

Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

# BAKALÁŘSKÁ PRÁCE



Peter Šimora

## **Optimalizace transportu**

Katedra aplikované matematiky

Vedoucí bakalářské práce: prof. RNDr. Luděk Kučera, DrSc.

Studijní program: Informatika

Studijní obor: Obecná informatika

Praha 2012

Ďakujem pánovi prof. RNDr. Luděkovi Kučerovi, DrSc. za odborné vedenie tejto práce, za rady a za čas, ktorý mi počas vypracovávania tejto práce venoval.

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne a výhradne s použitím citovaných prameňov, literatúry a ďalších odborných zdrojov.

Beriem na vedomie, že sa na moju prácu vzťahujú autorské práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona v platnom znení, najmä skutočnosť, že Univerzita Karlova v Praze má právo na uzavretie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe dňa 16.07.2012

Peter Šimora

Název práce: Optimalizace transportu

Autor: Peter Šimora

E-mail autora: petersimora@gtsmail.sk

Katedra / Ústav: Katedra aplikované matematiky

Vedoucí bakalářské práce: prof. RNDr. Luděk Kučera, DrSc.

**Abstrakt:** Vzhledem ke klesajícím zásobám ropy a stoupající populaci je potřebné zabývat se otázkou logistiky. V naší práci jsme zvolili železniční dopravu. Její výhody spočívají ve vybudovaném traťovém systému a vozovém parku. Optimalizaci a měření vykonáváme na železniční síti České republiky. Jednotlivé balíky generujeme na základě počtu obyvatel míst, ve kterých stanice leží. Pro každý balík volíme optimální přepravní trasu. Balíky přiřazujeme k vlakům tak, abychom zmenšili celkový počet najetých kilometrů a ukládáme je do jednotlivých vagonů tak, aby se jich tam vešel co největší možný počet. Měřeními jsme zjistili, že při optimálním počtu vygenerovaných balíků za hodinu je celkový počet najetých kilometrů vlaky v našem algoritmu o 36% menší jako u jednoduchého algoritmu.

**Klíčová slova:** transport, optimalizace, síť, rozvrh, ukládání

**Title:** Transport optimization

**Author:** Peter Šimora

**Author's e-mail:** petersimora@gtsmail.sk

**Department:** Department of Applied Mathematics

**Supervisor:** prof. RNDr. Luděk Kučera, DrSc.

**Abstract:** Considering decreasing reserves of oil and increasing population is necessary to talk about the question of logistics. We have chosen the railway transport. The advantages of railway transport are built railway system and the existing fleet of vehicles. The optimization and measuring are made at the railway network of Czech Republic. Packages are generated with the information of population of town, in which stations lie. For every package is chosen the optimal route. Packages are assigned to trains regard to decreasing the total number of traveled kilometers. Packages are placed to wagons regard to placing the most packages. We have found out that the total number of traveled kilometers in our algorithm is less by 36% than in the simple algorithm, at the optimal number of generated packages per hour.

**Keywords:** transport, optimization, network, schedule, packing

# Obsah

<b>Úvod</b>	<b>1</b>
<b>1. Jednoduchý algoritmus a náš algoritmus</b>	<b>3</b>
<b>2. Železničná sieť</b>	<b>4</b>
2.1 Výber grafu	4
2.2 Popis grafu	4
2.3 Tvorba grafu	5
2.4 Typy tratí	5
<b>3. Generovanie</b>	<b>7</b>
3.1 Generovanie balíkov	7
<b>4. Prepravná trasa</b>	<b>9</b>
4.1 Najkratšia trasa	9
4.2 Najrýchlejšia trasa	10
4.3 Optimálna trasa	10
<b>5. Rozvrhnutie balíkov</b>	<b>13</b>
5.1 Potenciálne prekladacie stanice	13
5.2 Preprava balíka	13
<b>6. Uloženie balíka</b>	<b>16</b>
6.1 Uloženie balíka do vlaku	16
<b>7. Merania a výsledky</b>	<b>17</b>
7.1 Merania algoritmu	17
7.2 Výsledky meraní, diskusia	17
<b>Záver</b>	<b>19</b>
<b>Zoznam použitej literatúry</b>	<b>20</b>
<b>Príloha A: Typy železničných tratí</b>	<b>21</b>
<b>Príloha B: Programátorská dokumentácia</b>	<b>23</b>
Štruktúra programu	23
<b>Príloha C: Užívateľská dokumentácia</b>	<b>25</b>
Obsah CD	25
Práca s aplikáciou	25

# Úvod

Vzhľadom na klesajúce zásoby ropy vo svete a stúpajúcu populáciu ľudstva sa vynára problém logistiky tovarov na celom svete. Táto práca rieši daný problém z ekonomického i ekologického hľadiska.

Pre optimalizáciu prepravy tovarov sme si vybrali železničnú dopravu. Výhody železničnej dopravy spočívajú vo vybudovanom traťovom systéme, vozňovom parku a v nevyčerpacej elektrickej energii.

Súčasný prepravný systém často využíva na prepravu osôb či tovaru pravidelné linky. Naš systém spája prvky pravidelných i nepravidelných liniek tak, aby efektívne prepravoval tovary s ohľadom na prepravný čas, celkový počet najazdených kilometrov a uloženie tovaru.

Aby sme zaručili väčšiu autenticitu, generujeme jednotlivé požiadavky na prepravu v závislosti od počtu obyvateľov daného mesta. Pre každý prepravovaný predmet nájdeme vhodnú prepravnú trasu. Predmet pridelieme vlaku, ktorý ho bude prevážať. Predmet nemusí byť prepravovaný iba jedným vlakom po celej prepravnej trase, ale môže byť prekladaný do iných vlakov za účelom optimalizácie počtu najazdených kilometrov. Uložíme ho do jednotlivých vagónov vlaku tak, aby jeho preloženie či vyloženie nebolo príliš náročné a aby sa do vagónu zmestil čo najväčší možný počet predmetov.

Prvá kapitola vysvetľuje hlavné rozdiely medzi jednoduchým a našim algoritmom.

Druhá kapitola pojednáva o výbere grafu [1] nad ktorým budeme pracovať a uvedie, prečo sme daný graf vybrali. Popíše tvorbu grafu, popíše samotný graf, uvedie základné definície potrebné v nasledujúcom texte. Ukáže vzájomné rozdiely medzi rôznymi typmi železničných tratí a rozdiely medzi železničnými tranzitnými koridormi<sup>1</sup> a ostatnými železničnými traťami.

Tretia kapitola sa zaoberá generovaním balíkov. Pojednáva o veľkostiach jednotlivých balíkov, ako často sa budú balíky generovať a v závislosti na akých ukazovateľoch.

---

<sup>1</sup> Označenie pre hlavnú, modernú železničnú trať, určenú predovšetkým k diaľkovej a tranzitnej osobnej i nákladnej doprave. Niekedy sa slovom koridor označuje i peážna trať.

Štvrtá kapitola sa zameriava na určenie prepravnej trasy pre daný balík. Rozoberá rôzne druhy prepravných trás a vyberá tú najlepšiu z nich.

Piata kapitola popisuje rozdelenie prepravovaných balíkov medzi rôzne typy vlakov a prepravu balíka od počiatkovej až po cieľovú stanicu.

