BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Preněk Ondřej**	Personal ID number:	**460532**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Open Informatics**		
Branch of study:	**Computer and Information Science**		

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Intermodal Trip Planning Using the Metaplanning Approach

Bachelor’s thesis title in Czech:

Intermodální plánování cest s využitím metaplánovacího přístupu

Guidelines:

Intermodal trip planning is the type of door-to-door trip planning for passenger mobility that is capable of finding trip plans considering the full range of means of transport and their combinations. The metaplanning approach to intermodal trip planning aims to deliver intermodal trip planning capability by integrating multiple single-mode trip planners, each specialized for a single means of transport.
Instructions:
1. Familiarize yourself with intermodal trip planning, in particular with the metaplanning approach to intermodal trip planning [1].
2. Survey single-modal trip planners suitable for integration into the metaplanning approach.
3. Implement trip metaplanner core and connectors to selected single-modal trip planners.
4. Deploy the implemented trip metaplanner on a selected test region.
5. Evaluate the performance of the metaplanner on a selected test region.

Bibliography / sources:

[1] NYKL, J: Multimodal route planner integration based on abstracted transport system representation. Diploma Thesis, Czech Technical University, February 2016
[2] BAST, H. et al.: Route Planning in Transportation Networks. Technical report, Microsoft Research, 2014.
[3] DELLING, D. et al.: Computing Multimodal Journeys in Practice. In Experimental Algorithms, 7933. Springer Berlin Heidelberg, 2013. s. 260-271.

Name and workplace of bachelor’s thesis supervisor:

doc. Ing. Michal Jakob, Ph.D., Artificial Intelligence Center, FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **14.01.2018** Deadline for bachelor thesis submission: **25.05.2018**

Assignment valid until: **30.09.2019**

doc. Ing. Michal Jakob, Ph.D.	doc. Ing. Tomáš Svoboda, Ph.D.	prof. Ing. Pavel Ripka, CSc.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics

Bachelor Thesis

Intermodal Trip Planning Using the Metaplanning Approach

Ondřej Preněk

Supervisor: prof. Ing. Michal Jakob

Study Programme: Open Informatics

Field of Study: Computer and Information Science

May 24, 2018

Aknowledgements

Foremost, I would like to express my gratitude to Jan Nykl for his continuous support throughout the whole project period, for his patience and motivation. His guidance helped me in all the times of research and writing of this thesis. My sincere thanks also go to my advisor Michal Jakob. Beside guidance during the research, he provided me an opportunity to join their team as an intern during the summer. I thank my university classmates for all the fun we have had in the last three years. Last but not the least, I would like to thank my family for providing me with unfailing support and continuous encouragement throughout my years of study and writing the thesis.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague 24.5. 2018 ....

Abstract

The low city center efficiency and high transport costs of car transport are factors that have influenced the growing popularity of intermodal transport during recent years. By combining existing single-mode planners I have created an intermodal journey planner, I have designed, implemented and subsequently evaluated strategies for the use of single-modal planners to make the process of constructing journey planner that combines multiple modes of transport as efficient as possible and thus the resulting journey planner quality as high as possible. I evaluate intermodal journey planner on real data in the Prague area. I tested the quality of the strategy using the quality of the resulting plans. The results of the following evaluation confirmed that the strategies using advanced techniques of using single-mode planners were much more effective.

Key words: intermodal journey planning, journey planning, metaplanning, abstracted transport network, shortest path problem

Abstrakt

Nízká efektivita v centru měst a vysoké přepravní náklady pomocí aut jsou faktory, které ovlivnily rostoucí popularitu intermodální dopravy v posledních letech. Kombinací více již existujících singlmodálních plánovačů jsme vytvořili intermodální plánovač, navrhli jsme a následně implementovali strategie použití singlmodálních plánovačů tak, aby vytvoření plánovače kombinující více módů dopravy bylo co nejefektivnější a tím tedy i výsledný plánovač co nejkvalitnější. Následnou evaluaci jsme provedli na reálných datech v oblasti Prahy. Kvalitu strategie jsme otestovali pomocí kvality výsledných plánů. Výsledky evaluace potvrdili, že strategie využívající pokočilé techniky použití singlmodálních plánovačů byli mnohem efektivnější.

Klíčová slova: intermodální plánování, plánování cesty, metaplánování, abstrahovaná dopravní síť, problém nejkratší cesty

Contents

Chapter 1

Introduction

1.1 Motivation

As a result of increasing population alongside urbanization, using cars in cities has become very inefficient. The transport infrastructure is unable to handle such a large number of cars and driving a car during rush hours is a nightmare. In addition to inefficiency, cars pollute the air in cities. The effect of air pollution on human health is devastating. It is a worldwide cause of many deaths and diseases, such as the respiratory disease, cancer and heart disease.

Nowadays because of many available modes of transport, one of the possible solutions is to combine these modes to use each mode in place, where it is most efficient. This method is not so widely used because it costs a lot of effort and time to plan manually *intermodal* journey - journey combining multiple means of transport.

Moreover, it usually disregards other factors, such as current traffic or closures. Instead of wasting time on manual journey planning, I will automize routing process by journey planner. That planner can take these factors into account when planning a journey and saves time, transport costs and can hence using the planner makes cities ”greener”.

To combine multiple modes of transport, I need to create an advanced journey planner that computes journey plans combining multiple modes of transport. That planner is referred to as *intermodal.*

A significant part of strategies constructing the intermodal journey planner required perfect trasnport data for the selected region. The primary motivation for choosing the metaplanning approach was an opportunity to construct intermodal journey planner without them. The data are difficult to reach or they are even unavailable, so we chose a metaplanning approach that uses API services rather than data.

1.2 Aim of the Thesis

This thesis aims at developing a journey planner capable of finding journey plans considering the full range of means of transport and their combination.

First of all, I define concepts of intermodal journey planner and approach of using already existing single-mode journey planners. Then, I focus on increasing efficiency of constructing intermodal journey planner and its quality. Next, I aim at implementation problems such as connecting single-mode journey planners and concepts for finding the intermodal path. Furthermore, I deploy the intermodal journey planner on a selected test region and last not least, I evaluate its performance.

Chapter 2

Related Work

In this chapter, I will present an overview of studies that have been previously dealing with journey planning.

First of all, I will focus on unimodal journey planning. I will mention the widespread algorithms, that are used for planning an unimodal journey. Then, I will continue with intermodal planning, where I will focus on recent studies and I will describe used techniques in detail. Finally, I will aim at journey planners integration, on which the thesis is mainly focused.

2.1 Unimodal journey planning

Since the unimodal journey planning has motivated a large number of theoretical works, there are multiple approaches, which made a huge impact on its improvement. A variety of techniques provides different trade-offs between preprocessing effort, space requirements, and query time.

Despite disadvantages of high space requirements and high preprocessing time, non-graph-based methods are widely used in real world. They are more than a million times faster than the Dijkstra's algorithm [8]. The table lookup is impossible to use due to enormous space requirements. However, the labeling algorithms seem to be the optimal alternative. HL-∞ is only five times slower (0.25 μs) and the HL nine times slower (0.56 μs) than table lookup (0.06 μs).

The RAPTOR (Round-bAsed Public Transit Optimized Router) algorithm is explicitly developed algorithm for finding shortest path on public transport network [11]. The RAPTOR minimizes arrival time and the number of transfers by traversing each route (such as a bus line) at most once per transfer. An approach of dynamical programming is applied for this procedure.

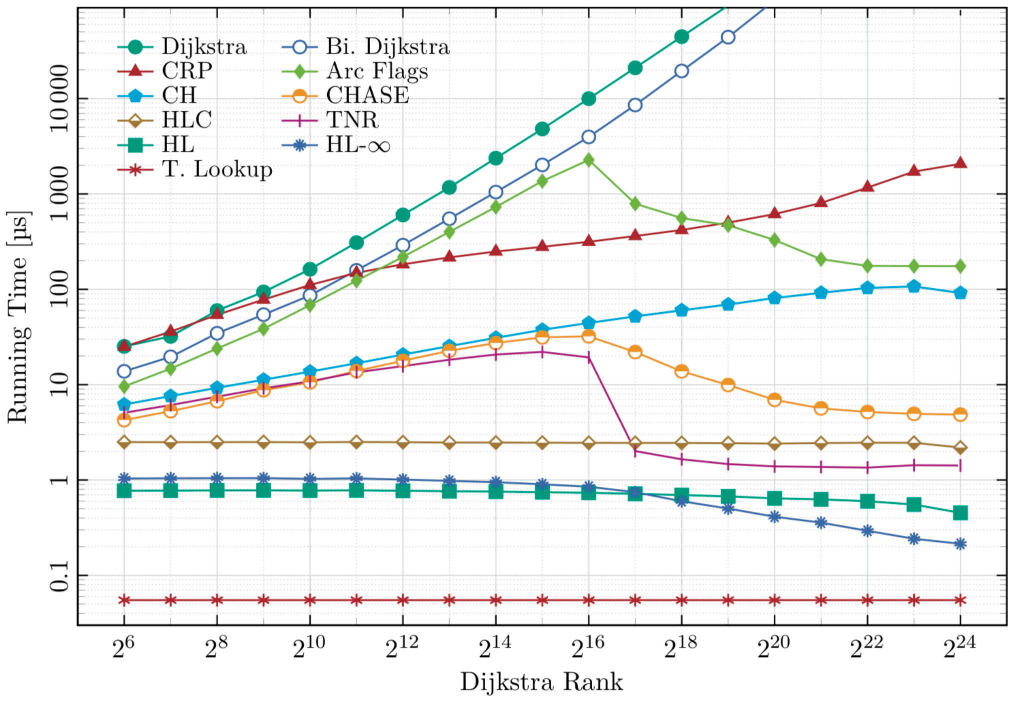


Figure 2.1: Performance of speedup techniques for various Dijkstra ranks [8]

2.2 Intermodal journey planning

The intermodal journey planning is a more complicated problem for which does not exist so many established techniques as in the case of the unimodal journey planning.

