

**VYSOKÁ ŠKOLA EKONOMICKÁ V PRAZE**

**FAKULTA INFORMATIKY A STATISTIKY**



Název bakalářské práce:

**Metoda TTT pro porovnání  
randomizovaných heuristik**

|                |                                      |
|----------------|--------------------------------------|
| Autor:         | <b>Petra Novotná</b>                 |
| Katedra:       | <b>Katedra ekonometrie</b>           |
| Obor:          | <b>Matematické metody v ekonomii</b> |
| Vedoucí práce: | <b>prof. RNDr. Jan Pelikán, CSc.</b> |

**Prohlášení:**

Prohlašuji, že jsem bakalářskou práci na téma „Metoda TTT pro porovnání randomizovaných heuristik“ zpracovala samostatně. Veškerou použitou literaturu a další podkladové materiály uvádím v seznamu použité literatury.

V Praze dne 8. ledna 2016

.....

Petra Novotná

**Poděkování:**

Ráda bych touto cestou poděkovala prof. RNDr. Janu Pelikánovi, CSc. za vedení bakalářské práce a cenné rady. Rovněž pak děkuji panu Ing. Janu Fábrymu, Ph.D. za odbornou konzultaci.

## Abstrakt

*Název práce:* Metoda TTT pro porovnání randomizovaných heuristik

*Autor:* Petra Novotná

*Katedra:* Katedra ekonometrie

*Vedoucí práce:* prof. RNDr. Jan Pelikán, CSc.

Pro praktické využití úlohy listonoše je navrhováno velké množství heuristik, které sice dosahují jen přibližného řešení, ovšem v reálném čase. Vzhledem k počtu možností řešení problému listonoše je pro řešitele důležitá výkonnost heuristiky, a proto je cílem práce porovnání výkonnosti algoritmů. První část práce je věnována teorii potřebné k pochopení způsobu porovnávání, kterým dále srovnávám výkonnost pěti modifikací randomizované heuristiky řešící rozvozní úlohy pomocí problému listonoše s kapacitami. Varianty řešení jsou spuštěny na několika různě složitých úlohách s nepovinnými hranami a s více vozidly vyjíždějícími z jednoho depa. Výsledky jsou zobrazeny v grafech a pro přesnější vyhodnocení je vypočtena pravděpodobnost vypovídající o schopnosti jedné modifikace dosáhnout zadané cílové hodnoty v kratším čase než druhá modifikace. Předpokladem je různá efektivnost při řešení úlohy malého a velkého rozsahu.

**Klíčová slova:** metoda time-to-target, kapacitní úloha listonoše, randomizovaná heuristika

## Abstract

*Title:* TTT method for the comparison of randomized heuristics

*Author:* Petra Novotná

*Department:* Department of Econometrics

*Supervisor:* prof. RNDr. Jan Pelikán, CSc.

There are many heuristics proposed for practical uses of the postman problem, which reach only approximate solutions, however in real time. Regarding the number of solutions of the postman problem, the efficiency of the heuristic is important to the solver, and that is the reason why the goal of this thesis is to compare the efficiency of different algorithms. In the first part of the thesis, the theory for understanding the comparison is described, which is further used for comparing the efficiency of five modifications of randomized heuristics solving delivery problems using the postman problem with capacities. Variants of solutions are launched on several differently complicated problems with optional edges and with multiple vehicles departing from a single depot. The results are shown in plots. A probability of the ability of one modification to reach the given target value in shorter time than the second modification is computed for more precise evaluation. The assumption is different efficiency when solving a problem of small and large scale.

**Keywords:** time-to-target method, capacitated postman problem, randomized heuristic

# Obsah

|                                                              |           |
|--------------------------------------------------------------|-----------|
| <b>Úvod.....</b>                                             | <b>1</b>  |
| <b>1 Formulace metody pro porovnání heuristik .....</b>      | <b>2</b>  |
| 1.1 Srovnávání algoritmů .....                               | 2         |
| 1.2 TTT grafy .....                                          | 4         |
| 1.3 Q-Q grafy .....                                          | 5         |
| <b>2 Arc routing problem .....</b>                           | <b>7</b>  |
| 2.1 Neorientované řešení pro neorientovaný graf.....         | 8         |
| 2.2 Heuristické algoritmy pro vytvoření cyklických tras..... | 9         |
| <b>3 Testování modifikací heuristiky .....</b>               | <b>13</b> |
| 3.1 Vytvoření TTT grafů nad konkrétními úlohami .....        | 15        |
| 3.2 Porovnávání modifikací .....                             | 19        |
| <b>Závěr.....</b>                                            | <b>22</b> |

# Úvod

Práce se zabývá způsobem porovnání výkonnosti algoritmů vzhledem k výpočetnímu času. Složitost rozvozních úloh je v dnešní době pro mnohé společnosti zásadní otázkou při snižování nákladů. Pro konkrétní aplikaci lze vybírat z mnoha heuristik řešících tento problém nebo navrhnout vlastní modifikaci, kterou je nutno porovnat z hlediska výpočetního času, neboť i ten s sebou nese určité náklady.

Cílem je porovnávání výkonnosti pěti modifikací dále popsané randomizované heuristiky na různě složitých úlohách a vyhodnocení výsledků testování, které by mělo pomoci při volbě způsobu řešení praktického problému, a to jak z hlediska výpočetního času, tak i přiblížení se optimální hodnotě účelové funkce. Předpokladem pro srovnávání je různá efektivnost při řešení úlohy malého a velkého rozsahu.

První část práce je věnována způsobu porovnávání za použití time-to-target grafů. Pro vyčíslení výkonnosti jsou uvedeny výpočty pravděpodobností pro párové porovnání algoritmů. Dále je popsána heuristika problému listonoše s kapacitami, jejíž modifikace jsou ve třetí části testovány na úlohách s povinnými a nepovinnými hranami, různými počty vozidel a s různými kapacitami v každé úloze. V závěru práce jsou shrnuty výsledky a komentována výkonnost modifikací v porovnání s ostatními.

# 1 Formulace metody pro porovnání heuristik

K porovnání modifikací budou v práci použity grafy zobrazující rozdělení výpočetního času neboli TTT<sup>1</sup> grafy, které jsou velmi užitečné pro charakteristiku času výpočtu stochastických heuristických algoritmů. Ribera a Rosseti [1] popsali nový nástroj pro srovnávání dvou takových algoritmů. Odvozují pravděpodobnost, že jeden algoritmus najde řešení alespoň tak dobré, jako je cílová hodnota v kratším výpočetním čase než druhý algoritmus.

Grafy zobrazují na svislé ose pravděpodobnost, že jeden algoritmus najde řešení alespoň tak dobré jako cílová hodnota během výpočetního času zobrazeného na ose vodorovné. Během mnoha implementací [2] bylo vypořizováno, že náhodná proměnná *time to target value* je exponenciálně rozdělená, popřípadě má posunuté exponenciální rozdělení. Hoos a Stützle [2] se domnívají, že to platí pro všechny heuristiky založené na stochastickém lokálním vyhledávání.

## 1.1 Srovnávání algoritmů

Za předpokladu, že výpočetní časy dvou různých algoritmů mají exponenciální rozdělení (nebo posunuté exponenciální), bylo vyvinuto [3] číslo udávající pravděpodobnost, že jeden z algoritmů nalezne cílovou hodnotu v kratším výpočetním čase než druhý. Dále je popsán postup pro výpočet pravděpodobnosti, že jeden z algoritmů najde lepší řešení než druhý ve stejném výpočetním čase.

Předpokládá se existence dvou stochastických algoritmů lokálního vyhledávání  $A_1$  a  $A_2$  pro přibližné řešení některého kombinatorického optimalizačního problému. Mimo to se předpokládá, že jejich výpočetní časy mají exponenciální (nebo posunuté exponenciální) rozdělení. Označíme  $X_1$  (resp.  $X_2$ ) spojitou náhodnou veličinu reprezentující čas potřebný algoritmem  $A_1$  (resp.  $A_2$ ) k nalezení řešení alespoň tak dobrého jako zadaná cílová hodnota. Pro  $f(X_1)$  a  $f(X_2)$  představující hustoty pravděpodobnosti platí [1]:

$$f(X_1) = \begin{cases} 0, & \tau < T_1 \\ \lambda_1 e^{-\lambda_1(\tau-T_1)}, & \tau \geq T_1 \end{cases}$$

$$f(X_2) = \begin{cases} 0, & \tau < T_2 \\ \lambda_2 e^{-\lambda_2(\tau-T_2)}, & \tau \geq T_2 \end{cases}$$

kde  $T_1$ ,  $\lambda_1$ ,  $T_2$  a  $\lambda_2$  jsou parametry ( $\lambda_1$  a  $\lambda_2$  definují tvar každého posunutého exponenciálního rozdělení, přičemž  $T_1$  a  $T_2$  udávají, jak moc je každé z nich posunuté).

