

**Univerzita Pardubice
Fakulta ekonomicko-správní**

**Problém obchodního cestujícího pomocí
metody Mravenčí kolonie**

Petra Pokorná

Bakalářská práce

2008

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky
Akademický rok: 2007/2008

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Petra POKORNÁ**

Studijní program: **B6209 Systémové inženýrství a informatika**

Studijní obor: **Informatika ve veřejné správě**

Název tématu: **Problém obchodního cestujícího pomocí metody Mravenčí kolonie**

Z á s a d y p r o v y p r a c o v á n í :

1. Problém obchodního cestujícího
2. Charakteristika metoda mravenčí kolonie
3. Algoritmus metody mravenčí kolonie
4. Praktická aplikace Mravenčí kolonie

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Nešetřil, J. Teorie grafů. SNTL, Praha 1979


Plesník, Ján: Grafové algoritmy. VEDA, Bratislava 1983

Habiballa Hashim: Umělá inteligence. Ostravská Univerzita 2004. Distanční studijní opora.

Vittorio Maniezzo, Luca Maria Gambardella, Fabio de Luigi - Ant Colony Optimization (pdf)

Marco Doroto, Luca Maria Gambardella - Ant colonies for the traveling salesman problem. Université Libre de Bruxelles. Belgium (pdf)

Vedoucí bakalářské práce:


Ing. Jan Panuš

Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce:


30. října 2007

Termín odevzdání bakalářské práce:

19. května 2008


prof. Ing. Jan Čapek, CSc.
děkan

L.S.


doc. Ing. Pavel Petr, Ph.D.
vedoucí ústavu

V Pardubicích dne 30. října 2007

SOUHRN

Bakalářská práce je věnována především problému obchodního cestujícího. Zaměřuje se na jeho historii, definici, na jeho různé varianty a modifikace. Je v ní také uváděn celistvý pohled na metodu mravenčí kolonie, a to zejména na biologickou inspiraci této metody a její optimalizaci. Biologická inspirace v sobě zahrnuje experimenty s živými mravenci, ale popisuje i mravence umělé. Dále se práce zabývá oblastí algoritmu pro problém obchodního cestujícího pomocí mravenčí kolonie. V závěru popisuje různé modely pro řešení úloh problému obchodního cestujícího pomocí metody mravenčí kolonie i jiných metod.

KLÍČOVÁ SLOVA

problém obchodního cestujícího, travelling salesman problem, mravenčí kolonie, ant colony optimization, ant system, ant colony system, úlohy NP, NP-úplné úlohy, meta-heuristické algoritmy, rojová inteligence, stigmergie

TITLE

Travelling salesman problem per method Ant colony

ABSTRACT

The bachelor work takes care of a travelling salesman problem. It targets the history, definition, different sort of alternate and modification. In work is overview on method Ant colony, first biological inspiration and optimization. Next, work deals with algorithm for travelling salesman problem per method Ant Colony. At the close describes work models of solution travelling salesman problem per various methods.

KEYWORDS

travelling salesman problem, ant colony, ant colony optimization, ant system, ant colony system, NP problems, NP-complete problems, meta-heuristic algorithms, swarm intelligence, stigmergy

OBSAH

ÚVOD	1
1. PROBLÉM OBCHODNÍHO CESTUJÍCÍHO.....	2
1.1. HISTORIE.....	2
1.1.1. <i>Historie v datech</i>	3
1.2. DEFINICE PROBLÉMU OBCHODNÍHO CESTUJÍCÍHO	4
1.2.1. <i>Třída úloh NP (nedeterministicky polynomiální)</i>	6
1.2.1.1. NP-úplné úlohy	7
1.2.2. <i>Problém nalezení hamiltonovské cyklu (cesty)</i>	7
1.3. VARIANTY TSP	8
1.4. MODIFIKACE ÚLOHY OBCHODNÍHO CESTUJÍCÍHO.....	9
1.4.1. <i>Úloha obchodního cestujícího s časovými okny (TSP with Time Windows)</i>	9
1.4.2. <i>Úloha s více obchodními cestujícími (Multiple TSP)</i>	10
2. MRAVENČÍ KOLONIE	11
2.1. META-HEURISTICKÝ ALGORITMUS.....	11
2.2. OPTIMALIZACE POMOCÍ MRAVENČÍ KOLONIE	11
2.2.1. <i>Rojová inteligence</i>	12
2.2.2. <i>Ant system</i>	13
2.2.2.1. <i>Ant colony system</i>	15
2.2.2.2. <i>MAX-MIN Ant System (MMAS)</i>	15
2.3. MRAVENČÍ KOLONIE – BIOLOGICKÁ INSPIRACE	16
2.3.1. <i>Získávání potravy</i>	16
2.3.1.1. <i>Sociální hmyz</i>	17
2.3.2. <i>Umělí mravenci</i>	18
2.3.3. <i>Mravenčí armáda</i>	19
2.4. ŘEŠENÍ TSP POMOCÍ ACO	19
2.4.1. <i>Pravidlo přechodu</i>	21
2.4.2. <i>Globální aktualizace feromonových stop</i>	22
2.4.3. <i>Lokální aktualizace feromonových stop</i>	22
2.5. EXPERIMENTY SE SKUTEČNÝMI MRAVENCI	23
2.6. ACO ALGORITMUS PRO ŘEŠENÍ TSP	25
2.6.1. <i>Vysvětlivky k algoritmu</i>	27
3. VÝVOJOVÝ DIAGRAM ACO	28
4. PRAKTICKÁ APLIKACE	29
4.1. MODEL Y PRO ŘEŠENÍ ÚLOHY TSP POMOCÍ ACO I JINÝCH ALGORITMŮ.....	29
4.2. MODEL PRO ŘEŠENÍ TSP POMOCÍ ACO.....	35
ZÁVĚR	39
LITERATURA	40
SEZNAM OBRÁZKŮ.....	43
SEZNAM TABULEK	43
SEZNAM GRAFŮ.....	44

