

1	Úvod.....	3
1.1	Vymezení tématu práce a důvod výběru tématu.....	3
1.2	Stručná charakteristika problémové oblasti v ČR a ve světě.....	3
1.3	Cíle práce	3
1.4	Předpoklady a omezení práce	3
1.5	Předpokládané vlastní přínosy práce	3
1.6	Určení práce.....	4
1.7	Struktura práce	4
2	Dopravní úlohy, jejich charakteristika a rozdělení	4
2.1	Dopravní úloha jazykem teorie grafů	5
2.2	Dopravní úloha jazykem teorie složitosti	5
2.3	Charakteristiky obecné dopravní úlohy	7
2.4	Komponenty dopravní úlohy	9
2.4.1	Vozidla /Vehicles/.....	9
2.4.2	Zákazníci /Customers/	9
2.4.3	Dopravní síť /Road network/	9
2.5	Základní typy dopravních úloh.....	9
2.5.1	Úloha obchodního cestujícího /Travelling Salesman Problem/.....	9
2.5.2	Kapacitní dopravní úloha /Capacitated Vehicle Routing Problem/ (CVRP).....	10
2.5.3	Dopravní úloha s více skladišti /Multiple Depot VRP/ (MDVRP).	10
2.5.4	Periodická dopravní úloha /Periodic/ VRP (PVRP)	10
2.5.5	Spojená dopravní úloha /Split Delivery VRP/	11
2.5.6	Stochastická dopravní úloha /Stochastic VRP/ (SVRP).....	12
2.5.7	VRP se zpětnou dopravou /VRP with Backhauls	12
2.5.8	Dopravní úloha s časovými okny /VRP with Time Windows/ (VRPTW).....	14
2.5.9	Dopravní úloha s přejímkou a dodávkou /VRP with Pickup and Delivery/ (VRPPD) 16	
2.5.10	Časově závislá varianta dopravní úlohy s přejímkou a dodávkou /Time Dependent Vehicle Routing Problem with Time Windows/ (TDVRPTW)	17
2.5.11	Dynamická dopravní úloha (DVRP).....	17
2.5.12	Ostatní typy dopravní úlohy	18
3	Metody řešení dopravních úloh	18
3.1	Hejnová inteligence /swarm intelligence/	19
3.1.1	Vlastnosti hejna.....	19
3.1.2	Sociální chování společenského hmyzu	19
3.1.3	Příčiny vzniku sociálního chování.....	20
3.2	Chování sociálního hmyzu jako základ pro výpočetní modely	21
3.2.1	Včely	21
3.2.2	Mravenci	22
3.2.3	Rozdíly mezi přirozenými a umělými mravenci.....	22
3.2.4	Postup aplikace ACO metaheuristiky na konkrétní úlohu:.....	23
3.3	Přehled ACO algoritmů	24
3.3.1	Ant System (AS) (Dorigo, Maniezzo et al. 1996)	24
3.3.2	Ant Colony System (ACS) (Dorigo 1997)	24
3.3.3	MAX-MIN Ant System (Stutzle and Hoos 1996; Stutzle and Hoos 2000).....	25
3.3.4	Graph based Ant System (Gutjahr 2000).....	25

3.3.5	Vztah ACO algoritmů a evolučních algoritmů	26
4	Software pro ACO optimalizaci	26
5	Praktická část	27
5.1.1	Popis použitého programu	27
5.1.2	Použití programu.....	32
5.1.3	Data	32
5.2	Výsledky	32
5.2.1	Vliv parametrů α a β na výkonnost úlohy.....	33
5.2.2	Vliv parametrů α a β - vyhodnocení.	37
5.2.3	Vliv rozhodovacího mechanismu	37
5.2.4	Vztah mezi TSP a VRP z hlediska volby vhodných parametrů.	42
5.2.5	Další zajímavá pozorování.....	45
6	Diskuse.....	47
7	Závěr	48
7.1	Shrnutí výsledků	48
7.2	Zhodnocení splnění stanovených cílů.....	48
7.3	Přínos autora práce k řešené problematice	48
7.4	Zhodnocení využitelnosti dosažených výsledků.....	49
7.5	Další náměty pro řešení.	49
8	Literatura.....	50

1 Úvod

1.1 Vymezení tématu práce a důvod výběru tématu

V poslední době jsou hodně studovány různé typy dopravních úloh. Objevují se nové a nové varianty řešení těchto úloh a další a další algoritmy. Algoritmy pro řešení dopravních úloh jsou jednak specializované výpočetní postupy zaměřené na konkrétní typ dopravní úlohy, jednak obecné metaheuristiky použitelné v širším rozsahu. K takovým metaheuristicám patří využití metod hejnové inelligence. V práci jsem se zabývala těmito pokročilými metodami, konkrétně metodami odvozenými z chování mravenců.

1.2 Stručná charakteristika problémové oblasti v ČR a ve světě

Problematika je intenzivně studovaná. Přehled dopravních úloh můžeme nalézt například v časopise Operations Research ISSN: 1526-5463, dále Journal of the Operational Research Society ISSN: 0160-5682 a v celé řadě dalších časopisů.

Hejnová inelligence je v současné době také velmi horkým tématem majícím řadu aplikací jak v civilní (řízení telefonního provozu, řízení leteckého provozu), tak ve vojenské oblasti (skupiny autonomních robotů, rozhodování na bitevním poli). Na toto téma probíhá v současnosti řada konferencí a vznikají nové časopisy zaměřené na tuto oblast, např.

Swarm Intelligence, Springer New York ISSN: 1935-3812

1.3 Cíle práce

Cílem práce je aplikace metod ACO optimalizace na úlohu obchodního cestujícího a dopravní úlohu. V teoretické části práce je provedena rešerše těchto metod, zhodnocení jejich výhod a nevýhod a v bodech je uveden postup který je třeba dodržovat chceme - li navrhnout ACO optimalizační algoritmus. V teoretické části práce jsem se snažila podat detailně i biologické pozadí ACO optimalizace. V experimentální části práce bylo mým cílem zjištění vlivu parametrů na ACO optimalizaci a zjištění možností využití daného programu napsaného v prostředí Matlab pro studium vlastností ACO algoritmu.

1.4 Předpoklady a omezení práce

Podmínkou pro úspěšné provedení práce je zvládnutí teoretického základu a vhodné softwarové vybavení pro provedení experimentů. Zde musím konstatovat, že VŠE na rozdíl od ČVUT nemá přístup (tedy mi alespoň není známo že by měla) do důležitých databází jako je Science Direct, Springer, The ACM Digital Library a k řadě dalších informačních zdrojů. Veškeré literární podklady byly proto získány prostřednictvím RNDr. Miroslava Pavelky PhD. pracujícího na ČVUT.

Po praktické stránce v současné době neexistuje vhodný volně dostupný program vhodný pro provádění ACO experimentů. Ke své práci jsem proto využila prostředí napsaném v programu Matlab, vycházejícího z AS algoritmu, viz příslušná kapitola.

1.5 Předpokládané vlastní přínosy práce

V teoretické části práce bych jako její přínos uvedla plánované propojení biologického pozadí z praktickými aplikacemi, značný přehled úloh a jejich řešení a uvedení souvislostí které nejsou na první pohled patrné (například vzájemná převoditelnost úloh). Dopravní

úloha je pojednávána z více hledisek, optimalizační techniky založené na hejnové inteligenci jsou popsány včetně chování sociálního hmyzu jako základu pro tvorbu algoritmů. V praktické části práce vidím hlavní přínos v ověření funkčnosti ACO algoritmu.

1.6 Určení práce

Práce může sloužit jako vstupní kurz pro zájemce o aplikace ACO optimalizací na dopravní úlohu. Poskytuje dosti podrobný návod jak postupovat při návrhu ACO algoritmu. Pokud jde o experimenty, tak současná implementace algoritmu může sloužit pro obeznámení s danou problematikou. Pro podrobnější studium vlivu parametrů na ACO optimalizaci je stávající řešení příliš pomalé (výpočty trvají příliš dlouho).

1.7 Struktura práce

V první části práce podávám přehled typů dopravních úloh a způsobů jejich řešení. Jelikož spektrum dopravních úloh je rozmanité a metod jejich řešení je značné množství a navíc v tomto směru probíhá bouřlivý výzkum, tak jsem se omezila jen na základní fakta a uvádím pouze některé metody. Některé algoritmy jsem v práci krátce popsala, jiné jsou jen zmíněny. Pokud to bylo možné, u všech algoritmů jsem se snažila uvést alespoň jeden relevantní odkaz na další literaturu.

Podrobněji se věnuji skupině metod, které jsou obecně označovány jako metaheuristiky. Jejich vývoj byl inspirován přírodními ději (mutace, selekce, přírodní výběr, fyzikální procesy při žíhání, a v neposlední řadě kolektivní chování živých organismů). Mezi optimalizačními technikami, které si vzaly inspiraci z kolektivního chování živých organismů hrají v současné době zřejmě nejvýznamnější úlohu metody optimalizace pomocí kolonií umělých mravenců nazývané /Ant Colony Optimization/ (ACO). Problematice řešení dopravní úlohy a úlohy obchodního cestujícího pomocí této techniky se věnuji také v experimentální části práce.

Práce vychází z anglicky psané literatury. Jelikož mi oficiální překlad některých odborných termínů není znám, a navíc některé odborné termíny zřejmě ani svůj překlad nemají, proto uvádím všechny relevantní termíny v textu ještě v původním anglickém znění, které jsem dala do lomených závorek.

2 Dopravní úlohy, jejich charakteristika a rozdělení

Problematika dopravních úloh je složitá a různorodá. Na jedné straně vychází z teorie grafů, na druhé straně z teorie složitost, z teorie rozhodování a z teorie optimalizace. Obecná dopravní úloha /Vehicle Routing Problem/ (VRP) je optimalizační úloha. Mějme daný počet vozidel, z nichž každé má danou kapacitu a daný počet zákazníků, z nichž každý má daný požadavek. Hledáme optimální řešení ve formě minimální ceny nebo ujeté vzdálenosti potřebné k uspokojení požadavků zákazníků s použitím daného počtu vozidel.

2.1 Dopravní úloha jazykem teorie grafů

Grafem se rozumí objekt popsáný množinou vrcholů a množinou hran. Mějme graf $G = (V, E)$, $E \subseteq [V]^2$, kde V je množina vrcholů, můžeme ji označit jako $V(G)$, a E je množina hran, můžeme ji označit jako $E(G)$ ¹.

Graf, jehož hrany a (nebo) vrcholy jsou opatřeny číselnými (nebo jinými) hodnotami, nazýváme ohodnoceným grafem. Hrana e_{ij} spojující vrchol V_i a vrchol V_j může být orientovaná (směřující z vrcholu i do vrcholu j , případně z vrcholu j do vrcholu i), nebo neorientovaná ($e_{ij} = e_{ji}$). V našem případě se bude jednat o prosté grafy (orientované nebo neorientované), v nichž násobnost každé hrany (=počet hran mezi vrcholy V_i a V_j je nejvýše rovna jedné²). Úplným grafem rozumíme neorientovaný graf v kterém je každý vrchol spojen hranou se všemi ostatními vrcholy.

Posloupnost vrcholů a hran $v_0 h_1 v_1 h_2 v_2$ se nazývá cesta (orientovaná případně neorientovaná). Je-li cesta uzavřená, nazývá se kružnice (uzavřená neorientovaná cesta) nebo cyklus (uzavřená orientovaná cesta).

Hamiltonovská cesta je cesta, která obsahuje každý vrchol grafu (tj. každý vrchol právě jednou). Je-li graf neorientovaný, mluvíme o Hamiltonovské kružnici.

Eulerovský tah (Eulerovská cesta) je tah, který obsahuje všechny hrany grafu (tj. každou hranu právě jednou).

Graf zobrazujeme nejčastěji pomocí incidenční matice. Incidenční matice grafu G o n vrcholech je matice $A(n, n)$ obsahující „1“ pokud hrana e_{ij} z vrcholu i do vrcholu j existuje a „0“ pokud neexistuje. V úplném grafu obsahuje matice samé „1“ a na diagonále „0“.

Podobně matice vzdáleností D obsahuje ohodnocení hran e_{ij} pokud hrana e_{ij} existuje, jinak „0“. Grafy G_1 a G_2 s incidenčními maticemi A_1 a A_2 jsou izomorfní, pokud existuje permutační matice P taková že platí $PA_1P^{-1} = A_2$. Permutační matice je matice obsahující v každém řádku a v každém sloupci právě jednu „1“, jinak matice obsahuje „0“.

Nejjednodušším typem dopravné úlohy je úloha obchodního cestujícího, tedy nalezení takové Hamiltonovské kružnice, aby součet ohodnocení hran byl minimální. Modifikací této úlohy dostaneme další typy dopravních úloh. Standardní dopravní úlohu můžeme definovat jako nalezení maximálně M cyklů (uzavřených cest, které mají právě jeden společný vrchol V_0 v terminologii dopravních úloh označovaný jako depo /depot/ a jejichž součet ohodnocení hran je minimální. Kladením si dalších podmínek a omezení dostaneme další typy dopravních úloh. Například ohodnocením vrcholů $C: C(G)$ a podmínkou pro každý cyklus (uzavřenou cestu) aby suma ohodnocení c_k pro k -tý cyklus nepřesáhla jistou hodnotu dostaneme kapacitní dopravní úlohu. Tak by se dalo pokračovat.

2.2 Dopravní úloha jazykem teorie složitosti

Dopravní úloha patří mezi úlohy kombinačně optimalizační. Jsou to takové typy úloh, u kterých hledáme nejlepší řešení z množiny přípustných řešení, která je diskrétní. Pro úlohu obchodního cestujícího roste množina přípustných řešení úměrně s faktoriálem počtu vrcholů $n!$, (počet jedinečných přípustných řešení pro úlohu o n vrcholech je roven $((n-1)!)/2$). Matematický základ pro řešení úloh tohoto typu tvoří teorie složitosti.

¹ Někdy mluvíme o trojici $G = (V, H, \varepsilon)$ kde ε je zobrazení incidence.

² Mezi těmiž dvěma vrcholy může existovat i více hran téhož typu (např. paralelní vedení apod.), to v definici prostého grafu není zachyceno (protože množina v matematickém smyslu je definována výčtem navzájem různých prvků).

Složitost úloh je posuzována z hlediska spotřeby výpočetních zdrojů. Výpočetní zdroje jsou zdroje použité výpočetním modelem pro řešení úlohy. Nejznámějšími zdroji jsou deterministický výpočetní čas (DTIME) a deterministický výpočetní prostor (DSPACE), který je reprezentován množstvím paměti potřebné k vyřešení úlohy.

Výpočetním modelem se často rozumí Turingův stroj³. Deterministický Turingův stroj odpovídá současným počítačům. Některé výpočetní úlohy se dobře analyzují s použitím neobvyklých výpočetních modelů jakým je nedeterministický Turingův stroj, umožňující provádět neomezené větvení úlohy.

Třídou složitosti /complexity class/ rozumíme skupinu všech takových výpočetních úloh které mohou být řešeny použitím určitého množství výpočetních zdrojů daného druhu. Nás budou zajímat dvě třídy úloh, úlohy P a NP . Úlohy třídy složitosti P mohou být řešeny deterministickým počítačem v polynomiálním čase. Úlohy třídy NP mohou být řešeny nedeterministickým počítačem v polynomiálním čase.

"Těžkost" a "úplnost" úlohy.

Množina úloh X je těžká pro množinu úloh Y , pokud každé Y může být "snadno" převedeno na X tak, aby řešení zůstalo stejné. Tedy těžká úloha /hard/ má takovou vlastnost, že umožňuje rychle vyřešit nějakou úlohu z dané třídy složitosti. Ekvivalentně můžeme říct, že existuje redukce – zúžení každé úlohy dané třídy na tuto těžkou úlohu. Pokud je daná úloha pro danou třídu těžká, ale zároveň do dané třídy patří, říkáme že je pro danou třídu úplná /complete/. NP-těžké úlohy jsou důležitou skupinou těžkých úloh. NP-těžké úlohy nejsou všechny NP úlohy. Můžeme rozlišit tři typy úloh:

- **Rozhodovací úlohy /Decision problems/.** Vyžadují odpověď ANO / NE
- **Funkční úlohy /Function problems/** Vyžadují komplexnější odpověď než pouze ANO/NE
- **Optimalizační úlohy /Optimization problems/** Hledají minimum / maximum.

Pro každou optimalizační úlohu existuje odpovídající rozhodovací úloha zda řešení existuje (ANO/NE) V případě úlohy obchodního cestujícího mluvíme o rozhodovací verzi úlohy (Úlohou je zjistit zda existuje cesta (Hamiltonův okruh) jejichž celková délka je menší než k .), funkční verzi úlohy (najít pořadí navštívených vrcholů) a optimalizační verzi (najít minimální vzdálenost).

Úloha obchodního cestujícího je NP těžká. Rozhodovací verze úlohy obchodního cestujícího je NP-úplná. Omezená úloha obchodního cestujícího (úloha obchodního cestujícího s úzkým hrdlem) /Bottleneck Traveling Salesman Problem/ je také NP těžká⁴. Zúžená úloha obchodního cestujícího tak, že vzdálenosti odpovídají euklidovským

³ Turingův stroj představuje "klasiku" v teorii složitosti. Turingův stroj tvoří šestici (6-tuple)

$M = (Q, \Sigma, q_0, \Pi, A, \delta)$, kde Q je konečná množina stavů, Σ je konečná množina symbolů (abeceda), q_0 je počáteční stav, $\Pi \in \Sigma$ je prázdný symbol, A je množina konečných stavů, $A \subseteq Q$, δ je jedno nebo vícehodnotová funkce $\delta: Q \setminus A \times \Sigma \rightarrow (Q \times \Sigma \times \{L, R\})$, kde L je posunutí vlevo a R vpravo. U deterministického Turingova stroje není δ vícehodnotová. Nedeterministický Turingův stroj proto umožňuje větvení výpočetních úloh. (funguje tak jako byl v každém kroku zkopírován na několik větví, současně prohledává všechny možnosti řešení. Rozhodovací mechanismus nedeterministického Turingova stroje pracuje jako nedeterministický konečný automat.

⁴ Najít takový Hamiltonův cyklus v grafu aby jeho nejdelší hrana byla minimalizovaná.

vzdálenostem v rovině (jsou symetrické a splňují trojúhelníkovou nerovnost) je také NP těžká. Vidíme, že různé úlohy jsou vzájemně na sebe převoditelné. Způsob formalizace dopravních úloh může být také různý. Přehled různých formulací zvláště pro kapacitní dopravní úlohu uvádí (Letchford and Salazar-González 2006). Autor uvádí různé dvou a tříindexové formulace dopravních úloh a diskutuje přednosti a nedostatky různých formulací. V experimentální části práce bude ukázáno jak stejný algoritmus může být použit pro řešení úloh se zdánlivě vzrůstající složitostí.

Transformacemi jedné úlohy v jinou se zabývala řada dalších autorů. Například Práce (Pisinger and Ropke 2007) se zabývá pěti typy dopravních úloh, pro které po transformaci⁵ je navržen jednotný adaptivní prohledávání velkého okolí /adaptive large neighborhood search/ (ALNS), což je rozšíření algoritmu publikovaného v práci (Ropke and Pisinger 2005). Podobně v práci (Beck, Prosser et al. 2002) autoři reformulovali (VRP) jako /open shop scheduling problem/ (SSP)

2.3 Charakteristiky obecné dopravní úlohy

Obecná okružní dopravní úloha je optimalizační úloha, kdy máme danou množinu zákazníků n , s kladnou celočíselnou poptávkou q , vozidlo s neomezenou kapacitou Q , jedno skladiště (depo, zdroj) a přepravní náklady mezi jednotlivými dvojicemi zákazníku a depem. Cílem je najít minimální náklady nebo minimální vzdálenost okruhu za podmínek, že vozidlo vyjede a vrátí se do depa, každý zákazník je obslužen právě jednou a celková kapacita pro daný počet vozidel, z nichž každé má danou kapacitu, daný počet zákazníků s požadavky a jejich celková kapacita na daném okruhu nepřekročí Q .

V matematickém modelu dopravního problému používáme bivalentní proměnné x_{ij} , $i=1,2,\dots,n$, $j=1,2,\dots,n$, kde hodnota „1“ udává že mezi místem A_i a A_j bude cesta v rámci trasy, a hodnota „0“ udává, že mezi těmito místy cesta nebude. Jelikož platí podmínka, že každé místo musí být navštíveno pouze jednou, bude součet řádkových a součet sloupcových vektorů x_{ij} roven jedné. Základní matematický model dopravní úlohy pak můžeme pak zapsat ve tvaru:

$$z = \sum_{i=1}^n \sum_{j=1}^n c_{ij}x_{ij},$$

který minimalizujeme. Za podmínek:

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, 2, \dots, n$$

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, 2, \dots, n$$

$$x_{ij} = 0(1), \quad i, j = 1, 2, \dots, n.$$

⁵ Jednalo se o následující úlohy: /vehicle routing problem with time windows/ (VRPTW), /capacitated vehicle routing problem/ (CVRP), /multi-depot vehicle routing problem/ (MDVRP), /site-dependent vehicle routing problem/ (SDVRP), (zákazník může být obslužen jen podmnožinou vozidel) a /open vehicle routing problem/ (OVRP), (vozidla se nemusí vrátit do depa). Všechny tyto úlohy byly transformovány na úlohu /rich pickup and delivery problem with time windows/ (RPDPTW)

Při řešení různých modifikací dopravních úloh se vychází obvykle z tohoto modelu, s obměnou dalších podmínek. Řešení dopravních úloh je součástí vědní disciplíny označované jako operační výzkum.