Šiesta kapitola sa zaoberá uložením balíkov do jednotlivých vagónov čo najvhodnejším spôsobom.

V siedmej kapitole je popísané meranie experimentov, výsledky a porovnanie oproti jednoduchému algoritmu.

## 1. Jednoduchý algoritmus a náš algoritmus

Jednoduchým algoritmom rozumieme algoritmus, ktorý využíva k preprave balíkov iba nepravidelné vlakové linky. Balík prepravuje ihneď po jeho vygenerovaní najrýchlejšou prepravnou trasou z počiatočnej do cieľovej stanice. Balíky nie sú vo väčšom počte združované do jedného vlaku, ale každý je prepravovaný samostatne.

Náš algoritmus využíva pravidelné i nepravidelné vlakové linky za účelom zmenšenia celkového počtu najjazdených kilometrov. Balíky sú buď prepravované najrýchlejšou trasou alebo trasou, ktorá bude využívať železničné koridory. Zvolená je tá trasa, ktorá je pre prepravu balíka výhodnejšia. Balíky sú do vagónov vlaku ukladané takým spôsobom, aby sa ich zmestil čo najväčší možný počet.

## 2. Železničná sieť

Kapitola o železničnej sieti hovorí o tom, aký graf vyberáme pre simuláciu a prečo. Udáva základné definície, popisuje graf a jeho tvorbu. Pojednáva o vzájomných rozdieloch medzi traťami.

### 2.1 Výber grafu

Simulácia prepravy tovarov železničnou dopravou môže byť uskutočnená na rôznych typoch grafov. Napríklad:

- bipartitné grafy,
- stromy,
- kubické grafy,
- vrcholovo kritické grafy,
- hranovo kritické grafy.

Uvedené grafy sa často nepodobajú na železničnú sieť. Preto pre simuláciu prepravy tovarov použijeme graf železničnej siete Českej republiky.

### 2.2 Popis grafu

Definícia (Balík): Balíkom sa rozumie obdĺžnik s minimálnymi rozmermi  $10 \times 10 \text{cm}$  a maximálnymi rozmermi  $300 \times 200 \text{cm}$ .

Definícia (Vlak): Vlakom sa rozumie vozidlo poháňané elektrickou energiou, ktoré prepravuje balíky.

Definícia (Stanica): Stanicou rozumieme miesto, v ktorom môžu byť do vlaku balíky naložené, preložené alebo z neho vyložené.

Označenie (Množina staníc): Množinu všetkých staníc budeme označovať  $S$ .

Definícia (Trať): Traťou rozumieme usporiadanú dvojicu  $(u, v)$ , kde  $u, v \in S, u \neq v$ .

Označenie (Množina tratí): Množinu všetkých tratí budeme označovať  $R$ .

Definícia (Koridor): Koridorom rozumieme hlavnú, modernú železničnú trať, určenú predovšetkým k diaľkovej a tranzitnej osobnej i nákladnej doprave.

Označenie (Množina koridorov): Množinu všetkých koridorov budeme označovať  $C$ .

Definícia (Obyčajná trať): Obyčajnou traťou chápeme trať, ktorá nie je koridorom.

Definícia (Železničná sieť): Železničnou sieťou rozumieme graf  $G(S,R)$ . Graf  $G$  bude súvislý a medzi každými dvoma stanicami nebude existovať žiadna trať alebo práve dve trate, aby mohli vlaky jazdiť medzi týmito stanicami súčasne oboma smermi.

Pre každú stanicu uchováваме tieto údaje:

- identifikačné číslo,
- názov,
- zemepisnú šírku,
- zemepisnú dĺžku,
- počet obyvateľov mesta, v ktorom stanica leží.

Pre každú trať uchováваме nasledujúce údaje:

- identifikačné číslo,
- vzdialenosť v kilometroch,
- identifikačné čísla staníc, ktoré trať spája,
- typ trate,
- čísla tranzitných železničných koridorov do ktorých trať patrí.

### 2.3 Tvorba grafu

Železničnú sieť tvoríme pomocou geografického informačného systému s využitím zemepisných súradníc miest, v ktorých stanice ležia a Obr. 1 v prílohe A. Mapu upneme do súradnicového systému Krovak, pridáme všetky stanice a trate spolu s údajmi, ktoré budeme uchovávať.

### 2.4 Typy tratí

Traťové triedy [2] dovoleného zaťaženia v Českej republike (zaťaženie na nápravu / na bežný meter):

- A1 (16 t / 5 t),
- B1 (18 t / 5 t),
- B2 (18 t / 6,4 t),
- C2 (20 t / 6,4 t),
- C3 (20 t / 7,2 t),

- C4 (20 t / 8 t),
- D3 (22,5 t / 7,2 t),
- D4 (22,5 t / 8 t),
- trať mimo prevádzku,
- úzkorozchodná trať,
- trať vo vlastníctve iných subjektov.

Trať mimo prevádzku budeme označovať X1, úzkorozchodnú trať X2, trať vo vlastníctve iných subjektov X3.

Uvedeným typom tratí pridáme nasledovné maximálne povolené rýchlosti:

- X1, X2 – 40 km/h
- A1, B1, B2, C2, C3, C4, X3 – 50 km/h
- D3 – 120 km/h
- D4 – 160 km/h

Trať X1 napriek tomu, že je mimo prevádzku, berieme do úvahy. Podrobné rozdelenie tratí celej Českej republiky do jednotlivých typov je uvedené v prílohe A, na Obr. 1.

*Definícia (Tranzitný železničný koridor):* Tranzitným železničným koridorom rozumieme cestu  $(s_1, c_1, s_2, c_2, \dots, c_{n-2}, s_{n-1}, c_{n-1}, s_n)$ , kde  $\{s_1, s_2, \dots, s_{n-1}, s_n\} \subseteq S$ ,  $\{c_1, c_2, \dots, c_{n-1}\} \subseteq C$  a  $c_1, c_2, \dots, c_{n-2}, c_{n-1}$  majú zhodné číslo koridoru.

V Českej republike sú štyri tranzitné železničné koridory, ich rozdelenie v rámci železničnej siete je podrobne uvedené v prílohe A, na Obr. 2.

### 3. Generovanie

Kapitola o generovaní rozoberá ako vygenerovať počiatočnú a cieľovú stanicu pre balík, ako vygenerovať rozmery balíka, maximálny čas na prepravu a ako často sa generovanie balíkov bude vykonávať.