---

<sup>1</sup> time-to-target

Jelikož se oba algoritmy zastaví, když najdou řešení alespoň tak dobré jako cílová hodnota, můžeme říci, že algoritmus  $A_1$  pracuje lépe než algoritmus  $A_2$ , zastaví-li se první dříve než druhý. Proto je potřeba odhadnout pravděpodobnost, že náhodná proměnná  $X_1$  má hodnotu menší nebo rovnou  $X_2$ , tj. vypočítat  $P(X_1 \leq X_2)$ . V závislosti na hodnotě  $X_2$  a použitím věty o úplné pravděpodobnosti dostaneme [1]:

$$\begin{aligned} P(X_1 \leq X_2) &= \int_{-\infty}^{\infty} P(X_1 \leq X_2 | X_2 = \tau) f(X_2)(\tau) d\tau \\ &= \int_{T_2}^{\infty} P(X_1 \leq X_2 | X_2 = \tau) \lambda_2 e^{-\lambda_2(\tau-T_2)} d\tau \\ &= \int_{T_2}^{\infty} P(X_1 \leq \tau) \lambda_2 e^{-\lambda_2(\tau-T_2)} d\tau \end{aligned}$$

Nechť  $v = \tau - T_2$ . Potom,  $dv = d\tau$  a

$$P(X_1 \leq X_2) = \int_0^{\infty} P[X_1 \leq (v + T_2)] \lambda_2 e^{-\lambda_2(v)} dv. \quad (1.1)$$

Použitím vzorce kumulativní pravděpodobnosti náhodné proměnné  $X_1$  uvádí v [1]:

$$P[X_1 \leq (v + T_2)] = 1 - e^{-\lambda_1(v+T_2-T_1)}. \quad (1.2)$$

Nahrazením (1.2) v (1.1) a integrováním dospěli v [1] k závěru:

$$P(X_1 \leq X_2) = 1 - e^{-\lambda_1(T_2-T_1)} \frac{\lambda_2}{\lambda_1 + \lambda_2}. \quad (1.3)$$

Tento výsledek může být lépe interpretován přepsáním výrazu (1.3) jako:

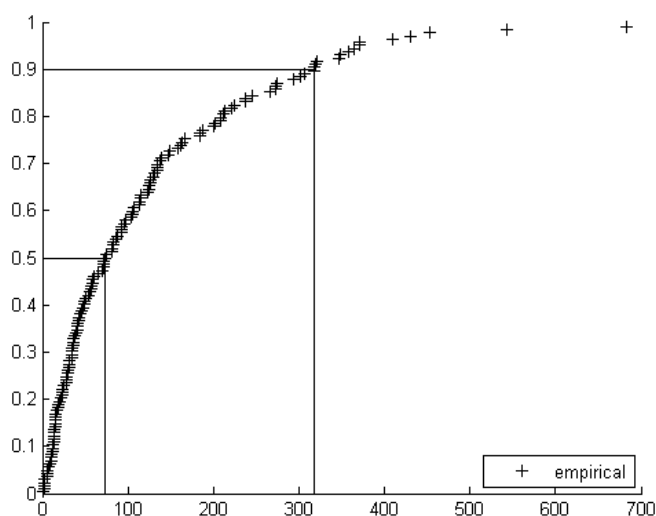
$$P(X_1 \leq X_2) = 1 - e^{-\lambda_1(T_2-T_1)} + e^{-\lambda_1(T_2-T_1)} \frac{\lambda_1}{\lambda_1 + \lambda_2}. \quad (1.4)$$

## 1.2 TTT grafy

Modifikace heuristiky jsou spuštěny  $n$ -krát na stejném příkladu se zadanou hodnotou cílového řešení. Pro každé z  $n$  spuštění je generátoru náhodného čísla nastavena počáteční hodnota s jiným zdrojem, čímž je splněn předpoklad, že spuštění musí být nezávislá. Pro srovnání naměřeného a teoretického rozdělení je použita grafická metodologie pro analýzu dat [4].

### Kumulativní rozdělení pravděpodobnosti

Pro vykreslení TTT grafů byl použit perl program Aiex a spol. [3]. Pro každou dvojici algoritmů jsou výpočetní časy seřazeny vzestupně. Každý  $i$ -tý výpočetní čas  $t_i$  je spojen s pravděpodobností  $p_i = (i - 1/2)/n$ , a jsou vykresleny body  $z_i = [t_i, p_i]$ , pro  $i = 1, \dots, n$ . Výběr  $p_i$  je komentován v následující podkapitole. Obr. 1 ilustruje kumulativní rozdělení pravděpodobnosti, která byla získána opakovanou aplikací heuristiky k nalezení řešení s cílovou hodnotou funkce alespoň tak dobrou jako zadaná cílová hodnota. Pro ilustrování výše uvedených výsledků vezmu v úvahu konkrétní případ, kdy je heuristika použita na příkladu s4-c z datového souboru dostupného na <http://www.uv.es/belengue/carp.html>. Heuristika byla spuštěna 150krát a cílová hodnota nastavena na 4050.



**Obrázek 1** Rozdělení pravděpodobnosti příkladu s4-c s cílovou hodnotou 4050

V obr. 1 vidíme, že nalezení řešení pomocí heuristiky alespoň tak dobrého jako cílová hodnota v nejvýše 72 generováních odpovídá pravděpodobnost okolo 50%, v nejvýše 318 generováních asi 90%.

### 1.3 Q-Q grafy

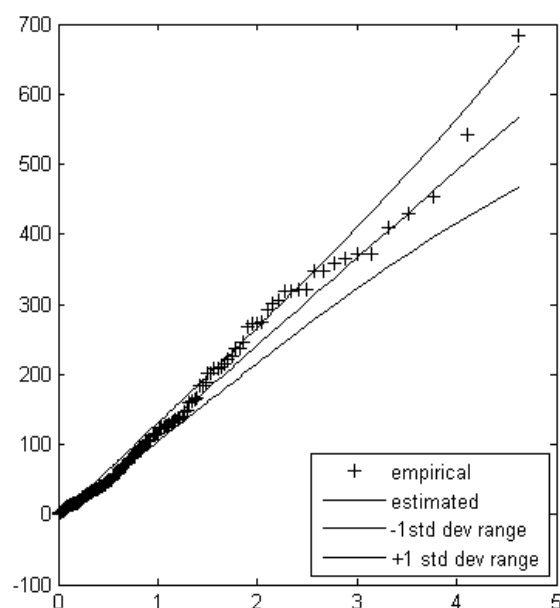
Body v TTT grafu nejvíce odpovídají distribuční funkci posunutého exponenciálního rozdělení, proto bude dále uveden výpočet pro odhad parametrů dvouparametrového exponenciálního rozdělení.

Dle [3] je postup odhadu parametrů  $\lambda$  a  $T$  posunutého exponenciálního rozdělení určen následujícím postupem. Pro každou hodnotu  $p_i$ ,  $i = 1, \dots, n$  spojujeme  $p_i$ -kvantil  $Qt(p_i)$  teoretického rozdělení. Pro dvouparametrové exponenciální rozdělení máme  $Qt(p_i) = -\lambda \ln(1 - p_i) + T$ . Kvantily dat empirického rozdělení jsou seřazené naměřené hodnoty. Teoretický kvantil-kvantilový (Q-Q) graf je získán vynesetím kvantilů dat empirického rozdělení proti kvantilům teoretického rozdělení. To zahrnuje tři kroky. Za prvé, naměřené časy jsou seřazené ve vzestupném pořadí. Za druhé, jsou získány kvantily teoretického exponenciálního rozdělení. A nakonec je vytvořen graf dat oproti teoretickým kvantilům [3].

Q-Q graf znázorněný v obr. 2 získáme vynesetím naměřených časů na osu y proti kvantilům teoretického dvouparametrového exponenciálního rozdělení na ose x. Aby bylo zabráněno případnému zkreslení způsobeného odlehlými hodnotami, neodhaduje se střední hodnota jako střední hodnota dat nebo lineární regresí. Namísto toho je sklon  $\hat{\lambda}$  odhadnut pomocí horního kvartilu  $q_u$  a dolního kvartilu  $q_l$  dat. Horní a dolní kvantily jsou  $Q(1/4)$  a  $Q(3/4)$  kvantily. Jako odhad sklonu bylo použito  $\hat{\lambda} = [z_u - z_l] / [q_u - q_l]$ , kde  $z_u$  a  $z_l$  jsou  $u$ -tý a  $l$ -tý body seřazených naměřených časů a pro odhad posunutí bylo použito  $\hat{T} = z_l - \hat{\lambda}q_l$ . Pro analyzování Q-Q grafů je pro každý vynesetý bod znázorněna plus minus jedna směrodatná odchylka ve svislém směru od přímky odhadnutého exponenciálního rozdělení. Odhad směrodatné odchylky pro bod  $z_i$ ,  $i = 1, \dots, n$  Q-Q grafu je  $\hat{\sigma} = \hat{\lambda}[p_i(1 - p_i)n]^{1/2}$ .