„Operační výzkum je možné charakterizovat jako vědní disciplínu nebo soubor relativně samostatných vědních disciplín, které jsou zaměřeny na analýzu různých typů rozhodovacích problémů. Mezi nástroje používané v rámci operačního výzkumu patří statistické metody, optimalizační a stochastické metody, teorie her, teorie front /queueing theory/ teorie grafů, teorie plánů a plánování, tvorba strategií, simulace.“ (Lagová, Jablonský (2004), "Lineární modely").

Hlavními aplikačními oblastmi operačního výzkumu jsou výstavba a zajištění materiálových toků v průmyslu, výstavba telekomunikačních sítí, veřejná hromadná doprava, navrhování elektronických obvodů, a další. Cílem je stanovit takovou úroveň provádění operací, aby bylo co možná nejlepší fungování celého systému.

V případě námi uvažovaného problému je úloha tvořená grafem skládajícím se z dvojice uzlů a hran $G = (V, E)$, kde $V = \{0, 1, \dots, n\}$ jsou uzly grafu, „0“ představuje depo a ostatní kladné hodnoty zákazníci, hrany E jsou vzdálenosti nebo náklady mezi jednotlivými uzly c_{ij} pro každou hranu (i, j) a přirozené Q . Chceme najít minimální náklady okruhu takové, že vozidlo vyjíždí a končí v depu, tj. v uzlu „0“, každý uzel i , kde $i = 1, \dots, n$ je zahrnut do okruhu právě jednou a počet uzlů, vyjma depa nesmí překročit Q . Pro $Q > 3$ je úloha NP-těžká. Kromě výše uvedených proměnných uvažujeme taky proměnnou Y , která představuje množství toku vycházející z depa a procházející uzly (i, j) . Přidáním toku do předchozího modelu s tím, že zde uvažujeme depo získáme model:

$$\min \rightarrow z = \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}$$

za podmínek:

$$\sum_{j=0}^n x_{ij} = 1, \quad i = 1, 2, \dots, n \quad (1)$$

$$\sum_{i=0}^n x_{ij} = 1, \quad j = 1, 2, \dots, n \quad (2)$$

$$\sum_{i=0}^n Y_{ij} - \sum_{i=0}^n Y_{ji} = 1, \quad j = 1, 2, \dots, n \quad (3)$$

$$Y_{ij} < Q x_{ij}, \quad i, j = 0, 1, 2, \dots, n \quad (4)$$

$$Y_{ij} > 0, \quad x_{ij} = 0(1), \quad i, j = 0, 1, 2, \dots, n$$

Pro zjednodušení neuvažujeme proměnné $X_{ii} (i = 0, \dots, n)$. Třetí omezení pro každý uzel nám zajišťuje, že řešení neobsahuje okruh zahrnující depo. Čtvrté omezení maximalizuje tok vycházející z uzlu „0“ - Q . Můžeme si všimnout, že tok uzlem (i, j) dává počet klientů, kteří mají být ještě navštíveni v rámci okruhu. Z toho důvodu toto omezení zaručuje, že okruh s více, než Q uzly nebude zahrnut v řešení.

2.4 Komponenty dopravní úlohy

Dopravní úloha je tvořena vozidly a jejich vlastnostmi, zákazníky a jejich požadavky, dopravní sítí a jejími vlastnostmi a případně dalšími omezujícími podmínkami.

2.4.1 Vozidla /Vehicles/

Vozidla často tvoří další omezující podmínky dopravního modelu. Vozový park může být homogenní nebo nehomogenní. V reálných aplikacích podmínka rovnocennosti použitých vozidel nebývá splněna. Limitující mohou být mechanické vlastnosti, jako například váha (některá vozidla nemohou využívat některé trasy, vybavenost vozidla (například někteří zákazníci mohou být obslouženi pouze vozidly která jsou vybavena automatickým nakladačem), vozidla mohou být omezena ve své kapacitě (druh a množství nákladu), a podobně.

2.4.2 Zákazníci /Customers/

Zákazníci požadují určité množství zboží (objednávka /order/) které musí být dodáno v určitém čase. Mluvíme o časovém okně /time window/ pro obsloužení zákazníka. Časové okno může být jen jedno, nebo může být více nespojitých časových oken (zákazník může být obsloužen buď mezi 7-8 hodinou ráno nebo mezi 4-5 hodinou odpoledne. Časové okno může být pevně dané /hard/ (to jest zákazník nemůže být obsloužen po jeho ukončení a vozidlo musí počkat na zákazníka pokud dorazí dříve), nebo může být časové okno volně dané /soft/. V tomto případě je obsloužení zákazníka v jinou než stanovenou dobu pokutováno časově závislou penalizací. Do účelové funkce je proto přidána pokutová funkce pro každého zákazníka. Nakonec do modelu mohou být zahrnuty i časy nakládky a vykládky zboží.

2.4.3 Dopravní síť /Road network/

Dopravní síť je reprezentovaná grafem, ve kterém zdroje (skladiště) a zákazníci představují uzly a hrany představují vzdálenosti, vyjádřené ve formě prostorových nebo časových vzdáleností. V případě, že jsou tyto vzdálenosti závislé na čase, což je v praxi běžný případ (většina prakticky řešených dopravních úloh musí brát denní dobu v potaz), mluvíme o časově závislém dopravním problému /time dependent VRP/.

2.5 Základní typy dopravních úloh

Kombinací různých prvků dopravní úlohy můžeme získat širokou škálu různých typů dopravních úloh, jejich přehled uvádí například (Toth and Vigo 2000).

2.5.1 Úloha obchodního cestujícího /Travelling Salesman Problem/

Všechny dopravní úlohy můžeme považovat za zobecnění klasické úlohy obchodního cestujícího. V této úloze k uspokojení zákazníků slouží jen jedno vozidlo. Navíc je zde podmínka že každý zákazník musí být navštíven právě jen jednou. Bohužel se ukázalo, že řešení problému obchodního cestujícího nemůže být základem pro řešení obecných dopravních úloh. (Algoritmy pro řešení rozsáhlých úloh obchodního cestujícího pro dopravní úlohy nelze použít (Bachem, Hochstattler et al. 1996))⁶.

⁶ Toto platí pro klasické výpočetní techniky. V případě metaheuristik může být situace poněkud jiná, viz experimentální část práce.

Úloha obchodního cestujícího může mít řadu modifikací, o některých jsem se již zmínila. Příbuzná úloha se nazývá **úloha čínského pošťáka** /Chinese postman problem/. Zde je požadavek najít nejkratší cestu v grafu tak aby byl každý vrchol navštíven minimálně jednou. Viz (Edmonds and Johnson 1973)

Příbuznou úlohou je také **úloha hada v krabici** /snake-in-the-box problem/. Úlohu poprvé popsal W. H. Kautz (Kautz 1958). Úloha je inspirována teorií samoopravných kódů /error correcting codes/. Cílem úlohy je vměstnat co nejdelšího hada do hyperkrychle tak, aby se nekřížil a nedotýkal. Podobnou úlohou je /coil-in-the-box problem/, kdy se do hyperkrychle vměšťává cyklus.

2.5.2 Kapacitní dopravní úloha /Capacitated Vehicle Routing Problem/ (CVRP)

Kapacitní dopravní úloha je základní verzí dopravní úlohy. Název úlohy je odvozen od omezené kapacity vozidel. Pokud toto kapacitní omezení odstraníme, a navíc pokud řekneme že všichni zákazníci musí být obslouženi během jediné cesty (musí ležet na jediné trase), dostaneme klasickou úlohu obchodního cestujícího /Travelling Salesman Problem/, viz výše. V základní verzi kapacitní dopravní úlohy jsou požadavky zákazníků pevně dané a jsou známy předem. Dodávky jednotlivým zákazníkům nemohou být rozděleny, to jest k obslužení zákazníka nemůže být použito více než jedeného vozidla. Vozový park je homogenní a v úloze je jen jediné depo. Kritériem úlohy je minimalizace ceny, která je vyjádřena vzdáleností nutné k obslužení všech zákazníků.

Řešením kapacitních dopravních úloh se zabývá řada autorů, namátkou můžeme uvést práce (Vigo 1996; Augerat, Belenguer et al. 1998)

2.5.3 Dopravní úloha s více skladišti /Multiple Depot VRP/ (MDVRP).

Zákazníci jsou uspokojováni s využitím více zdrojů (skladišť). Vhodné algoritmy uvádějí například práce (Dumas, Desrosiers et al. 1991; Wu, Low et al. 2002; Nagy and Salhi 2005). Když jsou skupiny zákazníků shluknuty kolem jednotlivých skladišť, problém může být formulován jako více klasických dopravních úloh. Avšak když tomu tak není a zákazníci jsou promícháni se skladišti, tak lze úlohu považovat za dopravní úlohu s více skladišti a podle toho taky řešit. MDVRP pak spočívá v přiřazení jednotlivých zákazníků ke skladištím. Vozidla jsou pak k dispozici v každém skladišti. Každé vozidlo vyjede ze skladiště, obsluží přiřazené zákazníky a vrátí se do toho samého skladiště.

Minimalizujeme počet vozidel a ujetou vzdálenost. Model opět vychází ze základu, navíc zde uvažujeme více skladišť. Množinu vrcholů trasy označíme jako $V = \{v_1, \dots, v_n\} \cup V_0$, kde $V_0 = \{v_{01}, \dots, v_{0d}\}$ jsou vrcholy reprezentující skladiště, v_i jsou jednotliví zákazníci.

Okruh je teď definován jako $R_i = \{d, v_1, \dots, v_m, d\}$, kde $d \in V_0$.

2.5.4 Periodická dopravní úloha /Periodic/ VRP (PVRP)

V klasické úloze je typickým plánovacím obdobím jeden den. V případě periodického problému je klasické plánování zobecněno na M dnů. Specifikem je zde, že vozidlo se nevrací do skladiště ve stejný den, ve který z něho vyjelo. Budeme minimalizovat počet vozidel a čas potřebný k obslužení všech zákazníků, tzn. náklady na všechny okruhy. Každý zákazník má danou denní poptávku, která musí být v jednom okruhu obslužená jedním vozidlem. Když $M=1$, jde o klasický dopravní problém. V případě periodického

problému se může poptávka jednotlivých zákazníků zařazovat do skupin, tvořící jednotlivé okruhy pro každý den. Každý zákazník musí být navštíven k -krát, kde $1 \leq k \leq M$. Lze uvažovat multi-úrovňový kombinatorický optimalizační problém:

Úroveň jedna – Cíl je vytvořit skupinu alternativ pro každého zákazníka, když např. perioda plánování je $t = 3$, $\{d_1, d_2, d_3\}$ pak možné kombinace návštěvy jsou $0 \rightarrow 000$, $1 \rightarrow 001$, $2 \rightarrow 010$, $3 \rightarrow 011$, $4 \rightarrow 100$, $5 \rightarrow 101$, $6 \rightarrow 110$, $7 \rightarrow 111$. Výsledky kombinací jsou v tabulce Tab. 1

Zákazník	Denní poptávka	Počet návštěv	Počet kombinací	Možné kombinace
1	30	1	3	1, 2, 4
2	20	2	3	3, 5, 6
3	20	2	3	3, 5, 6
4	30	2	3	1, 2, 4
5	10	3	1	7

Tab. 1 Periodická dopravní úloha

V druhé úrovni musíme vybrat jednu alternativu pro každého zákazníka, tak aby denní požadavky byly uspokojeny. Pak musíme vybrat zákazníky, které máme každý den navštívit. Ve třetí úrovni řešíme dopravní problém na každý den.

2.5.5 Spojená dopravní úloha /Split Delivery VRP/

Zákazník může být obslužen více vozidly pokud to snížilo celkové náklady. Tohle je velmi důležité v případě, že objednávka zákazníka je tak velká jako kapacita vozidla, nebo větší. Cílem je opět minimalizace nákladů. Nejjednodušší cestou řešení je rozdělení požadavku zákazníka na menší nedělitelné celky.

Uvažujme proměnnou x_{ij} jako binární definovanou pro $i \neq j$ a v případě optimálního řešení se rovná „1“. Vozidlo označíme jako v , jedoucí z i do j . Y_{iv} označuje i -tého zákazníka obsluženého vozidlem v . Formulace daného problému může být následující:

$$\min \rightarrow \sum_{i=0}^n \sum_{j=0}^n \sum_{v=1}^{\bar{m}} C_{ij} x_{ijv}$$

za podmínek

$$\sum_{i=0}^n x_{ikv} - \sum_{j=0}^n x_{kjh} = 0, \quad (k = 0, \dots, n; v = 1, \dots, \bar{m}) \quad (1)$$

$$\sum_{v=1}^{\bar{m}} y_{ij} = 1, \quad (i = 1, \dots, n) \quad (2)$$

$$\sum_{i=1}^n q_i y_{iv} \leq Q_v, \quad (v = 1, \dots, \bar{m}) \quad (3)$$

$$\sum_{j=0}^n x_{ijv} \geq y_{iv}, \quad (i = 1, \dots, n; v = 1, \dots, \bar{m}); \quad (4)$$

$$x_{ijv} \in \{0, 1\}, \quad (i, j = 0, \dots, n; 1, \dots, \bar{m}), \quad (5)$$

$$0 \leq y_{iv} \leq 1, (i = 1, \dots, n; v = 1, \dots, \bar{m}), \quad (6)$$

$$\sum_{v=1}^{\bar{m}} \sum_{j=0}^n x_{ij} \geq \sum_{v=1}^{\bar{m}} y_{iv} = 1, (i = 1, \dots, n). \quad (7)$$

V prvním omezení jsou ošetřeny podmínky toku. Druhé omezení znamená, že poptávka kteréhokoliv zákazníka je kompletně uspokojená. Třetí omezení zajišťuje nepřekročení kapacity vozidla, zatímco čtvrté omezení garantuje že když je zákazník navštíven vozidlem v , tak to samé vozidlo taky opouští navštíveného zákazníka. Páté a šesté omezení jsou tzv. subnávrštvní podmínky. Sedmé omezení zaručuje, že každý zákazník i bude navštíven minimálně jednou.

2.5.6 Stochastická dopravní úloha /Stochastic VRP/ (SVRP)

Stochastická dopravní úloha je úloha s jednou nebo více náhodnými komponenty:

Náhodným zákazníkem – každému zákazníkovi v_i je přiřazená pravděpodobnost p_i .

Stochastická poptávka – poptávka každého zákazníka d_i je náhodná proměnná

Stochastický čas – Čas obsluhy δ_i a čas příjezdu a odjezdu τ_i jsou náhodné proměnné.

Uvažují se zde dvě úrovně řešení:

První je vypočteno před uskutečněním náhodné proměnné

Druhá úroveň uvažuje již změnu náhodné proměnné a opravuje řešení o její změnu.

Minimalizujeme zde počet vozidel a čas potřebný k uspokojení zákaznické poptávky s náhodnými proměnnými. Z toho taky vyplývá, že když máme nějaké náhodné proměnné, nelze uspokojit všechny požadavky v přesně daných požadavcích. V tom případě může rozhodovatel buď požadovat satisfakci za některá omezení s danou pravděpodobností, nebo začlenit do modelu opravnou akci když omezení je vyvoláno.

Minimalizujeme
$$\sum_{i < j} c_{ij} x_{ij} + Q(x),$$

kde x_{ij} je celočíselná proměnná rovnající se časovému intervalu (v_i, v_j) z prvního řešení.

Když $i, j > 1$, pak x_{ij} může nabývat pouze hodnoty „0“ nebo „1“, když $i = 1$, $x_{ij} = 2$ když vozidlo udělá okruh mezi skladištěm a v_j . $Q(x)$ je druhá pomocná funkce. Závisí na partikulární volbě pomocné akce. Např.

Vrátit se do skladiště když vozidlo je plné, vyložit a pak znovu naložit dle plánu.

Vrátit se do skladiště, když vozidlo je plné a preoptimalizovat zbývající část okruhu.

Plánovaný preventivní návrat do skladiště, i když vozidlo není plné. V takovém případě může být rozhodnutí závislé na již navštívených zákaznicích, a vzdáleností do skladiště.

Proveditelnost: Pokud jsou některá data náhodná, pak už není možné zajistit proveditelnost úlohy za splnění všech omezujících podmínek pro všechny realizace náhodných proměnných. Proto buď přijmeme že omezující podmínky budou splněny jen s jistou pravděpodobností, nebo musí být do modelu zahrnuty opravné mechanismy pro případ že omezující podmínky budou překročeny.

2.5.7 VRP se zpětnou dopravou /VRP with Backhauls

Se zpětnou dopravou /VRP with Backhauls/ (VRPB). Pokud po uspokojení požadavků musí být od zákazníků něco odebráno. Viz například (Goetschalckx and Jacobsblecha 1989; Thangiah, Potvin et al. 1996; Wade and Salhi 2002; Zhong and Cole 2005).

Všechno zboží musí být doručeno před vyzvedáváním jiného, ani v jednom z případů nesmí být překročena kapacita. Cílem je najít takový okruh, který minimalizuje ujetou vzdálenost. Formulace: Ke klasickému VRP si definujeme p jako vektor zákazníků s doručovací poptávkou.

Doručení:

Máme daný okruh $R_i = \{v_0, v_1, \dots, v_{m+1}\}$ vozidlo přidělené danému okruhu s kapacitou C . Tohle omezení můžeme matematicky zapsat jako $C_d(v_k) \leq C$ a $C_d(v_{k+1}) > C$, kde $C_d(v_k)$ je celkové množství zboží dodáno všem zákazníkům na daném okruhu vycházející z depa v_0 a končící u zákazníka v_k : $C_d(v_k) \leq \sum_{v_i \in P(1, v_k)} d_i$. $P(1, v_k)$ označuje zákazníky navštívené v rámci jednoho okruhu z depa do v_k , zahrnující zákazníka v_k .

Zákazníci s požadavkem vyzvednutí:

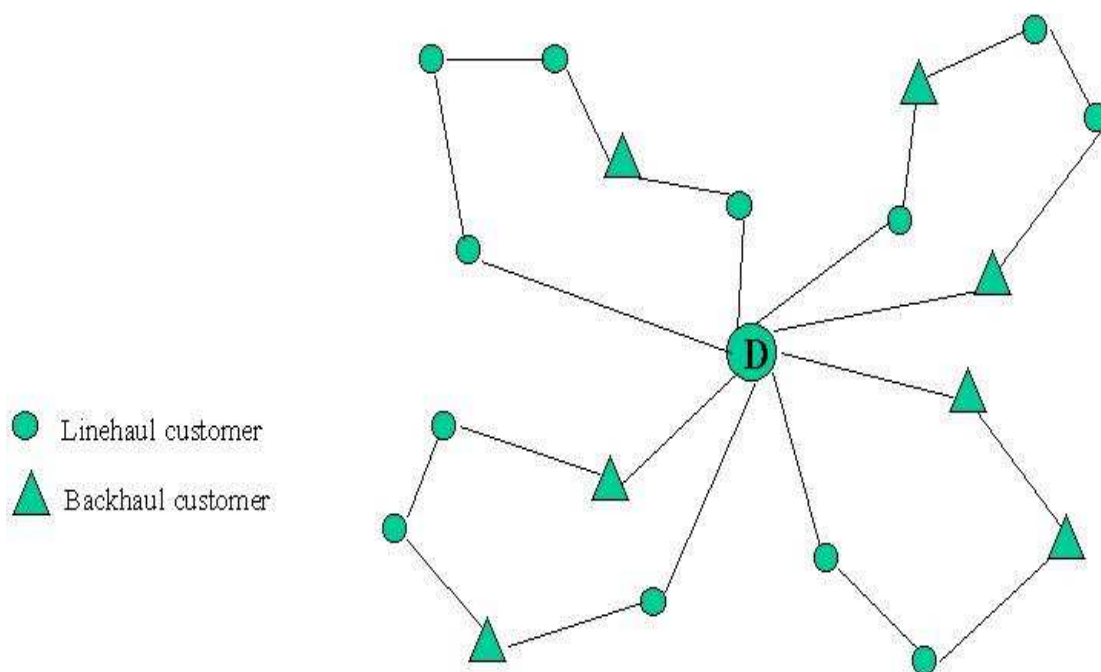
Tohle omezení zaručuje, že vozidlo má dostatečnou kapacitu na vyzvednutí zboží od všech zákazníků daného okruhu.

$C_p(v_k) \leq C$ a $C_p(v_{k+1}) > C$, kde $C_p(v_k)$ je celkové množství vyzvednutého zboží od všech zákazníků na daném okruhu zahrnující uzel v_k : $C_p(v_k) \leq \sum_{v_i \in P(1, v_p)} p_i$.

Nákladové omezení:

Nákladové omezení může být porušeno v jakémkoliv uzlu. Takové porušení závisí na pořadí zákazníků. Mějme $L(v_k)$ náklad zákazníka právě po zákazníkovi v_k .

Předpokládejme, že vozidlo opouští depo s počátečním nákladem $L(1) \leq C$. Pak náklad auta v jakémkoliv bodě okruhu je $L(v_k) = C_p(v_k) + L(1) - C_d(i_k)$. Náklad daný touto rovnicí může převyšovat kapacitu vozidla. Tohle znamená, že okruh má neomezené (nepřípustné) řešení, protože vozidlo nemůže obsloužit následujícího zákazníka v_{k+1} na okruhu. Okruh je tak měřitelný pouze při dalším omezení $L(v_k) \leq C$ a $L(v_{k+1}) \leq C$. Níže můžeme vidět schéma VRP se zpětnou dopravou.