To plan journeys that reasonably combine different transport modes, one may use penalties (mostly the travel time) in the objective function of the algorithm for path optimization. Aifadopoulou et al. [3] present a linear program that computes intermodal journeys. Another example that is based on this approach is the TRANSIT algorithm [4], which uses a linear utility function.

The *label-constrained shortest paths* [5] approach computes journeys that explicitly obey certain constraints on transport modes, and the approach is even solvable in deterministic polynomial time. The problem of computing shortest label-constrained paths is tractable for regular languages, which suffice to model reasonable transport mode constraints in intermodal journey planning [6].

One of the biggest drawbacks of label-constraints' approach is that the alternative journeys are not computed. To obtain a set of alternative journeys, *multicriteria optimization* has been considered.

Bast et al. [7] use MLS [14] to compute intermodal multicriteria journeys on a metropolitan scale. To identify the significant journeys of the Pareto set, they propose a method called Types aNd Thresholds (TNT) summarizing what users would consider mostly as unreasonable intermodal paths.

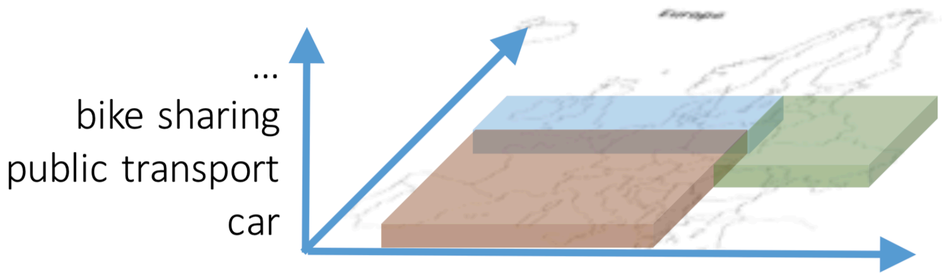
Delling et al. [10] consider metropolitan-scale networks and use multiple criteria as representatives for "convenience". To achieve better query performance than in the case of the MLS algorithm, they extend the RAPTOR, which results in the intermodal multicriteria RAPTOR algorithm (MCR).

2.3 Journey Planners Integration

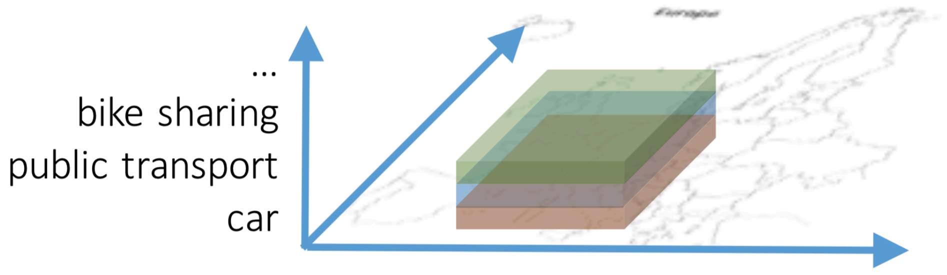
Integration of travel planners has not been extensively discussed in the prior research. Most of the journey-planning studies are focusing on unimodal or intermodal journey planning. Both fields of studies assume that transport data are available. Compared to that, the journey planners integration is focusing on planning with incomplete data - journey planners integration uses services rather than data.

Let me divide journey planners integration into two types. The first type is a geographical integration. The geographical integration provides availability to plan a journey on the covered area - at least one journey planner is available for each piece of the covered area. Probably the best-known project providing geographical integration of journey planners is the Rome2Rio [1], which offers the worldwide coverage. Rome2Rio provides a simplified first step, but a more useful system would take into account real-time traffic and transit information, historic patterns, schedule constraints, and monetary costs. Another project is EU-SPIRIT [16] that offers the geographical integration on a few areas of Europe.

The second type is modal integration. Modal integration provides intermodality in the covered area - it means that all journey planners are available for each piece of the covered area. This field has been studied only by Jan Nykl [15] whose work I follow up.



(a) Geographical integration of journey planners.



(b) Modal integration of journey planners.

Figure 2.2: Different types of journey planners integration [15]

Chapter 3

Problem Specification

This chapter defines certain key terms and problems, which are used in thesis. First of all, I will explain the concept of intermodal journey planner and subsequently a metaplanning approach.

To specify intermodal journey planner, I will describe transport network and intermodal transport network model. Furthermore, I will describe the shortest path problem, which is used for finding intermodal journey. I will define journey planning capable of combining multiple transport modes.

Next, I will describe the process of metaplanning. I will specify abstracted transport network, its construction strategies and determining its quality. Last but not least, I will describe a metasearch and a refinement.

3.1 Transport Network

Transport network consists of roads, crossroads, and sidewalks. These components allow a specific transport mode to move. In urban areas, there are usually more modes of transport such as bike/car sharing, taxi and much more. In my model I will employ cars, public transport, bicycles and walk. Variety of public transport depends on the specific area, but mostly it consists of buses, trains and it might be supplemented by trams or metros in large urban areas.

3.2 Intermodal Transport Network Model

In large urban areas, there is a wide variety of transport modes. Transport network with a wide variety of transport modes allows me to use multiple modes of transport when I want to get from point A to point B. This network is widely known as intermodal.
Let me define the transport directed graph $G = (V, E, M, \epsilon, \phi, \psi, \tau)$ where

- V is a set of vertices.
- $E = E_t + E_n$ is a set of edges. E_t are edges, that can be used for transport on given time. Such example might be edge from point A to point B by bus, that runs at ten-minute intervals. E_n are edges, that can be used anytime (e.g. walking edges).

- $M = M_t + M_a$ is set of transport modes. M_t is set of modes, that has time constraint, M_a are modes without time constraint.
- $\epsilon : V \times V \to E$ is a function, that assign ordered pair of vertices to edge.
- $\phi : E \to \mathbb{R}_0^+$ is edge's cost function.
- $\psi : E \to M$ is a function, that assign transport mode to edge.
- $\tau : E_t \to \vec{t}; \quad t_i \in \mathbb{R}_0^+ \times \mathbb{R}_0^+$ is a function, that assign each edge $e \in E_t$ vector of time intervals $\vec{t} = t_1, ..., t_k$, which are valid for edge e.

One of the main factors that influence the final planned route R is the cost function ϕ. Since the route is the edge sequence, summarising cost of each edge $e_i \in R$ will give me the route cost C.

$$R = \{e_1, e_2, ..., e_n\} \tag{3.1}$$

$$C = \sum_{i=1}^{n} \phi(e_i) \tag{3.2}$$

Despite the fact that edge's cost can depend on several criteria - duration, distance and the price I decided to determine the edge's cost according to its duration only.

3.3 Shortest Path Problem

The graph theory defines the shortest path problem as finding the path P between origin vertex v_o and destination vertex v_d, where the cost of path C (2.1) is the lowest possible. The resulting path P is represented by the ordered sequence of n subpaths:

$$P = (p_1, ..., p_n) \tag{3.3}$$

$$p_i = (e_i, \tau) \qquad\qquad i \in 1, ..., n \tag{3.4}$$

where $e_i \in V \times V$ is the edge between origin and destination vertex of subpath p_i and $\tau \in \mathbb{R}_0^+ \times \mathbb{R}_0^+$ is the departure and arrival time for subpath p_i.

3.4 Intermodal Journey Planning

Journey planning that combines multiple modes of transport is referred to as *intermodal*. There are several criteria, which could determine the optimal journey such as travel time, travel cost, number of transfer, et cetera. We take into account only the travel time.

The component capable of performing intermodal journey planning is reffered to as *intermodal journey planner*.

3.5 Metaplanning

Metaplanning is a process of finding the shortest path. Firstly, I perform searching the shortest path on abstracted transport network - *metasearch*, then I improve founded shortest path - *metaplan* by *refining* it. This process is shown below in Figure 3.1.

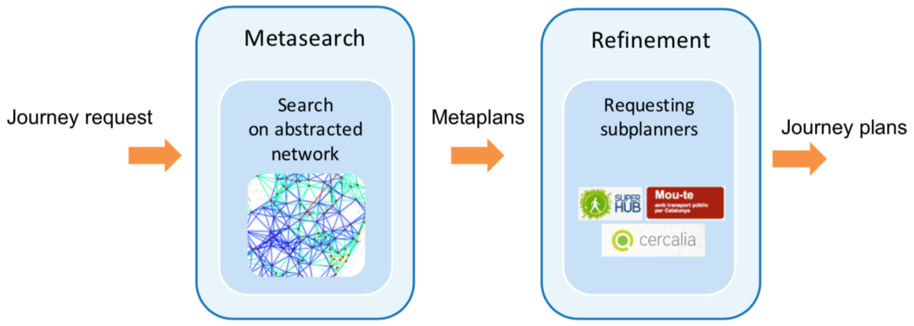


Figure 3.1: Illustration of the metaplanning process [15]

3.5.1 Abstracted Transport Network

Abstracted transport network is an imperfect form of the transport network. It is imperfect because it does not include all roads or even transport modes in the deployed area.