ÚVOD

Tématem mé bakalářské práce je problém obchodního cestujícího pomocí metody mravenčí kolonie. Problém obchodního cestujícího je velice zajímavý tím, že ačkoliv ho svět zná již dvě století, stále neexistuje jeho účinné řešení. Naopak metoda mravenčí kolonie je relativně mladá metoda inspirovaná živými mravenci, která vychází z modelu jejich chování. Je proto velice poučné sledovat aplikaci nové mladé metody na takový složitý problém, jakým bezpochyby úloha obchodního cestujícího je.

Za cíl své práce jsem si zvolila charakteristiku problému obchodního cestujícího a metody mravenčí kolonie. Také jsem se rozhodla zmapovat a popsat, jak funguje aplikace problému obchodního cestujícího na metodu mravenčí kolonie pomocí již známých výsledků, konkrétních autorů, kteří se tímto problémem zabývali přede mnou.

V první části bakalářské práce se budu zabývat teorií problému obchodního cestujícího, zahrnujícího jeho historii, definici a též jeho modifikaci. Pak přistoupím k charakteristice nové metody mravenčí kolonie, kde hlavním bodem bude biologická inspirace této metody. Po těchto charakteristikách popíši vývojový diagram a algoritmus pro řešení úlohy obchodního cestujícího pomocí mravenčí kolonie. Na závěr práce uvedu modely řešení úlohy obchodního cestujícího pomocí různých metod včetně metody mravenčí kolonie.

1. Problém obchodního cestujícího

1.1. Historie

Počátky problému obchodního cestujícího (TSP¹) jsou nejasné. Matematické problémy týkající se problému obchodního cestujícího byly poprvé zpracovány roku 1800 irským matematikem W. R. Hamiltonem a britským matematikem T. P. Kirkmanem. Hamiltonova hra Icosian vyžaduje, aby hráči dokončili cestu skrz 20 bodů použitím jen specifických spojení [17].

Diskuzi k ranému dílu Hamiltona a Kirkmana nalezneme v knize „Teorie grafů 1736-1936“ [Graph Theory] od N. L. Biggs, E. K. Lloyd, a R. J. Wilson, Clarendon Press, Oxford, 1976.

Všeobecný model TSP vypadá, že byl nejprve studován matematiky. Počínaje třicátými léty dvacátého století ekonomem Karlem Mengerem, který ho zveřejnil mezi svými kolegy ve Vídni a na Harvardu. Ve čtyřicátých letech byl problém studován statistiky (Mahalanobis (1940), Jessen (1942), Gosh (1948), Marks (1948)) [14].

TSP byl později podpořen Hasslerem Whitneyem a Merrillem Floodem z Princetonu. Průlom přišel, když G. Dantzing, R. Fulkerson a S. Johnson (1954) vydali popis metody pro řešení problému obchodního cestujícího. Sílu této metody objasnili tím, že řeší příklad s 49 městy v efektivním čase.

Detailní zacházení se souvislostmi mezi Mengerem a Whitneyem a růst TSP jako téma studie, můžeme nalézt ve spisech Alexandra Schrijvera „Historie kombinatorické optimalizace (do roku 1960)“ [On the history of combinatorial optimization]. Nakonec si TSP získal proslulost jako prototyp obtížného problému v kombinatorické optimalizaci.

¹ TSP – travelling salesman problem

1.1.1. Historie v datech

- § 1954 – G. Dantzing, R. Fulkerson a S. Johnson řešili příklad s 49 městy v efektivním čase.
- § 1962 – Firma Procter&Gamble vyhlásila soutěž, ve které požadovala nalezení optimální cesty pomocí TSP při specifikovaných 33 městech. Vítězem soutěže se stal profesor G. Thompson z Carnegie Mellon University.
- § 1977 – Groetschel našel optimální cestu 120 měst, ze kterých se později stalo Západní Německo.
- § 1987 – Padberg a Rinaldi našli optimální cestu pro 532 AT&T přepínacích míst v USA.
- § 1987 – Groetschel a Holland našli optimální cestu 666 zajímavých míst na světě.
- § 1987 – Padberg a Rinaldi našli optimální cestu skrz strukturu 2 392 bodů, kterou obdrželi od Tektronics Incorporated.
- § 1994 – Applegate, Bixby, Chvátal, a Cook našli optimální cestu pro 7 397 měst. TSP vznikl v programovatelném hradlovém poli² aplikace v AT&T Bell laboratořích.
- § 1998 – Applegate, Bixby, Chvátal, a Cook našli optimální cestu pro 13 509 měst v USA s populací větší než 500.
- § 2001 – Applegate, Bixby, Chvátal, a Cook našli optimální cestu 15 112 měst v Německu.
- § 2004 – Applegate, Bixby, Chvátal, Cook, a Helsgaun našli optimální cestu 24 978 měst ve Švédsku.