Obr. 1 Dopravní úloha se zpětnou dodávkou. Viz http://neo.lcc.uma.es/radi-ae/ WebVRP/index.html?/Problem_Descriptions/MDVRPDesc.html

2.5.8 Dopravní úloha s časovými okny /VRP with Time Windows/ (VRPTW)

V případě dopravní úlohy s časovými okny nadále zůstávají kapacitní omezení. Mimo tato omezení popsaná výše v úloze CVRP je tu další omezení – zákazníci mohou být obslouženi jen v předem daném časovém intervalu. Časová okna jsou svázána s horizontem plánování. (Pokud plánujeme 5 dnů dopředu, pak je těžké udělat rozvrh pro časová okna v trvání minut.) Časové okno je přiděleno každému zákazníkovi $v \in V$ v intervalu $[e_v, l_v]$ ve kterém musí být zákazník obsloužen. Intervalu $[e_0, l_0]$ v depu se říká „plánovací horizont“.

Formální popsání problému je následující: Cílem je minimalizovat počet vozidel a celkový čas potřebný k obsluze zákazníků v jejich požadovaném čase. Omezení jsou následující:

Problém se stává nepřipustným když je zákazník obsloužen v čase přesahujícím horní hranici časového okna

Vozidlo přichází před dolní hranici intervalu časového okna a způsobí to čekání na okruhu

Každý okruh musí začínat a končit v časovém okně přiděleném depu

V případě nepevného časového okna pozdější obsluha nepůsobí na přípustnost řešení, ale přidává se do formulace modelu penalizační funkce

Z výše uvedeného plyne následující:

Označme b_v začátek obsluhy zákazníka v . Pro okruh $R_i = \{v_0, v_1, \dots, v_{m+1}\}$ bude přípustné řešení pouze s přidáním podmínek $e_{v_i} \leq b_{v_i} \leq l_{v_i}$, $1 \leq i \leq m$, a $b_{v_m} + \delta_{v_m} + c_{v_m,0} \leq l_0$. Opatřit to, že vozidlo jede k následujícímu zákazníkovi co nejdříve po ukončení obsluhy

aktuálního zákazníka b_{v_i} , může být rekurzivně zapsáno jako

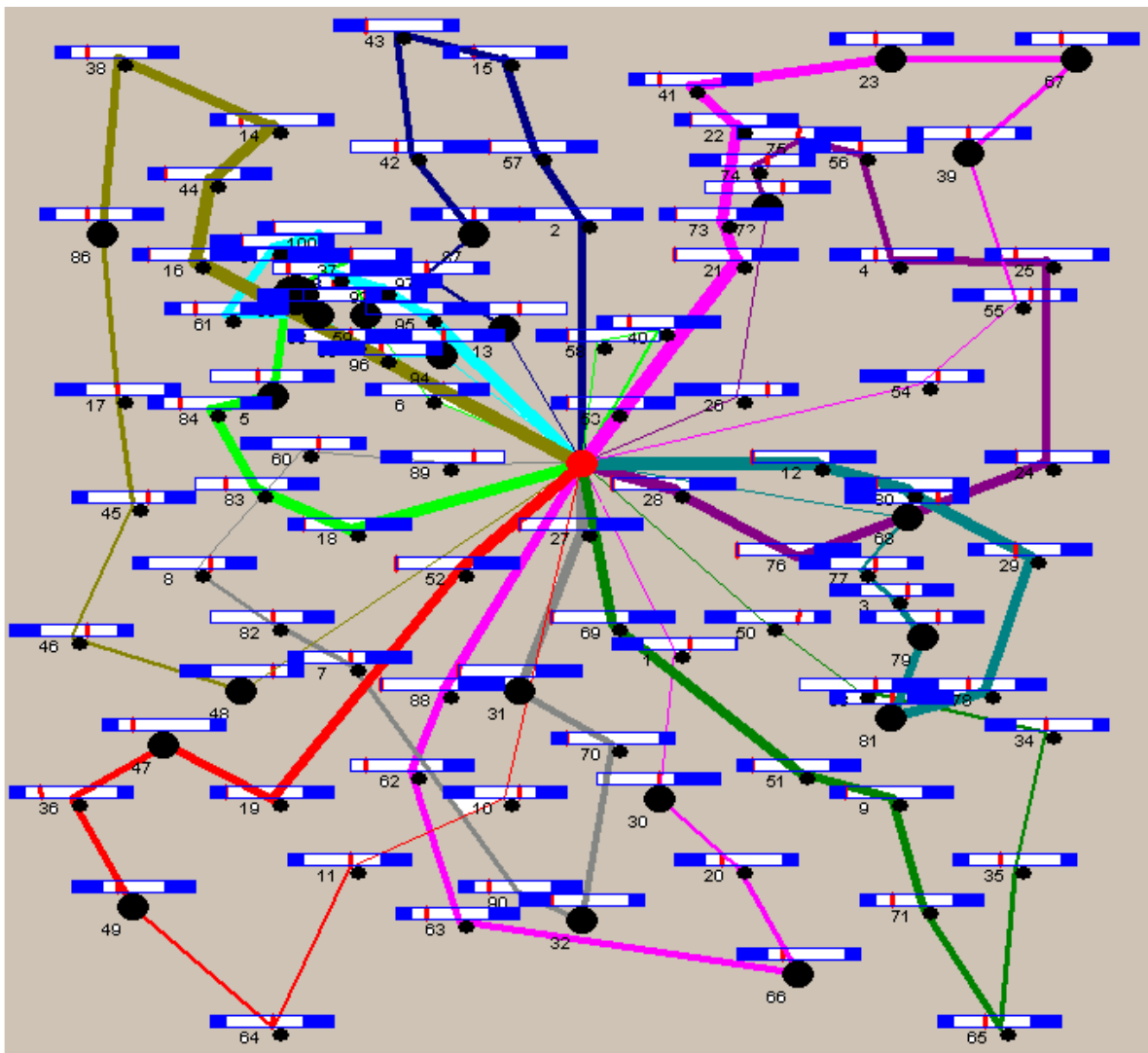
$b_{v_i} = \max\{e_{v_i}, e_{v_{i-1}} + \delta_{v_{i-1}} + c_{v_{i-1}, v_i}\}$ s $b_0 = e_0$ a $\delta_0 = 0$. Takto může být doba čekání u zákazníka v_i následující $w_{v_i} = \max\{0, b_{v_i} - b_{v_{i-1}} + \delta_{v_{i-1}} + c_{v_{i-1}, v_i}\}$. Náklady okruhu i jsou

v tomto případě následující: $C_{VRPTW}(R_i) = \sum_{i=0}^m c_{i, i+1} + \sum_{i=1}^m \delta_i + \sum_{i=0}^m w_{v_i}$. Pro řešení S s okruhem

R_1, \dots, R_m náklady řešení S jsou dány $F_{VRPTW}(S) = \sum_{i=1}^m C_{VRPTW}(R_i) + M$, kde M je veliké

číslo. M je přidáno z důvodu minimalizace velikosti vozidla je hlavní cíl VRPTW. S se říká přípustné když všechny okruhy patřící S jsou přípustnými a všichni zákazníci jsou obsluženi v rámci jednoho okruhu. Předpokládáme že na počátku všechna vozidla opouští depo v nejdřívejším možném čase e_0 . Abychom získali řešení pro VRPTW, přizpůsobíme čas odjezdu z depa každého vozidla tak, abychom počítali s odhadnutým nezbytně nutným časem čekání na okruhu.

Pro dopravní úlohu s časovými okny (VRTPW) je využíváno řady heuristik a metaheuristik. Příkladem může být lokální vyhledávání (Li and Lim 2003; Braysy, Hasle et al. 2004), tabu vyhledávání (Lau, Sim et al. 2003; Ho and Haugland 2004), simulované žíhání (Tan, Lee et al. 2001) a další. Přehled úloh s časovými okny spolu se způsobem jejich řešení uvádí (Cordeau 2000).



Obr. 1 Příklad řešení dopravní úlohy s časovými okny. Modrá a bílá pásma představují časová okna, v bílém pásmu je možné zákazníka obsloužit. Červený svislý proužek ukazuje dobu kdy k obslužení daného zákazníka skutečně dojde. Zdroj: <http://neo.lcc.uma.es/radi-aeb/WebVRP/images/map-vrptw.png>

2.5.9 Dopravní úloha s přejímkou a dodávkou /VRP with Pickup and Delivery/ (VRPPD)

V této dopravní úloze jsou zdroje pro uspokojování zákazníků rozptýleny v dopravní síti. Úlohy se také nazývají úlohami typu "dial-a-ride", což znamená, že je dané nejen časové okno pro dodání (a nebo i příjem) zboží, ale také je v úloze zahrnuta penalizace pokud dodavatel nebo zákazník musí čekat. Úlohou se zabývá řada autorů, namátkou můžeme uvést práce (Dumas, Desrosiers et al. 1991; Shang and Cuff 1996; Nanry and Wesley Barnes 2000; Tzoreff, Granot et al. 2002). Stochastickou a dynamickou variantou úlohy se zabývá (Swihart and Papastavrou 1999)

Nejjednodušší situace nastává když všechny požadavky jsou uspokojovány prostřednictvím vozidla startujícího z depa a všechny produkty odvezené od zákazníků zpět do depa přivezeny. V takovém případě neprobíhá výměna zboží mezi jednotlivými zákazníky.

Druhou alternativou je uvolnění podmínek kdy každý zákazník musí být vozidlem navštíven právě jednou. Další variantou je situace kdy každé vozidlo musí nejprve rozvézt veškeré zboží a teprve pak přijímat produkty od zákazníků.

Objektivní funkcí může být minimalizace potřebného počtu vozidel a minimalizace součtu dob jízdy. S omezením že vozidla musí mít dostatečnou kapacitu pro dopravu zboží k zákazníkům a pro zpětnou dopravu jejich produktů.

Úloha je proveditelná pokud celkové množství zboží přiřazené k dané trase nepřesáhne kapacitu obsluhujícího vozidla, a pokud vozidlo má dostatečnou kapacitu pro zpětný odvoz produktů od zákazníků.

2.5.10 Časově závislá varianta dopravní úlohy s přejímkou a dodávkou /Time Dependend Vehicle Routing Problem with Time Windows/ (TDVRPTW)

Jedná se o rozšíření dopravního problému s časovými okny (VRPTW) a jeho případné aplikace na městské prostředí, kdy je ohodnocení hran grafu závislé na čase. V této úloze může docházet k zajímavému narušení FIFO⁷ časové souslednosti (vozidla která dřív vyjela mohou v důsledku proměnlivé propustnosti silnic dorazit později než vozidla která vyjela za nimi).

Dynamickou variantou této úlohy se zabývá (Swihart and Papastavrou 1999). Práce zmíněných autorů proto spadá také do následujícího bodu, nicméně jej uvádím zde.

Požadavky na služby v čase jsou obvykle rozděleny podle Poissonova procesu⁸.

Rozmístění zastávek pro požadavky jsou nezávislé a rovnoměrně rozptýlené. Jednotlivé vozidlo musí odvést požadavek ze zastávky do místa určení. Jakmile je požadavek vybrán, musí být uvolněn pouze v místě určení. Místa určení jsou rovnoměrně rozmístěna. Cílem autorů bylo minimalizovat čas potřebný pro uspokojování požadavků. V práci je diskutován případ kdy má vozidlo jednotkovou kapacitu a případ kdy má vozidlo kapacitu větší a může najednou uspokojit více požadavků.. Také je diskutována situace za slabého a silného provozu.

2.5.11 Dynamická dopravní úloha (DVRP).

V tomto případě nejsou požadavky zákazníků předem zcela známé, ale přichází postupně v průběhu procesu. Jelikož se nové požadavky postupně objevují, ke změnám tras vozidel dochází průběžně. Předpokladem je že mezi řidičem a dispečerem existuje on-komunikační systém pracující v reálném čase. V takovém systému má řidič jen částečnou informaci o své budoucí trase. Řešení úlohy tohoto typu popisuje (Kilby, Prosser et al. 1998), (Ichoua,

⁷ První přijde – první odejde /first in – first out/

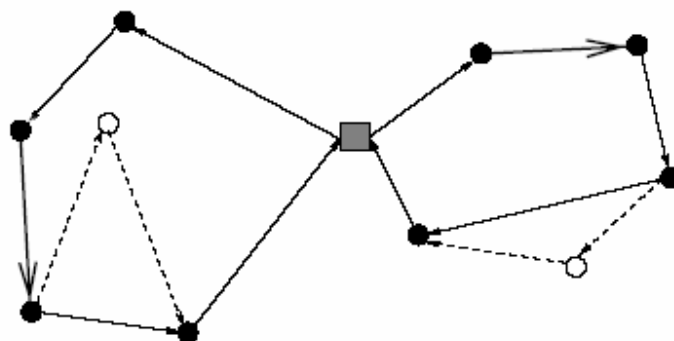
⁸ Jde o takový proces, při němž jsou změny možné pouze přechodem do nejbližší vyššího stavu. Vycházíme z následujících předpokladů:

1. Pravděpodobnost přechodu ze stavu n do stavu $n+1$ v časovém intervalu $(t, t+\Delta t)$ je λ .
2. Pravděpodobnost setrvání v témže stavu v časovém intervalu $(t, t+\Delta t)$ je $1-\lambda$.
3. Pravděpodobnost ostatních přechodů je v porovnání s předešlými zanedbatelná.
4. Stav „ -1 “ neexistuje

Vztahy určující rozložení pravděpodobnosti v poissonově procesu viz:

<http://www.fm.vslib.cz/~ksi/cz/mater/oa/obsluha/html/poisson.html>

Gendreau et al. 2000) a (Gendreau 1999). Přehled algoritmů a citací k dané problematice uvádí také (Wu, Beck et al. 2004).



Obr. 2 Varianta dopravní úlohy s 8 statickými (tmavý kruh) a 2 dynamickými zákazníky (světlý kruh). Výraznou šipkou je označena aktuální poloha vozidla, čtverec značí depo, přerušované šipky značí plánované, a plné šipky stávající trasy. Podle práce (Larsen, Madsen et al. 2002).

Většina výzkumu stochastického směřování vozidel je zaměřena na minimalizaci očekávané dopravní vzdálenosti. Obecně jednoduchá rekurzivní funkce (například navrácení se do depa) /depot/ je možné v průběhu /pokud jsou překročeny limitující podmínky/. Rekurzivní funkce přidává jistou cenu k očekávání (viz původní práce (Bertsimas 1992), (Laporte, Louveaux et al. 1992)), anebo přehled dané problematiky, různých modelů a přístupů uvádí (Gendreau, Laporte et al. 1996).

2.5.12 Ostatní typy dopravní úlohy

Různých modifikací dopravních úloh je téměř neomezené množství, můžeme jmenovat například dopravní úloha se satelitní navigací. dále sem patří různé modifikace předchozích úloh.

3 Metody řešení dopravních úloh

Jelikož jak již bylo v úvodu zmíněno, je dopravní úloha těžko řešitelná a pro řešení ani těch nejjednodušších úloh neexistují konkrétní postupy. Přímým přístupem, to jest spočítáním všech řešení a vybráním nejlepšího z nich lze řešit jen jednoduché dopravní úlohy.

Příkladem může být metoda K-stromů použitá Fisherem (Fisher 1994) pro řešení dopravní úlohy malého rozsahu.

Další skupinou algoritmů pro řešení dopravních úloh jsou různé heuristiky a metaheuristiky. Heuristiky představují postupy omezeného prohledávání stavového prostoru, založené na zkušenostech nebo znalostech experta. Metaheuristiky jsou heuristiky pro úlohy velkého rozsahu do kterých je zahrnuta neurčitost (stochastické heuristiky). Patří sem genetické algoritmy, simulované žíhání, zakázané prohledávání /tabu search/ a další.

3.1 Hejnová inteligence /swarm intelligence/

Hejnová, nebo také rojová inteligence je nové výpočetní paradigma používané k řešení složitých, výpočetně náročných nebo distribuovaných úloh. Výpočetní model k řešení úlohy je založen na fenoménu zvaném emergence. Emergencí se rozumí vznik organizace vyššího řádu, systému vyššího řádu nebo chování vyššího řádu (vznik složitějšího chování) a to procesem samoorganizace velkého počtu entit nižšího řádu. Příkladem jsou hejna ptáků, ryb, včelstva a mravenčí kolonie. Ve všech těchto případech společenstva organismů umí řešit úlohy, které jsou na úrovni jednotlivců neřešitelné.

3.1.1 Vlastnosti hejna

- Je založeno na chování /behaviour based/
- Je adaptivní – adaptuje se na prostředí a jeho změny
- Je situováno do prostředí (svázáno s prostředím)
- Je reaktivní (Na signály z prostředí a na změny prostředí reaguje rychle)
- Je autonomní, samostatné
- Chování hejna přesahuje chování jedince, schopnost řešit úlohy na úrovni hejna které nejsou řešitelné na úrovni jedince
- Pravidla interakce mezi jedinci v hejnu jsou jednoduchá
- Interakce jedinec – jedinec jsou mnohočetné
- Jedinci jsou plně autonomní, nicméně jejich vybavení je minimalistické. Informace dostupné jedinci jsou omezené a mají lokální povahu (jsou dostupné jen informace o bezprostředním okolí)
- Hejno má plně distribuované řízení. Není žádný centrální řídicí mechanismus hejna, žádný trvale vůdčí jedinec.
- Hejno je schopno měnit prostředí.
- Komunikace přímá nebo skrz změny prostředí (stigmergie)
- Je škálovatelné. Hejno může tvořit několik desítek, ale také několik tisíc jedinců
- Je robustní. V důsledku nadbytečnosti jedinců a jejich minimalistického vybavení.
- Globální struktura která je součinností jedinců vytvořena vykonává nějakou funkci, nejčastěji umožňuje skupině zúčastněných jedinců vyřešit nějaký úkol. Vytvořená struktura je flexibilní, (může se adaptovat na změny), robustní (nic se nestane pokud při jejím vytváření několik jedinců vypadne případně pokud některý jedinec udělá chybu)

3.1.2 Sociální chování společenského hmyzu

Interakce mezi organismy mohou být na různé úrovni složitosti. Mezi nejjednodušší typy chování patří chování hejn ryb a ptáků. Přestože hejna ryb a ptáků mohou utvářet zajímavé dynamické struktury, nejvíce studováno je chování sociálního hmyzu, jakým jsou včely, vosy, mravenci a termiti. Včely, vosy a mravenci patří do společné skupiny blanokřídlých (Hymenoptera). Termiti tvoří samostatnou skupinu (Isoptera).

Jakékoliv sociální chování mimo sexuálního chování je klasifikováno jako presociální.

U sociálního hmyzu jedinci, starající se o své potomstvo spolu kooperují. Prakticky sterilní jedinci, ať už z fyziologických nebo sociálních důvodů, pracují ve prospěch jiných jedinců

kteří se rozmnožují. Společně žijí různé generace a přezpívají ve prospěch společenstva tak že potomstvo pomáhá rodičům. Pokud tyto znaky jsou přítomny, pak se toto chování nazývá eusociální.

Další charakteristické vlastnosti sociálního hmyzu jsou následující:

- Přítomnost specializovaných jedinců (kasty) kteří se liší chováním a někdy i morfologií
- Hnízda jsou vysoce strukturovaná
- Vyvinutý systém vzájemné komunikace
- Komunikace také pomocí prostředí /stigmergie/ zanechávání feromonů⁹.
- Kolonie často prochází vývojem, začíná reprodukčním párem a končí vysoce diferencovaným společenstvem.
- Sociální hmyz je dominantní skupina v mnoha ekosystémech

3.1.3 Příčiny vzniku sociálního chování

Eusociální chování se vyvíjelo u různých skupin blanokřídlého hmyzu nezávisle na sobě. Jeho základem je altruistické chování jedinců. Odpověď na otázku o vzniku sociálních struktur a altruistického chování u sociálního hmyzu nám dává genetika (Pavelka and Smetana 2000; Flegr 2005)¹⁰

„Většina organismů je diploidních. To znamená, že mají dvě stejné sady chromozomů, a pohlaví se odlišují v jediném, pohlavím, chromozomu. Například u savců mají samci jeden chromozom X a jeden Y, samice mají oba chromozomy X (značíme XX) U ptáků a motýlů je to obráceně, samci mají identické a samice neidentické pohlavní chromozomy. Nicméně všechny diploidní organismy mají všechny ostatní chromozomy u obou pohlaví ve dvou identických sadách (2n). U blanokřídlých je tomu jinak. Samice mají dvě sady chromozomů (2n), samci pouze jednu (1n). To znamená že samci se líhnou z neoplozených vajíček, a samice (matky a dělnice) z vajíček oplozených. Takovým organismům říkáme haplodiploidní.

Situace, kdy se samci líhnou z vajíček která nebyla oplozena a samice z vajíček oplozených má paradoxní a na první pohled nezřejmé důsledky

Samec haplodiploidního organismu:

- *Nemá syny, pouze vnuky.*
- *Nemá otce, jen dědečka.*
- *Samčí potomstvo rodičovského páru nemá k otci žádný příbuzenský vztah.*

Samice haplodiploidního organismu:

- *Má pouze nemanželské syny kteří jsou jen její. (jsou z neoplozených vajíček, "jen její")*
- *Sestry jsou si příbuznější než bratři (mají více společných genů - 3/4 - viz Tab. 2 a také jsou si příbuznější než matka s dcerou.*

⁹ Feromony jsou látky které hmyz používá ke vzájemné komunikaci. Mravenci a některé další skupiny sociálního hmyzu si značkují místa výskytu potravy a cesty které k potravě vedou, ale také shromaždiště a místa v okolí vchodu do hnízda, aby hnízdo mohlo být rychleji nalezeno.

¹⁰ Koncept vychází z teorie sobeckého genu Richarda Dawkinse. Nejedná se tedy o původní citace, ve zmíněných pracích je myšlenka jen vysvětlena.