Let me define full transport network as $G = (V, E, M, \epsilon, \phi, \psi, \tau)$. Then abstracted transport network G' is defined as:

$$\begin{aligned} & G' = (V', E', M', \epsilon, \phi, \psi, \tau) \\ & \text{subject to} \qquad V' \subseteq V \\ & \qquad\qquad\qquad E' \subseteq E \\ & \qquad\qquad\qquad M' \subseteq M \end{aligned} \tag{3.5}$$

3.5.1.1 Abstraction quality metric

Abstraction quality metric determines how well the abstracted transport network is constructed. I proposed to measure path quality on abstracted transport network due to its importance in the subsequent planning.

Let $G = (V, E, M, \epsilon, \phi, \psi, \tau)$ be the transport network and let $G' = (V', E', M', \epsilon, \phi, \psi, \tau)$ be its abstraction. Shortest path problem is executed for each route $r \in R$ on G and its abstraction G' afterward. Resulting abstraction quality is determined as:

$$Q = \sum_{r \in R} |SPA(r, G') - SPA(r, G)| \tag{3.6}$$

where $SPA(r, G)/SPA(r, G')$ is duration of the shortest path r founded on transport network G/G'.

By improving abstraction, the value of Q is minimized. In other words, I optimize the value of $SPA(r, G')$ that ideally equals to $SPA(r, G)$.

3.5.2 Metasearch

Metasearch solves the shortest path problem on the abstracted transport network G', where the origin of the metasearch is the closest node $o_m \in V'$ to the origin location o. The same applies to the destination - destination of the metasearch is the closest node $o_d \in V'$ to the destination location d. I mention, that o and d are specified in the path request $q = (o, d, m)$. Metasearch has been similarly defined by Jan Nykl [15].

3.5.3 Refinement

Refinement is a process of improving metasearch path P by re-finding its subpaths p_i:

$$\begin{aligned} & P = (p_1, ..., p_n) \\ & p_i = (e_{i_1}, .., e_{i_m}) \\ & \text{subject to} \\ & \qquad \forall j \in [1, m] : e_{ij} \in E' \\ & \qquad \forall p_i \in P : \psi(e_{i_1}) = \psi(e_{i_2}) = ... = \psi(e_{i_m}) \\ & \qquad \forall p_i, p_{i+1} \in P : \psi(p_i) \neq \psi(p_{i+1}) \end{aligned} \tag{3.7}$$

I divide path P to the ordered sequence of n subpaths, where each subpath is *unimodal* - using only one mode of transport. For each subpath, I call subplanner request between subpath's origin and destination point. This procedure enables me to find other and better edges. Moreover, it enables me to eliminate gaps and insufficiently dense places in abstracted transport network (missing nodes, or better edges between them).

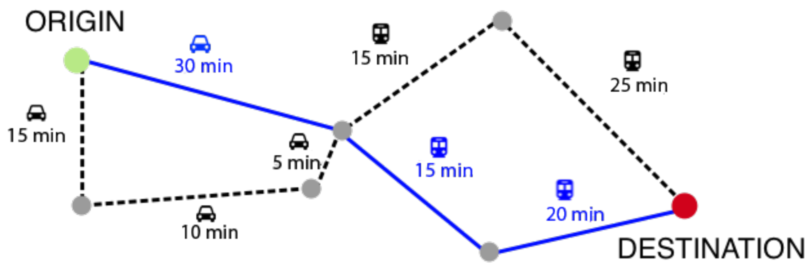
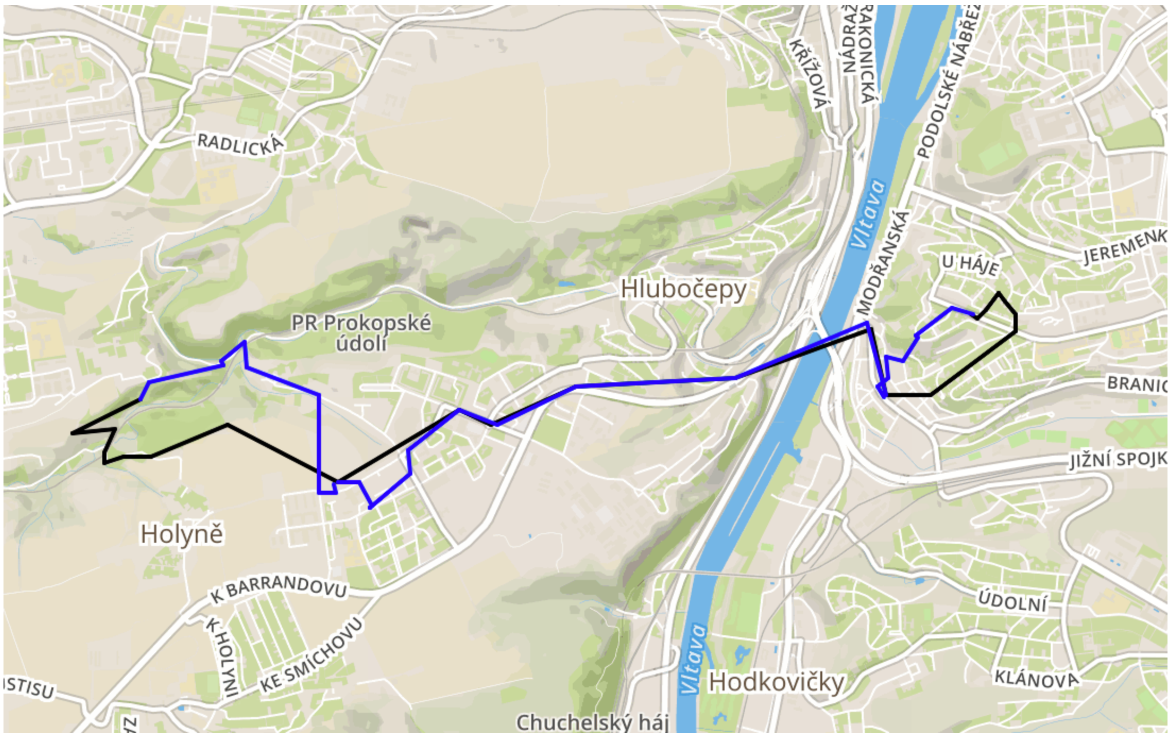


Figure 3.2: Theoretical example of refinement. Black dash line represents path before refinement and blue continuous line represents the path after refinement.



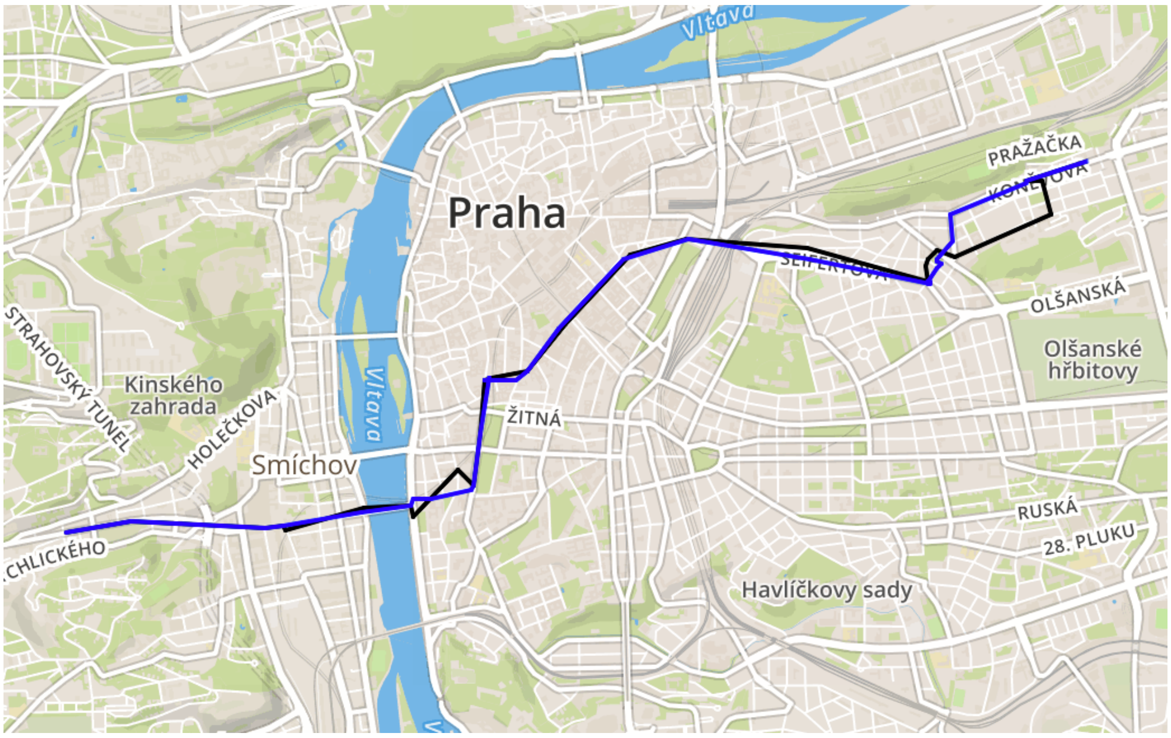


Figure 3.3: Examples of refinement on the selected test region - Prague. Black line represents path before refinement and blue line represents the path after refinement

Chapter 4

Solution Approach

Now, all the necessary terms were mentioned for understanding the approach that leads to the solution- finding intermodal journey. In this chapter, I will describe that approach in detail.