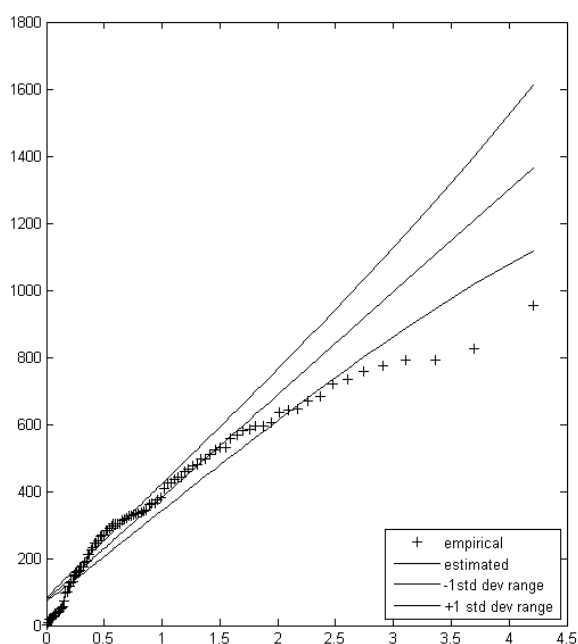
Po seřazení naměřených časů jsem vytvořila sloupec hodnot pravděpodobností dle  $p_i = (i - 1/2)/n$ , tedy například pro první naměřený čas z testování s 200 spuštěními jsem dosadila následovně  $p_1 = (1 - 1/2)/200$ . Dále jsem vytvořila sloupec s hodnotami vypočtenými dle  $Qt(p_i) = -\ln(1 - p_i)$ , které budou vykresleny v Q-Q grafu. Dle [3] jsem následně odhadla posunutí  $\hat{T}$  a sklon  $\hat{\lambda}$  rozdělení. Například pro 1000 spuštění odpovídá  $Q(1/4)$  250. hodnota a  $Q(3/4)$  je 750. hodnota a pro odhad jsem tedy to rovnice  $\hat{\lambda} = [z_u - z_l] / [q_u - q_l]$  dosadila  $z_u = 750$  a  $z_l = 250$  a zbylé dvě proměnné nahradila konkrétním počtem kroků, která danému kvantilu odpovídá. Dále jsem hodnoty dosadila i do rovnice  $\hat{T} = z_l - \hat{\lambda}q_l$ .

Obr. 2 znázorňuje příklad Q-Q grafu z příkladu s4-c použitím druhé modifikace.



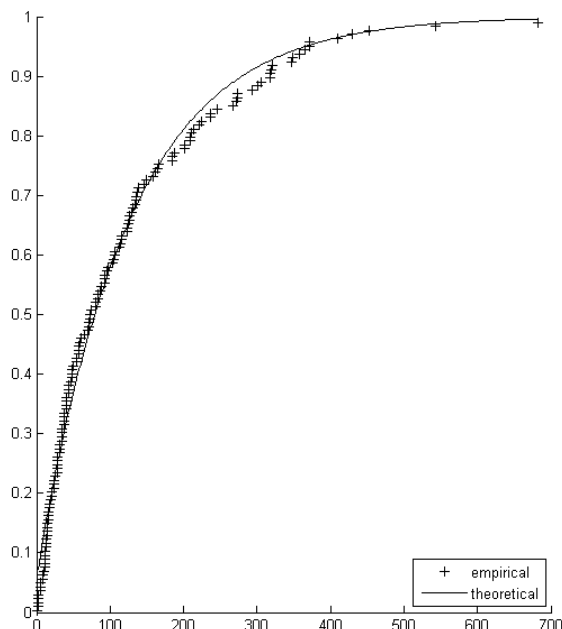
**Obrázek 2** Q-Q graf pro příklad s4-c s cílovou hodnotou 4050

Pokud se jeden vynesení bod odchyluje o více než jednu směrodatnou odchylku, je zde velká pravděpodobnost, že celá řada dalších také. Proto pokud se v Q-Q grafu standardní odchylky výrazně liší, je třeba problému věnovat pozornost. Jak lze pozorovat v grafu na obr. 3, směrodatná odchylka bodů v blízkosti horního konce je podstatně větší než směrodatná odchylka na druhém konci.



**Obrázek 3** Q-Q graf pro příklad e1-b s cílovou hodnotou 5000 při použití 3. modifikace

Poté, co jsou odhadnuty dva parametry rozdělení, může být vytvořen graf empirického a teoretického rozdělení. Obr. 4 zobrazuje graf odpovídající Q-Q grafu na obr. 3.



**Obrázek 4** Graf pro příklad s4-c s cílovou hodnotou 24400 při použití 5. modifikace

## 2 Arc routing problem

Pro pojem Arc Routing Problems se v českém jazyce dosud nevžil žádný vhodný ekvivalent, a proto se problémy tohoto typu přiřazují k původní úloze listonoše. Původně se jednalo o úlohu, v níž je v neorientovaném grafu definována množina ohodnocených hran, které je všechny nutné v cyklu projet alespoň jednou, tedy najít takový uzavřený sled, který obsahuje každou hranu alespoň jednou. [7]

Z habilitační práce [7] je čerpán následující popis problému. Pro následující klasifikaci úloh označme graf  $G = \{U, H\}$ ,  $U$  množinu všech uzlů,  $H$  množinu všech hran,  $E$  množinu všech neorientovaných hran,  $A$  množinu všech orientovaných hran a  $R$  množinu všech povinných hran (tj. hran, které je nutné projet či projít). V neorientovaném problému listonoše předpokládáme všechny hrany neorientované, tedy  $R = E = H$ .

Kapacitní úloha listonoše<sup>2</sup> je obdobou kapacitních rozvozních úloh. Je zde nutné vzít v potaz kapacitu obsluhujícího vozidla vzhledem k nenulovým požadavkům obsluhovaných hran. Belenguer a Benavent [5] formulovali matematický model celočíselného programování pro neorientovaný graf.

Úloha CARP je definována pro orientované i pro neorientované grafy [6]. V této práci se budu zabývat pouze neorientovanými grafy.

## 2.1 Neorientované řešení pro neorientovaný graf

V této části bude popsán matematický model, jehož řešením získáme množiny hran, které jsou součástí cyklické trasy, bez určení její orientace. Orientovaná trasa je nalezena až následně například Fleuryho algoritmem.

Pro všechny hrany  $(i, j) \in E$  jsou definovány jejich délky  $c_{ij} = c_{ji}$ , pro hrany  $(i, j) \in R$  jsou navíc definovány požadavky na svoz (rozvoz)  $d_{ij} > 0$ . Uzel č. 1 představuje depo, v němž je umístěno  $K$  vozidel o kapacitě  $V$ . Je zavedena celočíselná proměnná  $x_{ij}^k$ , jejíž hodnota představuje počet průjezdů  $k$ -tého vozidla hranou  $(i, j) \in E$  v jakémkoli směru. Binární proměnná  $y_{ij}^k$  je rovna 1 v případě, že hrana  $(i, j) \in R$  je obsloužena  $k$ -tým vozidlem, tj. požadavek  $d_{ij}$  je při svozu naložen na  $k$ -té vozidlo, jinak je proměnná rovna 0. Obě proměnné jsou navíc definovány pouze pro  $i < j$ .

Matematický model lze pak formulovat následovně [7]:

$$\text{minimalizovat} \quad z = \sum_{k=1}^K \sum_{(i,j) \in E} c_{ij} x_{ij}^k \quad (2.1)$$

za podmínek

$$y_{ij}^k \leq x_{ij}^k, \quad \forall (i, j) \in E, \quad k = 1, 2, \dots, K, \quad (2.2)$$

$$\sum_{k=1}^K y_{ij}^k = 1, \quad \forall (i, j) \in R, \quad (2.3)$$

$$\sum_{(i,j) \in R} d_{ij} y_{ij}^k \leq V, \quad k = 1, 2, \dots, K, \quad (2.4)$$

$$\sum_{(i,j) \in E} x_{ij}^k + \sum_{(j,i) \in E} x_{ji}^k = 2f_j^k, \quad \forall j \in U, \quad k = 1, 2, \dots, K, \quad (2.5)$$

$$x_{ij}^k \in \{0, 1\}, y_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in E, \quad k = 1, 2, \dots, K, \quad (2.6)$$

---

<sup>2</sup> capacitated arc routing problem (CARP)

$$\sum_{\substack{i \in S \\ (i,j) \in E}} \sum_{\substack{j \notin S \\ (j,i) \in E}} x_{ij}^k + \sum_{\substack{j \in S \\ (j,i) \in E}} \sum_{\substack{i \notin S \\ (i,j) \in E}} x_{ji}^k \geq \frac{1}{M} \sum_{i \in S} \sum_{j \in S} x_{ij}^k, \quad k=1,2,\dots,K, \quad S \subseteq U - \{1\}, S \neq \emptyset, \quad (2.7)$$

kde  $M$  je vysoká konstanta.

Kapacitní úloha listonoše patří mezi NP-obtížné problémy, proto při větším rozměru úlohy nelze očekávat získání optimálního řešení. V takových případech je nutné použít přibližné metody.

## 2.2 Heuristické algoritmy pro vytvoření cyklických tras

Dále je popsán heuristický algoritmus, který při vytváření tras využívá náhodný výběr hran. V následujícím textu budeme předpokládat existenci povinných a nepovinných hran [7].

