that cost of dead-heading trip between the depots is significantly lower than the cost of extra vehicle and crew operation. If the assumption is correct, then the proposed algorithm can be utilized without further loss of generality for lowering the necessary amount of vehicles as a preprocessing step for the multi-depot vehicle scheduling task.

In case that there are further requirements posed on the depots, for example some of the trips can be performed by vehicles from given depot, then there is no guarantee that this requirement will not create the need for extra vehicle over the resulting set of trips provided by the proposed algorithm. If the multi-depot 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 scheduling task. When crossing the vehicle schedules in between the depots, the proposed algorithm can for each depot give suggestions on which of the tasks to be accomplished by another depot, so that within given depot, 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.

### 7.1.2 Multi-vehicle extension

In case of multi-vehicle extension of the vehicle scheduling problem, utilizability of the proposed algorithm is even more complex than in the multi-depot extension.

Similarly as described in previous section, it can be used for weakly conditioned problems, but there is no guarantee that the resulting savings for multi-vehicle problem would be the same as for single-vehicle one. Also, the proposed algorithm could be used as a part of genetical heuristic for the multi-vehicle scheduling task, with the main idea similar to what was described within previous section.

There is one case, however, where the proposed algorithm for critical trips identification can be straightforwardly used – if there is not enough vehicles of given type within a fleet, or there is exactly the number of vehicles that is needed, which gives very little space for vehicle maintenance. In such case, we can use the proposed optimization only on the trips which are supposed to be covered by the given vehicle type.

# 8 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. 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, the summary is provided within table 1. We can observe from table 1 that most of the routes contain trips with multiple different durations and multiple different start stops. In order to compute uniformity coefficient for each route within table 1, we focused on timetables of one central station within Liberec, which is Fügnerova station. All the trips except for exactly three trips from routes 34, 97 and 98 run through Fügnerova station. Each of the routes 34, 97 and 98 contain only one trip, therefore, the uniformity coefficient is undefined by default. For routes which have both directions defined, the uniformity coefficient is computed for one direction only.

Figure 9 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), which is also described within chapter 2. We can observe in figure 9 that the number of necessary vehicles is almost flat between 8:30 and 18:30, with a minor peak around 13:00.

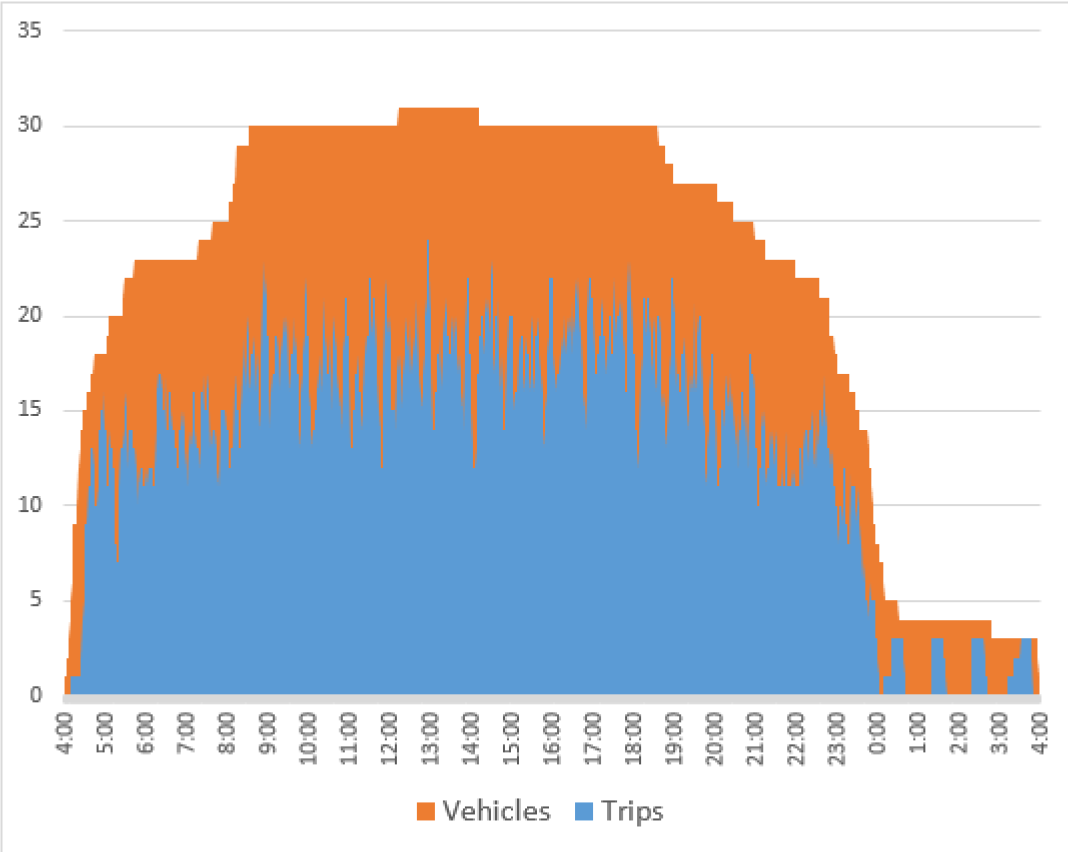


Figure 9: Number of trips and necessary vehicles for Saturday timetable[9]