4.1 Approach Overview

To find an intermodal journey, I need to create a component, that takes all the possible connections and find the optimal route in them. That component has been described previously as intermodal journey planner. To find the optimal route, intermodal journey planner has to use one of several techniques, which solve shortest path problem. The need to know all of the connections leads to the most important problem, which is the construction of the transport network. The transport network is a theoretical concept, and I am constructing only its abstraction. To develop a high-quality intermodal journey planner, I need to construct a high-quality abstracted transport network. I will achieve that by choosing the most appropriate strategy combined with a high number of requests.

4.2 Abstracted Transport Network construction

Abstracted Transport network can be obtained by using subplanners. I have chosen that approach.

I use subplanners for obtaining shortest path between multiple OD pairs with specific transport mode for each of them in given radius (corresponding to the size of the selected test region). I take obtained paths and create directed graph from them, where all points on every path are graph's vertices and subpaths between them are graph's edges. The way OD pairs are selected is determined by the chosen strategy.

4.2.1 Subplanner Request

I introduce the concept of the subplanner request because I need a benchmark for determining the rate of single-mode planners (hereinafter the *subplanners*) use. Abstracted network G' is constructed using transport data received from subplanners and subplanners' data are obtained from requesting path between origin and destination. I referred each of this request q_m as *subplanner request*:

$$q_m = (o, d, m) \qquad\qquad o, d \in \mathbb{R} \times \mathbb{R} \tag{4.1}$$

where o is an origin location, d is a destination location and m is a transport mode. o and d are represented as a tuple of geographic latitude and longitude.

4.3 Construction Strategies

Construction strategy of abstracted transport network determines, how the subplanners will be requested - strictly speaking how the *Origin-Destination pairs* (hereinafter the OD pairs) will be chosen. The better the strategy is, the fewer subplanner requests are needed, and thus it takes less time and it is less resource-intensive to construct abstracted transport network of a certain quality.

There are a lot of strategies for picking OD pairs. Below, I mention only these, which would be deployed and evaluated afterward.

4.3.1 Uninformed Strategies

Uninformed strategies have their strategy fixed regardless of the data obtained during the execution. Generally, these strategies have low complexity.

4.3.1.1 Random OD Pairs

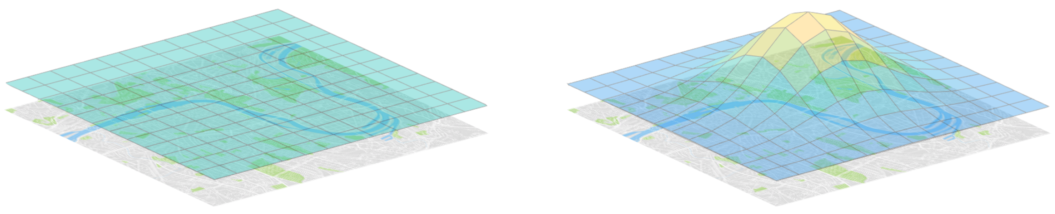
OD pairs are chosen randomly in selected test region without any other constraints. It is the simplest strategy to develop.

4.3.1.2 Random OD Pairs Constrained by Minimum Mutual Distance

OD pairs are chosen randomly too, but the distance between origin and destination point is set. It will prevent from calling unnecessary requests, that looks for the path for OD pair with very small mutual distance and thus it reduces the number of total requests needed to construct the abstracted network with specified quality.

4.3.1.3 Chaining Random OD Pairs

This strategy aims to achieve consistency of the abstracted network - providing its full connectivity. To do that, every two consecutive planned paths have common origin or destination.



(a) Uniform distribution (b) Normal distribution

Figure 4.1: Sample multivariate distribution on the selected test region- Prague

4.3.2 Informed Strategies

Informed strategies use the data obtained during the execution for improving abstraction quality. For example, it can continuously detect low-density places and improve its density. Basically, they consist of the uninformed-strategy core with a connection to more advanced techniques.

4.3.2.1 Distribution of Nodes

Following strategy regulates picking OD pairs to maintain the chosen distribution of nodes. I have decided to deploy and evaluate the efficiency of uniform and normal distribution. Due to multi-dimensional coordinate system (Latitude,Longitude) I have to use multi-dimensional (referred as *multivariate*) distribution too. Sample multivariate distribution on the selected test region is shown in Figure 4.1.

4.3.2.2 Distribution of Edges

This strategy picks OD pair to maintain the chosen distribution of edges for each transport mode separately. As in the previous case, I deploy and evaluate multivariate uniform and multivariate normal distribution.

4.3.3 Supervised Strategies

If I do evaluation in area, which is known in advance, supervised strategies use the knowledge to improve abstraction quality.

4.3.3.1 Density increase for key locations

This strategy is focusing on places, which play a key role in the transport for a specific area (bus station, railway station, .etc). These places are used for increasing the network's density in their surroundings.

4.4 Abstraction quality metric

Let $G = (V, E, M, \epsilon, \phi, \psi, \tau)$ be the transport network and let $G' = (V', E', M', \epsilon, \phi, \psi, \tau)$ be its abstraction. Abstraction quality is determined as:

$$Q = \sum_{r \in R} |SPA(r, G') - SPA(r, G)| \tag{4.2}$$

where $SPA(r, G)/SPA(r, G')$ is duration of the shortest path r founded on transport network G/G'. The transport network G is not known, so I can not use G to determine abstraction quality. Thus I need to find some alternative way how to measure it.

4.4.1 Refinement Deviation

The alternative way is to measure the metasearch path quality by comparing its quality before and after refinement. The metasearch path is evaluated on the abstracted network, which quality I want to determine. To be able to deduce abstraction quality, I need to perform comparison many times. Otherwise, the measured quality would not be accurate. I, therefore, summarise a duration difference of thousand metasearched routes to determine abstraction quality.

Now, abstraction quality Q can be determined by subtracting route cost C of original paths $p \in P$ and route cost C' of refined paths $p' \in P'$:

$$\begin{aligned} Q &= C - C' \qquad &(4.3)\\ &= \sum_{i=1}^{1000} \phi(p_i) - \phi(p'_i) \qquad &(4.4)\\ &= \sum_{i=1}^{1000}\sum_{j=1}^{m} \phi(e_{ij}) - \sum_{i=1}^{1000}\sum_{j=1}^{k} \phi(e'_{ij}) \qquad &(4.5) \end{aligned}$$

The lower value of Q, the higher is the quality of abstraction.

Chapter 5

Implementation

In the previous chapter, I have focused on the theoretical approach, how to create the abstracted transport network to make the best possible intermodal journey planner. In this chapter, I will describe certain algorithms and techniques, which are used in my work and lead to the solution.

At the beginning, I will describe the intermodal journey planner, that performs metaplanning, where I will focus on metasearch that solves the shortest path problem using the Dijkstra's algorithm. Furthermore, I will describe construction of the abstracted transport network construction. I will focus on the construction strategies of the abstracted transport network and measuring abstraction quality using the refinement. In the end, I will mention the software development tools used in this project.

5.1 Intermodal Journey Planner

To create the intermodal journey planner means to create a component capable of doing metaplanning, on which I will focus.

5.1.1 Metaplanning

Metaplanning process consists of two parts - metasearch and refinement. The pseudocode of metaplanning is shown below 1.

Algorithm 1 Metaplanning procedure

```
1: function METAPLANNING(ORIGIN, DESTINATION, GRAPH)
2:     $n_o \leftarrow findNearestNode(origin, graph)$
3:     $n_d \leftarrow findNearestNode(destination, graph)$
4:     $metaplan \leftarrow metasearch(n_o, n_d, graph)$
5:     $journeyPlan \leftarrow refineMetaplan(metaplan)$
6: return $journeyPlan$
```

In the beginning, to be even able to perform metasearch, I need to find the nearest node for given origin and destination location. Due to representing the graph's node in two-dimensional coordinate system, the procedure of finding nearest node is not that trivial and it is implemented by *KD Tree* data structure [9].

Furthermore, I perform the metasearch, and I refine obtained metaplan afterward.

5.1.2 Metasearch

To perform metasearch, I need to solve the shortest path problem. There are multiple known algorithms on how to obtain the shortest path, and I decided to use the Dijkstra's algorithm.

5.1.2.1 Dijkstra's algorithm

The Dijkstra's algorithm [12] is the most notorious algorithm for solving the shortest path problem. The Dijkstra's original implementation runs in $\mathcal{O}(|V|^2)$ time (where $|V|$ is the number of vertices). It is convenient to use priority queue data structure due to its lower time complexity [2]. The implementation, what I have used is based on the min-priority queue using Fibonacci heap and running in $\mathcal{O}(|E| + |V|log|V|)$ (where $|E|$ is the number of edges). Pseudocode of the algorithm is shown below 2.