### Algoritmus založený na náhodném výběru hran

Metoda spočívá v opakovaném generování tras, kdy jsou hrany do trasy zařazovány na základě jejich náhodného výběru. Počet vygenerovaných řešení  $N_{max}$  je předem zvolený a je použitý pro definování ukončovacího pravidla simulačního běhu. Během vytváření trasy se generuje posloupnost navštívených uzlů  $U_k = \{u_1^k, u_2^k, \dots, u_{h_k}^k\}$ , kde  $u_1^k = u_{h_k}^k = 1$  (1 je výchozí uzel), index  $k$  označuje pořadové číslo simulačního kroku, během něhož získáme jedno vygenerované řešení, a  $h_k$  je počet uzlů v posloupnosti  $U_k$ . Necht'  $P$  je množina nejkratších cest mezi uzly  $i, j \in U$ , jejichž délky označíme  $p_{ij}$ . Hodnota  $F_k$  představuje délku vytvořené trasy a pro délku nejkratší dosažené trasy platí [7]:

$$F^* = \min_{k=1,2,\dots,N_{max}} F_k \quad (2.8)$$

Nejlepší trasu označíme  $U^*$ . Hodnota  $L_t$  představuje hodnotu nákladu na vozidle po zařazení uzlu  $u_t^k$  do posloupnosti  $U_k$ . Pro každou povinnou hranu  $(i,j) \in R$  je definovaná binární proměnná  $serv_{ij}$ , jejíž hodnota je rovna 0 v případě, že hrana v generované trase dosud nebyla obsloužena, jinak nabývá hodnoty 1. Cílem je tedy dosáhnout rovnosti [7]:

$$\sum_{(i,j) \in R} serv_{ij} = |R|, \quad (2.9)$$

kde  $|R|$  je počet povinných hran.

Algoritmus se celkem skládá z devíti kroků. V prvním krku je nastaveno pořadové číslo simulace na hodnotu 1 a prozatím nejlepší hodnota účelové funkce je nastavena na nekonečno.

Krok 1:  $k=1$ ;  $F^* = +\infty$ .

Vozidlo vyjíždí z depa, v druhém kroku je tedy nastaven index prvního uzlu řešení na hodnotu 1. Hodnota účelové funkce je nastavena na hodnotu 0 a stejně je tomu pro hodnotu nákladu, vozidlo zatím nemá žádný náklad. Hodnoty pomocných proměnných jsou také nastaveny na nulovou hodnotu. Žádná povinná hrana zatím nebyla obsloužena.

Krok 2: Jestliže  $k > N_{max}$  jdi na Krok 9

$$t = 1; u_1^k = 1; F_k = 0; L_t = 0; n_R = |R|; n_1 = n_2 = 0; serv_{ij} = 0 \forall (i, j) \in R.$$

Za podmínky, že se vozidlo nachází v depu, je v dalším kroku vyložen náklad. Nenachází-li se v depu, jsou vyhledávány neobsloužené povinné hrany incidentní s uzlem, ve kterém se vozidlo nachází, a jejich požadavek ještě může být naložen. Pokud takové nejsou nalezeny, přejde algoritmus k sedmému kroku, a vozidlo jede do depa.

Krok 3: Jestliže  $n_R = 0$  pak jdi na Krok 7

$$R_t = \{(u_t^k, j) \in R \mid serv_{u_t^k j} = 0, L_t + d_{u_t^k j} \leq V\}; n_1 = |R_t|;$$

pokud  $n_1 = 0$  jdi na Krok 5.

Z nalezených hran je náhodně vybrána jedna. Do účelové funkce je přičtena vzdálenost mezi uzly, k nákladu jsou přičteny obsloužené požadavky a hrana je započtena do vygenerované trasy.

Krok 4: Vyber hranu  $(u_t^k, j^*)$  z množiny  $R_t$  náhodným způsobem

$$u_{t+1}^k = j^*; F_k = F_k + c_{u_t^k u_{t+1}^k}; L_{t+1} = L_t + d_{u_t^k u_{t+1}^k}; t = t + 1;$$

$$n_R = n_R - 1; \text{jestliže } n_R = 0 \text{ jdi na Krok 7;}$$

pokud  $F_k \geq F^*$  pak

$$k = k + 1; \text{jdi na Krok 2;}$$

$$\text{jestliže } u_t^k = 1 \text{ nastav } L_t = 0;$$

jdi na Krok 3.

Pokud žádnou povinnou neobslouženou hranu incidentní s uzlem, kde se aktuálně nachází a kterou uveze, nenalezne, přesouvá se na krok 5. V pátém kroku heuristika prohledává hrany, které jsou incidentní s uzly sousedící s aktuálním uzlem. Pokud se v okolí nalezne povinná neobsloužená hrana, kterou vozidlo uveze, pokračuje krokem 6. Pokud ne, jde na krok 7.

Krok 5:  $P_t = \{(u_t^k, i) \in P \mid \exists (i, j) \in R, \text{serv}_{ij} = 0, L_t + d_{ij} \leq V\}; n_2 = |P_t|;$

pokud  $n_2 = 0$  jdi na Krok 7.

V kroku 6 se náhodným výběrem zvolí jedna z neobsloužených povinných hran z množiny  $P_t$ , s akceptovatelným požadavkem z nejbližšího okolí a nejkratší cestou se k ní jede. Přičte se vzdálenost do účelové funkce. Načte se nový uzel do cesty řešení, přesune se na něj a vrací se na krok 3.

Krok 6:  $p_{u_t^k j^*} = \min_{(u_t^k, j) \in P_t} p_{u_t^k j}; u_{t+1}^k = j^*; F_k = F_k + p_{u_t^k u_{t+1}^k}; t = t + 1;$

pokud  $F_k \geq F^*$  pak

$k = k + 1$ ; jdi na Krok 2;

jestliže  $u_t^k = 1$  nastav  $L_t = 0$ ;

jdi na Krok 3.

V sedmém kroku je vozidlo posláno do depa. K účelové funkci je přičtena nejkratší vzdálenost z aktuálního uzlu do depa a první uzel je zaznamenán do vygenerované trasy. Náklad bude vyložen, a pokud je některá z povinných hran neobsloužená, vrací se algoritmus na krok 3 a dále prohledává graf. Jsou-li všechny povinné hrany obslouženy, algoritmus pokračuje krokem 8.

Krok 7: Jestliže  $u_t^k \neq 1$  pak

$u_{t+1}^k = 1; F_k = F_k + p_{u_t^k 1}; t = t + 1;$

$L_t = 0$ ;

pokud  $F_k \geq F^*$  pak

$k = k + 1$ ; jdi na Krok 2;

pokud  $n_R = 0$  jdi na Krok 8, jinak jdi na Krok 3.

V osmém kroku se porovnává účelová funkce s její nejlepší hodnotou. Pokud je aktuální řešení menší, přepíše se nejlepší hodnota účelové funkce a vektor její cesty. Přechází se na další simulaci. Pokud všechny simulace doběhly, algoritmus končí.

Krok 8: Pokud  $F_k < F^*$  pak

$$F^* = F_k; U^* = U_k;$$

$k = k + 1$ ; jdi na Krok 2.

Krok 9: Konec.

V kroku 4 je hrana  $(u_t^k, j^*)$  z množiny  $R_t$  vybrána náhodně pomocí generátoru pseudonáhodných čísel, která jsou následně transformována na hodnoty náhodné veličiny. V této části je možný výběr mezi pěti modifikacemi, které jsou zároveň dále použity pro porovnávání. Hodnoty náhodné veličiny lze volit s následujícím pravděpodobnostním rozdělením [7]:

- 1) Rovnoměrné rozdělení – každá hrana je vybrána s pravděpodobností

$$pst_{ij} = \frac{1}{n_1}, (i, j) \in R_t. \quad (2.10)$$

- 2) Nerovnoměrné rozdělení, pravděpodobnosti jsou odvozeny z velikosti požadavku:

$$pst_{ij} = \frac{d_{ij}}{\sum_{(r,s) \in R_t} d_{rs}}, (i, j) \in R_t \quad (2.11)$$

- 3) Nerovnoměrné rozdělení, pravděpodobnosti jsou odvozeny z délky hrany:

$$pst_{ij} = 1 - \frac{c_{ij}}{\sum_{(r,s) \in R_t} c_{rs}}, (i, j) \in R_t \quad (2.12)$$

- 4) Pokud není nalezena neobsloužená povinná hrana incidentní s uzlem, ve kterém se vozidlo nachází (krok 3), je zvolena neobsloužená povinná s nejkratší vzdáleností od depa. V opačném případě algoritmus pokračuje čtvrtým krokem a konkrétně náhodným výběrem s rovnoměrným rozdělením (2.10).
- 5) Pokud není nalezena neobsloužená povinná hrana incidentní s uzlem, ve kterém se vozidlo nachází (krok 3), je zvolena neobsloužená povinná s nejkratší vzdáleností od depa. V opačném případě algoritmus pokračuje čtvrtým krokem a konkrétně náhodným výběrem s nerovnoměrným rozdělením dle velikosti požadavku (2.11).