Příbuzný	samice		samec	
matka	1/2		1	=1/2
otec	1/2		0	
sestra	3/4	=1/2	1/2	
bratr	1/4		1/2	
dcera	1/2		1	=1/2
syn	1/2		0	

Tab. 2 Pravděpodobnosti výskytu genu u haplodiploidního organismu (Podle (Pavelka and Smetana 2000))

Abychom tyto vztahy lépe pochopili, podíváme se na věc z hlediska genu, a budeme sledovat, jakou máme pravděpodobnost, že kopie našeho genu jsou u našich příbuzných. Pokud jsme samice, tak pravděpodobnost že kterýkoliv náš gen je také v genomu matky je 1/2. Pokud jsme samec, tak máme 100% jistotu že náš gen bude i u matky. Pokud jsme samec, pak pravděpodobnost že kterýkoliv náš gen bude také u naší sestry závisí na tom, zda daný gen pochází od otce či od matky. Pokud gen pochází od otce, pak jeho kopie bude u všech sester. Pokud pochází od matky, pak jeho kopie bude u poloviny sester (druhá polovina sester bude mít kopii genu z protějšího chromozomu). Pravděpodobnost že náš gen pochází od otce je 50% (od matky také 50%). Když tyto hodnoty spočteme, vychází nám 75% pravděpodobnost, že kopie genu nacházejícího se v samičím jedinci je u libovolné jeho sestry. Vidíme, že sestry jsou si mezi sebou více příbuzné než s matkou. Takže je pro ně "výhodné" pomáhat vychovávat své sestry. Altruismus tedy můžeme chápat jako starost o vlastní geny, umístěné v jiném jedinci.“(Pavelka and Smetana 2000)¹¹.

3.2 Chování sociálního hmyzu jako základ pro výpočetní modely

U sociálního hmyzu je nejvíce studováno chování při sběru potravy /foraging behaviour/, chování při stavbě hnízda, vzájemná komunikace a vztah k riziku. Některé chování se stalo inspirací pro výpočetní metody, viz níže, jiné je zatím studováno jen teoreticky

3.2.1 Včely

Výpočetní model je založen na chování včel při sběru potravy /foraging behaviour/. Včelstvo musí efektivně využívat dostupné zdroje potravy a z dostupných zdrojů potravy musí vybírat ty nejvýnosnější. Rozdělování dělnic musí odpovídat výnosnosti jednotlivých zdrojů potravy a musí být dynamické. V minulosti se předpokládalo že existuje vedení ("vedoucí včela") která má tuto dělbu práce na starosti. Nicméně experimenty nic takového nepotvrdily. Bylo zjištěno, že délka trvání tance včelí dělnice po jejím návratu do úlu odpovídá profitabilitě zdroje. (Profitabilitu zdroje hodnotí každá dělnice individuálně.) Každá nezaměstnaná dělnice má pravděpodobnostní práh odpovědi na tanec. Proto déle trvající tanec má vyšší pravděpodobnost naverbování nezaměstnané dělnice. Nicméně u nevydatného zdroje je i při krátce trvajícím tanci jistá pravděpodobnost naverbování nových dělnic. Klíčem k úspěšnosti včelí strategie je redundance - existence zaměstnaných

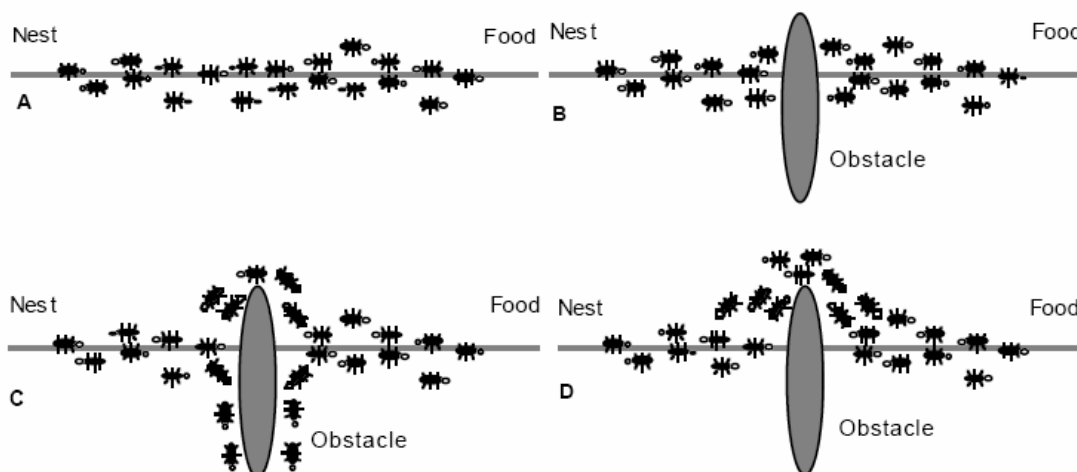
¹¹ Na rozdíl od blanokřídlých u termitů hraje roli inbreeding. Po opakovaném příbuzenském křížení je vzájemná příbuznost potomstva termitů dokonce vyšší než u blanokřídlých. (Lacy 1980).

i nezaměstnaných dělnic. Zaměstnané dělnice po návratu do úlu buď zalarmují další dělnice ke sběru, nebo se stanou nezaměstnanými, a to podle vydatnosti zdroje¹².

3.2.2 Mravenci

Také optimalizace pomocí agentů-mravenců patří do skupiny úloh využívajících inteligenci vzniklou seskupením organismů. /swarn intelligence/. Výpočetní model je založen na schopnosti mravenců nacházet ke zdrojům potravy co nejkratší cestu.

Mravenci-agenti náhodně prohledávají zkoumaný prostor, při čemž berou ohled na "atraktivitu" cesty. Atraktivita cesty je parametr, který simuluje feromonovou stopu, kterou vytváří mravenci při sběru potravy. Při nálezů lokálního řešení je "atraktivita" cesty mravence-agenta zvýšena. Aby nedocházelo ke globálnímu vzrůstu atraktivity nalézáných globálních řešení, je v systému implementován mechanismus "vypařování feromonů", neboli poklesu atraktivity cesty vedoucí k lokálnímu řešení v průběhu času. Tímto mechanismem je zajištěno, že zůstanou pouze řešení s nejkratší cestou, podobně jako je tomu u skutečných mravenců.



Obr. 3 Chování mravenců. A) Mezi potravou a hnízdem není překážka, B) Poté co se objeví překážka se mravenci rozhodují se stejnou pravděpodobností zda jít vpravo nebo vlevo C) více feromonů je ukládáno na kratší cestě k potravě, D) tato cesta je nakonec preferována

Ze všech skupiny sociálního hmyzu je optimalizace pomocí mravenců nejlépe rozpracována a má nejvíce aplikací.

3.2.3 Rozdíly mezi přirozenými a umělými mravenci

Umělí mravenci - mravenci agenti jsou nejen abstrakcí přírodních mravenců, ale také se od mravenců v přírodě v mnoha směrech liší. Je to zejména prohledávání: Zatímco mravenci v přírodě prohledávají dvourozměrný prostor (pohybují se po povrchu) umělí mravenci pracují na grafech. Jejich prostor je multidimenzionální a diskrétní (v každém uzlu se

¹² Délka včelího tance je úměrná vydatnosti zdroje. Čím déle včela tančí, s tím větší pravděpodobností získá dělnice které jí budou ke zdroji následovat.

otevřít konečný celočíselný počet dalších možností). Jako další odlišnosti můžeme jmenovat následující:

- U mravenců v přírodě je množství feromonů ovlivněno kvalitou zdroje potravy, u umělých mravenců je úměrné kvalitě dosaženého řešení.
- Mechanismus vypařování feromonové stopy je jiný u umělých mravenců. Musí se zabránit uvěznění v lokálním řešení.
- Také mechanismus aktualizace feromonové stopy je u mravenců v přírodě jiný
- Mravenci v přírodě mají i další způsoby komunikace než jen stigmergií.
- U umělých mravenců je často prováděno vylepšení pomocí lokální optimalizace nebo lokálního prohledávání.
- Tzv. činnost démona /daemon actions/ tedy globální zásahy do systému nemá u přírodních mravenců analogii.

Vidíme, že výpočetní model založený na mravencích se od skutečných mravenců značně liší stejně jako se umělé neuronové sítě liší od skutečných neuronů.

3.2.4 Postup aplikace ACO metaheuristiky na konkrétní úlohu:

- 1) Reprezentujeme úlohu jako úlohu nalezení cesty v grafu s ohodnocenými hranami.
- 2) Ujasníme si význam feromonové stopy τ_{ij} a jak feromonová stopa ovlivňuje globální řešení. Tento krok je pro vyřešení úlohy klíčový.
- 3) Správně nadefinujeme lokální heuristiky přechodu mravence mezi vrcholy grafu, tj. nadefinujeme lokální omezení spojená z přechodem z vrcholu do vrcholu a způsob konstrukce trasy. Lokální heuristika je zásadní v případě že nemůžeme uplatnit lokální prohledávání.
- 4) Pokud je to možné, implementujeme lokální prohledávání, nejlepší známá řešení jsou kombinací globálního a lokálního prohledávání.
- 5) Vybereme některý z dostupných ACO algoritmů s tím že aplikujeme všechny předchozí body.
- 6) Vyladíme parametry pro daný ACO algoritmus. Začneme s těmi parametry které byly použity k řešení obdobné úlohy, postupně je vyladíme aby byly pro řešení naší úlohy optimální.
- 7) Na základě znalostí získaných v průběhu řešení úlohy revidujeme všechny výše uvedené kroky. Nejdůležitější jsou první čtyři body, pokud tyto nedodržíme pak vyladění parametrů nevhodně zvoleného algoritmu nám příliš nepomůže.

3.3 Přehled ACO algoritmů

3.3.1 Ant System (AS) (Dorigo, Maniezzo et al. 1996)

Byl to první ACO algoritmus. Původně měl tři varianty: /AS-density, AS-kvantity, AS-cycle/, které se lišily podle aktualizace feromonové stopy, poslední z variant kdy byla feromonová stopa aktualizována po dokončení cesty dávala nejlepší výsledky a byla nakonec v algoritmu použita.

Aktualizace feromonové stopy je prováděna následujícím způsobem: Vypařování feromonové stopy je konstantní, úměrná množství feromonů, konstanta úměrnosti je p .

Feromonová stopa je aktualizována podle vztahu: $\tau_{ij} \leftarrow (1 - p)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k$, kde p je

rychlost vypařování, m je počet mravenců a $\Delta\tau_{ij}^k$ je množství feromonu uloženého na hranu (i,j) k -tým mravencem

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if } k \text{ used edge } ij \\ 0 & \text{otherwise} \end{cases}, \text{ kde } Q \text{ je konstanta a } L_k \text{ je délka trasy kterou si } k\text{-tý}$$

mravenec zvolil. Rozhodování mravence v i -tém vrcholu, kdy už má hotovou část řešení,

toto částečné řešení označíme jako s^p , je dáno vztahem:
$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{c_{ij} \in N(s^p)} \tau_{ij}^\alpha \eta_{ij}^\beta} & \text{if } c_{ij} \in N(s^p) \\ 0 & \text{otherwise} \end{cases},$$

kde p_{ij}^k je pravděpodobnost že následující navštívené město bude město j můžeme, $N(s^p)$ je množina přípustných řešení, v hraně (i,l) představuje l dosud nenavštívený vrchol.

Koeficienty α a β představují relativní význam feromonů a heuristické informace η_{ij} . Tato

heuristická informace η_{ij} je dána jednoduchým vztahem $\eta_{ij} = \frac{1}{d_{ij}}$, kde d_{ij} je vzdálenost

mezi vrcholy i a j . Pokud $\alpha=0$, dostaneme klasické prohledávání, pokud $\beta=0$, pak se mravenci řídí pouze feromonovou stopou.

Autoři algoritmus vylepšili zavedením akce démona, tedy dodatečné uložení feromonů nejlepším mravencem.

3.3.2 Ant Colony System (ACS) (Dorigo 1997)

Algoritmus ACS se liší od AS v několika bodech:

ACS má jiné přechodové pravidlo, nazývané náhodné poměrné přechodové pravidlo /pseudorandom proportional rule/. Pokud je k -tý mravenec v i -tém vrcholu, pak mějme parametr $q_0 \in \langle 0;1 \rangle$ a náhodně zvolené číslo $q \in \langle 0;1 \rangle$, pak následující vrchol je zvolen náhodně podle pravidla:

$$q \leq q_0 : p_{ij}^k = \begin{cases} 1 & \text{if } j = \arg \max(\tau_{il}^\alpha, \eta_{il}^\beta), l \in N(s^P) \\ 0 & \text{otherwise} \end{cases} . \text{Tedy, pravidlo si vysvětlíme tak že v}$$

$$q > q_0 : \text{the same as AS}$$

prvním případě, pokud $q \leq q_0$ pak vybereme nejlepší řešení z ohledem na heuristickou informaci a feromonovou stopu, ve shodě s předchozí symbolikou je l nenavštívený vrchol a $N(s^P)$ je množina přípustných řešení. V druhém případě je aplikované řízené prohledávání ve shodě s AS

Rozlišujeme lokální a globální aktualizaci feromonové stopy. Lokální aktualizace feromonové stopy je prováděna každým mravencem na konci každého konstrukčního kroku podle vztahu: $\tau_{ij} = (1 - \varphi)\tau_{ij} + \varphi\tau_0$, kde $\varphi \in (0,1]$ je koeficient vypařování feromonů a τ_0 je počáteční hodnota koncentrace feromonů. Globální aktualizace je prováděna po dokončení trasy jen tím nejlepším mravencem, a to podle vztahu

$$\tau_{ij} \leftarrow \begin{cases} (1-p)\tau_{ij} + p \cdot \Delta\tau_{ij} & \text{if } \{i, j\} \subset \text{best tour} \\ \tau_{ij} & \text{otherwise} \end{cases} . \text{Hlavní předností tohoto algoritmu je}$$

kombinace lokální a globální aktualizace feromonové stopy (Dorigo 1997; Dorigo and Gambardella 1997). Algoritmus ACS je odvozen od algoritmu Ant-Q, který je publikován v práci (Gambardella and Dorigo 1995). ACS je stejně výkonný jako Ant-Q ale je jednodušší.

3.3.3 MAX-MIN Ant System (Stutzle and Hoos 1996; Stutzle and Hoos 2000)

Jedná se o jeden z nejlepších algoritmů. Základní AS algoritmus vylepšuje v následujících oblastech:

Podobně jako u ACS, je celková feromonová stopa aktualizována jen "nejúspěšnějším" mravencem. Na rozdíl od AS a ACS je stanovena maximální a minimální koncentrace

feromonů T_{max} a T_{min} . Aktualizace feromonové stopy: $\tau_{ij} \leftarrow [(1-p)\tau_{ij} + \Delta\tau_{ij}^{best}]_{T_{min}}^{T_{max}}$,

operátor $[x]_a^b$ znamená

$$[x]_a^b = \begin{cases} a & \text{if } x > a \\ b & \text{if } x < b \\ x & \text{otherwise} \end{cases}$$

Minimální a maximální koncentrace feromonů T_{max} a T_{min} sou závislé na konkrétní úloze a získávají se empiricky, nicméně existují i jistá doporučení jak tyto hodnoty určit analyticky. Počáteční koncentrace feromonů je rovna T_{max} .

3.3.4 Graph based Ant System (Gutjahr 2000)

Algoritmus je založen na AS. Přijatelné řešení je reprezentováno jako cesta v orientovaném grafu. Autor se zabývá otázkou zda je nalezeno optimální řešení při dostatku zdrojů, tedy zda a za jakých podmínek úloha konverguje.

3.3.5 Vztah ACO algoritmů a evolučních algoritmů

ACO algoritmy a evoluční algoritmy mají společné prvky ve využití náhodných proměnných a v dosahování řešení prostřednictvím populací relativně nezávislých agentů. Informace o stavu úlohy je v ACO uchovávána vně agentů, zatímco v evolučních algoritmech je tato informace uchovávána v populaci organismů. Jeden typ evolučních algoritmů - EDA /Estimation of Distributions Algorithms/ je velmi podobný ACO algoritmům v tom že uchovává pravděpodobnostní rozložení parametrů úlohy. Jeden z nejlepších algoritmů v této skupině, algoritmus PBIL /Population Based Incremental Learning/ (Baluja 1994), případně (Rastegar and Hariri 2006) je také ACO velmi podobný.

4 Software pro ACO optimalizaci

ACO optimalizační techniky jsou relativně nové, a tomu odpovídá i dostupné programové vybavení. Není mi známo že by existoval dostupný komerční programový balík, nicméně výzkumníci zabývající se ACO optimalizací naprogramovali řadu volně dostupných nástrojů. Hlavní nevýhodou těchto nástrojů je že vyžadují nutnost pochopit vnitřní funkci programu (nežádá i zdrojový kód), jejich ovládání je neintuitivní a složité a vyžadují značný čas ke svému zvládnutí. V následujícím uvádím jejich přehled, který si ani zdaleka neklade nároky na úplnost¹³.

<http://neo.lcc.uma.es/Software/MHTB/index.html>

Toolbox pro metaheuristiky v Matlabu. Neobsahuje ACO, nicméně součástí projektu jsou knihovny v C++ kde je i ACO.

ACOTSP.V1.0

<http://iridia.ulb.ac.be/~mdorigo/ACO/downloads/ACOTSP.V1.0.tar.gz>

Autorem je Thomas Stützle

Implementuje některé ACO algoritmy, použití pro symetrickou úlohu obchodního cestujícího (TSP). Algoritmy: AS, EAS, MMAS, ACS a další. Pod licencí GPL
Vytvořeno v ANSI C pod linuxem.

AntNet

<http://www.omnetpp.org/filemgmt/singlefile.php?lid=48>

Autor: Muddassar Farooq

Založeno na práci (Di Caro and Dorigo 1998)

Programovací jazyk: Omnet++.

HC-MMAS

<http://iridia.ulb.ac.be/~mdorigo/ACO/downloads/hc-mmas-ubqp.tar.gz>

Autor: Christian Blum

Implementace MMAS, hyperkrychle, pro řešení Unconstrained Binary Quadratic
Podle práce (Blum and Dorigo 2004)

Programovací jazyk: C++

¹³ Při rychlém rozvoji v této oblasti se co nevidět objeví i sofistikovanější nástroje.

GUIAnt-Miner

<http://sourceforge.net/projects/guiantminer/>

Programovací jazyk: Java

Je to nástroj pro extrakci klasifikačních pravidel z dat. Program vytvořen podle aktualizované verze algoritmu Ant-Miner (Ant Colony-based Data Miner), který byl publikován v práci (Parpinelli, Lopes et al. 2002)

Program je kompatibilní se systémem VEKA¹⁴

JOpt.NET - Vehicle Routing Software Library

<http://www.soft-articles.com/about/genetic-algorithms.html>

Genetické algoritmy, umělá inteligence. Freeware

Je třeba zmínit i další programy, které s optimalizací pomocí ACO nesouvisí nicméně pro řešení VRP úloh jsou užitečné. Je to například komerční toolbox pro řešení TSP-like úloh v matlabu: <http://www.geatbx.com/>, dále pak Sheduling toolbox

<http://rtime.felk.cvut.cz/scheduling-toolbox/>

Pro řešení kvadratické přiřazovací úlohy /Quadratic Assigment Problem/ může být použita knihovna QLIB <http://www.seas.upenn.edu/qaplib/>

5 Praktická část

V praktické části jsem se zaměřila na ověření funkčnosti programu který napsal RNDr. Miroslav Pavelka v programovém prostředí Matlab. Program umožňuje značnou volnost pro studium chování AS algoritmu a vlivu parametrů na jeho výkonnost. Nevýhodou programu v jeho současné verzi je jeho značná pomalost, což do jisté míry limituje jeho využití, které je prakticky možné jen ke studijním účelům.

Proto jsem se hlavně jsem se zaměřila na studium vlivu parametrů α a β na optimalizaci úlohy obchodního cestujícího. Volba vhodných parametrů α a β je podle literárních údajů {Dorigo, 1996 #1334} klíčová. Na základě výsledků u úlohy obchodního cestujícího (dobrá funkce v širokém rozmezí parametrů) bylo od tohoto testování na dalších typech úloh upuštěno. Místo toho jsem se dále zaměřila na chování algoritmu pro různé typy dopravních úloh při předem daných parametrech.

ACO optimalizace poskytuje nebývale široký prostor pro studium vztahů a závislostí, čemuž odpovídá obrovský počet prací v tomto oboru.

5.1.1 Popis použitého programu

Pro experimenty s mravenci bylo vytvořeno prostředí v programu Matlab. Vlastní programování provedl RNDr. Miroslav Pavelka, já jsem se zaměřila na testování funkčnosti programu a provádění experimentů. Nicméně zde toto programové prostředí stručně popíši. Zdrojové kódy jsou k dispozici.

Prostředí bylo vytvořeno na základě programu optimalizace pomocí ACO. Jedná se o implementaci AS algoritmu na základě práce {Dorigo, 1996 #1334} který byl publikován <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=14543&objectType=FILE>

Tento původní program byl upraven pro různé modifikace dopravní úlohy, viz níže.

¹⁴ Jedná se o dosti těžkopádný nástroj pro data mining implementovaný v jazyce JAVA.