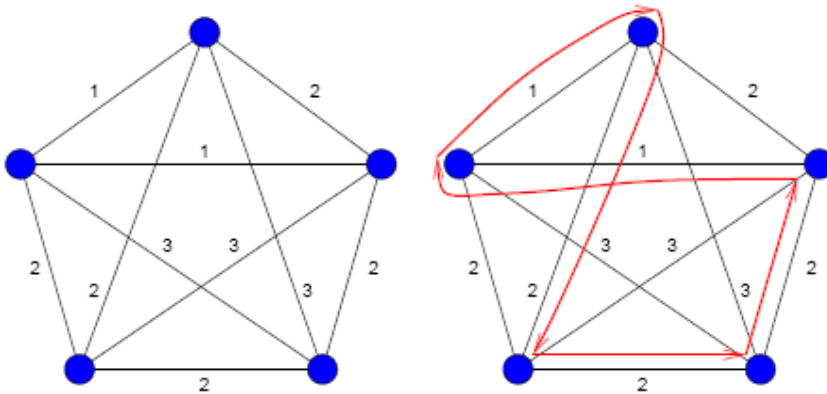
Přes intenzivní výzkum matematiků, počítačových vědců, operačních výzkumníků a jiných, za posledních padesát let, tu stále zůstává otevřená otázka, zda existuje nebo ne, účinné obecné řešení této metody.

² Hradlové pole - je elektronická součástka používaná pro vytváření digitálních obvodů.

1.2. Definice problému obchodního cestujícího

a) Definice využívající teorii grafů

Problém obchodního cestujícího vyžaduje nalezení nejkratší cesty, která je určena k návštěvě souboru měst a návratu do počátečního bodu. Neorientovaná úloha může být uváděna v teorii grafů následovně. Necht' $G = (V, A)$ je graf, kde $V = \{v_1, \dots, v_n\}$ je vrchol (nebo uzel) a $A = \{(v_i, v_j) | v_i, v_j \in V, i \neq j\}$ je hrana s nezápornými náklady (nebo vzdálenostmi) matice $C = (c_{ij})$ spojená s A . TSP spočívá v určení minima nákladů hamiltonovského cyklu v úloze grafu. Souměrné implicitní používání neorientované hrany, spíše než orientovaného oblouku, může být také výslovně stanovené vztahem $c_{ij} = c_{ji}$. Úloha TSP patří mezi NP-úplné úlohy.



Obrázek 1 - Problém obchodního cestujícího (zdroj: [16])

b) Definice TSP pomocí n -měst

TSP může být snadno uveden následovně. Obchodní cestující chce navštívit n různých měst a poté se navrátit domů. Chce stanovit pořadí cesty, tak aby cestovní vzdálenost byla minimalizovaná, když navštíví každé město ne více než jednou. Ačkoli je TSP pojmově jednoduchý, je obtížné získat optimální řešení.

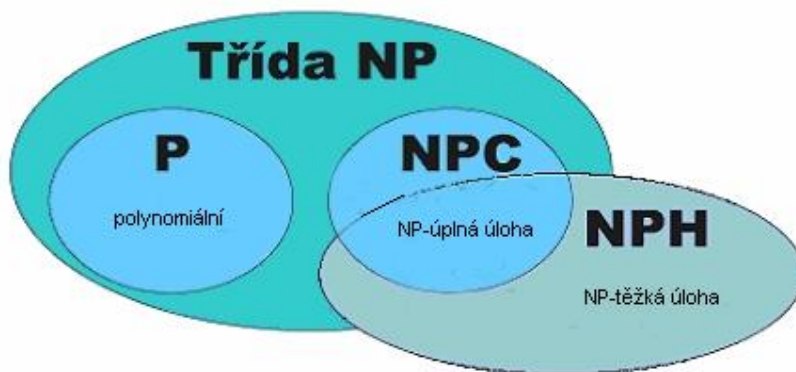
V n -městské situaci, kterákoliv permutace n měst přináší možné řešení. V důsledku toho $n!$ možných cest musí být ohodnoceno ve vyhledávacím prostoru. U uvedených proměnných x_{ij} označující cestu obchodního cestujícího z města i do města j , může být jedno z obecných celočíselných programovacích formulací pro TSP napsáno následovně [12]:

1.2.1. Třída úloh NP (nedeterministicky polynomiální)

V typických NP-úlohách se ptáme, zda existuje něco, co se možná i těžko hledá, ale snadno se ověřuje, že jsme to našli. Konkrétním příkladem je úloha, zda v daném grafu existuje kružnice, která prochází všemi vrcholy grafu (tzv. hamiltonovská kružnice). Certifikátem zde je ona kružnice. Ověřovací algoritmus ověřuje, že je to opravdu kružnice a že opravdu prochází přes všechny vrcholy grafu.

Poznamenejme, že u NP-úloh je velmi důležité, jak je položena otázka, která se v úloze řeší. Je to proto, že certifikát a potvrzení požadujeme pouze pro kladnou odpověď. Obrácením otázky u nějaké NP-úlohy tedy dostaneme úlohu zcela jinou, která již do třídy NP nemusí patřit (může být těžší).