Algorithm 2 Dijkstra's algorithm using Fibonacci heap

```
1:  function DIJKSTRA(GRAPH,ORIGIN,DESTINATION)
2:      $dist[origin] \leftarrow 0$
3:      for each vertex v in graph do
4:          if $v \neq origin$ then
5:              $dist[v] \leftarrow \infty$
6:              $prev[v] \leftarrow undefined$
7:              $Q.addWithPriority(dist[v])$
8:      while Q is not empty do
9:          $u \leftarrow Q.extractMin()$
10:         if u is destination then terminate
11:         for each neighbour v of u do
12:             $alt \leftarrow dist[u] + length(u, v)$
13:             if $alt < dist[v]$ then
14:                 $dist[v] \leftarrow alt$
15:                 $prev[v] \leftarrow u$
16:                 $Q.decreasePriority(v, alt)$
        return $dist[], prev[]$
```

5.2 Abstracted Transport Network Construction

The way the network is constructed depends on the chosen construction strategy. The process of constructing a network using different strategies is almost the same and the only thing that differs is how the OD pair is selected. The general procedure of constructing abstracted transport network is shown below 3.

Algorithm 3 Abstracted transport network construction

```
1: function CONSTRUCT(REQUESTCOUNT, STRATEGY, AREA, BESTLOCATIONS, MINDISTANCE)
2:     $graph \leftarrow initEmptyGraph()$
3:     for each transportMode do
4:         $remainingRequests \leftarrow requestCount$
5:         while $remainingRequests > 0$ do
6:             $[loc1, loc2] \leftarrow pickOD(graph, area, lastLocation, bestLocations, minDistance)$
7:             $route1 \leftarrow findRoute(loc1, loc2, transportMode)$
8:             $route2 \leftarrow findRoute(loc2, loc1, transportMode)$
9:             $graph \leftarrow expandGraph(graph, [route1, route2])$
```

Let me clarify the procedure above. First of all, to construct abstracted transport network, I initialize empty graph. Then, I expand the graph by requesting subplanners for finding a route between the OD pair. I can not use an undirected graph due to unwanted ignoring one-way roads. Instead of this, I request route in two directions, which is performed in line 6 and 7. The OD pair is chosen by given construction strategy. This expansion is repeated until I exceed the number of subplanner requests.

5.2.1 Connecting Subplanners

By connecting multiple subplanners I will get a multi-modal transport network, which allows me to use multiple transport modes in further planning. However, each subplanner is using different request/response standards. To simplify that process, I need to create generic interface known as *Gateway* 5.1. Using the Gateway, I will be able to send requests and receive responses from subplanners in a unified form.

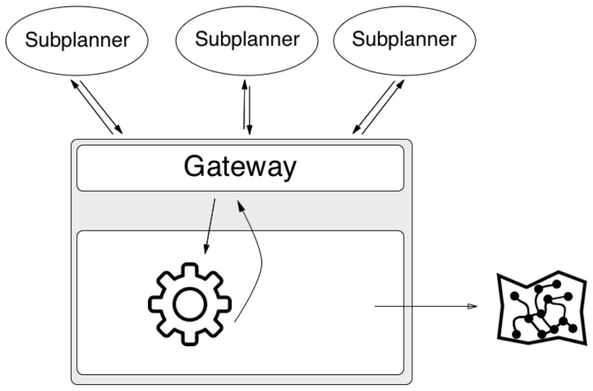


Figure 5.1: Scheme of connecting subplanners using Gateway interacting multiple times with network-constructing component, which resulting output is transport network

5.2.2 Construction Strategies

Construction strategies of abstracted transport network determine how the OD pair is selected. All construction strategies consist of the uninformed-strategy core, which is extended by using advanced techniques in non-uninformed strategies. To measure the abstraction quality, I have decided to compare the quality of metaplan before refinement and after refinement. The abstraction quality metric is described in detail previously in Section 4.4.

5.2.2.1 Uninformed Strategies

Uninformed strategies are the simplest strategies to implement because I do not have to analyze obtained data during strategy execution (as in *informed strategies*) or before executing strategy (as in *supervised strategies*). I implement and later evaluate three, very similar uninformed strategies. They all have been described in detail previously in 4.3.1, so I mention only their pseudocodes below 4, 5 and 6. I do not mention all arguments of *pickOD* method because they are not needed.

Algorithm 4 Random OD strategy

```
1: function PICKOD(AREA)
2:     $loc1 = getRandomLocation(area)$
3:     $loc2 = getRandomLocation(area)$
4:     $ODpair \leftarrow [loc1, loc2]$
5: return $ODpair$
```

Algorithm 5 Random OD Min Mutual Distance strategy

```
1: function PICKOD(AREA, MINDISTANCE)
2:     $[loc1, loc2] = getRandomLocations(count : 2, area, minDistance)$
3:     $ODpair \leftarrow [loc1, loc2]$
4: return $ODpair$
```

Algorithm 6 Chaining Random OD strategy

```
1: function PICKOD(AREA, LASTLOCATION)
2:     $loc1 \leftarrow lastLocation$
3:     $loc2 \leftarrow getRandomLocation(area)$
4:     $ODpair \leftarrow [loc1, loc2]$
5: return $ODpair$
```

5.2.2.2 Informed Strategies

All used informed strategies are based on comparing the actual and the ideal distribution of some graph's elements. To be able to compare it, I need to create an auxiliary grid for which I introduce the concept of the *distribution grid*.

The distribution grid divides selected test region into cells. In each cell, I compare the actual number of graph's elements and the ideal number by chosen distribution. Depending on whether the number of graph's elements match the ideal distribution, I evaluate the cell as valid and invalid vice versa. Then I execute expansion in invalid cells to meet the ideal distribution. Illustration of distribution grid is shown in the Figure below 5.2.

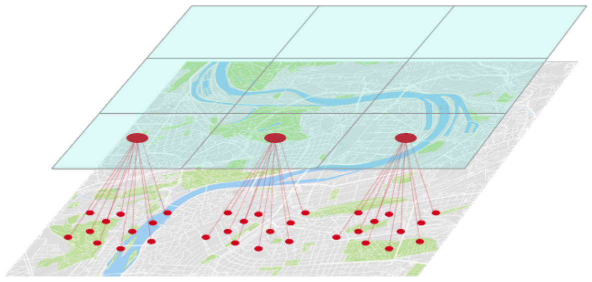


Figure 5.2: Example of collecting graph's elements in each cell of the distribution grid.

In used strategies, I observe multivariate uniform distribution and multivariate normal distribution of graph's nodes and graph's edges. Pseudocodes of the strategies are shown below 7, 8, 9 and 10. I do not mention all arguments of *pickOD* method because they are not needed.

Algorithm 7 Nodes' Uniform Distribution strategy

```
1: function PICKOD(GRAPH, AREA)
2:     $nodes \leftarrow getAllNodes(graph)$
3:     $actualDist \leftarrow actualDistOfNodesOnArea(nodes, area)$
4:     $uniformDist \leftarrow uniformDistOfNodesOnArea(nodes, area)$
5:     $invalidAreas \leftarrow getInvalidDistributedAreas(actualDist, uniformDist)$
6:     $ODpair \leftarrow pickODUsingUninformedStrategy(graph, invalidAreas)$
7: return $ODpair$
```

Algorithm 8 Nodes' Normal Distribution strategy

```
1: function PICKOD(GRAPH, AREA)
2:     $nodes \leftarrow getAllNodes(graph)$
3:     $actualDist \leftarrow actualDistOfNodesOnArea(nodes, area)$
4:     $normalDist \leftarrow normalDistOfNodesOnArea(nodes, area)$
5:     $invalidAreas \leftarrow getInvalidDistributedAreas(actualDist, normalDist)$
6:     $ODpair \leftarrow pickODUsingUninformedStrategy(graph, invalidAreas)$
7: return $ODpair$
```

Algorithm 9 Edges' Uniform Distribution strategy

```
1: function pickOD(graph, area)
2:     $edges \leftarrow getAllEdges(graph)$
3:     $actualDist \leftarrow actualDistOfEdgesOnArea(edges, area)$
4:     $uniformDist \leftarrow uniformDistOfEdgesOnArea(edges, area)$
5:     $invalidAreas \leftarrow getInvalidDistributedAreas(actualDist, uniformDist)$
6:     $ODpair \leftarrow pickODUsingUninformedStrategy(graph, invalidAreas)$
7: return $ODpair$
```

Algorithm 10 Edges' Normal Distribution strategy

```
1: function pickOD(graph, area)
2:     $edges \leftarrow getAllEdges(graph)$
3:     $actualDist \leftarrow actualDistOfEdgesOnArea(edges, area)$
4:     $normalDist \leftarrow normalDistOfEdgesOnArea(edges, area)$
5:     $invalidAreas \leftarrow getInvalidDistributedAreas(actualDist, normalDist)$
6:     $ODpair \leftarrow pickODUsingUninformedStrategy(graph, invalidAreas)$
7: return $ODpair$
```

5.2.2.3 Supervised Strategies

Supervised strategies use knowledge of the selected test region and make the network more dense in places, where it is needed. In the area around all important traffic junctions, I execute network expansion, and then I continue normally with executing uninformed strategy in the whole selected test region. I use strategy, which at the beginning select important traffic junctions as origin/destination point. Afterward, I continue normally using uninformed strategy. The pseudocode is shown below, too 11. I do not mention all arguments of *pickOD* method because they are not needed.