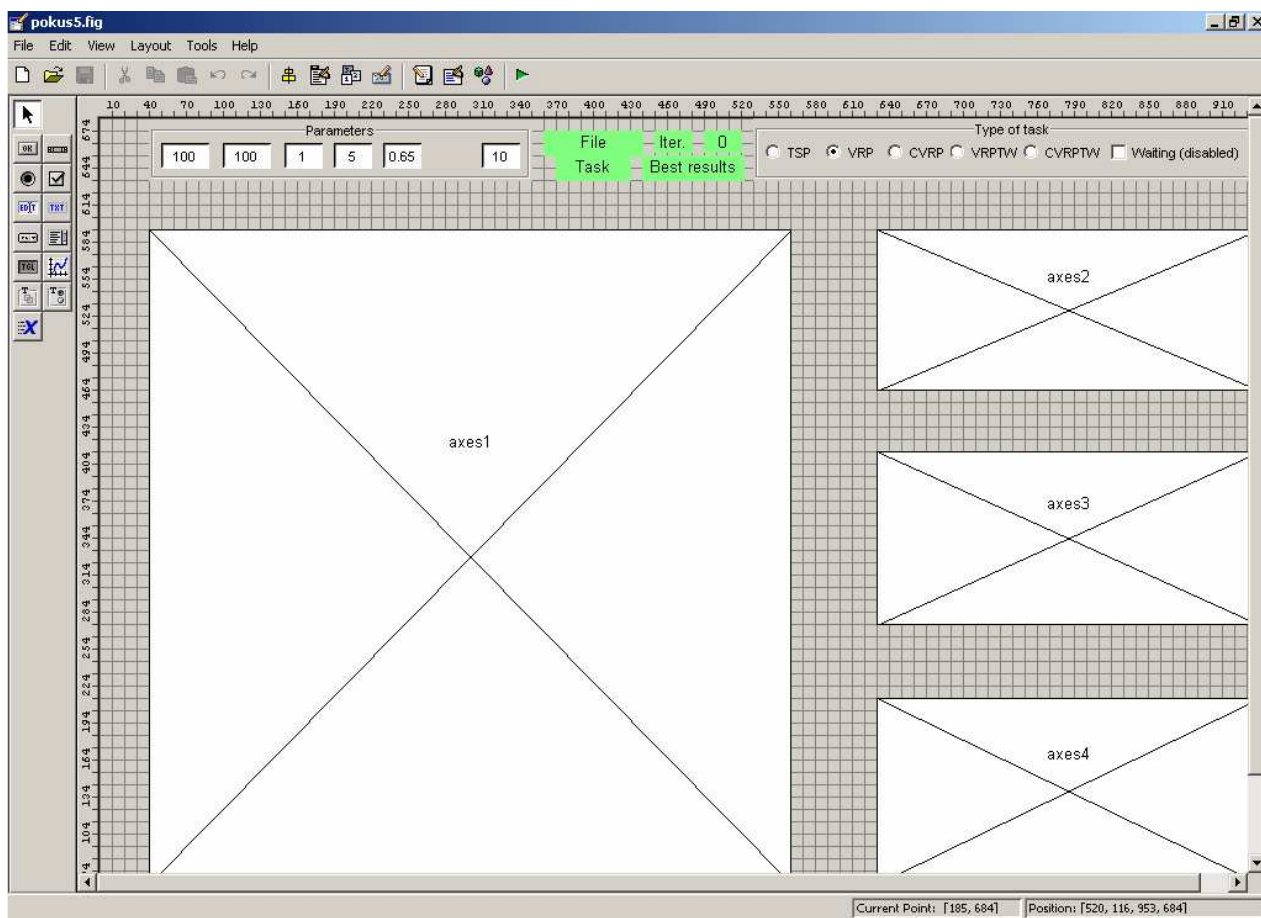
(další informace k původnímu programu viz <http://blogs.mathworks.com/pick/2007/05/04/ant-colony-optimization/> **Program byl napsán pro studium ACO optimalizace s ohledem na funkčnost, nikoliv na uživatelskou přívětivost. K programu není dokumentace, je k dispozici zdrojový kód.** Uvedená implementace ACO optimalizace vyžaduje nastavení některých parametrů. I když je většina parametrů volitelná, z praktických a časových důvodů se experimenty zaměřily jen na parametry α a β (určující vliv koncentrace feromonů a vliv vzdálenosti) Prostředí umožňuje nastavení i dalších parametrů. Přehled všech parametrů v úloze použitých vidíme v tabulce Tab. 3. Některé z těchto parametrů je možné nastavit přímo (Alpha, Beta, Rho, MaxIteration), viz popis k obrázku Obr. 2, ostatní je potřeba měnit ve zdrojovém kódu programu.

NumberOfAnts	Počet Mravenců:	100
Alpha:	Relativní důležitost stopy	0-5
Beta:	Relativní důležitost viditelnosti (vzdálenosti)	0-5
Rho:	Trvanlivost stopy (Hodnotu 1-Rho lze chápat jako rychlost vypařování stopy)	0.65
MaxIterations:	Maximální počet opakování	100
Q:	konstanta globálně určující množství feromonů	10
C:	Počáteční hodnota feromonové stopy	100
Lambda:	Převzatý parametr, pro vlastní výpočet redundantní, používá se pro výpočet ANB (average node branching)	0.15

Tab. 3 Význam jednotlivých parametrů a jejich přednastavené hodnoty

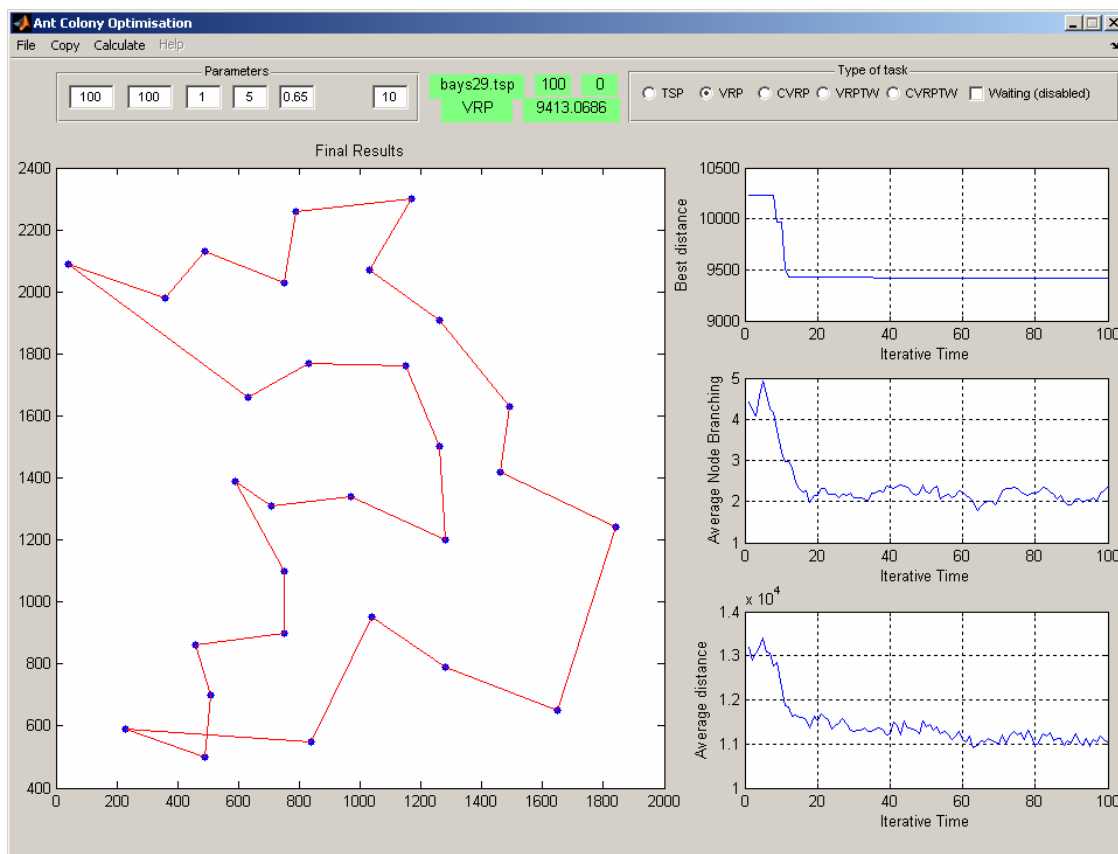
Na obrázku Obr. 2 je uživatelské rozhraní v prostředí guide¹⁵. Běžící program vidíme na Obr. 3.

¹⁵ Prostředí "guide" slouží v Matlabu k pohodlnému vytváření programů obsahujících grafické uživatelské rozhraní.



Obr. 2 Náhled na prostředí použité pro studium ACO optimalizace. V horní části grafu jsou v jednotlivých editačních polích přednastavené následující hodnoty: Počet mravenců (100), počet iterací (100), Alfa (1), Beta (5) Rho (0,65), počet opakování (pro výpočet průměrných hodnot – 10) . V zelených polích jsou hodnoty indikující aktuální stav výpočtu (vybraný sobor, aktuální stav výpočtu, atd).

Můžeme řešit úlohu obchodního cestujícího nebo některou z dopravních úloh. Pokud data která si vybereme neobsahují limitující podmínky (kapacitu vozidel, šířku okna), a my si vybereme některou z dopravních úloh, pak jsou tyto limitující podmínky nastaveny na maximální hodnoty tak aby úloha neměla omezení. Nicméně je použit příslušný algoritmus. Tímto způsobem byla zjišťována kontrola správnosti – úlohy TSP a CVRPTW bez omezení by měly dát stejný výsledek. V praxi tomu tak vždy nebylo, viz kapitola diskuse.



Obr. 3 Program použitý pro studium ACO optimalizace po dokončení činnosti. V levém grafu je zobrazena vybraná nejlepší cesta. Vpravo nahoře je průběh nejlepší celkové dosažené vzdálenosti. Vpravo uprostřed je střední větvení /average node branching¹⁶. Vpravo dole je průměr délky tras všech nasazených mravenců. (Na ose x jsou jednotlivá kola /Iterative Time/. V pravé části nahoře si můžeme zvolit jaký typ úlohy chceme řešit, zda úlohu obchodního cestujícího (TSP) případně některou z dopravních úloh (VRP, CVRP, VRPTW, CVRPTW)).

Úloha obchodního cestujícího a některé typy dopravních úloh jsou na sebe velmi dobře převeditelné. Pro všechny zde popsané typy úloh byl použit jednotný algoritmus. Jedinou nevýhodou tohoto komplikovanějšího provedení bylo zpomalení výpočtu.

Jednotlivé typy úloh (výběr pomocí tlačítek vpravo nahoře):

TSP: Úloha obchodního cestujícího. Standardní nastavení. Trasy všech mravenců mají konstantní předem danou délku (počet vrcholů +1, první a poslední vrchol jsou stejné)

VRP: Klasická dopravní úloha. Od TSP se liší jen tím, že při konstrukci trasy každého mravence je vždy možno vrátit se do depa (pokud depo není aktuálním právě navštíveným vrcholem). Tato situace působí problém, o kterém nevím že by byl v literatuře diskutován: Tím že je depo permanentní součástí volných vrcholů, jejichž počet v průběhu trasy klesá, pravděpodobnost toho že depo bude dalším navštíveným vrcholem se úměrně zvyšuje. To znamená že návštěvy depa jsou před dokončením okruhu častější než na začátku, což samo o sobě nepříspěvá k optimalizaci. Navazující problém je ten, že trasy mezi depem a

¹⁶ Tato hodnota byla převzata z původního programu. Její význam nebyl v literatuře podrobněji dokumentován a ani já jsem se jí nezabývala. Podrobnější vysvětlení viz zdrojový kód programu, proměnná ANB a funkce CalculateANB.

sousedícími vrcholy jsou neúměrně protěžovány a hromadí se na nich feromony. Řešení může být několikero:

- Návštěvy depa v průběhu konstrukce trasy penalizovat, případně při dosažení jistého minimálního počtu zbylých vrcholů je zcela zakázat dokud nebudou ostatní vrcholy navštíveny
- Zmenšit ukládání feromonů na hranách grafu sousedících s depem, případně na tyto hrany feromony vůbec neukládat.
- Nedovolit návštěvu depa dokud to omezující podmínky nevyžadují.

Pokud úloha nemá žádná další omezení, pak pokud jsou vzdálenosti mezi vrcholy euklidovské a trojúhelníková nerovnost je zachována, pak úloha obchodního cestujícího a jednoduchá dopravní úloha bez dalších omezení musí mít stejné optimální řešení. V programu který jsem používala se výše uvedený nedostatek projevil existencí "minitrasy", o 1-2 vrcholech a hlavní trasy kde byla všechny ostatní vrcholy.

CVRP: Kapacitní dopravní úloha. Kromě toho, že je umožněno kdykoliv navštívit depo, je ještě omezení kapacitní. Při konstrukci trasy se bere v úvahu kapacita vozidla a požadavky zákazníků, zákazníci s požadavky přesahujícími momentální kapacitu vozidla jsou z trasy vyřazeni a vozidlo (mravenec) s evrací do depa.

VRPTW: Dopravní úloha s časovými okny. Při konstrukci řešení se berou v úvahu časová omezení, a pro konstrukci trasy (zbylé části trasy) jsou použity jen vrcholy které vyhovují. Pokud je uvažována doba čekání (toto v programu v současné době není implementováno) je možné čekání převést na vzdálenost a vrcholy s čekáním tak penalizovat. Solomonovy úlohy které byly použity jsou konstruovány tak že všechny vrcholy jsou dosažitelné bez nutnosti čekání. V těchto úlohách jsou také kromě kapacitního omezení (viz níže) dány časy obslužení jednotlivých zákazníků.

CVRPTW: Kapacitní dopravní úloha s časovými okny. Omezující podmínky jsou implementovány při konstrukci trasy stejně jako v předchozích případech.

Je třeba říci, že zavedením omezujících podmínek vzrůstá složitost úlohy která není stávající implementací ACO mechanismu podchycena. Jedná se zejména o následující:

- Zahnutí rozhodování z hlediska kapacity vozidla. Toto by bylo pravděpodobně řešeno jako parametr na lokální úrovni (při konstrukci trasy)
- Zahnutí rozhodování z hlediska šířky okna a časové návaznosti (například preference zákazníků jejichž okna jsou úzká ale ještě v dosahu) Toto by bylo též nejsnáze představitelné jako lokální optimalizace.
- Optimalizace rekonstrukce celé trasy. Na tuto optimalizaci byl šlo využít opět ACO. Jde o to, že u dopravních úloha s několikero návštěvou depa každý mravenec musí zkonstruovat celou trasu, to jest obejít všechny vrcholy při čemž jeden vrchol (depo) navštíví několikrát. Takovou trasu je možné rozdělit na podtrasy depo-vrcholy-depo. a tyto podtrasy které byly jednotlivými mravenci vytvořeny je možné vzájemně kombinovat při zřejmých omezujících podmínkách (každý vrchol kromě depa se ve výsledné trase musí vyskytovat právě jednou) A dostáváme známou kombinačně optimalizační úlohu nazývanou úloha batohu /knapsack problem/, na kterou jsou ACO algoritmy též s úspěchem používány.

Jak je s výše uvedeného popisu patrné, jednotlivé úlohy jsou na sebe velmi dobře převoditelné a k jejich řešení lze použít stejného algoritmu.

5.1.2 Použití programu

Jak již bylo dříve uvedeno, program byl napsán pro studijní účely a proto jeho funkce nejsou dokumentovány. *Postup:*

- Po spuštění programu načteme soubor – menu File -> Open Máme na výběr dopravní úlohy – soubory s koncovkou *.vrp, případně úlohy obchodního cestujícího – soubory s koncovkou *.tsp.
- Výpočet provedeme pomocí menu Calculate -> Calculate task
- Výpočet můžeme přerušit pomocí menu Calculate -> Stop current calculation
- Výpočet můžeme obnovit pomocí menu Calculate -> Continue current calculation (objeví se nahoře místo původního Calculate task).
- Po ukončení výpočtu můžeme kopírovat jednotlivé grafy do schránky (v menu Edit zvolíme příslušné submenu) a také můžeme do schránky zkopírovat data
- Zkopírované grafy můžeme vložit do aplikace MSWord, data nejlépe vložíme do aplikace MSExcel.

V menu jsou i další položky které zde neuvádím, pro jejich využití je zapotřebí menší úprava zdrojového kódu programu.

5.1.3 Data

Pro studium dopravní úlohy je dobré mít standardizované datasety. Pro svou práci jsem použila pro úlohu obchodního cestujícího data z <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html>, dále jsem pro dopravní úlohy s časovými okny použila klasické Solomonovy úlohy (viz <http://web.cba.neu.edu/~msolomon/problems.htm>). Nutno říci, že těchto úloh je široká škála a jejich testování by se stávajícím programovým vybavením zabralo mnoho hodin výpočetního času. Proto jsem si vybírala vždy jen jednotlivé úlohy, označené úlohy je vždy na příslušném místě uvedeno. Ještě je třeba se zmínit o dalších datech používaných pro dopravní úlohu, např. benchmarky a nejlepší řešení jsou na <http://www.top.sintef.no/vrp/benchmarks.html>, případně <http://www.sintef.no/static/am/opti/projects/top/vrp/benchmarks.html>. Rizzoli {Rizzoli, 2004 #2228} uvádí že v posledních letech (od roku 2000) bylo dosaženo jen minimální zlepšení v klasických benchmarkích u dopravních úloh.

5.2 Výsledky

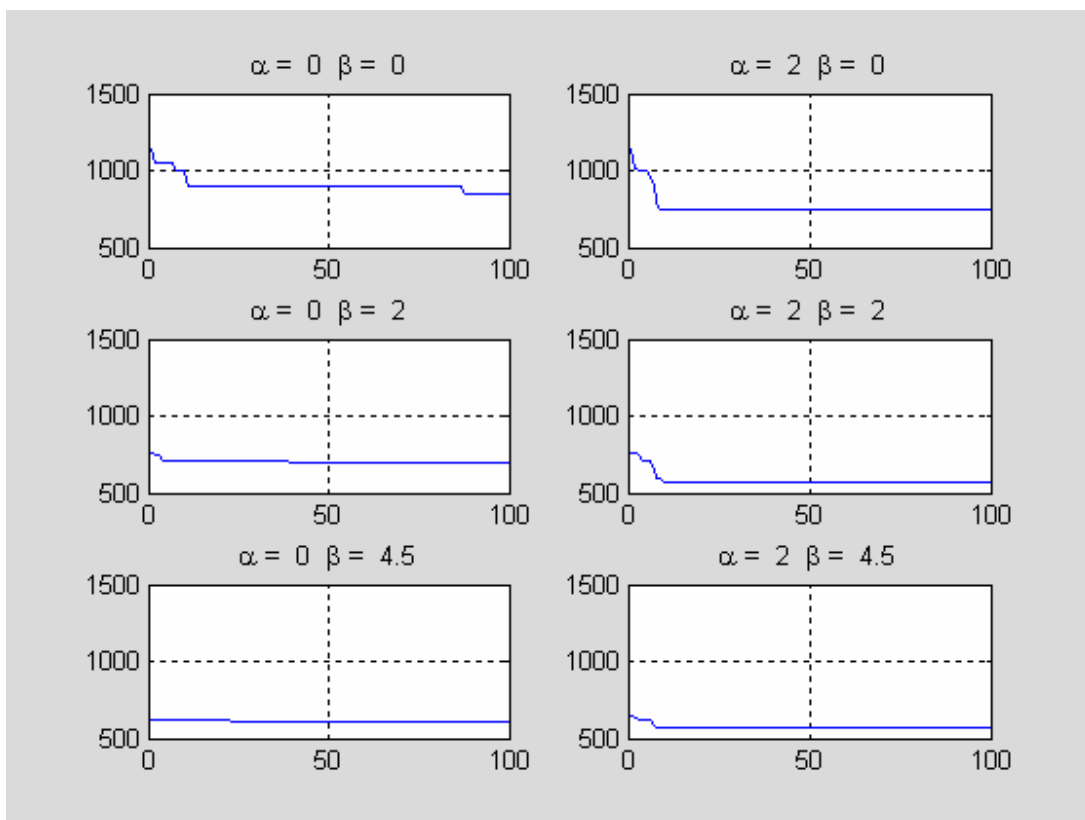
Hotový program umožňuje provést řadu pokusů a pozorování. Lze volit různé kombinace parametrů a s použitím téhož algoritmu lze řešit různé typy úloh. Možností studia chování kolonie umělých mravenců v průběhu optimalizace je nepřeberné. Z tohoto důvodu, aby rozsah práce nepřesáhl rozumnou míru a aby nebyla ztracena přehlednost, jsem se soustředila jen studium vlivu několika faktorů. Druhým důvodem tohoto omezení byl časový požadavek. Optimalizace pomocí kolonie mravenců vyžaduje takový výpočetní výkon, který nebyl k dispozici. Navíc programy napsané v Matlabu jsou mnohonásobně pomalejší než programy napsané v jiných programovacích jazycích. Prostředí je tak možno použít pouze ke studijním účelům, což jsem i učinila.

Vzhledem k charakteru výpočtu a výsledků jsem dala přednost grafickému znázornění a grafickému výstupu všude kde to bylo možné. Domnívám se že tento přístup zvýší přehlednost práce.