Každá úloha ze třídy P je zároveň ve třídě NP, neboť polynomiální algoritmus, který úlohu řeší, lze pokládat za algoritmus ověřovací, u něhož na certifikátu nezáleží. Platí tedy $P \subseteq NP$. V současné době není známo (nikdo neumí matematicky dokázat), zda platí rovnost, ale všeobecně se věří, že $P \neq NP$. Ve třídě NP jsou úlohy různé obtížnosti. [1]



Obrázek 2 - Hierarchie tříd složitosti (zdroj: [10])

1.2.1.1. NP-úplné úlohy

NP-úplná úloha (anglicky NP-complete) je taková úloha, která patří do NP a je ve třídě NP nejtěžší v tom smyslu, že jakákoli jiná NP-úloha je na ni P-redukovatelná. Třídu všech NP-úplných úloh značíme NP-C. Všechny NP-úplné úlohy jsou z hlediska P-redukce stejně těžké.

Kdyby se podařilo najít polynomiální algoritmus pro kteroukoli z nich, měli bychom ihned polynomiální algoritmy pro všechny NP-úlohy a platilo by $P = NP$. V podstatě nikdo nevěří, že se takový algoritmus najde, ale dokázat to prozatím nikdo nedovede. [1]

1.2.2. Problém nalezení hamiltonovské cyklu (cesty)

Problém nalezení hamiltonovské cesty je definován takto: necht' G je orientovaný graf s označeným počátečním uzlem A a koncovým B . Cesta z uzlu A do B se nazývá hamiltonovská právě tehdy, když obsahuje každý uzel grafu G právě jednou. Obecně je problém nalezení hamiltonovské cesty formulován jako rozhodnutí, zda daný orientovaný graf hamiltonovskou cestu obsahuje či ne.

Problém nalezení hamiltonovské cesty je NP-úplný. To znamená, že neexistuje jeho efektivní, v polynomiálním čase dosažitelné řešení, a že všechna jeho obecná řešení vedou na úplné prohledávání stavového prostoru. Počet kroků potřebných na vyřešení problému roste exponenciálně s velikostí grafu.

Jednou z aplikací je problém obchodního cestujícího. Obchodník vyjíždí ze svého města (počáteční uzel) a navštívuje všechna existující města (uzly) právě jednou. Při tom se smí pohybovat pouze po vyznačených cestách (hrany grafu). Svou pouť končí v předem určeném cílovém městě (koncový uzel). Problémem je zjistit, zda může obchodník navštívit všechna města. Pokud může, máme zjistit, po kterých cestách se má pohybovat. [15]

kde n je počet míst, které je třeba navštívit, $x_{ij} = 1$, jestliže hrana z uzlu i do uzlu j je zařazena do hledaného hamiltonova cyklu, v opačném případě je $x_{ij} = 0$. Podmínka (6) vyjadřuje skutečnost, že v každém uzlu i právě jedna hrana končí a právě jedna hrana z něj vychází. Zaručují tudíž, že každý uzel bude obsažen v nějakém cyklu, nezaručují však hamiltonův cyklus, protože připouštějí více cyklů, tj. parciální cykly. Smyčkovou podmínku (7) lze vyjádřit také jako $\sum_{i \in U} \sum_{j \in V-U} x_{ij} \geq 1$ pro $i \neq j, \forall U \in V, 2 \leq |U| \leq n-1$. Nelze ji však zapsat pomocí pořadových čísel uzlů, jak ji definovali Miller, Tucker a Zemlin. Poslední podmínka tohoto modelu představuje skutečnost, že proměnná x_{ij} je bivalentní.

Asymetrická úloha obchodního cestujícího má nesymetrickou matici nákladů C (v grafickém vyjádření jde o orientovaný graf). Existuje různé ohodnocení cest mezi dvěma vrcholy ($c_{ij} \neq c_{ji}$). Matematický model je uveden v předchozí části. [3]

§ **Geometrický** – ctí platnost trojúhelníkové nerovnosti $d(u,v) \leq d(u,w) + d(w,v)$;
 $\forall u,v,w$.

§ **Eukleidovský TSP** – města odpovídají bodům v d -dimenzionálním prostoru a funkce ceny koresponduje s eukleidovskou vzdáleností, tedy $\sqrt{\sum_{i=1}^d (x_i - y_j)^2}$.

§ Cesta existuje mezi každými dvěma městy versus z daného města vede cesta pouze do některých dalších měst. [10]

1.4. Modifikace úlohy obchodního cestujícího

1.4.1. Úloha obchodního cestujícího s časovými okny (TSP with Time Windows)

Obchodní cestující opět musí projít celou trasu přes všechny zákazníky, kdy navštíví každého právě jednou a vrací se do výchozího místa. Navíc je ale omezen tím, že každému zákazníkovi je přiřazeno tzv. časové okno (období, kdy ho musí obchodní cestující navštívit). Časové okno i tého zákazníka je tedy časový interval mezi nejdříve možným začátkem obsluhy a_i a nejpozději možným začátkem obsluhy b_i . Označme okamžik, kdy začíná obsluha

itého zákazníka t_i . Předpokládáme, že $t_i \in \langle a_i, b_i \rangle$ (více v matematickém modelu). Dále ještě potřebujeme definovat matici D , jejíž prvky d_{ij} vyjadřují dobu přejezdu (přechodu) mezi zákazníky i a j .

Matematický model této úlohy je shodný s matematickým modelem úlohy obchodního cestujícího. Místo podmínky $u_i - u_j + nx_{ij} \leq n - 1$, $i = 1, 2, \mathbf{K}, n$, $j = 2, 3, \mathbf{K}, n$, $i \neq j$, je smyčková podmínka definována $t_i + d_{ij} - M(1 - x_{ij}) \leq t_j$, $i = 1, 2, \mathbf{K}, n$, $j = 2, 3, \mathbf{K}, n$, $i \neq j$, kde M je vysoká konstanta.