Výhodou randomizovaného přístupu je získání velkého počtu tras, u kterých existuje možnost, že získáme lepší řešení než při použití deterministického algoritmu svázaného striktně danými pravidly pro výběr povinných hran, resp. cest k nim vedoucích.

### 3 Testování modifikací heuristiky

Celkem bylo testováno pět modifikací dané heuristiky na 24 příkladech na počítači HP ProBook 4510s s procesorem Intel Core 2 Duo CPU T5870 (2,00 GHz, RAM 4GB). Pro porovnávání byly použity příklady z datového souboru, který byl vytvořen na základě údajů získaných z aplikace zimního posypu ve městě Lancashire [8]. Datový soubor je dostupný na <http://www.uv.es/belengue/carp.html> ve formátu DAT.

Pro ukázkou je uvedena část jednoho ze souborů s překladem v závorkách:

|                      |                   |                    |    |             |    |
|----------------------|-------------------|--------------------|----|-------------|----|
| NOMBRE:              | egl-e1-A          | (název)            |    |             |    |
| VERTICES:            | 77                | (vrcholy)          |    |             |    |
| ARISTAS_REQ:         | 51                | (povinné hrany)    |    |             |    |
| ARISTAS_NOREQ:       | 47                | (nepovinné hrany)  |    |             |    |
| VEHICULOS:           | 5                 | (vozidla)          |    |             |    |
| CAPACIDAD:           | 305               | (kapacita vozidel) |    |             |    |
| COSTE_TOTAL_REQ:     | 1468              | (celkové náklady)  |    |             |    |
| LISTA_ARISTAS_REQ:   | (povinné hrany)   | (náklady)          |    | (požadavek) |    |
|                      | (1,2)             | coste              | 32 | demanda     | 32 |
|                      | (2,3)             | coste              | 14 | demanda     | 14 |
|                      | (2,4)             | coste              | 17 | demanda     | 17 |
|                      | (4,5)             | coste              | 56 | demanda     | 56 |
|                      | (9,10)            | coste              | 20 | demanda     | 20 |
|                      | (11,12)           | coste              | 32 | demanda     | 32 |
|                      | (12,16)           | coste              | 29 | demanda     | 29 |
| LISTA_ARISTAS_NOREQ: | (nepovinné hrany) |                    |    |             |    |
|                      | (5,6)             | coste              | 8  |             |    |
|                      | (5,7)             | coste              | 6  |             |    |
|                      | (7,8)             | coste              | 18 |             |    |
|                      | (8,9)             | coste              | 26 |             |    |
|                      | (10,11)           | coste              | 12 |             |    |
|                      | (14,15)           | coste              | 7  |             |    |

**Tabulka 1** Datový soubor dostupný na <http://www.uv.es/belengue/carp.html>

Úlohy jsou dostupné ve stejném formátu, vždy nejprve hlavička se základními informacemi o rozsahu úlohy a v druhé části jsou pak vypsány všechny hrany a jejich požadavky a náklady. Na konci je uvedeno, který uzel je výchozí.

Pro první porovnání uvádím v tabulce č. 2 hodnoty dosažené použitím 5. modifikace na všech 24 příkladech ze zmíněného datového souboru a hodnoty publikované v literatuře [6]. V tabulce jsou následující informace: označení příkladu, počet uzlů, počet hran, počet povinných hran, hodnota dolní meze z literatury (LB), nejlepší hodnota dosažená použitím zmíněné heuristiky po 1000 krocích (r1000) a 10000 krocích (r10000).

Byla použita 5. modifikace. Následně vždy po nejlepší dosažené hodnotě uvádím odchylku od LB.

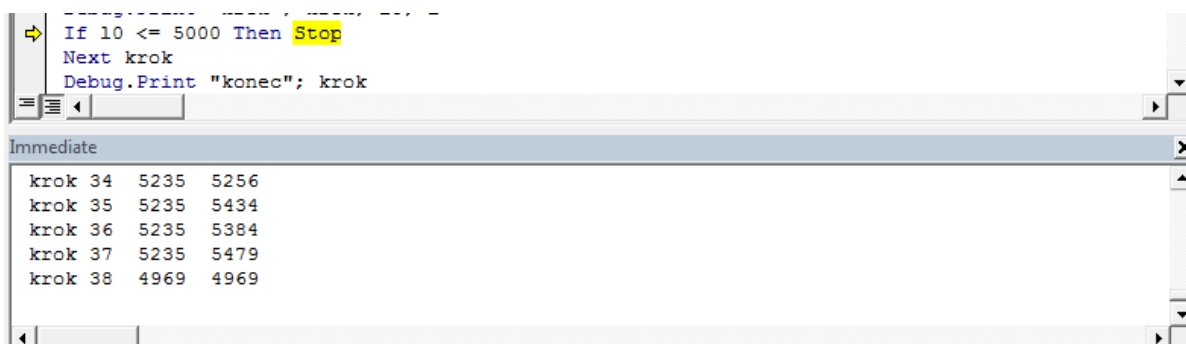
|      | V   | E   | R   | LB    | r1000 | %LB   | r10000 | %LB   |
|------|-----|-----|-----|-------|-------|-------|--------|-------|
| e1-a | 77  | 98  | 51  | 3548  | 4001  | 12,8  | 3944   | 11,2  |
| e1-b | 77  | 98  | 51  | 4468  | 4967  | 11,2  | 4956   | 10,9  |
| e1-c | 77  | 98  | 51  | 5542  | 6786  | 22,4  | 6596   | 19,0  |
| e2-a | 77  | 98  | 72  | 5011  | 5619  | 12,1  | 5606   | 11,9  |
| e2-b | 77  | 98  | 72  | 6280  | 7373  | 17,4  | 7171   | 14,2  |
| e2-c | 77  | 98  | 72  | 8234  | 9471  | 15,0  | 9456   | 14,8  |
| e3-a | 77  | 98  | 87  | 5898  | 6599  | 11,9  | 6590   | 11,7  |
| e3-b | 77  | 98  | 87  | 7697  | 8847  | 14,9  | 8830   | 14,7  |
| e3-c | 77  | 98  | 87  | 10163 | 11816 | 16,3  | 11632  | 14,5  |
| e4-a | 77  | 98  | 98  | 6395  | 7249  | 13,4  | 7209   | 12,7  |
| e4-b | 77  | 98  | 98  | 8884  | 10011 | 12,7  | 9970   | 12,2  |
| e4-c | 77  | 98  | 98  | 11427 | 13163 | 15,2  | 13077  | 14,4  |
| s1-a | 140 | 190 | 75  | 5014  | 5975  | 19,2  | 5941   | 18,5  |
| s1-b | 140 | 190 | 75  | 6379  | 8126  | 27,4  | 8110   | 27,1  |
| s1-c | 140 | 190 | 75  | 8480  | 9670  | 14,0  | 9460   | 11,6  |
| s2-a | 140 | 190 | 147 | 9824  | 11571 | 17,8  | 11411  | 16,2  |
| s2-b | 140 | 190 | 147 | 12968 | 15051 | 16,1  | 14852  | 14,5  |
| s2-c | 140 | 190 | 147 | 16353 | 19129 | 17,0  | 18905  | 15,6  |
| s3-a | 140 | 190 | 159 | 10143 | 12348 | 21,7  | 12072  | 19,0  |
| s3-b | 140 | 190 | 159 | 13616 | 16090 | 18,2  | 15871  | 16,6  |
| s3-c | 140 | 190 | 159 | 17100 | 20155 | 17,9  | 20128  | 17,7  |
| s4-a | 140 | 190 | 190 | 12143 | 14794 | 21,8  | 14733  | 21,3  |
| s4-b | 140 | 190 | 190 | 16093 | 19197 | 19,3  | 18947  | 17,7  |
| s4-c | 140 | 190 | 190 | 20375 | 24300 | 19,3  | 23980  | 17,7  |
|      |     |     |     |       |       | Ø16,9 |        | Ø15,7 |

**Tabulka 2** Porovnání dosažených hodnot s hodnotami v literatuře

Z tabulky je patrné, že u žádné z úloh heuristika nedošla k hodnotám z literatury a nejmenší procentuální odchylky od dat bylo dosaženo v úloze e1-b, a to 10,9%. Nejnižších odchylek bylo dosaženo v úlohách nejmenšího rozsahu. Varianty se od sebe liší jak počty uzlů, povinných a nepovinných hran, tak také počty vozidel, které pro přehlednost v tabulce neuvádím. Například úlohy e1-a e1-b, e1-c se od sebe liší počty vozidel (a jejich kapacitami) ve stejném pořadí: 5 (305), 7 (220), 10 (160). Naopak největší odchylky bylo dosaženo při výpočtu v úloze s1-b a to 27,1%. Celkově se výsledky odchylují od výsledků z literatury v průměru o 15,7 % po 10000 kroků spuštění.

Pro spuštění simulací heuristiky [9] byl použit editor programovacího jazyka VBA<sup>3</sup> v prostředí MS Excel, který přikládám na CD. Výsledné hodnoty byly uloženy přes software MATLAB ve formátu DAT potřebném pro spuštění perl programu [3]. Pro spuštění programu dostupného na <http://mauricio.resende.info/ttplots/> byl také zvolen software MATLAB. Potřebné grafy byly vykresleny ze souborů, které tento script vytvořil.