#### 3.1 Generovanie balíkov

Označme  $p$  generovaný balík. Označme  $\text{Pop}(s)$ , kde  $s \in S$ , počet obyvateľov mesta, v ktorom stanica  $s$  leží. Nech  $P = \sum_{i=1}^n \text{Pop}(s_i)$  je súčet všetkých obyvateľov miest, v ktorých stanice  $s_i, i \in \{1, 2, \dots, n-1, n\}$  ležia. Ďalej nech  $A \in R_{n \times n}$  je štvorcová matica s nulovou diagonálou taká, že hodnota na mieste  $a_{i,j}, i \in \{1, 2, \dots, n-1, n\}, j \in \{1, 2, \dots, n-1, n\}$  sa rovná pravdepodobnosti vygenerovania balíka  $p$  na doručenie zo stanice  $s_i$  do stanice  $s_j$ . Hodnotu na mieste  $a_{i,j}$  vypočítame

ako  $a_{i,j} = \frac{\text{Pop}(s_i) + \text{Pop}(s_j)}{2 \cdot (n-1) \cdot P}$ , aby sme zaručili, že  $\sum_{i,j \in \{1, \dots, n\}} a_{i,j} = 1$ . Generujeme náhodné

číslo  $r \in (0,1)$ . Počiatočnú a cieľovú stanicu zistíme nasledovne:

```
double sum = 0;
for(int i=0; i < n; i++){
    for(int j = 0; j < n; j++){
        sum += ai,j;
        if(sum >= r){
            return new int[] {i, j};
        }
    }
}
```

Programový kód 1

kde  $s_i$  je počiatočná stanica a  $s_j$  je cieľová stanica.

Označme  $p_w$  dĺžku balíka  $p$  a  $p_h$  šírku balíka  $p$ . Generujeme  $p_w \in \{10, 11, \dots, 300\}$  a  $p_h \in \{10, 11, \dots, 200\}$ .

Označme  $p_t$  maximálny čas v dňoch, za ktorý je potrebné balík  $p$  doručiť. Generujeme  $p_t \in \{1, 3, 5\}$ .

Čas medzi vygenerovaním dvoch bezprostredne za sebou nasledujúcich balíkov sa riadi exponenciálnym rozdelením [3] so strednou hodnotou 0,492 min

pri 122 vygenerovaných balíkoch za hodinu, 0,488 min pri 123 balíkoch za hodinu a 0,484 min pri 124 vygenerovaných balíkoch za hodinu.

Generovanie v našom systéme nezahŕňa sezónne požiadavky ani potreby priemyslu, ktoré sa značne odchyľujú od pravdepodobnosti určenej na základe počtu obyvateľov daného mesta, v ktorom sa stanica nachádza.

## 4. Prepravná trasa

Po vygenerovaní balíka je potrebné určiť jeho prepravnú trasu. V tejto kapitole sú rozoberané rôzne typy prepravných trás, ich hľadanie a výber optimálnej z nich.

### 4.1 Najkratšia trasa

*Definícia (Trasa):* Trasou rozumieme cestu  $(s_1, r_1, s_2, r_2, \dots, r_{n-2}, s_{n-1}, r_{n-1}, s_n)$ , kde  $\{s_1, s_2, \dots, s_{n-1}, s_n\} \subseteq S$ ,  $\{r_1, r_2, \dots, r_{n-2}, r_{n-1}\} \subseteq R$ .

Uvažujme železničnú sieť Českej republiky, množinu všetkých staníc  $S$ , množinu všetkých tratí  $R$ , počiatočnú stanicu  $s_{source}$  a cieľovú stanicu  $s_{target}$ . Ďalej označme  $length(r)$  dĺžku trate  $r$  v kilometroch. Problém hľadania najkratšej trasy riešime Dijkstrovým algoritmom [4][5]. Dijkstrov algoritmus vyžaduje pre správne fungovanie nezáporné ohodnotenie hrán, čo náš graf spĺňa. Použitie algoritmu:

```
function Dijkstra(G, s_source, s_target){
    //Inicializácia
    for(each station s in S){
        //Neznáma vzdialenosť z s_source do s
        dist[s] = infinity;
        //Predchádzajúca stanica v optimálnej trase z s_source
        previous[s] = undefined;
    }
    dist[s_source] = 0; //Vzdialenosť z s_source do s_source
    //Všetky stanice sú neoptimalizované, teda sú v Q
    Q = S;
    while(Q is not empty){
        u = station in Q with smallest distance in dist[];
        if(dist[u] = infinity){
            break; //Zostávajúce stanice sú z s_source nedostupné
        }
        if(u==s_target){
            break;
        }
        remove u from Q ;
        for(each neighbor v of u){
            alt = dist[u] + dist_between(u, v);
            if(alt < dist[v]){
                dist[v] = alt;
                previous[v] = u ;
                decrease-key v in Q;
            }
        }
    }
    return dist[];
}
```

Programový kód 2

Zostáva už len získať najkratšiu trasu iteráciou:

```
sp = empty sequence;  
u = starget;  
while(previous[u] is defined){  
    insert u at the beginning of sp;  
    u = previous[u];  
}
```

Programový kód 3

## 4.2 Najrýchlejšia trasa

Najrýchlejšiu trasu spočítame rovnakým spôsobom ako trasu najkratšiu, iba miesto dĺžok tratí použijeme čas, ktorý je potrebný na prejazd danou traťou ľubovoľným vlakom maximálnou rýchlosťou. Pre uvedený čas pre trať  $r$  platí, že  $\text{time}(r) = \frac{\text{length}(r)}{\text{speed}(r)}$ , kde  $\text{length}(r)$  je dĺžka trate  $r$  v kilometroch a  $\text{speed}(r)$  je maximálna povolená rýchlosť na trati  $r$ . Najrýchlejšiu trasu zo stanice  $u$  do stanice  $v$  budeme označovať  $\text{fastest}(u, v)$ .

## 4.3 Optimálna trasa

Aby sme optimalizovali celkový počet najjazdených kilometrov, ponúka sa myšlienka zväzať balík ku koridoru. Tu ho preložiť do pravidelnej vlakovej linky, jazdiacej výhradne po tranzitnom železničnom koridore. Balík koridormi prevážať do určitej stanice, odkiaľ bude preložený do iného vlaku a prevezený až do cieľovej stanice. Podrobné informácie o tom, ako balíky priradiť jednotlivým vlakom sú uvedené v nasledujúcej kapitole.

Ešte pred prijatím prvej požiadavky na prepravu balíka spočítajme najrýchlejšiu trasu medzi každými dvoma stanicami grafu  $G$  Floyd-Warshallovým algoritmom [6]. Floyd-Warshallov algoritmus vyžaduje nezáporné cykly, čo náš graf spĺňa, keďže všetky hrany sú ohodnotené nezáporne.

Nech existuje práve vygenerovaný balík  $p$  s počiatočnou stanicou  $s_{\text{source}}$  a cieľovou stanicou  $s_{\text{target}}$ . Nájďme všetky stanice  $u_1, u_2, \dots, u_{n-1}, u_n$  také, že  $\forall i \in \{1, 2, \dots, n-1, n\} \exists v \in S, u_i \neq v: (u_i, v) \in C$ . Pre každú takto vybranú stanicu dohľadáme najrýchlejšiu trasu z  $s_{\text{source}}$  do tejto stanice, ktorú sme spočítali Floyd-Warshallovým algoritmom. Označme  $\text{time}(j)$  čas, za ktorý prejde ľubovoľný

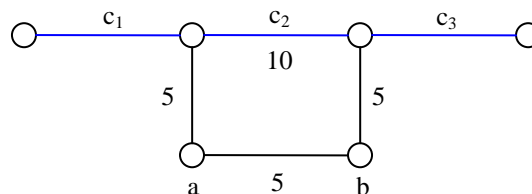
vlak trasu  $j$  maximálnou rýchlosťou. Zo všetkých staníc  $u_1, u_2, \dots, u_{n-1}, u_n$  vyberieme takú stanicu  $u_f$ , ktorej  $\text{time}(\text{fastest}(s_{\text{source}}, u_f))$  je minimálny.