Algorithm 11 Using Known Nodes as OD strategy

```
1: function pickOD(area, importantAreas)
2:     if $importantAreas$ is $Empty$ then
3:         $[loc2, loc2] = pickODUsingUninformedStrategy(area)$
4:     else
5:         $loc1 = importantAreas.poll()$
6:         $loc2 = getRandomLocation(area)$
7:     $ODpair \leftarrow [loc1, loc2]$
8: return $ODpair$
```

5.3 Software Development

The code is written in Java 9 where I am using Apache Maven[1] as a build automation tool. For code versioning I use Git[2] version control system, specifically the Github[3] repository.

5.3.1 Subplanners

I use Google Maps API [4] for car transport and for walking. For bike and public transport, I have created my own OpenTripPlanner [5] instance, for which I need to provide the data. I use OpenStreetMap[6] geographic data and public transport data of Prague [7] both freely available for public use. Creating my own subplanner is against the metaplanning approach and its creation is not needed in most regions, but there is no freely available public transport subplanner for the area of Prague.

5.3.2 Packages and modules

The project is based on two parts. The first part is a construction of the abstracted transport network and the second is the planning part. Package diagram of the project is shown below 5.3. Construction of abstracted network is based on *graph* and *subplanners* packages, while planning part is based on *planning* package.

Besides I create a module for requesting intermodal journey. The communication is performed via REST API [13], where the response is sent in JSON format. The module was created mainly because of communication with the front-end web application.

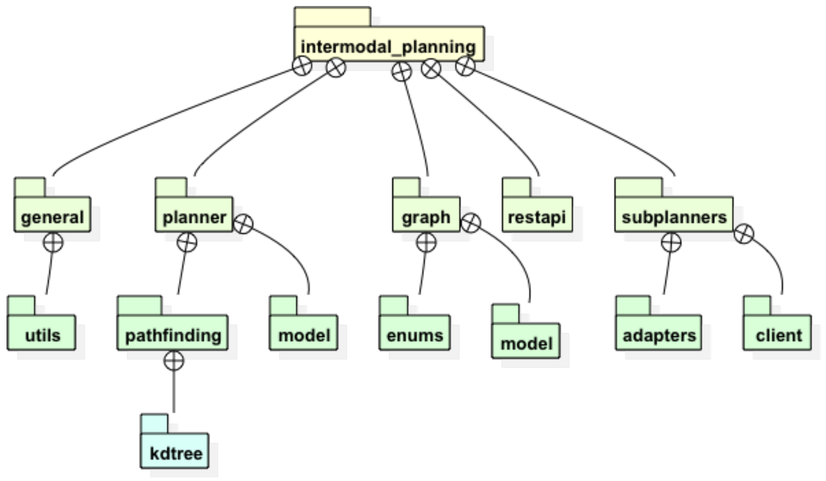


Figure 5.3: Package diagram

1<https://maven.apache.org>
2<https://git-scm.com>
3<https://github.com>
4<https://developers.google.com/maps/documentation/directions/intro>
5<http://www.opentripplanner.org>
6<https://www.openstreetmap.org/>
7<http://opendata.praha.eu/dataset/dpp-jizdni-rady>

5.4 Web Application

I created the web application[8] 5.4 for planning an intermodal journey, that communicates with self-implemented intermodal journey planner. The app was written in HTML. I am using CSS language for content styling and Javascript for ensuring whole app logic including client-server communication. To render interactive maps I am using Mapbox GL JS[9] library. The app allows to choose, which means of transport are to be combined, so it can be restricted to finding unimodal journey only. When the journey is planned, its polyline is shown on the map, and the planning box is expanded with detail description of the journey.

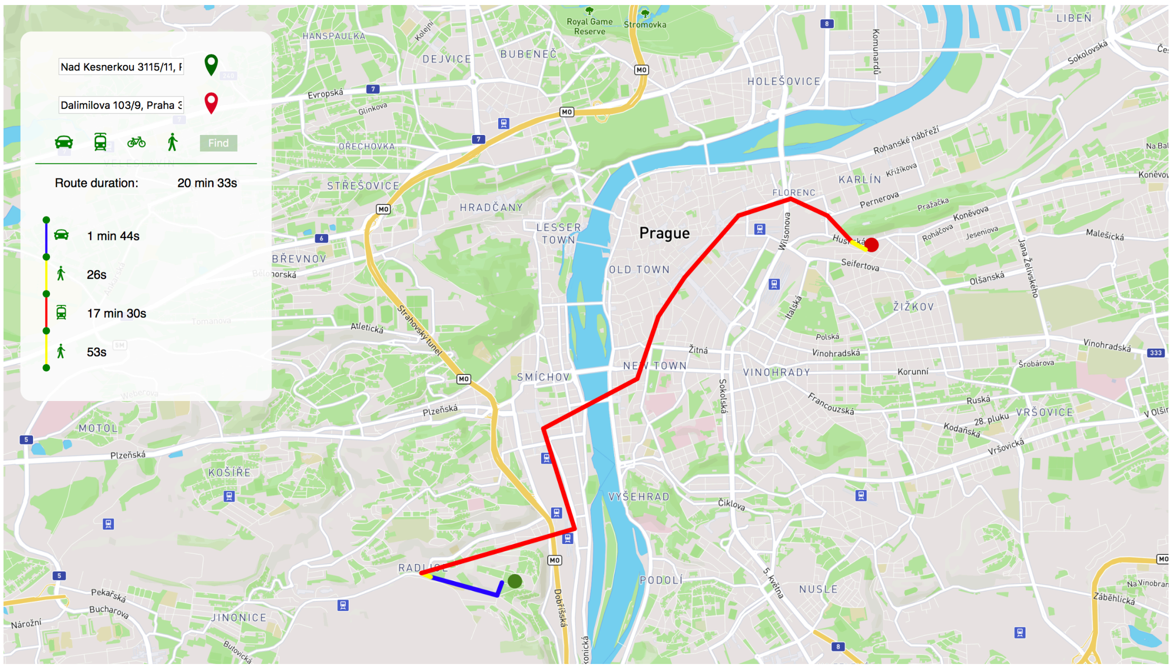


Figure 5.4: Intermodal journey planner web application available on <http://www.ondrejprenek.com/>

[8]<http://www.ondrejprenek.com/>
[9]<https://www.mapbox.com/mapbox-gl-js/api/>

Chapter 6

Evaluation and Deployment

In this chapter, I will evaluate intermodal journey planner on real-world data. Firstly, I will define the environment where the planner will be used. Then, I will show the importance of the intermodal planning in the world of transport. Then, I will construct the abstracted transport network using multiple strategies. Last but not least, I will measure the abstraction quality for every used construction strategy with a different number of subplanner requests.

6.1 Evaluation Settings

For planner testing, I have chosen Prague due to its place suitability. There is a very wide variety of transport modes. Moreover, there is very good infrastructure for public transport, there is an ability for further possible extension by taxis, bike/car sharing and much more.

6.2 Importance of Intermodal Planning

To show the importance of intermodal transport, I evaluate searching shortest path for one thousand randomly-selected OD pairs. Below, in the Figure 6.1, it can be seen that intermodal journey plans are often found as the fastest ones. The decisive factor for selecting the best journey was the travel time. However, if I take into account the travel costs, the intermodal route ratio will get even better as it is practically always cheaper than using the car.

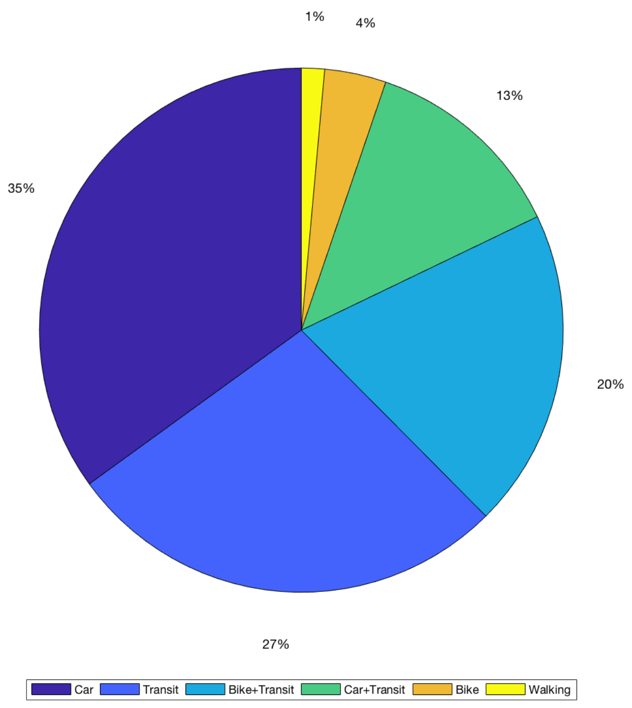


Figure 6.1: Ratio of used transport modes in one thousand journey plans for randomly-selected OD pairs, that show importance of intermodal planning in the world of transport

6.3 Abstracted transport network construction

I construct abstracted transport network using multiple strategies, for which I perform 5000 subplanner requests per each transport mode. I will be continually checking abstraction quality after 500, 1000, 2500 and 5000 subplanner requests. I determine abstraction quality by summarising duration difference of 1000 random routes obtained by metasearch before and after refinement. The lower the sum is, the better is the abstraction quality.