Při spuštění simulací heuristiky jsem ocenila, že je možné v editoru jazyka VBA sledovat průběh výpočtů v okně Immediate, které lze jednoduše přidat kombinací kláves Ctrl+G. Do tohoto okna lze nastavit zapisování pomocí Debug.Print "text"; promenna;. Když se algoritmus zastaví, například ve chvíli, kdy dosáhl cílové hodnoty, je velmi přehledné ihned vědět, kolikrát se do té doby výpočet spustil, popřípadě jaké hodnoty dosáhl po daném počtu kroků.



Obrázek 5 Okno Immediate ve VBA

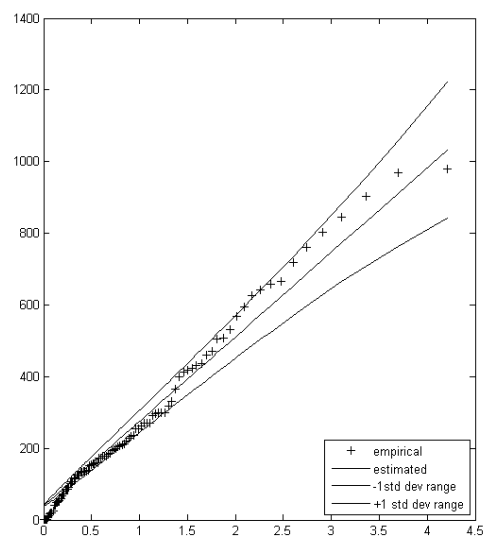
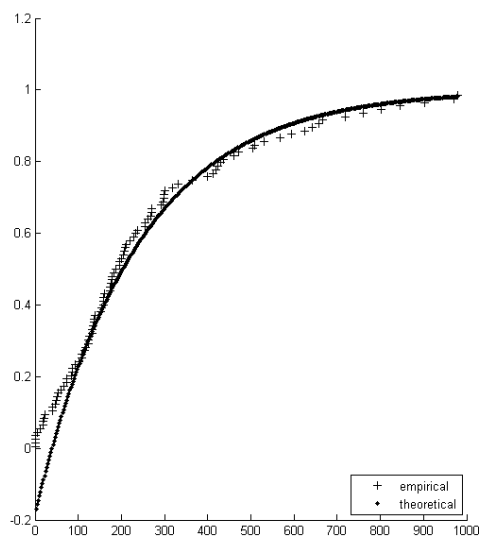
### 3.1 Vytvoření TTT grafů nad konkrétními úlohami

Použitím metody porovnávající heuristické algoritmy budu porovnávat čas výpočtu, a to konkrétně počty kroků algoritmu, které byly zapotřebí pro dosažení cílové hodnoty.

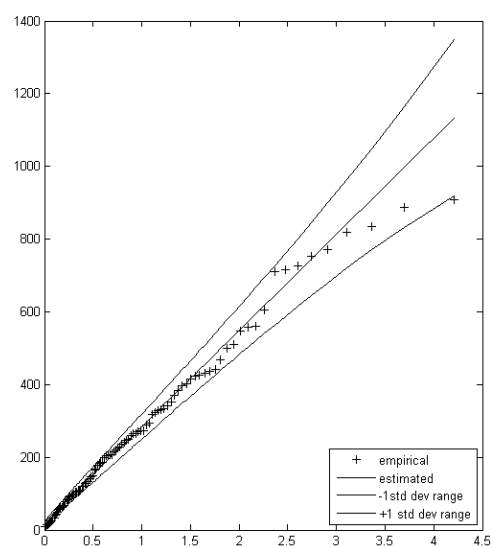
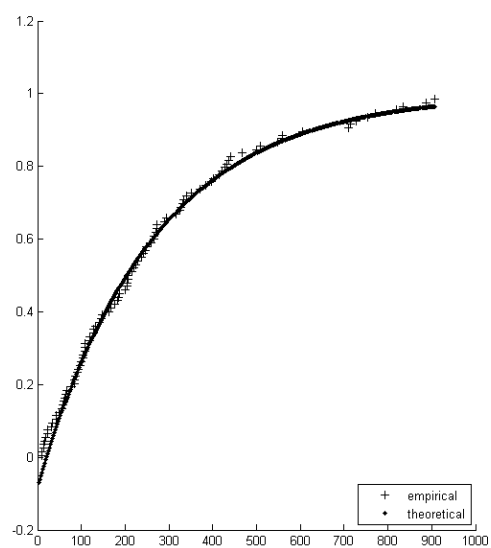
Pro příklad e1-b proběhlo celkem 100 spuštění výpočtu s cílovou hodnotou 5000. Pro upřesnění, úloha e1-b má 77 uzlů, 98 hran, 51 povinných hran a k dispozici je 7 vozidel s kapacitou 220 jednotek. Obrázky 6-11. ilustrují kumulativní rozdělení pravděpodobnosti, která byla získána opakovanou aplikací heuristiky k nalezení řešení s cílovou hodnotou funkce menší nebo rovnou 5000.

---

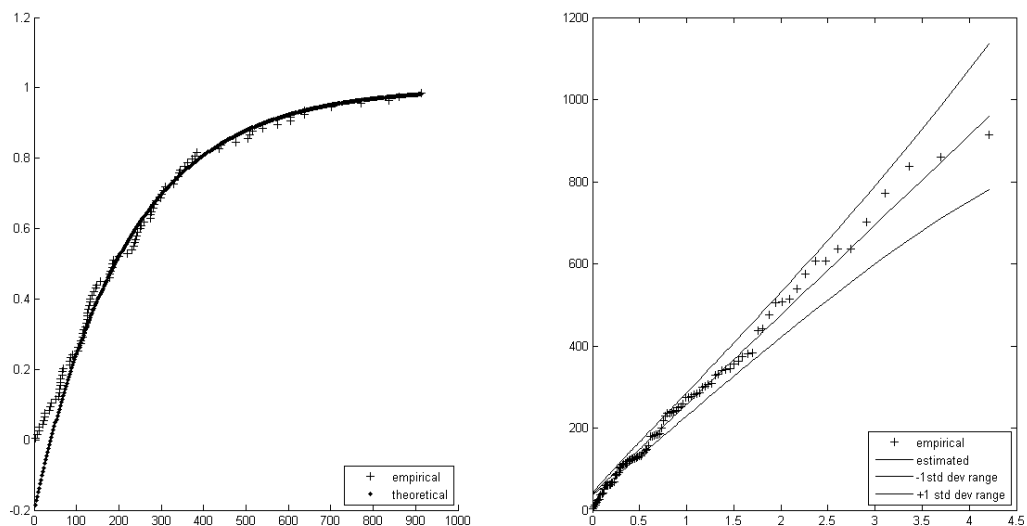
<sup>3</sup> Visual Basic for Applications



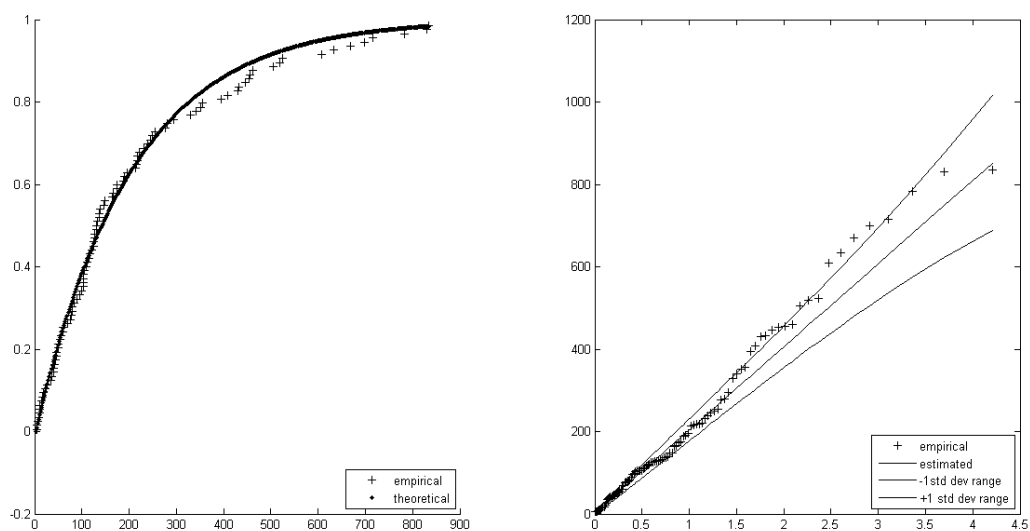
**Obrázek 6** Rozdělení výpočetního času 1. modifikace na příkladu e1-b