Ďalej nájdeme všetky stanice  $m_1, m_2, \dots, m_{n-1}, m_n$  také, že  $\forall i \in \{1, 2, \dots, n-1, n\} \exists o \in S, m_i \neq o : (m_i, o) \in C$ . Pre každú takto vybranú stanicu dohľadáme najrýchlejšiu trasu z tejto stanice do  $s_{\text{target}}$ . Zo všetkých staníc  $m_1, m_2, \dots, m_{n-1}, m_n$  vyberieme takú stanicu  $m_f$ , ktorej  $\text{time}(\text{fastest}(m_f, s_{\text{target}}))$  je minimálny.

Nakoniec nájdeme najrýchlejšiu trasu zo stanice  $u_f$  do stanice  $m_f$ :  $(u_f, q_1, q_2, \dots, q_{n-1}, q_n, m_f); (u_f, q_1) \in C, (q_1, q_2) \in C, \dots, (q_{n-1}, q_n) \in C, (q_n, m_f) \in C$  a označme ju  $\text{corridsPath}(u_f, m_f)$ .

Našli sme trasu  $(\text{fastest}(s_{\text{source}}, u_f), \text{corridsPath}(u_f, m_f), \text{fastest}(m_f, s_{\text{target}}))$ , ktorú budeme nazývať najrýchlejšia trasa s použitím koridoru.

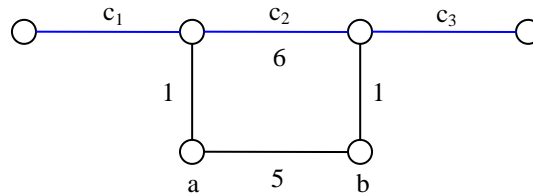
Najrýchlejšia trasa s použitím koridoru i najrýchlejšia trasa majú svoje nevýhody. Nevýhodou najrýchlejšej trasy s použitím koridoru je, že síce vždy využije koridor po ktorom premávajú pravidelné vlakové linky, ale môže zbytočne plytvať obyčajnými traťami, kde sa balíky prepravujú linkami nepravidelnými. Problém tejto trasy vidieť na Obr. 3. Nech  $c_1, c_2, c_3 \in C$ , nech čísla pri tratiach znamenajú čas, za ktorý vlak prejde trať maximálnou rýchlosťou. Vyhľadajme najrýchlejšiu trasu zo stanice  $a$  do stanice  $b$  s využitím koridorov. Súčet časov obyčajných tratí je rovno 10, pričom výhodnejšie by v tomto prípade bolo použitie najrýchlejšej trasy. Tu by súčet časov obyčajných tratí bol rovný 5. Časy na koridoroch zanedbávame, keďže po nich jazdia pravidelné vlakové linky neustále a je vhodné ich využiť, nie ale nutne vždy.



Obr. 3 – Problém najrýchlejšej trasy s použitím koridoru

Problémom najrýchlejšej trasy je, že niekedy plytvá použitím obyčajných tratí a bolo by lepšie využiť koridor. Problém tejto trasy je vidieť na Obr. 4. Vyhľadajme

najrýchlejšiu trasu zo stanice a do stanice b. Súčet časov obyčajných tratí je v tomto prípade 5, pričom výhodnejšie by bolo použitie najrýchlejšej trasy s využitím koridoru, kde je súčet časov obyčajných tratí rovný 2.



Obr. 4 – Problém najrýchlejšej trasy

Pre hore uvedené problémy uvažujeme vždy najrýchlejšiu trasu i najrýchlejšiu trasu s použitím koridorov a vyberáme tú z nich, ktorá má menší súčet časov obyčajných tratí. V prípade rovnosti použijeme najrýchlejšiu trasu.

## 5. Rozvrhnutie balíkov

Kapitola o rozvrhovaní balíkov definuje potenciálne prekladacie stanice a stavy balíka. Určuje ako balík prevážať pomocou nepravidelných vlakových liniek po obyčajných tratiach a pravidelných vlakových liniek po tranzitných železničných koridoroch.

### 5.1 Potenciálne prekladacie stanice

Označením  $\text{number}(c), c \in C$  označujeme číslo tranzitného železničného koridoru, ktorého je koridor  $c$  súčasťou.

Nech je pre balík  $p$  daná prepravná trasa  $j: (s_1, r_1, s_2, r_2, \dots, r_{n-2}, s_{n-1}, r_{n-1}, s_n)$ ,  $s_1, s_2, \dots, s_{n-1}, s_n \in S; r_1, r_2, \dots, r_{n-2}, r_{n-1} \in R$ , ktorú sme našli algoritmom popísaným v predchádzajúcej kapitole. Potom stanicu  $s_i, i \in \{1, 2, \dots, n-1, n\}$  nazveme potenciálnou prekladacou stanicou ak platí, že  $(r_i \in C \wedge r_{i+1} \in R) \vee (r_i \in R \wedge r_{i+1} \in C) \vee (\text{number}(r_i) \neq \text{number}(r_{i+1}), r_i \in C, r_{i+1} \in C)$ .

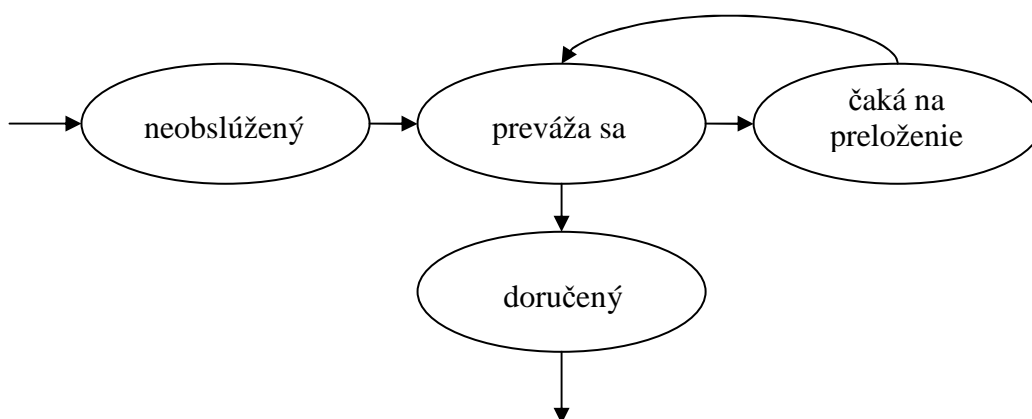
### 5.2 Preprava balíka

Po každom z tranzitných železničných koridorov neustále premáva vopred daný počet vlakov. Vlaky premávajúce po obyčajných tratiach doručujú balíky ku koridorom, od koridorov ku cieľovým staniciam alebo priamo z počiatočnej k cieľovej stanici. Vlaky premávajúce po obyčajných tratiach premávajú nepravidelne.

Pre každý balík definujeme štyri stavy:

- neobslužený – balík ešte nebol naložený do žiadneho vlaku,
- preváža sa – balík je naložený vo vlaku,
- čaká na preloženie – balík čaká na preloženie do iného vlaku na niektorej, z potenciálne prekladacích staníc na jeho prepravnej trase,
- doručený – požiadavka na doručenie balíka je splnená.