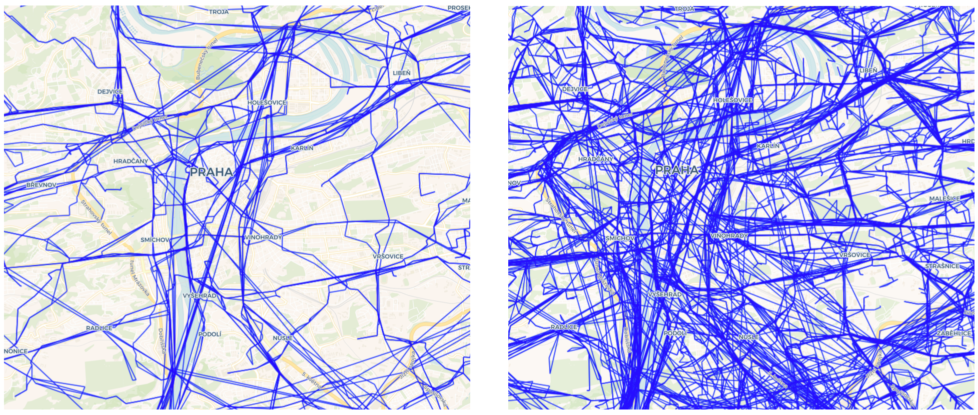

(a) 100 requests (b) 1000 requests

Figure 6.2: The progress of creating the abstracted transport network. One hundred requests are enough to map the main traffic junctions, but in the order of thousands requests are needed to create a quality abstraction

6.4 Uninformed strategies

The results 6.1 of the uninformed-strategies evaluation show that minimal mutual distance between the OD pairs nor chaining OD pairs does not help and their abstraction quality is even worse. On the other hand, in the Figure 6.3, it can be seen that quality difference between the *Random OD* strategy and the *Chaining Random OD* strategy is gradually decreasing, but for a smaller number of subplanner requests the abstraction quality of the *Random OD strategy* is much better.

Table 6.1: Evaluation results of the uninformed construction strategies of the abstracted transport network

Construction Strategy	500 requests	1000 requests	2500 requests	5000 requests
Random OD	58 064s	45 280s	28 612s	20 092s
Random OD Min Mutual Distance	72 450s	51 208s	40 075s	26 530s
Chaining Random OD	67 317s	49 543s	38 324s	22 122s

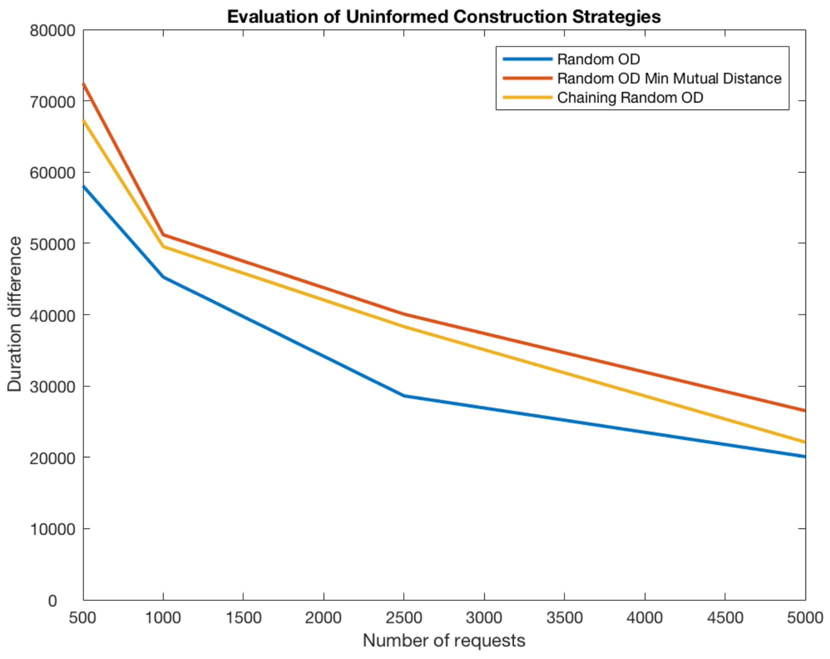


Figure 6.3: Figure of gradually increasing quality of the abstracted transport network, for which construction I use the uninformed strategies

6.5 Informed strategies

The results 6.2 of the informed strategies evaluation proved their expected high efficiency. Normal Distribution of Nodes proved to be the best. On the other hand, the worst informed strategy is the *Uniform Distribution of Nodes*.

As it can be seen, each informed strategy is significantly better than any other uninformed strategy. The Figure 6.4 indicates that it is advisable to follow edges' distribution than nodes' distribution. it is also noticeable that in general the uniform distribution suits better the construction with fewer requests. However, with the increasing number of requests, the normal distribution achieves better results.

Table 6.2: Evaluation results of the informed construction strategies of the abstracted transport network

Construction strategy	500 requests	1000 requests	2500 requests	5000 requests
Uniform Distribution of Nodes	55 436s	36 322s	21 456s	8 541s
Normal Distribution of Nodes	64 250s	34 372 s	18 900 s	4 209s
Uniform Distribution of Edges	36 560s	27 328s	8 734s	3790s
Normal Distribution of Edges	54 150s	25 135s	2 631s	618s

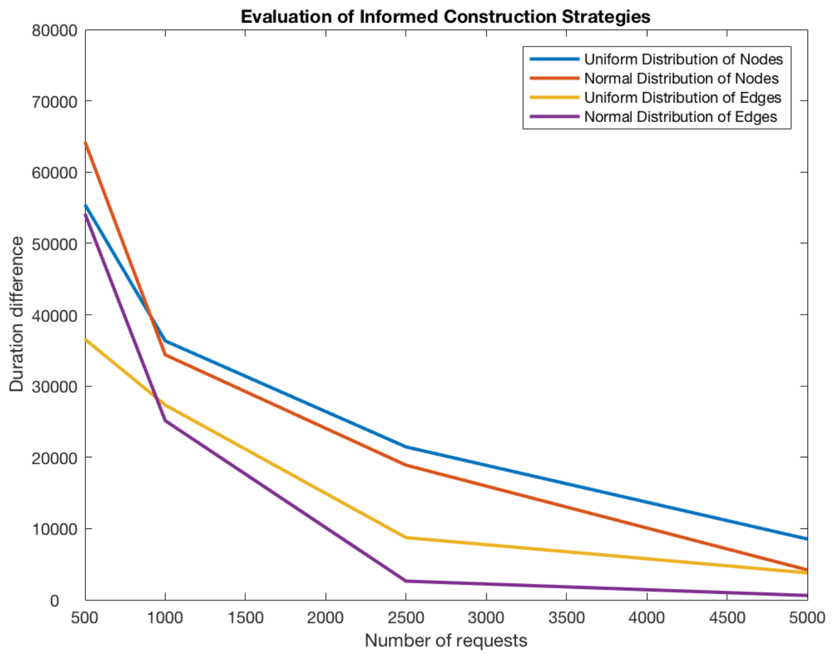


Figure 6.4: Figure of gradually increasing quality of the abstracted transport network, for which construction I use the informed strategies

6.6 Supervised strategies

The evaluation results 6.3 of the supervised strategy has shown the benefit of deployed region ahead-knowledge. The abstraction quality scored better comparing to any of the uninformed strategies. However, it does not achieve the quality of the informed strategies.

Table 6.3: Evaluation results of the supervised construction strategies of the abstracted transport network

Construction strategy	500 requests	1000 requests	2500 requests	5000 requests
Known nodes as OD	54 150s	43 239s	23 890s	19 128s

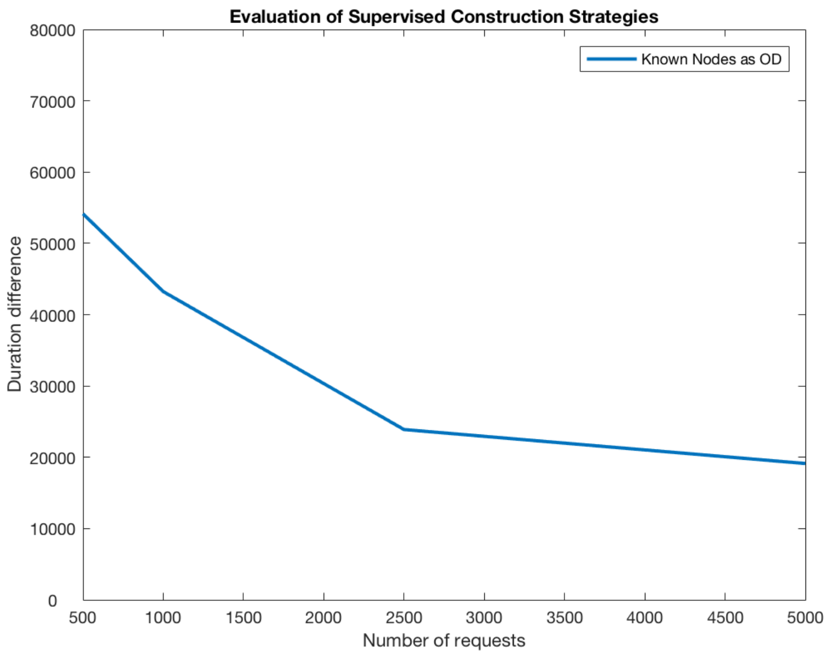


Figure 6.5: Figure of gradually increasing quality of the abstracted transport network, for which construction I use the supervised strategy

6.7 Summary

The evaluation proved that it is worth using advanced strategies. The results demonstrate a superior quality of abstraction for the informed strategies, specifically observing the normal distribution of graph's edges. On the other hand, I got the worst evaluation results for uninformed strategies, specifically for selecting OD pairs with minimal mutual distance. For the supervised strategy, I did not obtain so good result. The supervised strategy was combined with uninformed strategy, which did not lead to the good result, so it offers the opportunity to try supervised strategy in combination with some informed strategy in future work, due to its best evaluation results. It could even lead to the best construction strategy of the abstracted transport network. Evaluation results of all strategies 6.4 and their visual comparison 6.6 can be seen below.