[9]Data source: Saturday timetable of public transport company of Liberec and Jablonec nad Nisou, data are attached in Appendix B

| Route name | Number of trips | Number of durations | Duration range | Number of start stops | Uniformity coefficient |
|---|---|---|---|---|---|
| 12 | 140 | 11 | 12 - 31 | 4 | 0.96 |
| 13 | 36 | 2 | 18 - 33 | 3 | 0.69 |
| 14 | 38 | 1 | 24 | 2 | 0.00 |
| 15 | 40 | 3 | 10 - 14 | 3 | 0.83 |
| 16 | 38 | 6 | 13 - 34 | 4 | 0.24 |
| 17 | 14 | 1 | 14 | 1 | 0.92 |
| 18 | 39 | 6 | 9 - 31 | 4 | 0.11 |
| 19 | 33 | 4 | 11 - 18 | 3 | 0.79 |
| 20 | 40 | 3 | 14 - 22 | 3 | 0.28 |
| 21 | 40 | 3 | 6 - 16 | 2 | 0.92 |
| 22 | 117 | 19 | 8 - 42 | 7 | 0.48 |
| 24 | 78 | 4 | 13 - 35 | 4 | 0.22 |
| 25 | 107 | 4 | 8 - 22 | 4 | 0.90 |
| 26 | 39 | 4 | 15 - 34 | 3 | 0.39 |
| 27 | 4 | 1 | 11 | 2 | undefined |
| 30 | 29 | 2 | 11 - 13 | 2 | 0.58 |
| 31 | 6 | 2 | 15 - 16 | 2 | undefined |
| 32 | 2 | 2 | 7 - 10 | 2 | undefined |
| 34 | 1 | 1 | 11 | 1 | undefined |
| 38 | 8 | 1 | 5 | 2 | 1.00 |
| 90 | 3 | 1 | 22 | 1 | 1.00 |
| 91 | 3 | 1 | 18 | 1 | 1.00 |
| 92 | 3 | 1 | 20 | 1 | 1.00 |
| 97 | 1 | 1 | 25 | 1 | undefined |
| 98 | 1 | 1 | 18 | 1 | undefined |
| 99 | 1 | 1 | 37 | 1 | undefined |
| 500 | 59 | 2 | 7 - 10 | 2 | 0.85 |
| 600 | 77 | 2 | 9 - 10 | 2 | 1.00 |

Table 1: Routes and trips summary for Saturday timetable[10]

[10]Data source: Saturday timetable of public transport company of Liberec and Jablonec nad Nisou, data are attached in Appendix B

## 8.1 Methods

We use the methods proposed and described within previous chapters of this work. First, we detect critical trips using algorithm 1, and select the desired number of vehicles. If necessary, we obtain meaningful number of critical trips alternatives as per section 5.4. Then, in tight collaboration with subject matter experts from public transport company of Liberec and Jablonec nad Nisou and building on their input, for selected handling methods we evaluate the critical trips alternatives as per section 6.2. Based on the evaluation, the choice of given amount of critical trips alternatives for handling is made.

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 underlying timetabled data are attached as appendix B in GTFS format, see Google LLC (2006).

The necessary amount of vehicles and crew is computed by vehicle and crew scheduling system Kastor, see Palúch and Majer (2017), which is also described within chapter 2. Kastor is vehicle and crew scheduling system, therefore solution yielded by Kastor can generally use more vehicles than the minimal possible amount. However, here, the amount of vehicles yielded by Kastor is minimal possible, which will be evident from the results of algorithm 1 provided within next section.

## 8.2 Discussion

Relevant measures for the original result of vehicle and crew scheduling run upon the timetabled data are provided in table 2. Upon the timetabled data, we run algorithm 1 to obtain the critical trips, and algorithm 3 to obtain all the alternatives.

Figure 10 shows last 9 vehicle schedules after the last run of algorithm 1, i. e. the last 9 shortest

| Measure | Count |
|---|---|
| Number of trips | 997 |
| Number of vehicles | 31 |
| Number of crew shifts | 59 |

Table 2: Original results of vehicle and crew scheduling[11]

[11]Data source: Saturday timetable of public transport company of Liberec and Jablonec nad Nisou, data are attached in Appendix B

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. To summarize, we aim to remove 2 critical trips to lower the amount of necessary vehicles by 2 to 29. Algorithm 3 yielded 3 further alternatives of critical trips, colored in grey.