Prechod medzi stavmi balíka možno popísať konečným automatom [7] na Obr. 5.



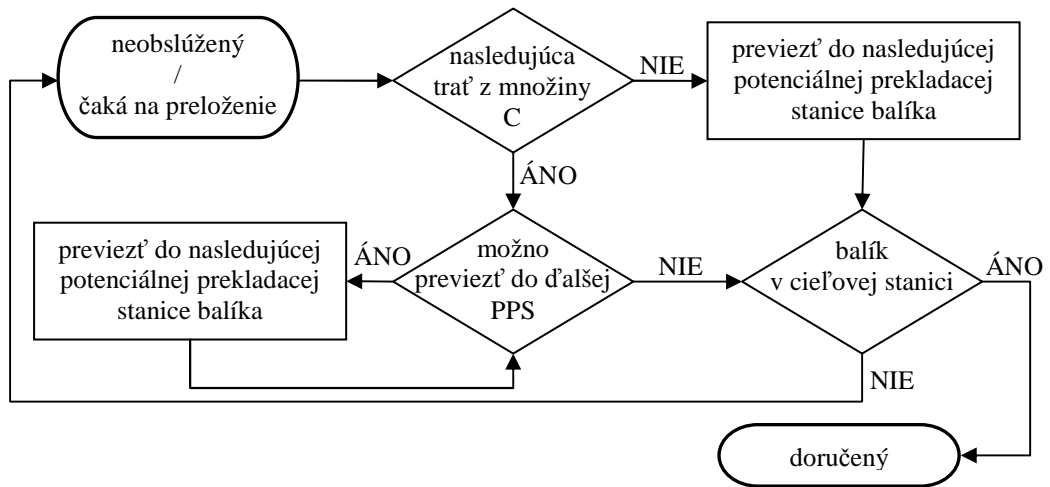
Obr. 5 – Prechod medzi jednotlivými stavmi balíka

Ak je nasledujúca trať prepravnej trasy neobslúženého balíka  $p$  z množiny  $C$ , bude tento balík prepravený vlakom premávajúcim po tranzitnom železničnom koridore, ak sa do neho zmestí, do najbližšej potenciálnej prekladacej stanice. Ide o pravidelnú vlakovú linku, ktorá nemusí balík obslúžiť ihneď. Po prepravení balíka  $p$  do danej potenciálnej prekladacej stanice, prepraví rovnaký vlak balík  $p$  aj do nasledujúcej potenciálnej prekladacej stanice, ak sa táto stanica nachádza na prepravnej trase vlaku. Vlak prepravuje balík  $p$  do nasledujúcej potenciálnej prekladacej stanice až dovtedy, pokiaľ sa táto stanica nachádza na prepravnej trase vlaku. V okamihu keď sa najbližšia potenciálna prekladacia stanica na prepravnej trase vlaku nenachádza a súčasne je vlak v inej potenciálnej prekladacej stanici, je balík vyložený. Ak sa jedná o cieľovú stanicu balíka, balík je doručený, v opačnom prípade balík čaká na preloženie.

Ak je nasledujúca trať prepravnej trasy neobslúženého balíka z množiny  $R$ , prepraví ho nepravidelná traťová linka, ak sa do vlaku zmestí, do najbližšej potenciálnej prekladacej stanice. Balík je obslúžený ihneď novým vlakom jazdiacim po obyčajných tratiach a prepraví ho do najbližšej potenciálnej prekladacej stanice balíka. Vlak štartuje z počiatkovej stanice balíka. V prvej potenciálnej prekladacej stanici balíka je balík vyložený. Ak sa jedná o cieľovú stanicu balíka, balík je doručený, inak balík čaká na preloženie.

Rovnaký postup z predchádzajúcich dvoch odsekov pre neobslúžený balík sa použije aj pre balík, ktorý čaká na preloženie.

Postup na prepravu balíka vidieť aj na vývojovom diagrame, na Obr. 6.



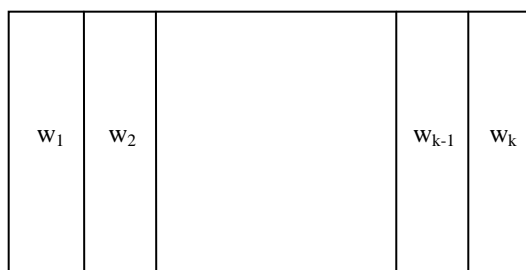
Obr. 6 – Postup na prepravu balíka

## 6. Uloženie balíka

Kapitola o uložení balíkov pojednáva o ukladaní balíkov do jednotlivých vagónov vlaku.

### 6.1 Uloženie balíka do vlaku

Každý vlak má vopred stanovený pevný počet vagónov. Vagónom chápeme obdĺžnik vopred stanovených rozmerov. Každé dva vagóny majú rovnaké rozmery. Každý vagón rozdelíme na zvislé obdĺžniky rovnakých rozmerov, ako je uvedené na Obr. 7. Celkový počet zvislých obdĺžnikov označíme  $n$ .



Obr. 7- Rozdelenie vagónu na časti

Balíky ukladáme do jednotlivých vagónov a jednotlivých častí vlaku algoritmom First-fit [8] nasledovne:

```
for (All objects  $i = 1, 2, \dots, m$ ){
  //hľadanie zvislého obdĺžnika, do ktorého sa balík  $i$  zmestí
  for (All bins  $j = 1, 2, \dots, n$ ){
    if (Object  $i$  fits in bin  $j$ ){
      Pack object  $i$  in bin  $j$ .
      break;
    }
  }
}
```

Programový kód 4

## 7. Merania a výsledky

Siedma kapitola popisuje meranie experimentov, uvádza výsledky meraní, a porovnáva dosiahnuté výsledky s jednoduchým algoritmom.

### 7.1 Merania algoritmu

Uvažujeme pravidelné traťové linky jazdiace po tranzitných železničných koridoroch a nepravidelné traťové linky jazdiace po obyčajných tratiach. Pre každý tranzitný železničný koridor uvažujeme práve jeden vlak (v praxi možno stanoviť väčší počet, podľa počtu požiadaviek na prepravu balíkov). Každý vlak jazdiaci po tranzitných železničných koridoroch obsahuje 60 zvislých obdĺžnikov zmienených v predchádzajúcej kapitole. Každý zvislý obdĺžnik má rozmery 300 x 200 cm. Pre každý vlak jazdiaci po obyčajných tratiach uvažujeme 5 takýchto obdĺžnikov. Merania uskutočňujeme postupne pre rôzne stredné hodnoty času medzi vygenerovaním dvoch bezprostredne za sebou nasledujúcich balíkov. A to 0,492 min, 0,488 min a 0,484 min. Čo je 122, 123, a 124 vygenerovaných balíkov za hodinu. Pozorujeme celkový počet najjazdených kilometrov, splnenie/nesplnenie času na doručenie balíkov. Výsledky nášho algoritmu porovnáваме vzhľadom k jednoduchému algoritmu. Kde jednoduchým algoritmom rozumieme algoritmus, ktorý každý balík síce prepravuje najrýchlejšou prepravnou trasou, ale balíky sa nezdrúžujú vo väčšom počte do jedného vlaku (každý balík je prepravovaný samostatne v jednom vlaku).