V případě, že bychom do modelu zavedli navíc dobu obsluhy itého zákazníka S_i , měla by smyčková podmínka tvar $t_i + S_i + d_{ij} - M(1 - x_{ij}) \leq t_j$, $i = 1, 2, \mathbf{K}, n$, $j = 2, 3, \mathbf{K}, n$, $i \neq j$. Dále přibude podmínka pro dodržení časového okna $a_i \leq t_i \leq b_i$, $i = 2, 3, \mathbf{K}, n$. V tomto tvaru je označována jako silné omezení („hard“). Pokud bychom povolili překročení tohoto intervalu za cenu penále, je označována jako slabé omezení („soft“). Nyní dodefinujeme proměnnou t_i . Z výchozího uzlu vyrazí obchodní cestující v čase nula, proto $t_1 = 0$. Pro ostatní časy příchodu k itému zákazníkovi platí $t_i \geq 0$, $i = 2, 3, \mathbf{K}, n$. [3]

1.4.2. Úloha s více obchodními cestujícími (Multiple TSP)

V této úloze mají všichni obchodní cestující jeden výchozí bod, do kterého se po absolvování svého okruhu vrací. Cílem úlohy s více obchodními cestujícími je minimalizovat délku všech tras absolvovaných všemi obchodními cestujícími. Opět každé město či zákazník musí být navštíven právě jednou.

Všichni obchodní cestující musí vyjít z výchozího bodu a také do něj přijít zpět. Jestliže k definici úlohy s více obchodními cestujícími přidáme jejich kapacitu, získáváme definici rozvozní. Tato úloha může být navíc kombinována s možností více výchozích míst (Multiple Depot), s časovými okny (with Time Windows), atd. [3]

2. Mravenčí kolonie

2.1. *Meta-heuristický algoritmus*

Meta-heuristické algoritmy jsou takové algoritmy, které aby vyskočily z lokálního optima, řídí některé základní heuristiky. Buď konstruktivní heuristika začíná od nulového řešení a přidáváním základních prvků vede k vytvoření dobrého kompletu, nebo lokální vyhledávací heuristika začíná od kompletního řešení a iteračně modifikující některé z prvků v postupu dosáhnout nějakého lepšího řešení.

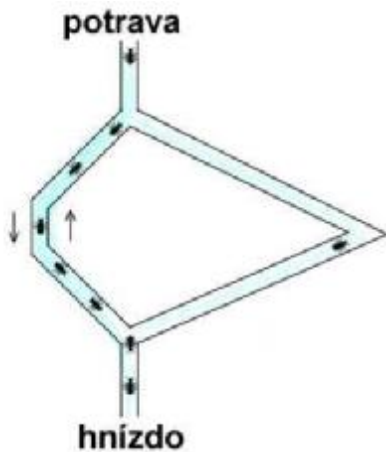
Meta-heuristická úloha povoluje nízkou úroveň heuristiky k získání lepšího řešení, než jakého by mohla dosáhnout sama, i když iterovaná. Obvykle kontrolní mechanismus dosáhne buď za omezujícího, nebo náhodně generovaného čísla sady z lokálního sousedního řešení zvažovaného v lokálním vyhledávání (to je případ Simulovaného žhání a Tabu prohledávání) nebo za kombinujících základních prvků vzatých za rozdílných řešení.

2.2. *Optimalizace pomocí mravenčí kolonie*

V téměř každém oboru lidské činnosti se mohou vyskytovat optimalizační problémy. Metoda mravenčí kolonie (ACO³) je založena na algoritmu rojové inteligence a je velice úspěšně aplikována na řešení optimalizačních úloh. [8]

ACO je nový přístup v navrhování meta-heuristických algoritmů pro kombinatorické optimalizační problémy. Základní rys ACO algoritmu je kombinace apriorní informace o struktuře příznivého řešení s posteriori informací o struktuře předchozího získaného správného řešení. Tento algoritmus je inspirován zkoumáním reálných mravenců.

³ ACO – Ant Colony Optimization



Obrázek 4 - Zvyšování pravděpodobnosti kratší cesty u metody ACO (zdroj: [10])

Tabulka 2 - Neúplný seznam úspěšných ACO algoritmů (v chronologickém pořadí)

Algoritmus	Autor (autoři)	Rok
Ant System (AS)	Dorigo a spol.	1991
Elitist AS	Dorigo a spol.	1992
Ant-Q	Gambardella & Dorigo	1995
Ant Colony System	Dorigo & Gambardella	1996
MAX-MIN AS	Stützle & Hoos	1996
Rank-based AS	Bullnheimer a spol.	1997
ANTS	Maniezzo	1999
BWAS	Cordón a spol.	2000
Hyper-cube AS	Blum a spol.	2001

2.2.1. Rojová inteligence

Rojovou inteligence můžeme chápat jako techniku umělé inteligence založené na studiu kolektivního chování decentralizovaných samoorganizujících se systémů. Systémy rojové inteligence se klasicky skládají z populace jednoduchých agentů interagujících⁴ lokálně mezi sebou a také s okolním prostředím. Komunikace mezi agenty může probíhat přímo, nebo

⁴ Interagující – vzájemně působící. Zdroj: KOLEKTIV AUTORŮ. *Slovník cizích slov*. 1. vydání. OTTOVO nakladatelství. Praha, 2000. s. 317. ISBN 80-7181-376-1.