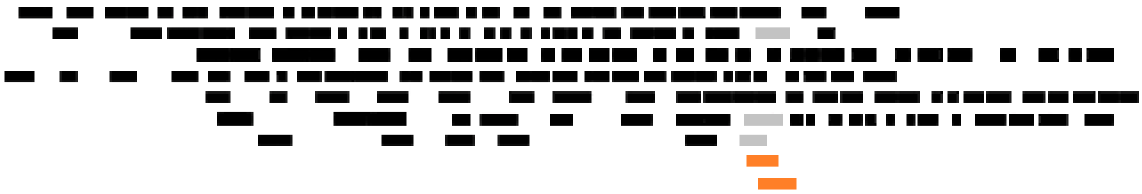

Figure 10: Segment of vehicle schedule for Saturday timetable yielded by algorithm 1 with critical trips identified in orange and critical trips alternatives in grey[12]

[12]Data source: Saturday timetable of public transport company of Liberec and Jablonec nad Nisou, data are attached in Appendix B

Let us denote the critical trips alternatives from figure 10 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 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. Table 3 summarizes the values of uniformity coefficients for each critical trip alternative.

| Critical trip alternative | Uniformity coefficient |
|---|---|
| $a_1$ | 0.96 |
| $a_2$ | 0.96 |
| $a_3$ | 0.24 |
| $a_4$ | 0.11 |

Table 3: Uniformity coefficients of critical trips alternatives[13]

[13]Data source: Saturday timetable of public transport company of Liberec and Jablonec nad Nisou, data are attached in Appendix B

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.

## 8.3 Conclusion of case study

Within this case study, we applied the proposed methods for critical trips identification, evaluation and handling on Saturday timetable of public transport company of Liberec and Jablonec nad Nisou.

Based on the amount of critical trips yielded by algorithm 1, we set the desired reduction of number of necessary vehicles to 2. We performed the search for all critical trips alternatives using algorithm 3, which yielded in total 5 alternatives for consideration, and together with subject matter experts from public transport company of Liberec and Jablonec nad Nisou we performed critical trips evaluation and handling. One critical trip got cancelled, and one rescheduled. Comparison of vehicle and crew scheduling results before and after critical trips handling is provided in table 4. 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 |
| Avg trips per vehicle | 32.2 | 34.3 |
| Avg trips per crew shift | 16.9 | 17.5 |

Table 4: Results of vehicle and crew scheduling before and after critical trips handling[14]

[14]Data source: Saturday timetable of public transport company of Liberec and Jablonec nad Nisou, data are attached in Appendix B

# 9 Conclusion

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, see attachment A. 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 problems.

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, 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.
- Decreasing the amount of drivers necessary for duty rostering.
- Utilization of knowledge of critical set of trips for optimizing the iterative process of

timetabling → vehicle scheduling → timetabling for higher efficiency and lower operating costs.

- One of the risk management auxiliary approaches to the case that some of the crew members get infected with covid 19 or are quarantined.

# References

Aarhaug, J. (2016). Taxis as a Part of Public Transport. *GIZ*.

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.

Daduna, J. and A. Wren (1988). Computer-aided transit scheduling. *(Springer) Berlin, Heidelberg, New York, London, Paris, Tokyo*.

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.

Goldberg, A. and R. Tarjan (2018). Finding minimum-cost circulations by canceling negative cycles. *Journal of the ACM* 16, pp. 1–4.

Google LLC (2006). *GTFS specification reference*. Accessed: 17. 4. 2021. URL: https://developers.google.com/transit/gtfs/reference/#general_transit_feed_specification_reference.

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.

Last Mile (2017). State-of-the-Art of regional public transport systems and particularly flexible systems. Accessed: 1. 3. 2021. URL: http://www.parcs.at/nphtt/pdf_public/2018/36064_20180316_073812_LM_stateoftheartanalysis_EASTTYROL.PDF.

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*.

Rousseau, J. (1985). Computer scheduling in public transport 2. *(North Holland) Amsterdam, New York, Oxford*.

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.

Sengo, R. and S. Garland (2020). Accelerating Bellman-Ford Single Source Shortest Path Algorithm on GPU using CUDA. *Term Paper, Parallel Algorithms, University of Texas, Austin*.

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. (1981). Computer scheduling in public transport. *(North Holland) Amsterdam, New York, Oxford*.

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.

# Attachment

**Attachment A** Session's best paper award from the WMSCI 2018 conference

**Attachment B** Supplementary data file with case study data

**Attachment A**

22nd World Multiconference on
Systemics, Cybernetics and Informatics

**Session's Best Paper Award**

Session: Computer Science and Engineering

*presented to*

***Kateřina Pastirčáková and Jaromír Pǔlc***

*for the paper entitled*

Detection of Minimal Set of Trips Causing the Necessity to Use Extra Vehicle for Vehicle Scheduling Problem

*July 11th, 2018*
*Orlando, Florida, USA*

Nagib C. Callaos
*General Chair*
*WMSCI 2018*

Belkis Sánchez
*Organizing Committee Chair*
*WMSCI 2018*

## Attachment B

Supplementary data file

Filename: GTFS.zip

Description: The accompanying GTFS file provides the timetabled data used within the case study. It is bus timetabled data valid for Saturday 13. 1. 2018.

The zipped file contains following files:

- agency.txt
- calendar.txt
- calendar_dates.txt
- reserves.txt
- routes.txt
- shapes.txt
- stop_times.txt
- stops.txt
- successors.txt
- transfers.txt
- trips.txt