### 7.2 Výsledky meraní, diskusia

Porovnanie celkového počtu najjazdených kilometrov u nášho algoritmu vzhľadom k jednoduchému algoritmu je uvedené v Tabuľke 1.

	Náš algoritmus	Jednoduchý algoritmus
122 balíkov/hodinu	64,42 %	100 %
123 balíkov/hodinu	60,56 %	100 %
124 balíkov/hodinu	57,10 %	100 %

Tabuľka 1 – Pomer celkového počtu najjazdených kilometrov nášho algoritmu vzhľadom k jednoduchému algoritmu

Vo všetkých uvedených prípadoch došlo v našom algoritme k zlepšeniu oproti jednoduchému algoritmu o viac ako tretinu.

Porovnanie počtu doručených balíkov v časovom limite v našom algoritme vzhľadom k jednoduchému algoritmu je uvedené v tabuľke 2.

	Náš algoritmus	Jednoduchý algoritmus
122 balíkov/hodinu	100 %	100 %
123 balíkov/hodinu	99,83 %	100 %
124 balíkov/hodinu	99,76 %	100 %

Tabuľka 2 – Pomer počtu doručených balíkov v našom algoritme vzhľadom k jednoduchému algoritmu

Z Tabuľky 2 vyplýva, že optimálne doručovanie balíkov v našom algoritme nastáva ak je počet vygenerovaných balíkov za hodinu medzi hodnotami 122 a 123. Čo znamená úsporu v celkovom počte najazdených kilometrov približne o 36 %. Skutočná úspora v celkovom počte najazdených kilometrov by mala byť ešte väčšia, pretože v reálnom svete majú balíky tri rozmery a balíky možno ukladať na seba.

## Záver

Vytvorili sme a popísali algoritmus, ktorý pre každý prepravovaný predmet nájde vhodnú prepravnú trasu. Predmet prideli medzi pravidelné a nepravidelné vlakové linky tak, aby bol celkový počet najazdených kilometrov nižší ako u jednoduchého algoritmu. Predmet je v každom vlaku, ktorý ho preváža uložený takým spôsobom, aby sa do vlaku zmestil čo najväčší možný počet prepravovaných predmetov.

Uskutočnili sme merania, ktoré ukázali, že vlaky premávajúce v našom algoritme najazdia pri optimálnom počte vygenerovaných balíkov približne o 36 % menej ako vlaky v jednoduchom algoritme. Výsledky v praxi by mali byť ešte lepšie, keďže balíky majú tri rozmery (možno ich ukladať na seba) a v našej práci sme pracovali iba s dvoma rozmermi.

V ďalšom pokračovaní práce možno pracovať s trojrozmernými balíkmi a pridať ďalšie heuristiky, ktoré by získané výsledky mohli vylepšiť.

## Zoznam použitej literatúry

- [1] MATOUŠEK, Jiří a NEŠETŘIL, Jaroslav. *Kapitoly z diskrétní matematiky*. 4., upr. a dopl. vyd. V Praze: Karolinum, 2009, s. 111-158. ISBN 978-80-246-1740-4.
- [2] Správa železniční dopravní cesty. [online]. [cit. 2012-06-13]. Dostupné z: <http://www.szdc.cz/o-nas/zeleznicni-mapy-cr.html>
- [3] ZVÁRA, Karel a ŠTĚPÁN, Josef. *Pravděpodobnost a matematická statistika*. Vyd. 4. Praha: Matfyzpress, 2006, s. 72, 96. ISBN 80-86732-71-7.
- [4] DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*. 1959, roč. 1, č. 1, s. 269-271. ISSN 0029-599x. DOI: 10.1007/BF01386390. Dostupné z: <http://www.springerlink.com/index/10.1007/BF01386390>
- [5] KUČERA, Luděk. *Algovize: aneb procházka krajinou algoritmů*. 1. vyd. Praha: Univerzita Karlova, 2009, s. 138-145. ISBN 978-80-902938-5-4.
- [6] FLOYD, Robert W. Algorithm 97: Shortest path. *Communications of the ACM*. roč. 5, č. 6, s. 345-. ISSN 00010782. DOI: 10.1145/367766.368168. Dostupné z: <http://portal.acm.org/citation.cfm?doid=367766.368168>
- [7] CHYTIL, Michal. *Automaty a gramatiky*. 1. vyd. Praha: Státní nakladatelství technické literatury, 1984, 331 s.
- [8] Berkey, J.O., and Wang, P.Y. (1985), "Two-dimensional finite bin-packing algorithms", *Journal of the Operational Society* 38, 423-429.
- [9] Java Downloads for All Operating Systems. [online]. [cit. 2012-07-15]. Dostupné z: <http://java.com/en/download/manual.jsp>