**Obrázek 7** Rozdělení výpočetního času 2. modifikace na příkladu e1-b

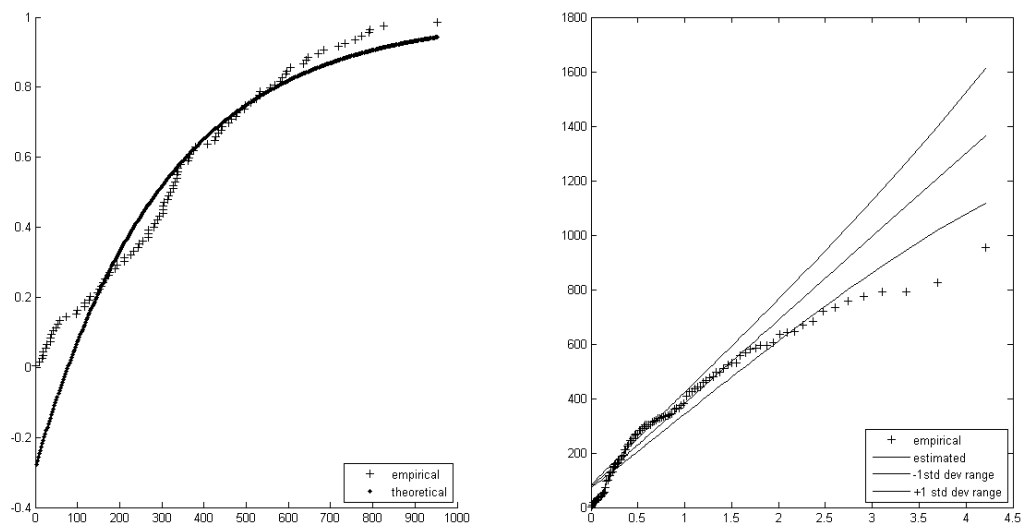


**Obrázek 8** Rozdělení výpočetního času 4. modifikace na příkladu e1-b



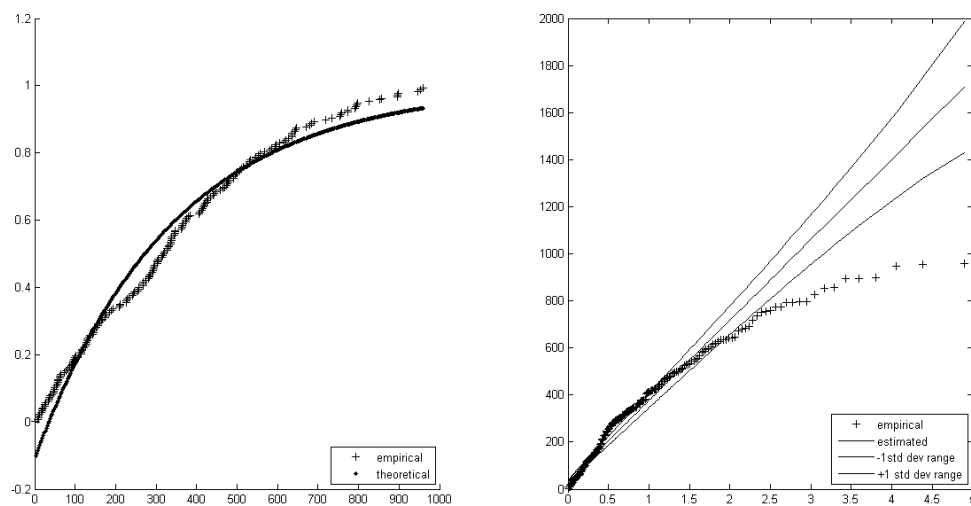
**Obrázek 9** Rozdělení výpočetního času 5. modifikace na příkladu e1-b

Z následujících dvou obrázků pro 1. a 3. modifikaci lze pozorovat, že se body neustále odchyľují o více než jednu směrodatnou odchylku od odhadu horního kvantilu (tj. mnoho bodů souvisejících s velkými výpočetními časy nespádá do hranice plus/minus jedna směrodatná odchylka). Můžeme tedy říci, že tato rozdělení výpočetního času nejsou exponenciální. V takovém případě neplatí výraz (1.3) a tyto modifikace nelze srovnávat tímto přístupem.



**Obrázek 10** Rozdělení výpočetního času 3. modifikace na příkladu e1-b při sto spuštěních

Obr. 10 zobrazuje rozdělení výpočetního času při 200 spuštěních třetí heuristiky. Lze vyčíst, že se body ještě více odchyľují, čímž se potvrzuje, že toto rozdělení výpočetního času neodpovídá exponenciálnímu rozdělení, což bylo potvrzeno při spouštění dalších úloh. Dále proto nebudeme ve srovnáních třetí modifikaci uvažovat.



**Obrázek 11** Rozdělení výpočetního času 3. modifikace na příkladu e1-b při dvě stě spuštěních

### 3.2 Porovnávání modifikací

Body v grafech nejvíce odpovídají distribuční funkci posunutého exponenciálního rozdělení. Proto budou dále uvedeny vypočtené odhadnuté hodnoty parametrů dvouparametrového exponenciálního rozdělení.

| <i>modifikace</i>            | <i>1</i> | <i>2</i> | <i>4</i> | <i>5</i> |
|------------------------------|----------|----------|----------|----------|
| tvar rozdělení ( $\lambda$ ) | 0,2315   | 0,2083   | 0,1792   | 0,2174   |
| posunutí ( $T$ )             | 0,0919   | 0,1655   | 0,1175   | 0,3538   |

**Tabulka 3** Odhad hodnot parametrů rozdělení

Pro srovnání modifikací heuristiky byla dále vypočtena pravděpodobnost, že jeden algoritmus najde řešení alespoň tak dobré jako cílová hodnota v kratším výpočetním čase než druhý algoritmus dle:

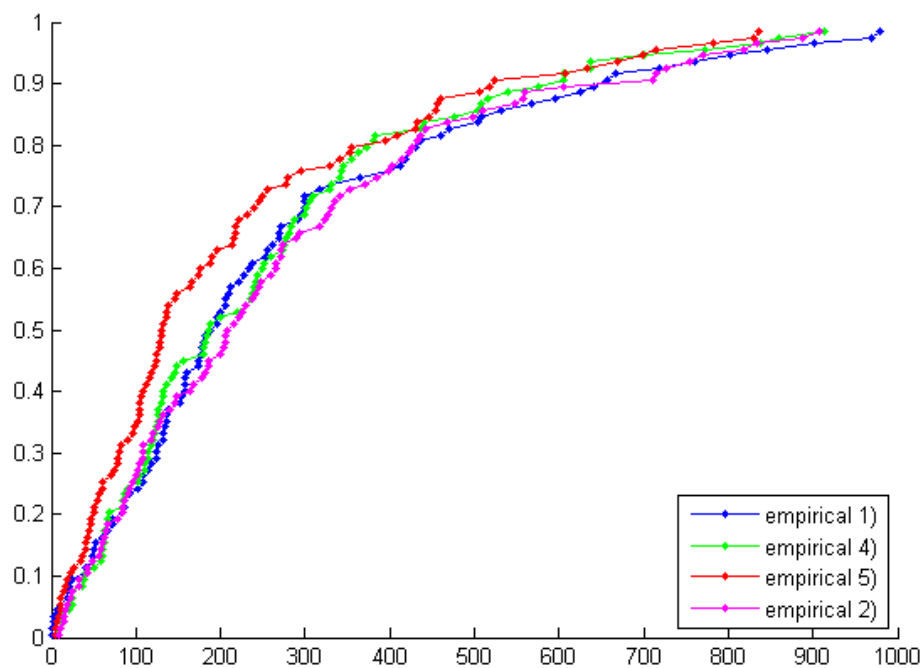
$$P(X_1 \leq X_2) = 1 - e^{-\lambda_1(T_2 - T_1)} + e^{-\lambda_1(T_2 - T_1)} \frac{\lambda_1}{\lambda_1 + \lambda_2}. \quad (1.4)$$

V tabulce č. 4 uvádím vypočítané pravděpodobnosti pro první, druhou, čtvrtou a pátou modifikaci, kdy označení porovnávaných modifikací je typu  $X_1 - X_2$ .

| <i>porovnávané modifikace</i> | <i>1 - 2</i> | <i>4 - 1</i> | <i>5 - 1</i> | <i>4 - 2</i> | <i>5 - 2</i> | <i>5 - 4</i> |
|-------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| $P(X_1 \leq X_2)$             | 0,524        | 0,509        | 0,544        | 0,533        | 0,566        | 0,534        |

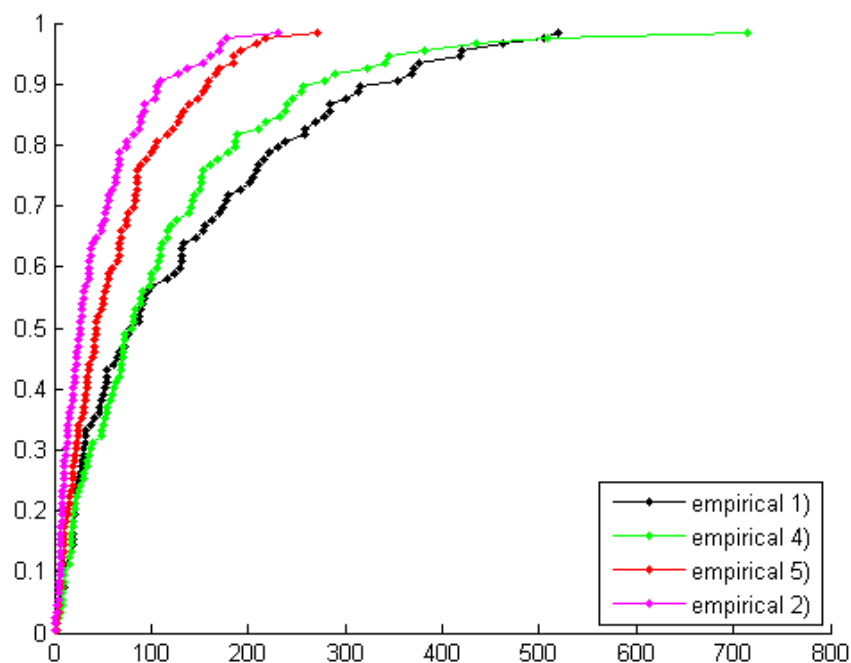
**Tabulka 4** Porovnání modifikací pomocí pravděpodobnosti

Pro vizualizaci srovnání ještě uvádím následující grafy. Rozdělení času výpočtu pro příklad e1-b je znázorněno tak, že na svislé ose je zaznamenána kumulativní pravděpodobnost a čas výpočtu (zde měřený v krocích, kolikrát byla heuristika spuštěna) na ose horizontální.