nepřímo působením na lokální prostředí. Přestože nemají tyto systémy žádnou centrální kontrolu chování agentů, lokální působení mezi agenty a jednoduché vzory chování agentů mohou vést k emergenci globálního chování. Takovéto systémy můžeme v přírodě pozorovat například u mravenců, bakterií nebo včel. [8]

2.2.2. Ant system

Význam originálního Ant System (Mravenčího systému) spočívá především v existenci prototypu množství mravenčích algoritmů, které všeobecně implementují ACO paradigma. Pohyb pravděpodobnosti rozdělení charakterizuje pravděpodobnosti p_{iy}^k rovny 0 pro všechny pohyby, které jsou nemožné (tj. jsou v tabu seznamu mravence k , tj. seznam obsahující všechny pohyby, které jsou nemožné pro mravence k začínajících ze stavu i), nebo jsou popsány prostřednictvím rovnice (8), kde a a b jsou parametry definované uživatelem ($0 \leq \alpha, \beta \leq 1$):

$$p_{iy}^k = \begin{cases} \frac{t_{iy}^a + h_{iy}^b}{\sum_{(iz) \notin \text{tabu}_k} (t_{iz}^a + h_{iz}^b)} & \text{if } (iy) \notin \text{tabu}_k \\ 0 & \text{jinak.} \end{cases} \quad (8)$$

V rovnici (8) tabu_k je tabu seznam mravence k , zatímco parametry a a b určují dopad cesty a atraktivnosti. Po každé iteraci t algoritmu, tj. když všichni mravenci mají hotové řešení, cesty jsou aktualizovány prostřednictvím následující rovnice: $t_{iy}(t) = \rho t_{iy}(t-1) + \Delta t_{iy}$ kde Δt_{iy} reprezentují součet přínosů všech mravenců, kteří používali pohyb (iy) k vytvoření svého řešení. ρ , $0 \leq \rho \leq 1$ jsou uživatelsky definované parametry nazývané odpařovacími koeficienty. Mravenčí přínosy jsou dosaženy úměrně ke kvalitě řešení, tj. čím lepší řešení je, tím vyšší přínosy budou z cest přidaných k pohybu používané.

Např. v příkladu TSP, pohyby korespondují s grafovým obloukem, čili stav i může korespondovat s cestou, která končí v uzlu (vrcholu) i , stav ψ se stejnou cestou, ale s obloukem (ij) přidaným na konec a pohyb by byl četností průchodů grafu (ij) .

Kvalita řešení mravence k by byla délka L_k cesty nalezená mravencem a rovnice

$$t_{iy}(t) = r t_{iy}(t-1) + \Delta t_{iy} \text{ by se stala } t_{ij}(t) = r t_{ij}(t-1) + \Delta t_{ij}, \text{ s } \Delta t_{ij} = \sum_{k=1}^m \Delta t_{ij}^k$$

kde m je počet mravenců a Δt_{ij}^k je součet cesty dané hrany (ij) u mravence k , která může být vypočtena jako

$$\Delta t_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if mravenec } k \text{ používá arc}(ij) \text{ v jeho cestě} \\ 0 & \text{jinak.} \end{cases} \quad (9)$$

kde v rovnici (9) je Q konstanta parametrů.

Ant System (Mravenčí systém) jednoduše iteruje hlavní cyklus, kde m mravenců vytváří své řešení, a poté aktualizují úroveň cesty. Výkon algoritmu závisí na správném vyladění několika parametrů, jmenovitě α , β , vzájemná důležitost cesty a atraktivnosti, ρ , cesta vytrvalosti, $\tau_{ij}(0)$, počáteční úroveň cesty, m , číslo mravence a Q , používané pro definování vysoké kvality řešení s nízkými náklady.

Algoritmus je následující [5]:

1. {Inicializace}

Inicializovat $\tau_{i\psi}$ a $\eta_{i\psi}$, $\forall (i\psi)$.

2. {Konstrukce}

For každý mravenec k (aktuálně ve stavu i) do

repeat

vybrat z pravděpodobnosti stav k přesunutí do.

přidat vybraný pohyb ke k -tému mravenčímu seznamu tabu_k .

3. {Aktualizace cesty}

For každý pohyb mravence $(i\psi)$ do

vypočítat $\Delta\tau_{i\psi}$

aktualizovat cestu matice.

end for

4. {Ukončovací podmínka}

If not (konec testování) go to krok 2.

2.2.2.1. Ant colony system

AS byl první algoritmus inspirovaný chováním živých mravenců. AS byl prvně použit na řešení problému obchodního cestujícího, ale nebyl schopný obstát na pozadí stavu techniky (na úrovni doby). Naproti tomu má podstatu věci představenou ACO algoritmy a ukazuje potenciální použití umělého feromonu a umělých mravenců přivedených k hledání vždy lepšího řešení složitějšího optimalizačního problému. Další zkoumání bylo motivováno dvěma cíly: první bylo zlepšení výkonnosti algoritmu a druhé bylo zkoumání a lepší vysvětlení chování. Gambardella a Dorigo navrhli v roce 1995 Ant-Q algoritmus k rozšíření AS, který začleňuje některé nápady z Q-learning a v roce 1996 Ant Colony System zjednodušená verze Ant-Q, která udržovala aproximaci stejné úrovně vykonávání, měřená u algoritmu složitosti a u výpočtových výsledků. Protože ACS je základ mnoha algoritmů definovaných v následujících letech, zaměříme naši pozornost na ACS jinou než Ant-Q. ACS se liší od předcházejícího AS kvůli třem hlavním aspektům: feromonu, stavu přechodového pravidla, hybridizaci a zlepšení výkonu.