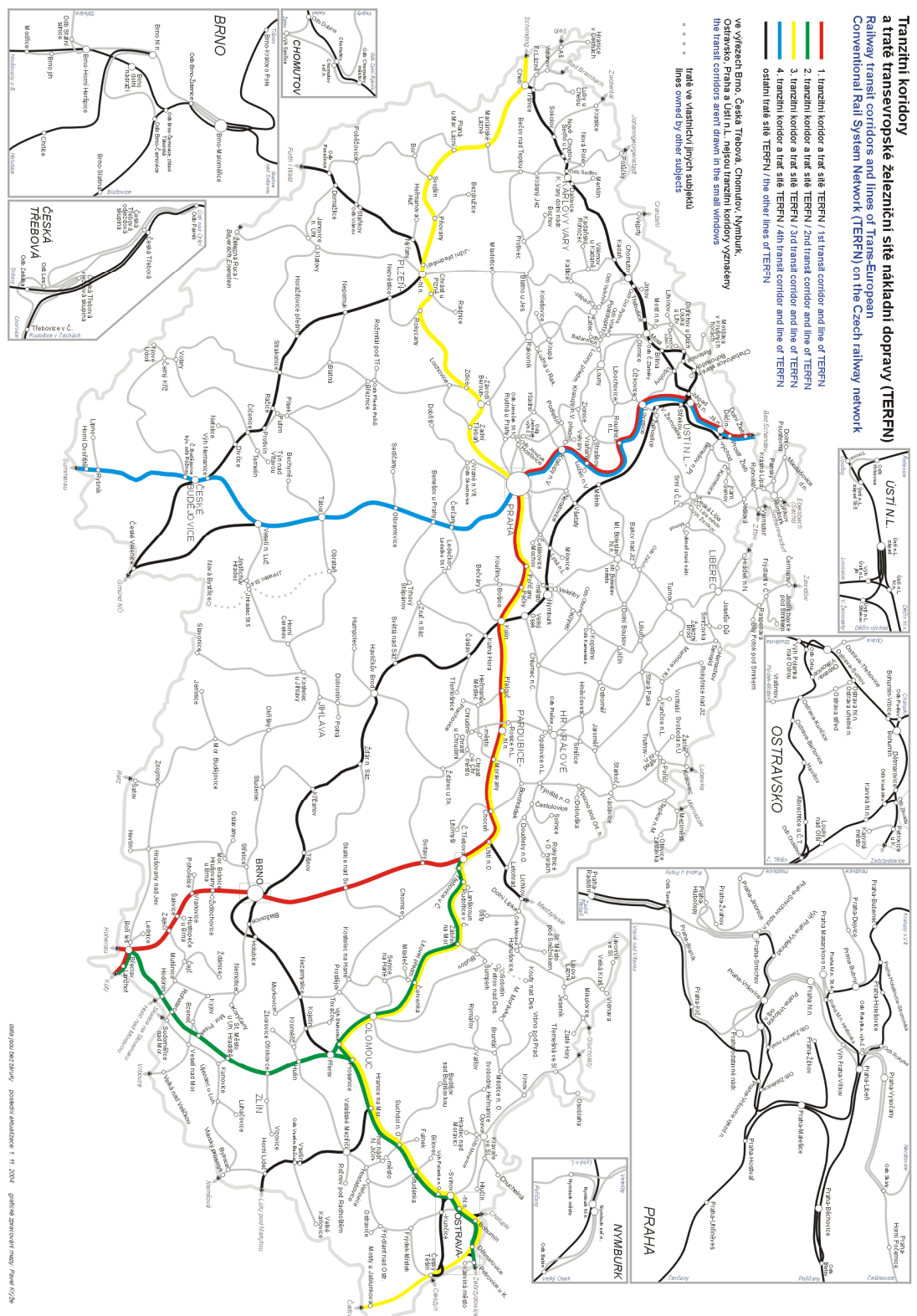


**Tranzitní koridory  
a trase trans-evropské železniční sítě nákladní dopravy (TERFN)  
Railway transit corridors and lines of Trans-European  
Conventional Rail System Network (TERFN) on the Czech railway network**

- 1. tranzitní koridor a trat' síť TERFN / 1st transit corridor and line of TERFN
- 2. tranzitní koridor a trat' síť TERFN / 2nd transit corridor and line of TERFN
- 3. tranzitní koridor a trat' síť TERFN / 3rd transit corridor and line of TERFN
- 4. tranzitní koridor a trat' síť TERFN / 4th transit corridor and line of TERFN
- ostatní tratě sítě TERFN / the other lines of TERFN

ve výřezech Brno, Česká Třebová, Chomutov, Nymburk,  
Ostava a Praha a Ústí n. L., nejsou tranzitní koridory vyznačeny  
the transit corridors aren't drawn in the small windows

trati ve vlastnických jiných subjektech  
lines owned by other subjects



Obr. 2 - Tranzitní koridory

odpovědný redaktor: projektová analýza 1. října 2014 grafická zpracování mapy: PAVEL VEJŠKA

## Príloha B: Programátorská dokumentácia

Program je napísaný v jazyku Java verzie 7. Je vytvorený a odladený v operačnom systéme Windows. Môže byť spustený i v operačnom systéme Linux, Solaris i Apple.

Neoddeliteľnou súčasťou programátorskej dokumentácie je Javadoc a komentáre v zdrojových kódach. Javadoc sa nachádza na priloženom CD v adresári *javadoc/*.

### Štruktúra programu

Program obsahuje nasledujúce balíčky:

- < default package > - Obsahuje ikony a súbory potrebné pre vytvorenie mapy. V súbore *boundaries.mif* sa nachádzajú súradnice bodov hraníc Českej republiky. Súbor *railways.bcp* obsahuje údaje o jednotlivých tratiach, ako sú čísla staníc ktoré trate spájajú, súradnice staníc, vzdialenosť v kilometroch, typ trate, čísla koridorov do ktorých trať patrí. V súbore *railways.mif* sa nachádzajú súradnice tratí potrebné pre ich zobrazenie na mape. Súbor *stations.bcp* obsahuje informácie o staniaciach ako sú súradnice staníc, názov, populácia.
- algorithms – Balíček obsahuje použité algoritmy pre generovanie balíkov, obsluhu balíkov, pre vyhľadanie najrýchlejšej trasy. Obsahuje i triedy, ktoré reprezentujú balíky a vlaky.
- constants – V balíčku sa nachádzajú referencie na fonty a ikony.
- extract – Triedy a algoritmy potrebné na parsovanie súborov z *default package*.
- gui – Grafické rozhranie, menu, kreslenie mapy, tlačidlá.
- gui.listeners – Triedy, ktoré obsahujú akcie, ktoré reagujú na rôzne udalosti aktivované užívateľom.
- preparation – Prípravné práce pred spustením simulácie ako je výpočet pravdepodobností s akou sa daná stanica stane počiatočnou alebo cieľovou stanicou pre vygenerovaný balík.
- transpotoptimization – main

V programe súčasne existuje niekoľko vláken pri simulácii. Jedno vlákno generuje požiadavky, ďalšie požiadavky obsluhuje a iné zobrazuje aktuálnu pozíciu vlakov na mape.

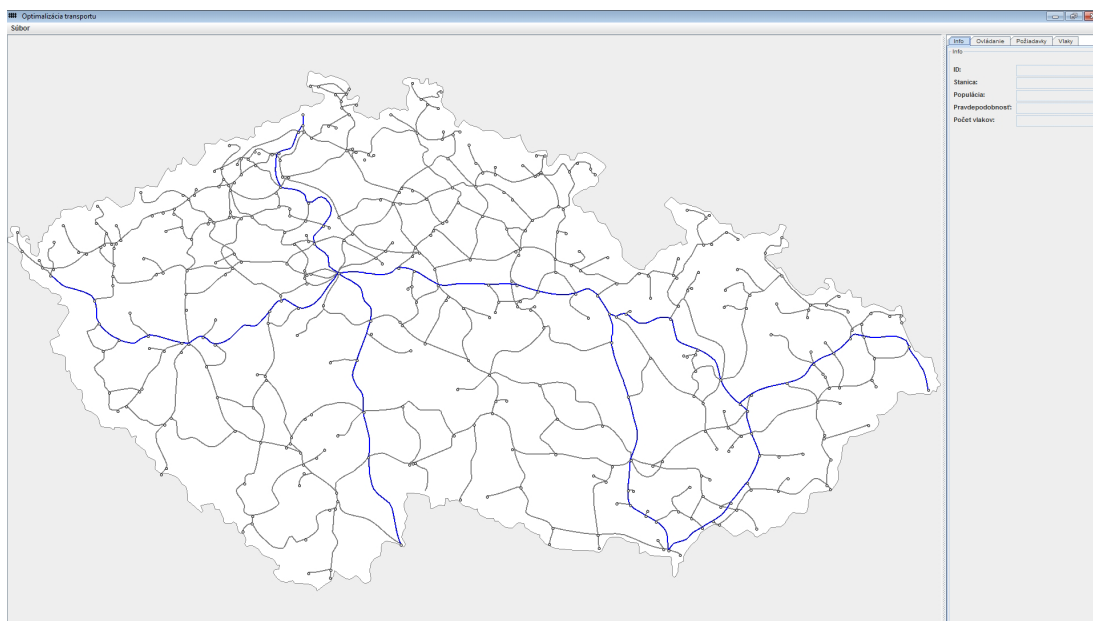
## Príloha C: Uživatelská dokumentácia

### Obsah CD

Pre správny chod aplikácie je potrebné nainštalovať Javu verzie 7, ktorá je umiestnená v adresári *install/java/*. Možno tu nájsť Javu pre operačný systém Linux i Windows. Podporu Javy pre ostatné operačné systémy možno nájsť na stránkach Javy [9]. Samotná aplikácia sa nachádza v adresári *install/*. Dokumentácia Javadoc sa nachádza v adresári *javadoc/*. Adresár *sources/* obsahuje zdrojové kódy aplikácie.