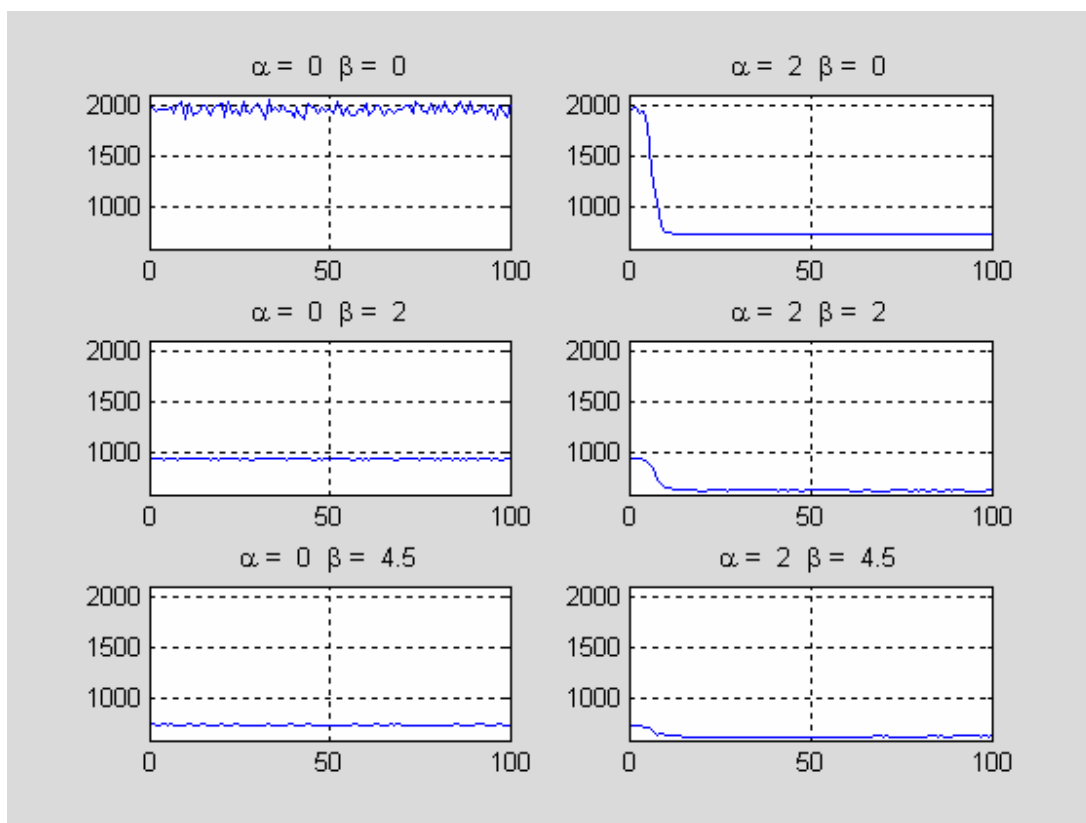
5.2.1 Vliv parametrů α a β na výkonnost úlohy

Prvním z pokusů bylo stanovení vlivu parametrů α a β na výkonnost algoritmu. K tomuto účelu byly kombinace testovány kombinace obou parametrů v rozmezí 0-5. Globální nejlepší dosažené vzdálenosti vidíme na Obr. 4. Na ose x je počet cyklů, na ose y je nejlepší vzdálenost (=vzdálenost dosažená nejlepším mravencem). Všimněme si důležitosti parametru β . Pokud je nulový, pak ani nejlepší řešení není optimální. Tedy heuristická informace nesmí chybět.

Lepší přehled o situaci - o vlivu parametrů α a β nám může dát nikoliv nejlepší vzdálenost, ale průměrná vzdálenost, tedy průměr tras všech mravenců. Situaci vidíme na Obr. 5. Zde vidíme důležitost obou parametrů. Pokud je parametr α roven 0, pak nevidíme vývoj. Tady se jasně ukazuje důležitost feromonové stopy.

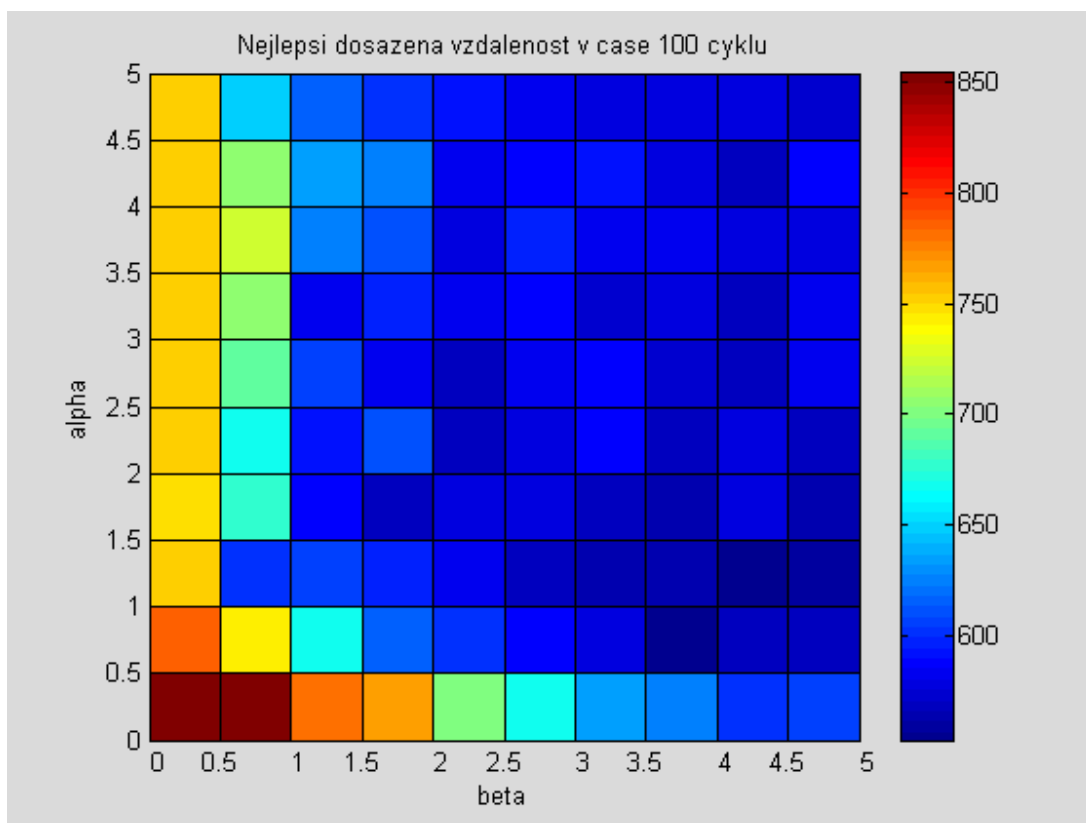


Obr. 4 Nejlepší globálně dosažené vzdálenosti v závislosti na parametrech α a β . Vidíme že je důležité aby oba parametry byly nenulové

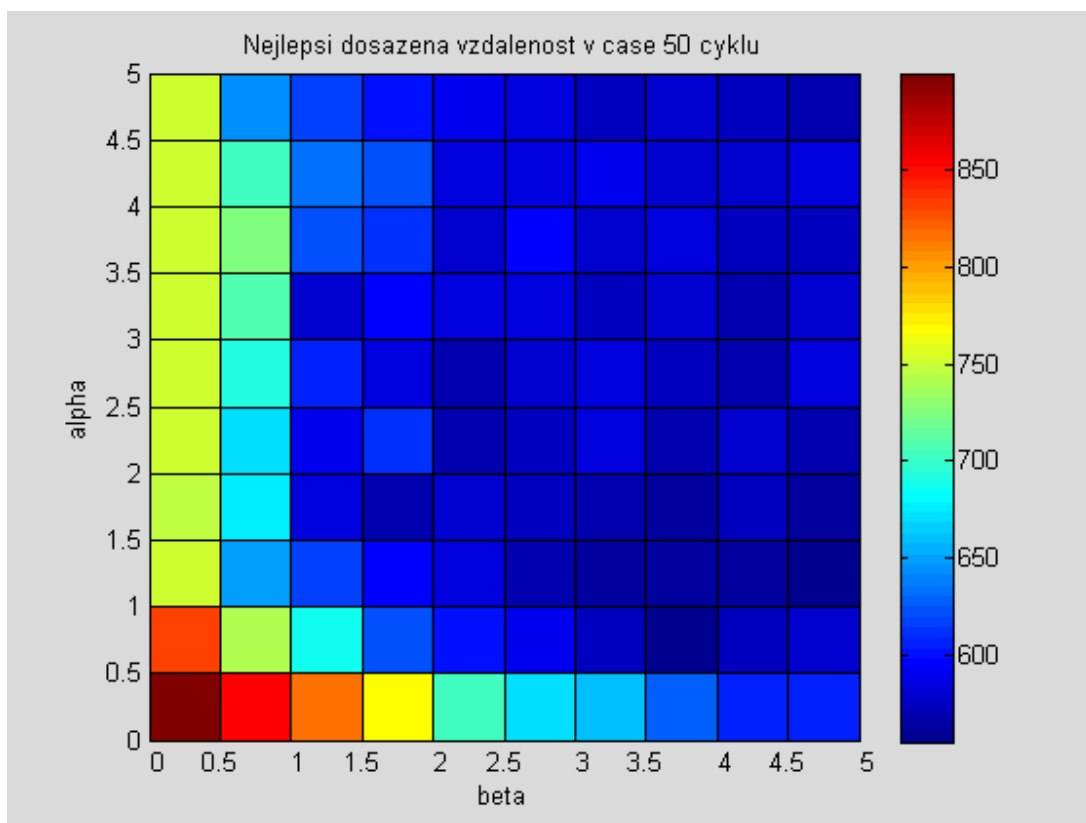


Obr. 5 Průměrná vzdálenost v závislosti na parametrech α a β . Vidíme nejen to, jak je důležité aby oba parametry byly nenulové, ale i význam parametru α . Pokud je tento parametr nulový, pak není možná komunikace mezi mravenci prostřednictvím feromonů. Tedy v systému neprobíhá vývoj.

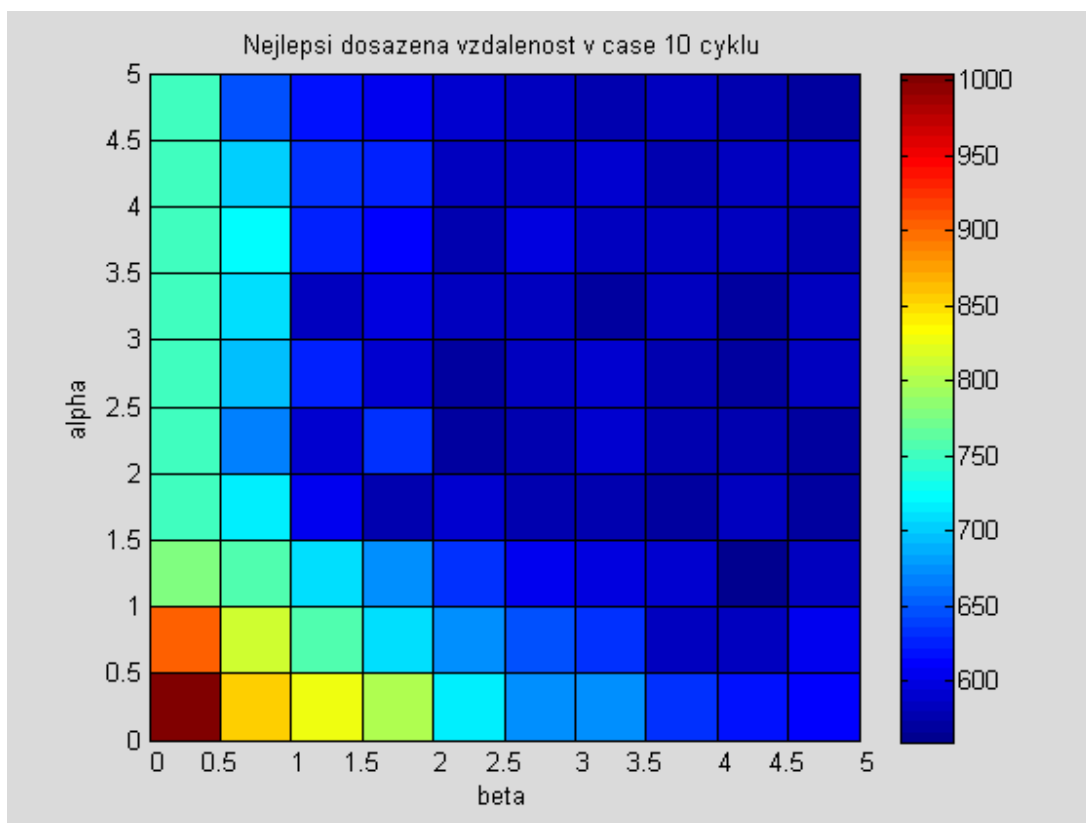
Dále se můžeme podívat podrobněji na situaci jaká nastala po 100, 50, a 10 cyklech. Situaci vidíme na Obr. 6, Obr. 7, Obr. 8. Všimněme si jasně modrého políčka (druhé zleva, třetí od spodu) na Obr. 6. Toto políčko má jasně modrou barvu, což ukazuje nízkou dosaženou globální vzdálenost při relativně nízkých parametrech α a β (0,5 a 1). Provedla jsem několik pokusů, výsledky byly vždy stejné. V této oblasti prostoru parametrů byly výsledky lepší než v bezprostředním okolí. Příčiny tohoto jevu si neumím vysvětlit. Z dalších pokusů vyplynulo, že tento jev možná souvisí s mechanismem generování náhodných čísel který je v Matlabu implementován.



Obr. 6 Úloha obchodního cestujícího, nejlepší dosažená vzdálenost po 100 cyklech v závislosti na parametrech.



Obr. 7 Úloha obchodního cestujícího, nejlepší dosažená vzdálenost po 50 cyklech v závislosti na parametrech



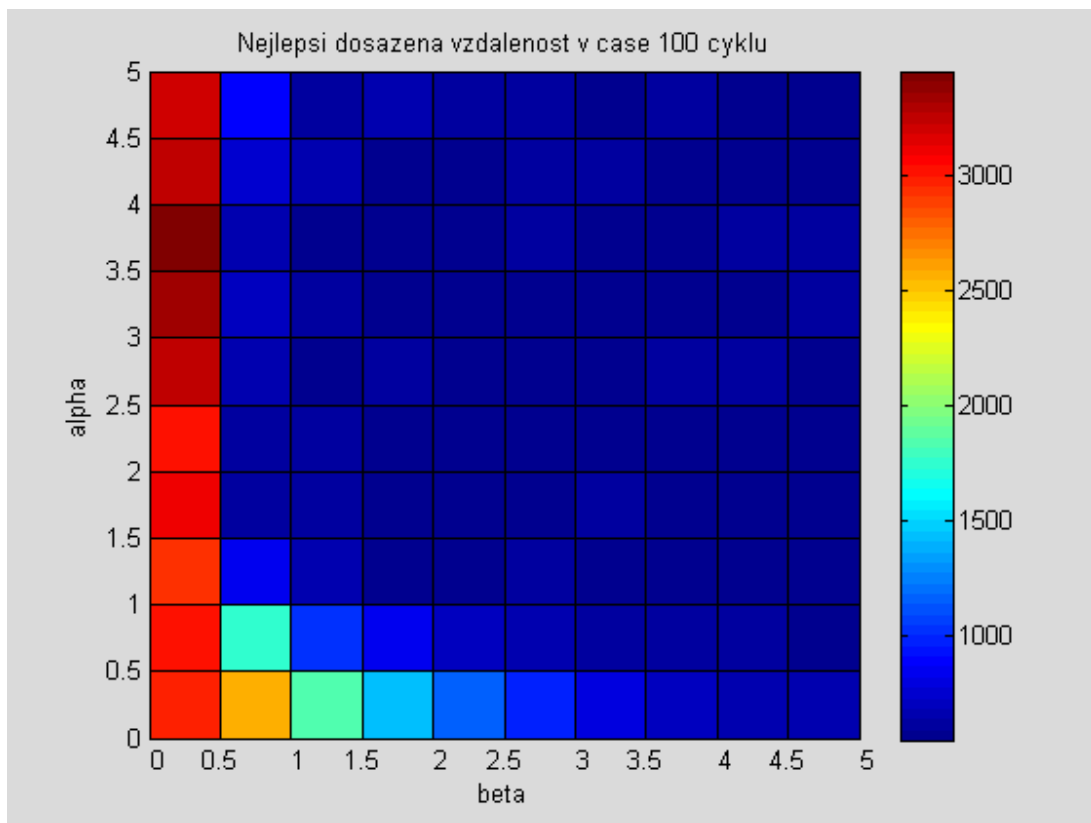
Obr. 8 Úloha obchodního cestujícího, nejlepší dosažená vzdálenost po 10 cyklech.

5.2.2 Vliv parametrů α a β - vyhodnocení.

Z výše uvedených obrázků je vidět, že pokud jsou parametry α a β nenulové a řekněme vyšší než 1, pak jsou výsledky v širokém rozsahu vyhovující (Velká modrá plocha vpravo nahoře). Tyto výsledky jsou v rozporu s publikovanými výsledky (Dorigo, Maniezzo et al. 1996), který uvádí úzké rozmezí parametrů kdy je řešení úlohy optimální. Podobné výsledky byly získány i u klasické dopravní úlohy. Parametry α a β jsou použitelné v širokém rozmezí hodnot. Lokální optimum pro $\alpha = 1$ a $\beta = 0,5$ bylo zmíněno výše, vysvětlení tohoto chování můžeme hledat ve způsobu implementace rozhodovacího mechanismu a v možná mechanismu generování náhodných čísel.

5.2.3 Vliv rozhodovacího mechanismu

Přestože použitý algoritmus byl velmi dobře popsán, výsledky kterých jsem dosáhla se lišily od publikovaných (Viz (Dorigo, Maniezzo et al. 1996), graf na str 14.). Jednou z možných příčin může být skutečnost, že ačkoliv algoritmus popsán je, jeho praktická implementace popsána není. Kritickým prvkem může být způsob rozhodování podle parametrů α a β . Toto rozhodování bylo v původní převzaté verzi programu implementováno ve funkci Roulette. Tato funkce byla přepsána, a výsledky byly poněkud odlišné Situaci po 100 cyklech vidíme na obrázku Obr. 9. Na tomto obrázku vidíme nejlepší dosaženou vzdálenost po 100 cyklech, tedy v ustálené situaci. Na dalších obrázcích vidíme stejnou situaci po 50, 10, a 5 cyklech. Vidíme, že i 10 cyklů stačí k tomu aby se vytvořila rozumně krátká cesta. K těmto, stejně jako k minulým pokusům byl použit soubor gr96.tsp.

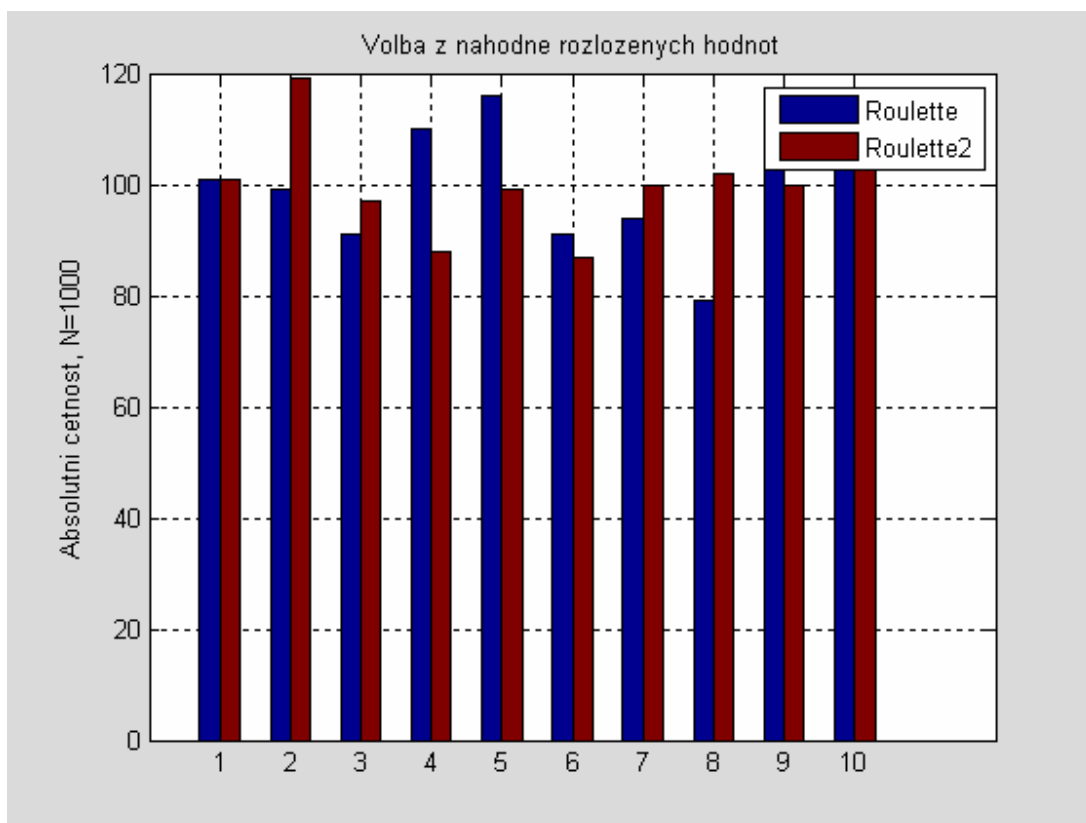


Obr. 9 Nejmenší dosažená vzdálenost po 100 cyklech, funkce roulette přepsána (viz roulette2)

Situace je v mnohém podobná jako v předchozích případech. Vidíme že pro $\beta=0$ jsou výsledky pro jakékoliv alfa špatné. Na druhé straně pokud je $\alpha=0$, pak se vzrůstajícím β se výsledky zlepšují. Jestliže je $\beta=0$, to znamená že chybí heuristická informace. Mravenci neprohledávají a řídí se jen dle feromonů. To znamená, že pokud je vytvořena feromonová cesta, tato je zachovávána všemi mravenci ať je jakkoliv špatná. Chybí lokální prohledávání. Odlišnost je v tom, že rozsah možných parametrů je ještě větší, a že chybí lokální struktura patrná na předchozích obrázcích (tím myslím to, že rozdíly v barvě sousedních políček jsou menší).