2.2.2.2. MAX-MIN Ant System (MMAS)

Tento algoritmus je zlepšením originálního Ant System (AS). Jeho charakteristické prvky jsou takové, že jen nejlepší mravenec aktualizuje feromonovou cestu a hodnota feromonu je mez (hranice). Feromonová aktualizace je realizována následovně:

$$t_{ij} \leftarrow \left[(1-r) \cdot t_{ij} + \Delta t_{ij}^{best} \right]_{t_{min}}^{t_{max}}$$

kde, τ_{max} a τ_{min} jsou příslušné maximální a minimální meze (hranice) uložené ve feromonu.

Operátor $[x]_b^a$ je definován takto:

$$[x]_b^a = \begin{cases} a & \text{if } x \in [a, b] \\ b & \text{if } x < a \\ x & \text{jinak.} \end{cases} \quad (10)$$

a Δt_{ij}^{best} je:

$$\Delta t_{ij}^{best} = \begin{cases} 1/L_{best} & \text{if } (i, j) \text{ náleží nejlepší cestě} \\ 0 & \text{jinak.} \end{cases} \quad (11)$$

L_{best} je délka cesty nejlepšího mravence. Ta může být (podřízená algoritmu tvůrci rozhodnutí) jednou nejlepší cestou nalezenou v aktuální iteraci – iterace best, L_{ib} – nebo řešením nalezeném od startu algoritmu – best-so-far, L_{bs} – nebo kombinací obou.

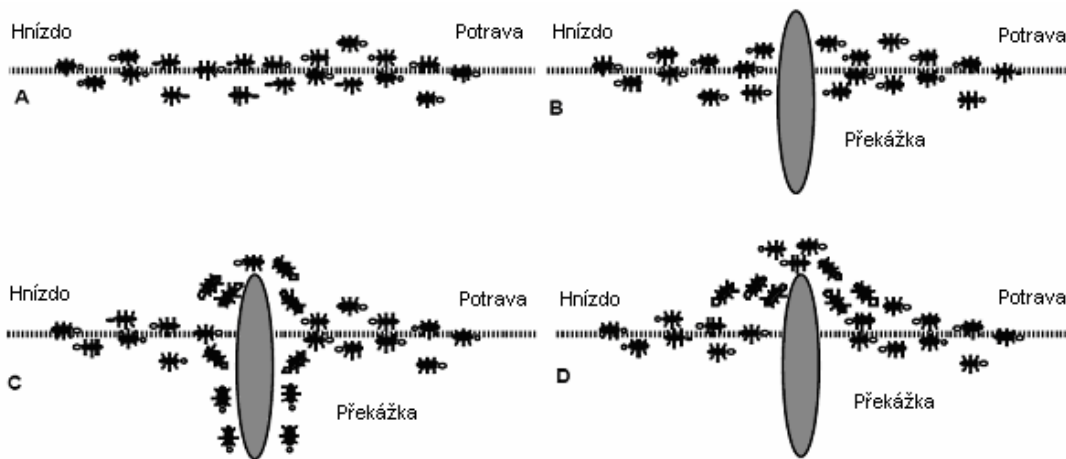
Maxima a minima mezí na feromonových hodnotách, τ_{max} a τ_{min} , jsou typicky obstarávány empiricky a vyladěná na specifický problém. Nicméně, některá vodítka se starají o definování τ_{max} a τ_{min} na základě analytické úvahy.

2.3. Mravenčí kolonie – biologická inspirace

Mravenčí kolonie jsou relativně mladou metodou optimalizace. Inspirace pochází z chování živých mravenců v kolonii. Aplikací jednoduchých pravidel, kterými se řídí jednotliví jedinci, vzniká komplexní chování celku, jenž je schopné řešit složité optimalizační úlohy. Počet úspěšných aplikací těchto algoritmů exponenciálně roste zejména v oborech jako jsou optimalizace, robotika a komunikační sítě. [8]

2.3.1. Získávání potravy

Mravence je možno považovat za sociální hmyz tzn., že jejich chování směřuje k zachování kolonie. Specifické chování při získávání potravy můžeme pozorovat u mnoha druhů mravenců. Mravenec, při pohybu od hnízda ke zdroji potravy a nazpět, ukládá chemickou látku, která je nazývána feromonem. Ostatní mravenci umí odhalit tyto feromony (a rozlišovat mezi jejich různými koncentracemi) zanechané svými soukmenovci a vybírají si cestu, která má největší koncentraci tohoto feromonu. Mechanismus této nepřímé komunikace je nazýván „stigmergie“, což je termín převzatý z výzkumu kolonií sociálního hmyzu. [8]



Obrázek 5 - Chování živých mravenců (zdroj: [6])

A – živí mravenci hledají cestu mezi hnízdem a zdrojem potravy.

B – objevila se překážka na cestě: mravenci volí možnost, zda se dát doleva nebo doprava, se stejnou pravděpodobností.

C – feromon je rychleji ukládán na kratší cestu.

D – všichni mravenci vybírají kratší cestu.