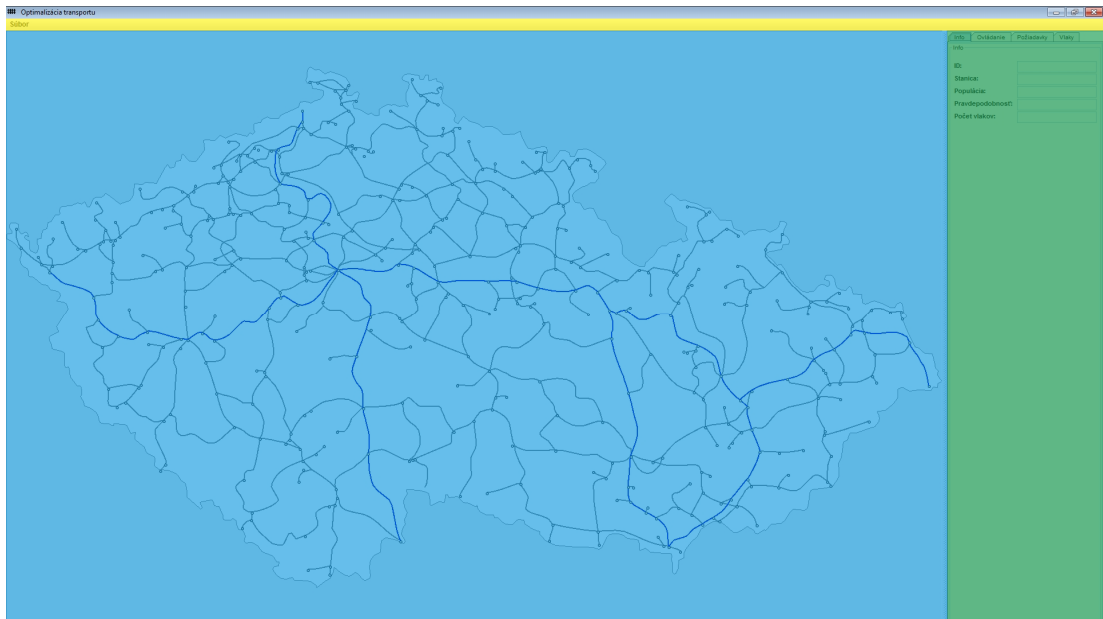
### Práca s aplikáciou

Po nainštalovaní Javy verzie 7, nainštalovaní samotnej aplikácie a jej spustení sa zobrazí okno uvedené na Obr. 8.



Obr. 8 – Vzhľad aplikácie

Okno na Obr.8 si pre jednoduchší popis rozdelíme na tri farebné časti (Obr. 9). Žltá časť obsahuje horné menu, modrá časť pokrýva mapu a zelená obsahuje viacero kariet.



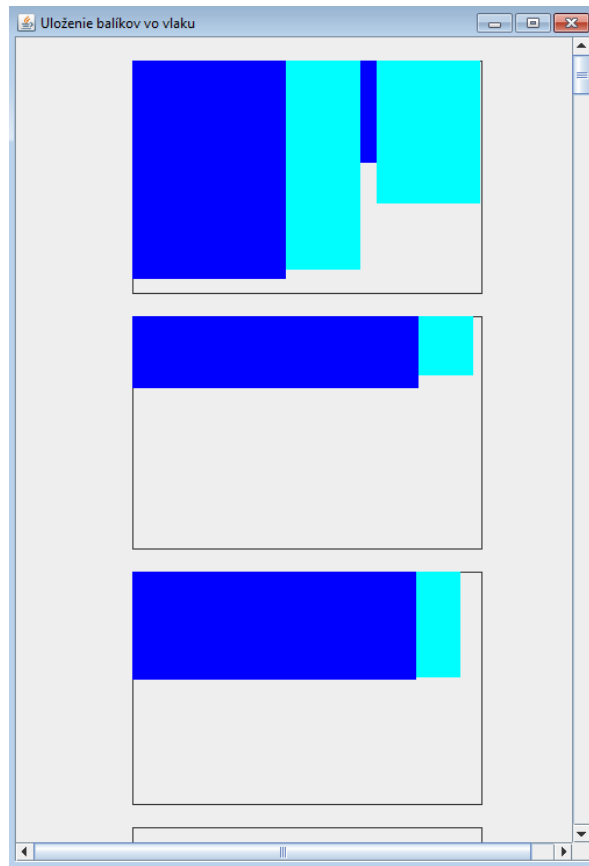
Obr. 9 – Rozdelenie okna na časti

V modrej časti možno na jednotlivé stanice kliknutím získať informácie o danej vlakovej stanici, ako sú: ID, názov, počet obyvateľov, pravdepodobnosť s akou sa stanica stane počiatkovou alebo cieľovou stanicou balíka, počet dostupných vlakov v stanici. Uvedené informácie sa zobrazia v zelenej časti na karte „Info“.

Pre spustenie simulácie generovania a doručovania balíkov je potrebné v zelenej časti na karte „Ovládanie“ nastaviť požadovaný počet generovaných balíkov za hodinu a rýchlosť programu. Rýchlosť 1 znamená generovanie a doručovanie balíkov rovnako rýchlo ako v realite. Rýchlosť 100 znamená, že program beží oproti skutočnosti 100-krát rýchlejšie. Po nastavení počtu generovaných balíkov a rýchlosti simulácie stlačíme tlačidlo „Štart“, ktorým sa zaháji simulácia. Simuláciu možno zastaviť tlačidlom „Stop“.

Vygenerované balíky sa zobrazujú v zelenej časti na karte „Požiadavky“, pri každom balíku sa nachádza tlačidlo „Info“, stlačením ktorého získame o balíku bližšie informácie. V prípade, že chceme tieto informácie aktualizovať, stlačíme tlačidlo „Aktualizácia“.

Obsluhujúce vlaky sa zobrazujú v zelenej časti na karte „Vlaky“. Pri každom možno kliknutím na tlačidlo „Info“ získať podrobnejšie informácie o obsluhujúcom vlaku. Otvorí sa tak okno s informáciami, kde možno zobrazit' i aktuálne uloženie balíkov vo vlaku podobné ako je na Obr. 10.



Obr. 10 – Ukážka uloženia balíkov vo vlaku

Prvé štyri vlaky jazdia po koridoroch vyznačených na mape modrou farbou. Ostatné vlaky jazdia po obyčajných tratiach.

Aktuálny stav simulácie (balíkov i vlakov), možno uložiť do textových súborov pomocou tlačidiel „Export požiadaviek“ a „Export vlakov“ umiestnených v hornom menu, v žltej časti. Výsledné súbory sú podobné ako súbory typu CSV, avšak na oddelenie jednotlivých údajov používame znak „#“ a nie čiarku. Takéto súbory možno ďalej spracovávať a uskutočňovať merania ako v kapitole 7.