<pre>function S = Roulette2(P) I=find(P~=0); q=P(I); if abs((sum(q)-1))>10e-10; error('Must be equal one'); end; q=[0 cumsum(q)]; pom=rand; J=find(q<pom); J=J(end); S=I(J);</pre>	<pre>function Select = Roulette(P,num) m = length(P); flag = (1-sum(P)<=1e-5); Select = zeros(1,num); rand('state',sum(100*clock)); r = rand(1,num); for i=1:num sumP = 0; j = ceil(m*rand); while (sumP<r(i)) && flag sumP = sumP + P(mod(j-1,m)+1); j = j+1; end Select(i) = mod(j-2,m)+1; end</pre>
--	--

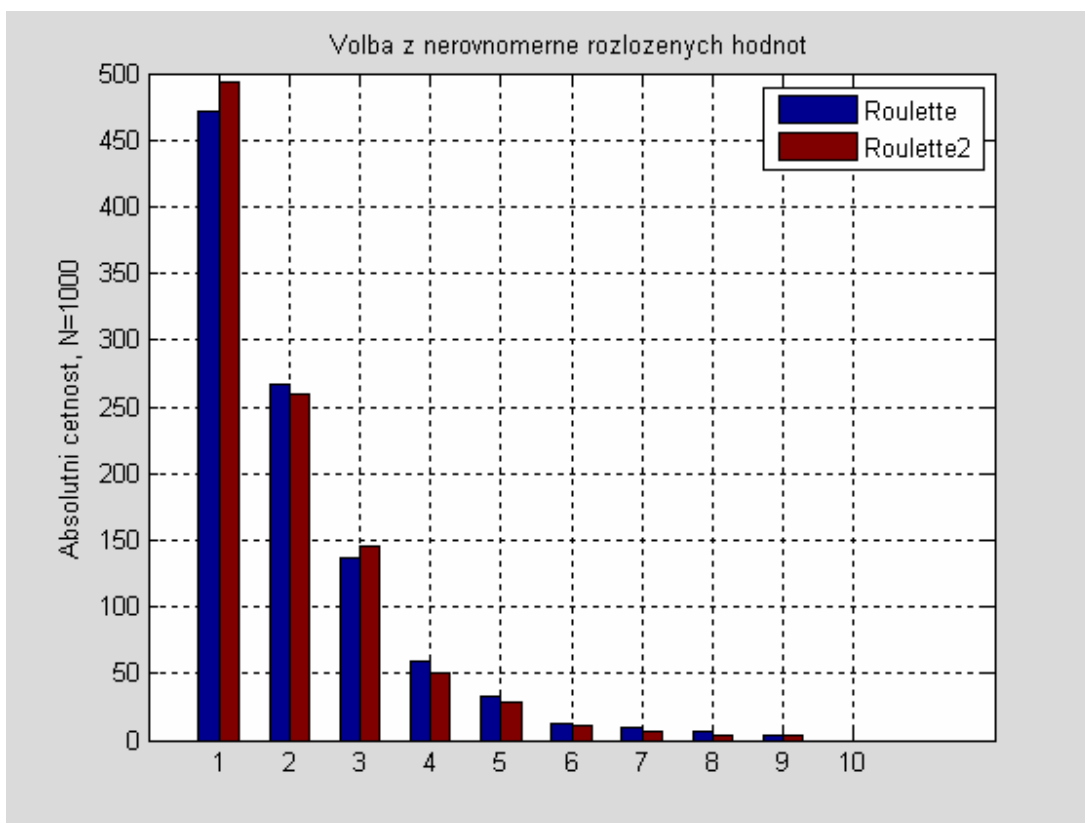
Obr. 10 Zdrojový kód použitých mechanismů. Původní mechanismus je funkce Roulette, modifikovaný mechanismus je Roulette2. Druhý parametr ve funkci Roulette nemá žádný význam, kód byl převzat tak jak je.



Obr. 11 Histogram volby hodnoty pomocí funkcí Roulette a Roulette2 Původní rozložení bylo rovnoměrné, 10 hodnot každá z pravděpodobností 0.1. Na ose x jsou hodnoty pravděpodobností 0.1,0.1....., 0.1 (desetkrát) Jejich součet je roven jedné.

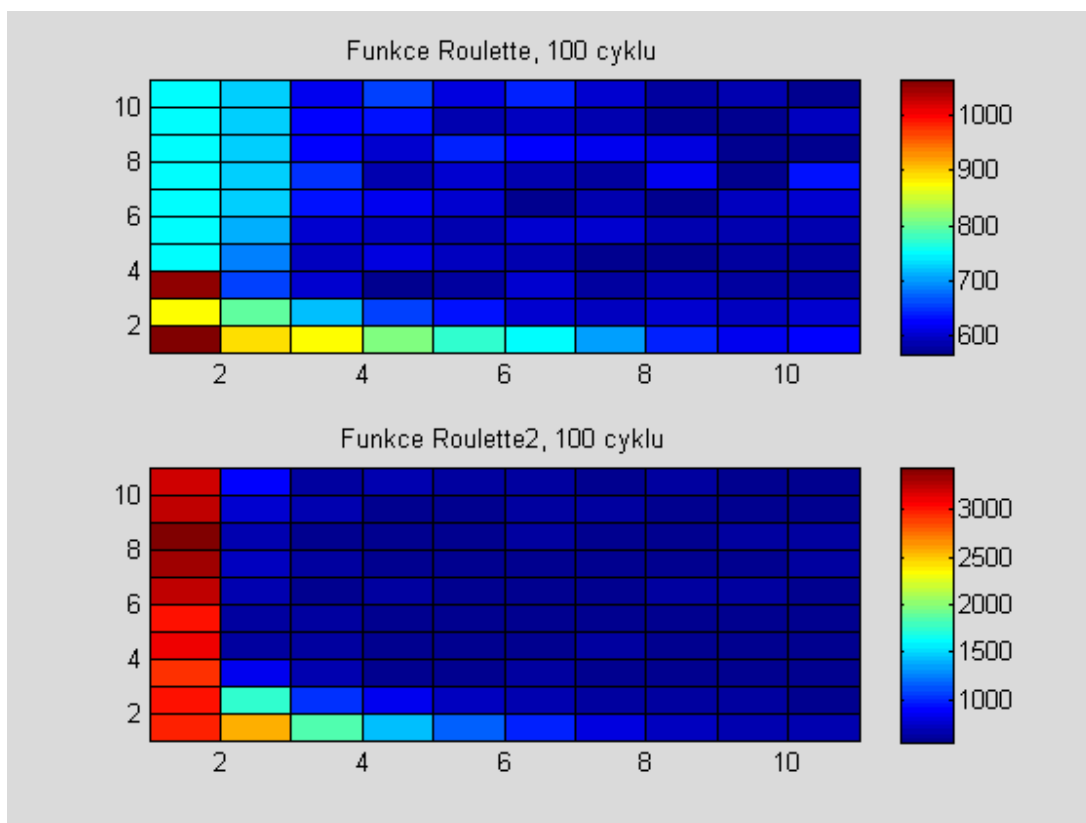
Na Obr. 11 a Obr. 12 vidíme chování funkcí Roulette a Roulette2. V prvním případě (Obr. 11) jsou původní hodnoty rozloženy rovnoměrně. Funkce by měla náhodně vybírat, výsledné rozložení by mělo být také rovnoměrné. To, do jaké míry je rovnoměrné vidíme na histogramu. Vidíme že výsledek vykazuje fluktuace. Zda výsledné fluktuace jsou náhodné (celkem 1000 pokusů, $N=1000$), nebylo statisticky testováno.

V druhém případě Obr. 12 jsou původní hodnoty – pravděpodobnosti rozloženy nenáhodně, (0.5005, 0.2502, 0.1251, 0.0626, 0.0313, 0.0156, 0.0078, .0039, 0.0020, 0.0010), jejich součet je stejně jako v předchozím případě roven jedné. Na histogramu jasně vidíme že nová funkce (Roulette2) původní rozložení pravděpodobnosti odráží lépe. (První tři červené bary jsou blíže očekávaným hodnotám než bary modré).

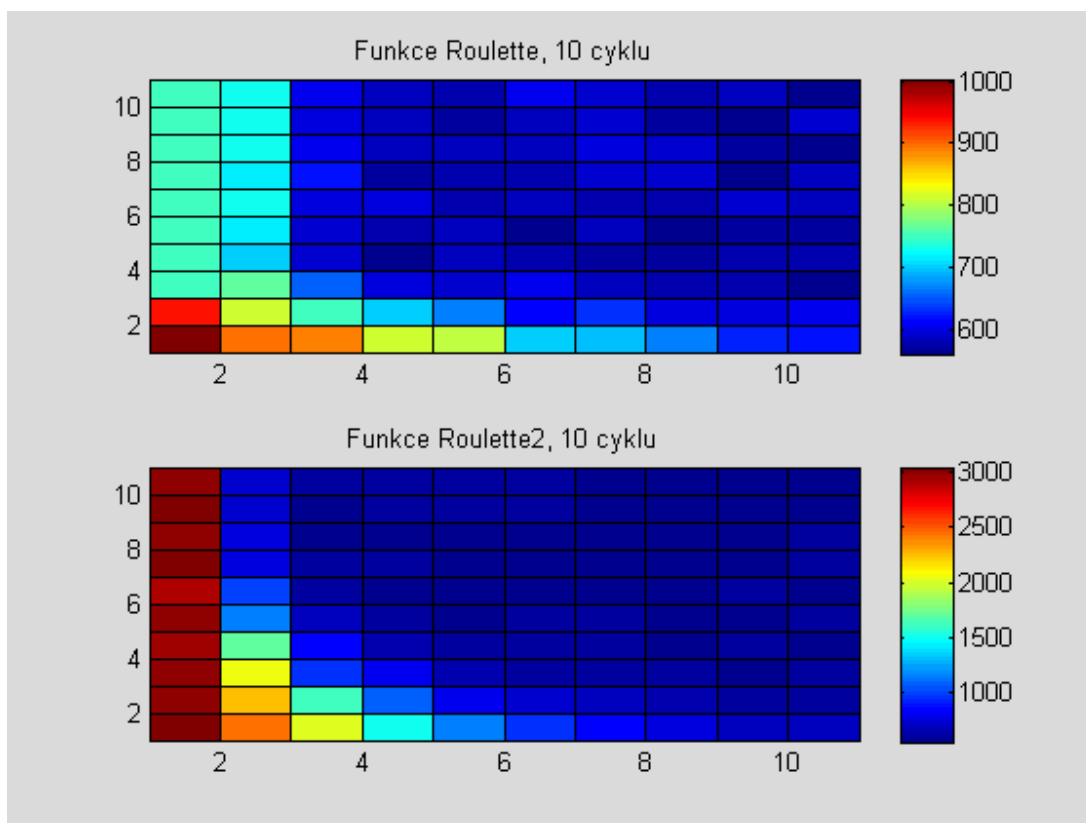


Obr. 12 Volba z nerovnoměrně rozložených hodnot. N ose x jsou hodnoty pravděpodobnosti (0.5005, 0.2502, 0.1251, 0.0626, 0.0313, 0.0156, 0.0078, .0039, 0.0020, 0.0010), jejich součet je stejně jako na Obr. 11 roven jedné.

Nicméně, nemůžeme tvrdit že by funkce Roulette a Roulette2 dávaly výrazně odlišné výsledky. Situaci po 100 cyklech si můžeme porovnat na následujícím obrázku



Obr. 13 Porovnání výsledku po 100 cyklech při použití funkce Roulette a Roulette2. Popis os x a y viz ostatní grafy, hodnoty zde nejsou škálovány.

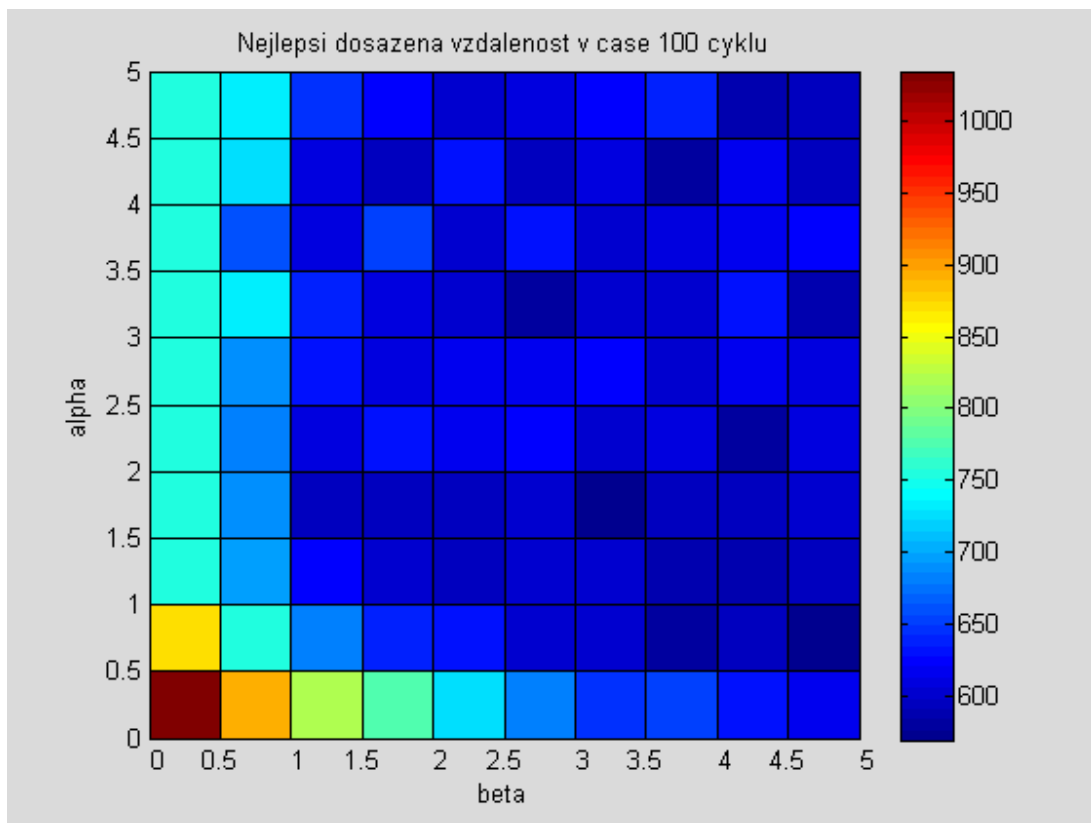


Obr. 14 Funkce Roulette a Roulette2, 10 cyklů.

Podobná situace je i při 10 cyklech. Vidíme, že 10 cyklů je většinou dostačujících. Všimněme si rovnoměrné modré plochy téměř v celém grafu. To znamená že parametry α a β , pokud nejsou nulové, na výsledek optimalizace téměř nemají vliv.

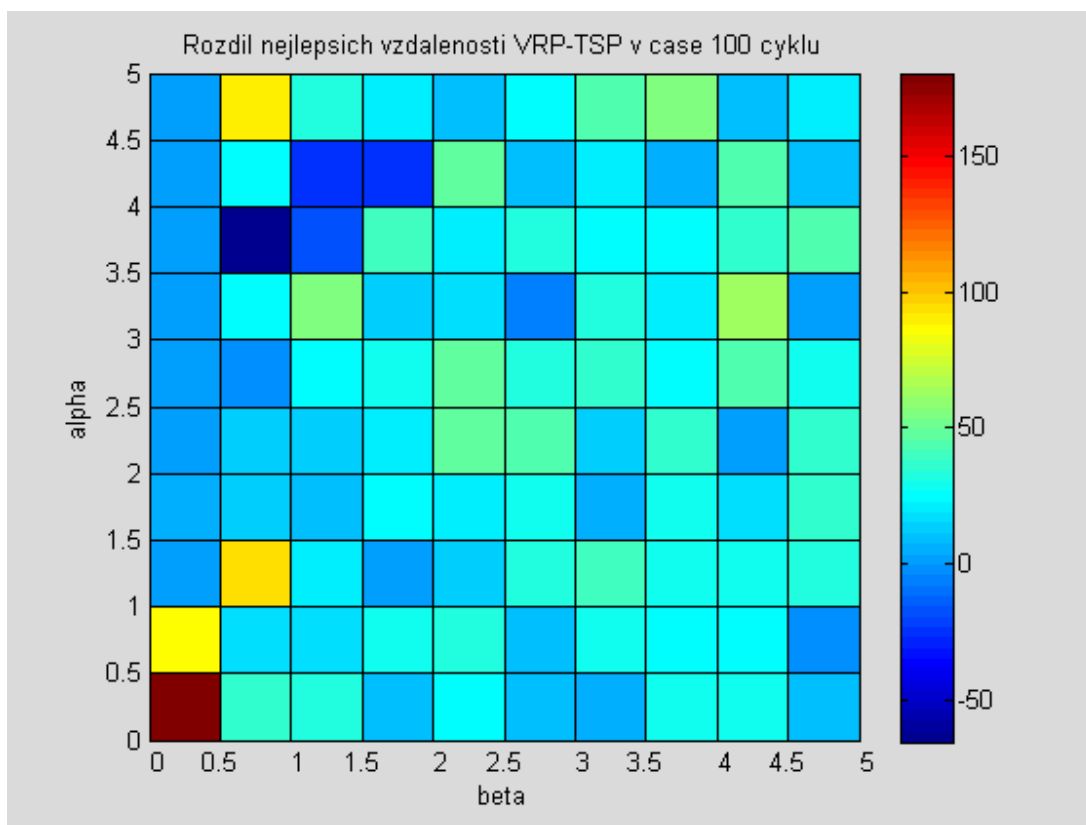
5.2.4 Vztah mezi TSP a VRP z hlediska volby vhodných parametrů.

V předchozích pokusech bylo ukázáno, že suboptimální řešení úlohy obchodního cestujícího je možné dosáhnout v širokém rozsahu parametrů. Situaci po 100 cyklech vidíme na Obr. 15. Tedy vidíme, že i zde, pokud jsou parametry α , β různé od 0, dosáhneme dobré řešení.

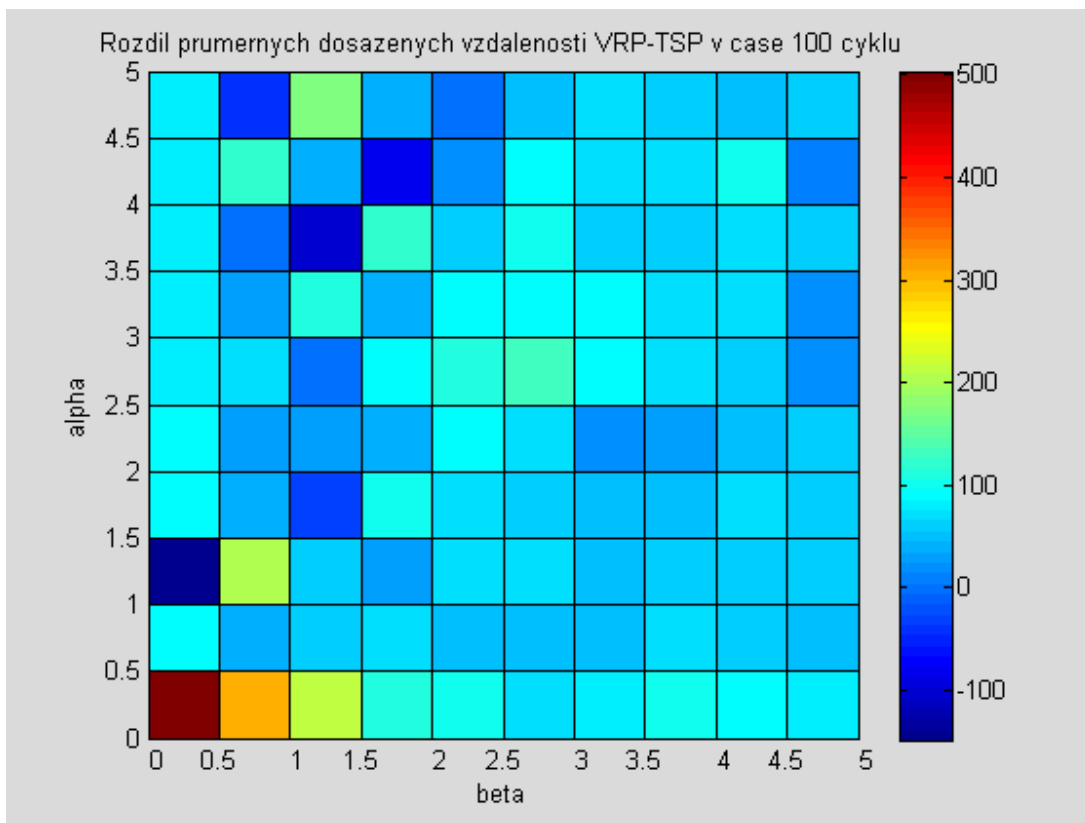


Obr. 15 Dopravní úloha, nejlepší řešení dosažené po 100 cyklech.

Můžeme si položit otázku, jak vypadá vzájemné porovnání úlohy obchodního cestujícího a té samé dopravní úlohy. Teoreticky bychom měli obdržet totožné výsledky. Jelikož dopravní úloha bez dalších omezení se od úlohy obchodního cestujícího liší pouze tím, že depo v dopravní úloze můžeme navštívit vícekrát, zatímco v úloze obchodního cestujícího jen jednou, je jasné, že výsledky dopravní úlohy budou horší, a nejlepší výsledky budou totožné. Podíváme se na situaci po 100 cyklech.



Obr. 16 Rozdíl v nejlepší vzdálenosti mezi dopravní úlohou a tou samou úlohou obchodního cestujícího po 100 cyklech.