**Obrázek 12** Porovnání výsledků výpočtů modifikací na příkladu e1-b

Pro ukázkou větší rozdílnosti ve výkonnosti daných modifikací je uveden obrázek č. 13 vytvořený nad rozsáhlejší úlohou s2-a.



**Obrázek 13** Porovnání výsledků výpočtů modifikací na příkladu s2-a

V grafu jsou znázorněny výpočetní časy 1., 2., 4. a 5. modifikace. Je patrné, za kolik kroků je která modifikace schopna dosáhnout cílové hodnoty. Například pravděpodobnost nalezení řešení alespoň tak dobrého jako cílová hodnota v nejvýše 100 krocích je u první modifikace okolo 57% a u druhé modifikace asi 89%.

Pro porovnání výkonnosti vzhledem k velikosti úlohy, tedy zvyšujícímu se počtu uzlů, byla každá modifikace spuštěna stokrát v několika úlohách. Výstupem tohoto testování je uvedena následující tabulka shrnující výsledky modifikací na čtyřech úlohách.

V tabulce č. 5 je srovnání, jak se pravděpodobnosti, že první modifikace nalezne cílovou hodnotu v kratším nebo stejném počtu kroků mění s rozměrem úlohy. První porovnávaná modifikace je vždy uvedena před lomítkem, tedy například pravděpodobnost, že druhá modifikace nalezne cílovou hodnotu za kratší nebo stejný počet kroků než první modifikace, je v úloze s2-a přibližně 77%. Lze vyčíst, že se dané pravděpodobnosti s velikostí úlohy zvětšují. Pro menší úlohu s názvem e1b byly modifikace dle výkonnosti seřazeny následovně (od modifikace, která dosáhne cílové hodnoty pravděpodobně nejdříve): 5,4,1,2. Pro složitější úlohu s2-a je pořadí následující: 2,5,4,1.

|      | 2/1   | 4/1   | 5/1   | 2/4   | 2/5   | 5/4   |
|------|-------|-------|-------|-------|-------|-------|
| e1-b | 0,475 | 0,509 | 0,544 | 0,467 | 0,434 | 0,534 |
| e4-a | 0,567 | 0,498 | 0,586 | 0,569 | 0,482 | 0,587 |
| s1-a | 0,519 | 0,569 | 0,536 | 0,449 | 0,482 | 0,466 |
| s2-a | 0,769 | 0,586 | 0,722 | 0,703 | 0,567 | 0,648 |

**Tabulka 5** Odlišnost výkonu modifikací na různě složitých úlohách

Závěrem tohoto testování bych shrnula, že v úlohách se 77 hranami a postupně se zvyšující náročností v počtu povinných hran se pro jednodušší úlohy jevila nejvýkonnější pátá modifikace, následně čtvrtá, první a nakonec druhá modifikace s nejhoršími výsledky. Zajímavým zjištěním bylo, že čím více úlohy narůstaly na náročnosti, druhá modifikace se dostávala do popředí a pro úlohy se skoro dvojnásobným počtem hran se pořadí změnilo pouze tak, že nejvýkonnější je pro takové úlohy druhá modifikace, dále pak pořadí zůstalo.

Odhad pravděpodobnosti  $P(X_1 \leq X_2)$  je vypočten pro  $N=100, 300, 500$  a  $1000$  nezávislých spuštění každé modifikace na úloze e4-a. Tabulka 6 zobrazuje výsledky výpočtů.

| $N$  | 2/1   | 4/1   | 5/1   | 2/4   | 2/5   | 5/4   |
|------|-------|-------|-------|-------|-------|-------|
| 100  | 0,567 | 0,498 | 0,586 | 0,569 | 0,482 | 0,587 |
| 300  | 0,545 | 0,532 | 0,498 | 0,512 | 0,546 | 0,466 |
| 500  | 0,523 | 0,505 | 0,472 | 0,518 | 0,551 | 0,467 |
| 1000 | 0,516 | 0,487 | 0,449 | 0,529 | 0,567 | 0,462 |

**Tabulka 6** Stabilizace pravděpodobnosti závislá na počtu spuštění výpočtu

Lze pozorovat, že se odhad  $P(X_1 \leq X_2)$  stabilizuje s rostoucím počtem spuštění.

# Závěr

Cílem práce bylo porovnávání výkonnosti pěti modifikací randomizované heuristiky na různě složitých úlohách a vyhodnocení výsledků testování, které by mělo pomoci při volbě způsobu výpočtu praktického problému.

Modifikace heuristiky byly testovány celkem na 24 úlohách o velikosti 77 a 140 uzlů s různými počty povinných a nepovinných hran, počty vozidel a jejich kapacitami. Na těchto úlohách bylo provedeno sto spuštění každé modifikace a výsledky postupně porovnávány z hlediska výpočetních časů. Při porovnání dosažených hodnot s hodnotami z literatury bylo dosaženo nejmenší odchylky 10,9% při 10 000 spuštění, největší 27,1% a v průměru se nejlepší hodnoty řešení lišily o 15,7%.

Dalším párovým porovnáváním modifikací pomocí pravděpodobnosti vypovídající o schopnosti jedné modifikace dosáhnout zadané cílové hodnoty při menším počtu kroků než druhá modifikace lze shrnout, se zvyšující náročností v počtu povinných hran se pro jednodušší úlohy jevila nejvýkonnější pátá modifikace, následně čtvrtá, první a nakonec druhá modifikace s nejhorsími výsledky. Čím více úlohy narůstaly na náročnosti, druhá modifikace se postupně řadila až k prvnímu místu.

Jako rozšíření této práce bych navrhovala zobecnění metody párového porovnávání algoritmů, které by umožnilo porovnat s ostatními i třetí modifikaci, jež byla z porovnání vynechána kvůli pravděpodobnostnímu rozdělení, které neodpovídá rozdělení exponenciálnímu, což je pro popsání způsobu srovnání nezbytné.

# Literatura

- [1] RIBERA, C.C., ROSETI, I., VALLEJOS, R.: Exploiting run time distributions to compare sequential and parallel stochastic local search algorithms. *Journal of global optimization*, Vol. 54, No.2, 2012, s. 405-429.
- [2] HOOS, H.H., STÜTZLE, T.: Towards a characterisation of the behaviour of stochastic local search algorithms for SAT. *Artificial Intelligence*, Vol. 112, 1999, s. 213-232.
- [3] AIEX, R.M., RESENDE, M.G.C., RIBERO, C.C.: TTTPLOTS: a perl program to create time-to-target plots. *Optimization Letters*, Vol. 1, 2007, s. 355-366
- [4] CHAMBERS, J.M., CLEVELAND, W.S., KLEINER, B., TUKEY, P.A.: Graphical Methods for Data Analysis. *Chapman Hall*, London (1983)
- [5] BENAVENT, E., CAMPOS, V., CORBERÁN, A., MOTA, E.: The capacitated arc routing problem. A heuristic algorithm. *Qüestió*, Vol. 14, 1990, s. 107-122.
- [6] LONGO, H., POGGI DE ARAGAO, M. V. S. and UCHOA, E.: Solving capacitated arc routing problems using a transformation to the CVRP. *Computers and Operations Research*, Vol. 33, No. 6, 2006, s. 1823-1837.
- [7] FÁBRY, J.: Okružní a rozvozní úlohy. Habilitační práce. *Dostupné v archivu FIS VŠE v Praze*, 2014, s. 89-99.
- [8] LI, L. Y. O. and EGGLESE, R. W.: An interactive algorithm for vehicle routing for winter-gritting, *Journal of the Operational Research Society*, vol. 47, no. 2, 1996, s. 217–228.
- [9] PELIKÁN, J.: Randomized heuristics for capacitated arc routing problem. In *Mathematical Methods in Economics (MME2014)*. Olomouc: Palacký University in Olomouc, 2014, s. 772-776. ISBN 978-80-244-4208-2.