Table 6.4: Evaluation results of all construction strategies of the abstracted transport network

Construction strategy	500 requests	1000 requests	2500 requests	5000 requests
Random OD	58 064s	45 280s	28 612s	20 092s
Random OD Min Mutual Distance	72 450s	51 208s	40 075s	26 530s
Chaining Random OD	67 317s	49 543s	38 324s	22 122s
Uniform Distribution of Nodes	55 436s	36 322s	21 456s	8 541s
Normal Distribution of Nodes	64 250s	34 372 s	18 900 s	4 209s
Uniform Distribution of Edges	36 560s	27 328s	8 734s	3790s
Normal Distribution of Edges	54 150s	25 135s	2 631s	618s
Known nodes as OD	54 150s	43 239s	23 890s	19 128s

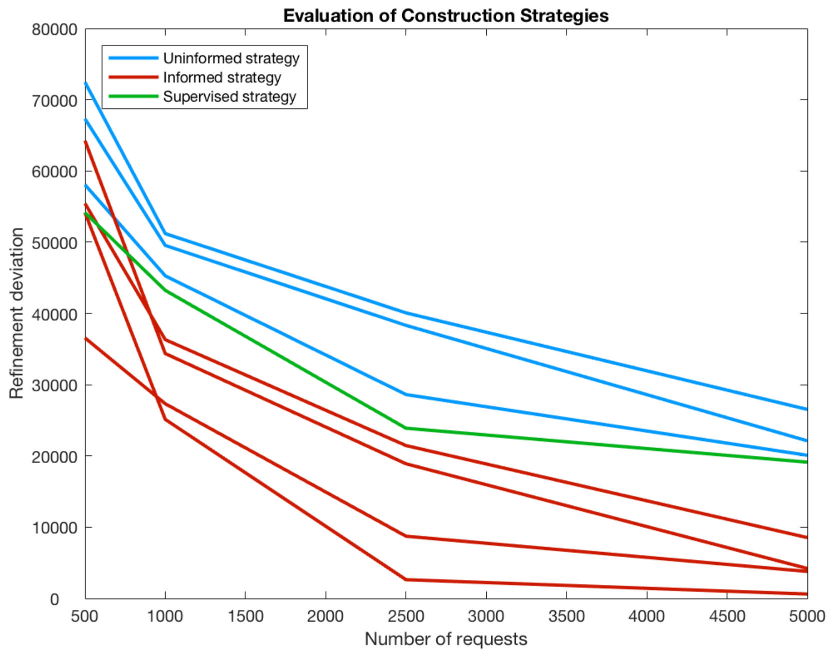


Figure 6.6: Visual comparison of all evaluated construction strategies

Chapter 7

Conclusion

I defined the concept of intermodal journey planner. I aimed to explain a transport network and the shortest path problem. Next, I defined a metaplanning approach consists of the metasearch and the refinement. I specified the abstracted transport network, how its quality is measured and what affects the construction strategy.

Furthermore, I proposed a theoretical approach how to construct component used for solving the shortest path problem - the abstracted transport network. I described all strategies, which I used for constructing abstracted transport network and I described the theoretical approach of measuring the abstraction quality of the abstracted transport network.

I described the process of constructing the abstracted transport network in detail. I aimed to elaborate metaplanning process, where I chose and further described the Dijkstra's algorithm for finding the shortest path.

Then, I evaluated all construction strategies and I measured abstraction quality of constructed abstracted transport network. The best abstraction quality came out for the informed strategies, specifically observing the normal distribution of graph's edges. The worst abstraction quality came out for uninformed strategies, specifically for selecting OD pairs with minimal mutual distance.

The evaluation results proved the importance of using advanced strategies for constructing the abstracted transport network. Their effectiveness was much higher than in case of simple strategies.

Bibliography

[1] Rome2rio: discover how to get anywhere. Dostupné z: <https://www.rome2rio.com/>.

[2] AHUJA, R. K. et al. Faster algorithms for the shortest path problem. *Journal of the ACM (JACM)*. 1990, 37, 2, s. 213–223.

[3] AIFADOPOULOU, G. – ZILIASKOPOULOS, A. – CHRISOHOOU, E. Multiobjective optimum path algorithm for passenger pretrip planning in multimodal transportation networks. *Transportation Research Record: Journal of the Transportation Research Board*. 2007, , 2032, s. 26–34.

[4] ANTSFELD, L. – WALSH, T. Finding multi-criteria optimal paths in multi-modal public transportation networks using the transit algorithm. In *Proceedings of the 19th ITS World Congress*, s. 32, 2012.

[5] BARRETT, C. – JACOB, R. – MARATHE, M. Formal-language-constrained path problems. *SIAM Journal on Computing*. 2000, 30, 3, s. 809–837.

[6] BARRETT, C. et al. Engineering label-constrained shortest-path algorithms. In *International conference on algorithmic applications in management*, s. 27–37. Springer, 2008.

[7] BAST, H. – BRODESSER, M. – STORANDT, S. Result diversity for multi-modal route planning. In *ATMOS-13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems-2013*, 33, s. 123–136. Schloss Dagstuhl Leibniz-Zentrum fuer Informatik, 2013.

[8] BAST, H. et al. Route planning in transportation networks. In *Algorithm engineering*. Springer, 2016. s. 19–80.

[9] BENTLEY, J. L. Multidimensional binary search trees used for associative searching. *Communications of the ACM*. 1975, 18, 9, s. 509–517.

[10] DELLING, D. et al. Computing multimodal journeys in practice. In *International Symposium on Experimental Algorithms*, s. 260–271. Springer, 2013.

[11] DELLING, D. – PAJOR, T. – WERNECK, R. F. Round-based public transit routing. *Transportation Science*. 2014, 49, 3, s. 591–604.

[12] DIJKSTRA, E. W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* December 1959, 1, 1, s. 269–271. ISSN 0029-599X. doi: 10.1007/BF01386390.

[13] FIELDING, R. T. – TAYLOR, R. N. *Architectural styles and the design of network-based software architectures.* 7. University of California, Irvine Doctoral dissertation, 2000.

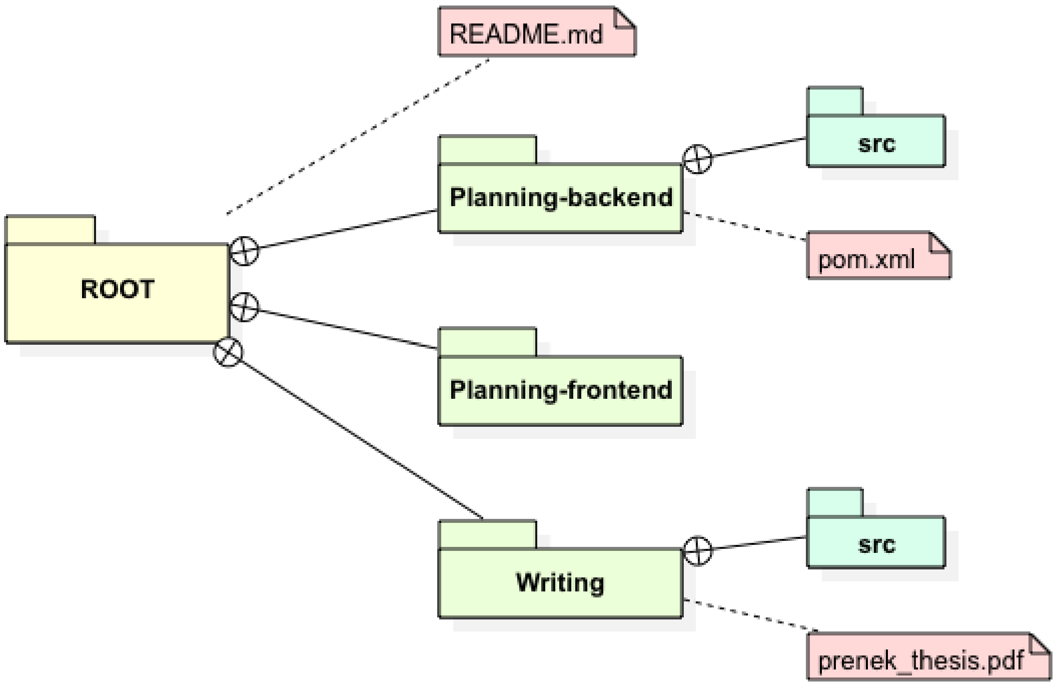
[14] MARTINS, E. Q. V. On a multicriteria shortest path problem. *European Journal of Operational Research.* 1984, 16, 2, s. 236–245.

[15] NYKL, J. Multimodal route planner integration based on abstracted transport system representation. Master's thesis, Czech Technical University, the Czech Republic, February 2016.

[16] web:euSpirit. EU-SPIRIT European travel information network. Dostupné z: <`http://www.eu-spirit.eu`>.

Appendix A

CD Content


README.md
src
Planning-backend
pom.xml
ROOT
Planning-frontend
src
Writing
prenek_thesis.pdf