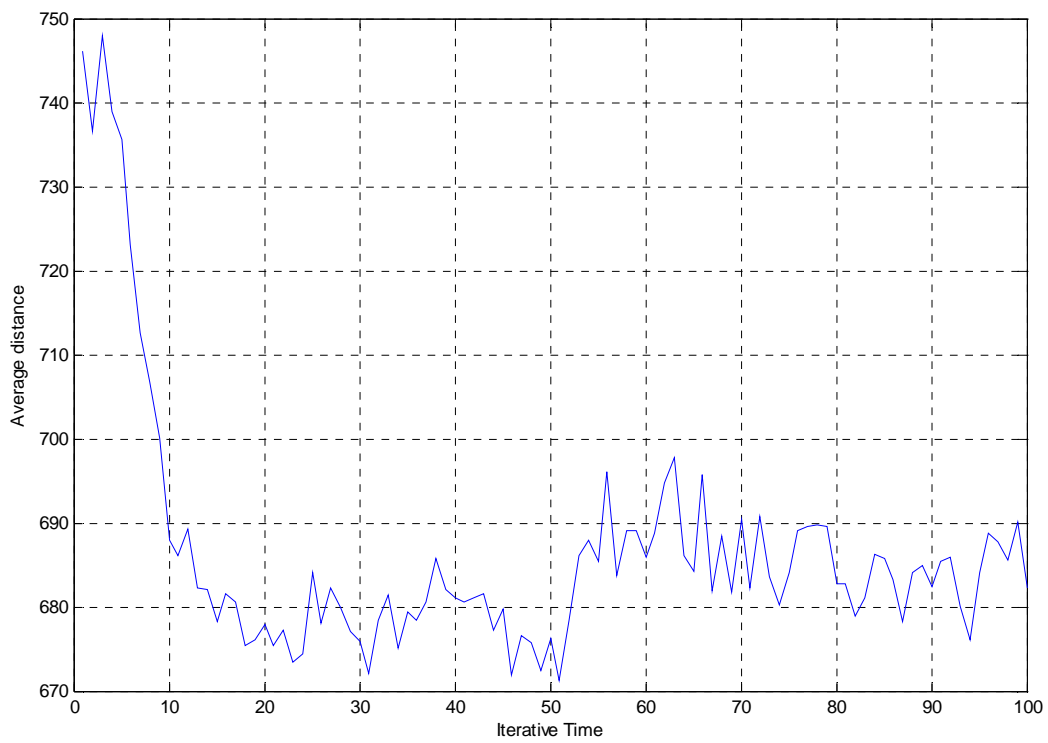


Obr. 17 Rozdíl průměrných dosažených vzdáleností mezi dopravní úlohou a úlohou obchodního cestujícího.

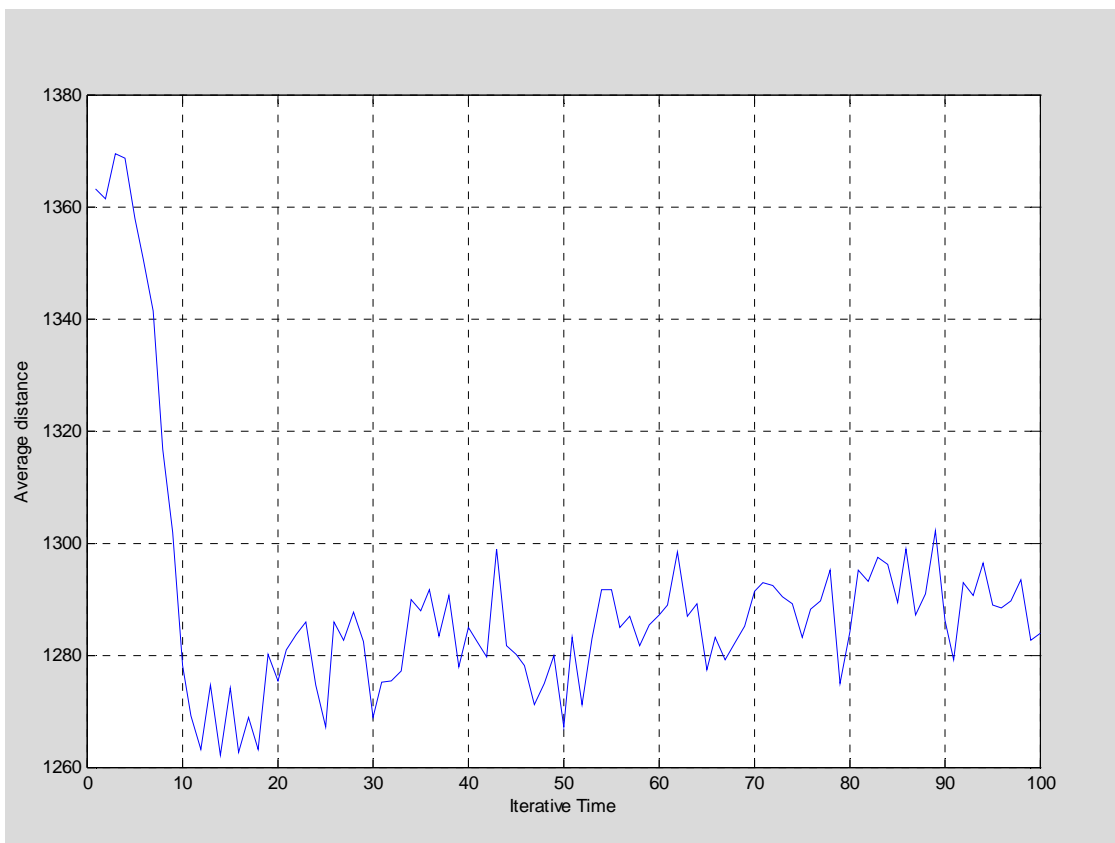
Jak je z Obr. 16 a Obr. 17 patrné, jsou u dopravní úlohy výsledky poněkud horší než u úlohy obchodního cestujícího, přestože by podle teorie měly být stejné. Tento rozdíl je ovšem v dobré shodě s realitou, kdy u dopravní úlohy musí mravenci řešit více možností, a kdy mají implicitně ztíženou situaci že se mohou vracet do depa, kdežto v úloze obchodního cestujícího toto nesmí. To znamená že v dopravní úloze mají horší počáteční podmínky, což se samozřejmě projeví.

5.2.5 Další zajímavá pozorování

Jak již bylo dříve řečeno, program na kterém jsem spolupracovala umožňuje provádět řadu pokusů z ACO optimalizací. Na tomto místě bych chtěla zmínit zajímavou skutečnost, že jsem si všimla postupného zhoršování průměrných výsledků s časem. Tato situace je v literatuře velmi dobře známá a je označována jako "deception", do češtiny přeložitelné snad jako selhávání. (Blum and Dorigo 2004; Blum and Dorigo 2005). Jedná se o analogickou situaci jako "přeučení" u neuronových sítí. Tohoto jevu jsem si všimla při některých kombinacích parametrů, tento jev by jistě zasloužil podrobnější zkoumání.



Obr. 18 Vidíme, jak se průměrná vzdálenost po 60tém cyklu prudce zhoršila...



Obr. 19 CVRP. Krásně vidíme jak po 30tém cyklu se průměrná výkonnost zhoršuje

Dalším zajímavým jevem bylo sledování dynamiky postupného zlepšování nejlepšího dosaženého řešení. V některých případech bylo nejlepší řešení dosaženo již po desíti cyklech, v jiných docházelo k jeho postupnému zlepšování až do konce. Bohužel se nepodařilo vyřešit vizualizaci rozložení feromonů v grafu, čímž by se tento jev stal srozumitelnějším.

6 Diskuse

V praktické části práce jsem provedla řadu pokusů na různých datech. Výsledky které se mi zdály nejzajímavější jsem uvedla v předchozích kapitolách. Věnovala jsem se studiu vlivu parametrů α a β na výkonnost systému. Na rozdíl od výsledků popsanych v literatuře (Dorigo, Maniezzo et al. 1996) jsem zjistila, že systém funguje v širokém rozsahu parametrů. O příčinách rozdílů mezi dosaženými výsledky a výsledky publikovanými lze diskutovat, domnívám se však, že hlavní příčinou je rozdílná implementace.

Vliv rozdílů v implementaci lze dobře dokumentovat na výkonnosti systému při různém způsobu generování výsledku v rozhodovacím mechanismu (funkce Roulette versus funkce Roulette2). Tato funkce by měla dávat výsledky s četností která odpovídá pravděpodobnosti původního rozložení. Je s podivem že odlišné implementace této funkce mají za následek rozdíly ve výkonnosti systému, tyto rozdíly jsou nejvýznamnější na okrajích, t.j. když je jeden z parametrů α nebo β je blízký nule.

Další zajímavou vlastností systému které jsem si všimla a kterou jsem zde i doložila je postupné zhoršování průměrné výkonnosti mravenčí kolonie v některých případech.

Tomuto jevu se říká "deception" , je studován například v pracích (Blum and Dorigo 2004; Blum and Dorigo 2005).

Nakonec ještě je potřeba zmínit implementační záležitosti. Program s kterým jsem experimentovala byl napsán v prostředí Matlab. Vzhledem k charakteru úlohy, která je ve své podstatě paralelní a vhodná k distribuovanému provádění výpočtů by stálo za úvahu použít odpovídajícího toolboxu (Distributed Computing Toolbox) matlabu a výpočet provádět na clusteru několika (desítek) počítačů. Stávající implementace byla příliš pomalá pro extenzivní studium problematiky.

7 Závěr

7.1 Shrnutí výsledků

V teoretické části práce jsem podala podrobný přehled různých typů dopravních úloh a současných algoritmů které se používají pro jejich řešení. V kapitole o výpočetní složitosti jsem zdůvodnila, proč je možné, a že je možné vzájemně na sebe převádět různé typy dopravních úloh. Hlavní část práce tvoří popis technik souvisejících s rojovou inteligencí. Věnovala jsem se biologickým základům vzniku společenského chování u sociálního hmyzu a vztahům mezi genetikou a vznikem sociálních struktur z pohledu teorie sobeckého genu. V bodech jsem uvedla rozdíly mezi přírodními a umělými mravenci a návod jak postupovat chceme –li pomocí agentů – mravenců řešit nějakou úlohu. Podala jsem přehled hlavních algoritmů používaných v ACO optimalizacích, stejně jako soupis některých v současnosti dostupných programů.

V experimentální části práce jsem využila dříve napsaného programu pro testování ACO algoritmů a provedla jsem studii vlivu parametrů vzdálenosti a feromonové stopy na výkonnost ACO algoritmu.

7.2 Zhodnocení splnění stanovených cílů

Domnívám se že v teoretické části práce jsem danou problematiku popsala dostatečně podrobně. Jedná se o problematiku značně rozsáhlou, proto bylo nutno přistoupit k zestručnění tak aby rozsah práce nepřekročil únosnou mez.

Pokud jde o praktickou část práce, zde jsem pocítila limitaci v pomalosti výpočtu. Z tohoto důvodu nebylo možné provádět detailní porovnání výkonnosti algoritmu na různých typech dopravních úloh, přestože jsem měla k dispozici mnoho datových souborů. Na druhé straně se ukázalo že parametry na které jsem se zaměřila, to jest vliv intenzity feromonů a vzdálenosti souseda dávají dobré výsledky v širokém rozsahu hodnot. Proto by detailní porovnávání různých datových souborů zřejmě nepřineslo nic nového.

Po stránce prezentace výsledků jsem dala přednost grafické prezentaci před uváděním konkrétních číselných hodnot. Jsem toho názoru, že hlavní dosažené výsledky mají kvalitativní povahu a uvádění statistik a tabulek by bylo v tomto kontextu jen matoucí a nepřehledné.

7.3 Přínos autora práce k řešené problematice

Přínos práce vidím v jejím širokém zaměření, ve spojení biologického pozadí s praktickými aplikacemi. Práce je ve značné míře multioborová, spojuje výsledky více vědních disciplin. V praktické části práce vidím svůj přínos v ověření funkčnosti publikovaného algoritmu, při čemž mé výsledky jsou částečně ve shodě s výsledky publikovanými (algoritmus se

chová tak jak očekáváme včetně "nectností" v podobě zhoršování /deception/ průměrného výkonu mravenců v čase), částečně se s publikovanými výsledky rozcházejí (algoritmus funguje v širším rozsahu parametrů než bylo publikováno). Zajímavý je též vliv náhodného prvku na celkovou výkonnost, viz porovnání funkce Roulette a Roulette2 Obr. 13 a Obr.14. Zhodnocení využitelnosti dosažených výsledků

Domnívám se že práce může dobře sloužit jako studijní pomůcka pro zájemce o danou problematiku. V praktické části je hlavní limitací konstrukce programu, který by vyžadoval přepsat aby jej bylo možné lépe využít.

7.4 Další náměty pro řešení.

Optimalizace pomocí kolonie představuje moderní technologii do které plynou nemalé finanční prostředky a která se velmi rychle vyvíjí. Problematika může být zkoumána i z mnoha dalších hledisek, jako je například vliv dalších parametrů (počet mravenců, vypařování feromonové stopy). V současné době jsou zkoumány především dynamické varianty dopravní úlohy, pro které tato program který jsem měla k dispozici není vhodný. Studium této problematiky totiž vyžaduje softwarové řešení, které je paralelní, distribuované¹⁷ a multiprocessorové.

¹⁷ Řešení rozdělené na vzájemně nezávislé komponenty jejichž zpracování probíhá na různých počítačích

8 Literatura

- Augerat, P., J. M. Belenguer, et al. (1998). "Separating capacity constraints in the CVRP using tabu search." European Journal of Operational Research **106**(2-3): 546-557.
- Bachem, A., W. Hochstattler, et al. (1996). "The simulated trading heuristic for solving vehicle routing problems." Discrete Applied Mathematics **65**(1-3): 47-72.
- Baluja, S. (1994). Population based incremental learning: a method for integrating genetic search based function optimization and competitive learning. Technical Report, Carnegie Mellon University.
- Beck, J. C., P. Prosser, et al. (2002). On the Reformulation of Vehicle Routing Problems and Scheduling Problems. Abstraction, Reformulation, and Approximation: 282.
- Bertsimas, D. J. (1992). "A Vehicle-Routing Problem with Stochastic Demand." Operations Research **40**(3): 574-586.
- Blum, C. and M. Dorigo (2004). "Deception in ant colony optimization." Ant Colony Optimization and Swarm Intelligence, Proceedings **3172**: 118-129.
- Blum, C. and M. Dorigo (2004). "The hyper-cube framework for ant colony optimization." Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics **34**(2): 1161-1172.
- Blum, C. and M. Dorigo (2005). "Search bias in ant colony optimization: On the role of competition-balanced systems." Ieee Transactions on Evolutionary Computation **9**(2): 159-174.
- Braysy, O., G. Hasle, et al. (2004). "A multi-start local search algorithm for the vehicle routing problem with time windows." European Journal of Operational Research **159**(3): 586-605.
- Cordeau, J. F. (2000). VRP with Time Windows., in . . . p. 157-193. The Vehicle Routing Problem. Philadelphia, USA, SIAM, Society for Industrial and Applied Mathematics.
- Di Caro, G. and M. Dorigo (1998). "AntNet: Distributed Stigmergetic Control for Communications Networks." Journal of Artificial Intelligence Research **9**: 317-365.
- Dorigo, M. (1997). "Ant Colony System: A cooperative learning approach to the traveling salesman problem,." IEEE Transactions on Evolutionary Computation **1**(1): 53-66.
- Dorigo, M. and L. M. Gambardella (1997). "Ant colonies for the travelling salesman problem." Biosystems **43**(2): 73-81.
- Dorigo, M., V. Maniezzo, et al. (1996). "Ant system: Optimization by a colony of cooperating agents." Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics **26**(1): 29-41.
- Dumas, Y., J. Desrosiers, et al. (1991). "The pickup and delivery problem with time windows." European Journal of Operational Research **54**(1): 7-22.
- Edmonds, J. and E. L. Johnson (1973). "Matching, Euler Tours, and the Chinese Postman." Math. Programm. **5**: 88-124.
- Fisher, M. L. (1994). "Optimal Solution of Vehicle Routing Problems Using Minimum K-trees." Operations Research **42**(626-642).
- Flegr, J. (2005). Evolucni biologie. Praha, Academia.

- Gambardella, L. M. and M. Dorigo (1995). Ant-Q: a reinforcement learning approach to the travelling salesman problem. Proceedings of the Twelfth International Conference on Machine Learning, Palo Alto, California, USA, Morgan Kaufmann.
- Gendreau, M. (1999). "Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching.." Transportation Science **33**(4): 381-390.
- Gendreau, M., G. Laporte, et al. (1996). "Stochastic vehicle routing." European Journal of Operational Research **88**(1): 3-12.
- Goetschalckx, M. and C. Jacobsblecha (1989). "The Vehicle-Routing Problem with Backhauls." European Journal of Operational Research **42**(1): 39-51.
- Gutjahr, W. J. (2000). "A Graph-based Ant System and its convergence." Future Generation Computer Systems **16**(8): 873-888.
- Ho, S. C. and D. Haugland (2004). "A tabu search heuristic for the vehicle routing problem with time windows and split deliveries." Computers & Operations Research **31**(12): 1947-1964.
- Ichoua, S., M. Gendreau, et al. (2000). "Diversion issues in real-time vehicle dispatching." Transportation Science **34**(4): 426-438.
- Kautz, W. H. (1958). "Unit distance error checking codes." IRE Transaction on Electronic Computers **7**: 179-180.
- Kilby, P., P. Prosser, et al. (1998). Dynamic VRPs: a study of scenarios.. Technical Report, University of Strathclyde.
- Lacy, R. C. (1980). "The Evolution of Eusociality in Termites: A Haplodiploid Analogy?" The American Naturalist **116**(3): 449-451.
- Lagová, Jablonský (2004), "Lineární modely"
- Laporte, G., F. Louveaux, et al. (1992). "The Vehicle-Routing Problem with Stochastic Travel-Times." Transportation Science **26**(3): 161-170.
- Larsen, A., O. Madsen, et al. (2002). "Partially Dynamic Vehicle Routing - Models and Algorithms." Journal of the Operational Research Society **53**: 638-646.
- Lau, H. C., M. Sim, et al. (2003). "Vehicle routing problem with time windows and a limited number of vehicles." European Journal of Operational Research **148**(3): 559-569.
- Letchford, A. N. and J.-J. Salazar-González (2006). "Projection results for vehicle routing." Mathematical Programming **105**(2 - 3): 251-274.
- Li, H. and A. Lim (2003). "Local search with annealing-like restarts to solve the VRPTW." European Journal of Operational Research **150**(1): 115-127.
- Nagy, G. and S. Salhi (2005). "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries." European Journal of Operational Research **162**(1): 126-141.
- Nanry, W. P. and J. Wesley Barnes (2000). "Solving the pickup and delivery problem with time windows using reactive tabu search." Transportation Research Part B: Methodological **34**(2): 107-121.
- Parpinelli, R. S., H. S. Lopes, et al. (2002). "Data mining with an ant colony optimization algorithm." Ieee Transactions on Evolutionary Computation **6**(4): 321-332.
- Pavelka, M. and V. Smetana (2000). Cmelaci. Valasske Mezirici, Cesky svaz ochrancu prirody: 115.
- Pisinger, D. and S. Ropke (2007). "A general heuristic for vehicle routing problems." Computers & Operations Research **34**(8): 2403-2435.

- Rastegar, R. and A. Hariri (2006). "The Population-Based Incremental Learning Algorithm converges to local optima." Neurocomputing **69**(13-15): 1772-1775.
- Ropke, S. and D. Pisinger (2005). "An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows." Transportation Science.
- Shang, J. S. and C. K. Cuff (1996). "Multicriteria pickup and delivery problem with transfer opportunity." Computers & Industrial Engineering **30**(4): 631-645.
- Stutzle, T. and H. H. Hoos (1996). Improving the Ant System: A detailed report on the MAX-MIN Ant System. Technical Report. TU Darmstadt, Germany, FG Intellektik, FB Informatik.
- Stutzle, T. and H. H. Hoos (2000). "MAX-MIN Ant System." Future generation computer systems **16**(8): 889-914.
- Swihart, M. R. and J. D. Papastavrou (1999). "A stochastic and dynamic model for the single-vehicle pick-up and delivery problem." European Journal of Operational Research **114**(3): 447-464.
- Tan, K. C., L. H. Lee, et al. (2001). "Heuristic methods for vehicle routing problem with time windows." Artificial Intelligence in Engineering **15**(3): 281-295.
- Thangiah, S. R., J.-Y. Potvin, et al. (1996). "Heuristic approaches to vehicle routing with backhauls and time windows." Computers & Operations Research **23**(11): 1043-1057.
- Toth, P. and D. Vigo (2000). An Overview of Vehicle Routing Problems. The Vehicle Routing Problem. Philadelphia, USA, SIAM, Society for Industrial and Applied Mathematics: 1-26.
- Tzoreff, E., T. Granot, et al. (2002). "The vehicle routing problem with pickups and deliveries on some special graphs." Discrete Applied Mathematics **116**(3): 193-229.
- Vigo, D. (1996). "A heuristic algorithm for the asymmetric capacitated vehicle routing problem." European Journal of Operational Research **89**(1): 108-126.
- Wade, A. C. and S. Salhi (2002). "An investigation into a new class of vehicle routing problem with backhauls." Omega **30**(6): 479-487.
- Wu, C. W., J. C. Beck, et al. (2004). Dynamic Vehicle Routing With Uncertain Customer Demand. International Workshop on Constraint Solving under Change and Uncertainty.
- Wu, T.-H., C. Low, et al. (2002). "Heuristic solutions to multi-depot location-routing problems." Computers & Operations Research **29**(10): 1393-1415.
- Zhong, Y. and M. H. Cole (2005). "A vehicle routing problem with backhauls and time windows: a guided local search solution." Transportation Research Part E: Logistics and Transportation Review **41**(2): 131-144.