2.3.1.1. Sociální hmyz

Eusocialitu u živočichů je možno nazývat takové společenské uspořádání, kde v jednom hnízdě žije ve vzájemné pospolitosti a spolupráci více jedinců téhož druhu. Eusocialita je vymezena těmito několika následujícími rysy:

- § dospělí jedinci se o potomstvo starají společně;
- § v jednom hnízdě spolu žijí alespoň dvě generace;
- § společnost hnízda je rozdělena na reprodukující se („královskou“) kastu a nereprodukující se („dělnickou“) kastu;
- § všichni jedinci jsou zcela oddáni celému organismu, chrání jej a dbají o jeho přežití.

[20]

Mravenci vytvářejí několik kast. Do základními kast můžeme zařadit královnu (gyne), samce (aner) a dělnice (ergate). Vojáky jsou nazývané velké dělnice s neobvykle vyvinutými kusadly (dinergate). Každá kolonie by měla mít alespoň jednu královnu a několik dělnic.

Mravenci se vyskytují téměř všude na planetě Zemi, v tropických deštných lesích mohou tvořit až 15 % celkové živočišné biomasy. V roce 2006 bylo známo 11 844 mravenčích druhů, většina z nich obývá oblasti s teplým podnebím. Mravenci potřebují pro svůj život vysokou vlhkost vzduchu, ale nežijí ve vysloveně zamokřených místech. Budují si hnízda na různých místech, jako například pod zemí, ve stromech a v rostlinách, někteří dokonce mezi listy stromů (*Oecophylla sp.*) či zcela volně.

Běžný mravenec lesní se může dožít věku 7 až 10 let a mravenčí královna dokonce 10 až 20 let. [21]

2.3.2. Umělí mravenci

Umělí mravenci byly vytvořeni pro účely implementace optimalizačních algoritmů. Jejich chování vychází z chování skutečných živých mravenců. Avšak oproti živým mravencům byly některé vlastnosti umělých mravenců posíleny natolik, aby došlo ke zlepšení výsledků algoritmů řešících konkrétní problémy.

Podobnosti se živými mravenci:

- § kolonie kooperujících mravenců;
- § feromonová stopa;
- § nepřímá komunikace mravenců zprostředkovaná pomocí feromonu;
- § pravděpodobnostní rozhodování, lokálnost strategie.

Rozdíly oproti živým mravencům:

- § diskrétní svět;
- § vnitřní stavy - osobní paměť zaznamenávající doposud vykonané akce;
- § umělí mravenci nejsou zcela slepí;
- § množství zanechaného feromonu je funkcí kvality nalezeného řešení;
- § problémově závislé časování ukládání feromonu.

Metody mravenčí kolonie můžeme použít pro řešení kteréhokoliv diskrétního optimalizačního problému, pro který lze použít konstruktivní heuristickou proceduru. Mravenci, které jsou použity v mravenčích koloniích, fungují jako stochastické konstruktivní procedury, které

vytvářejí řešení interaktivním přidáváním komponent do částečného řešení, přičemž při výběru každé další komponenty uvažují

- § heuristickou informaci o řešeném problému, která směřuje výpočet ke slibným řešením,
- § zkušenosti získané všemi mravenci od začátku výpočtu, reprezentované feromonovými stopami, které se během výpočtu neustále adaptují.

Vygenerování velkého počtu různých řešení umožňuje stochastická složka. [8]

2.3.3. Mravenčí armáda

Mravenčí armádu je možno zařadit mezi největší a nejvíce soudržné společnosti. Systém, jakým mravenci získávají potravu v koordinaci stovky tisíc jedinců pokrývající i tisíce metrů čtverečních. Účinné obrazce jsou dynamické. Základní struktura je vykazována stejně. Konkrétní tvar vytvořeného obrazce se shoduje s prostorovým rozložením potravy pro jednotlivé druhy.

Simulace prokazují, že stejný program (počítačový program s ekvivalentním chováním) může řešit různé problémy, nebo se adaptovat na různá prostředí. Z toho plyne, že jediný algoritmus založený na samoorganizaci je schopný generovat optimální struktury, nebo distribuční sítě, bez centrálního řízení. [28]

2.4. Řešení TSP pomocí ACO

Základní myšlenka algoritmu spočívá v použití mechanismu kladné zpětné vazby, založeném na analogii s označováním a sledováním cesty některými druhy mravenců. Virtuální feromon umožňuje uchovat v paměti dobrá řešení, odkud mohou být použita k doplnění ještě lepších řešení. Je samozřejmé se vyhnout tomu, aby některá dobrá, ale ne dostatečně dobrá řešení, vedla k předčasnému uvážnutí algoritmu. Proto je do algoritmu implementována záporná zpětná vazba, která spočívá ve vypařování feromonu, čímž je zahrnuto do algoritmu časové měřítko. Příliš dlouhé časové měřítko může vést k uvážnutí v lokálním extrému. Naopak příliš krátké měřítko může způsobit nevyužití efektu spolupráce. Spolupráce je dalším velice

důležitým pojmem. Vyhledávání pomocí různých řešení souboru identických mravenců využívá algoritmus mravenčích kolonií. Mravenci pracující v dané iteraci dobře, ovlivňují výzkum mravenců v budoucích iteracích, neboť ti využívají feromonovou cestu jako průvodce pro své výzkumy.

U problému obchodního cestujícího je cílem najít uzavřenou cestu minimální délky spojující n měst. Každé město musí být navštíveno právě jednou. Necht' d_{ij} je vzdálenost mezi městy i a j . Problém může být definován v Euklidovském prostoru, potom

$$d_{ij} = \left[(x_i - x_j)^2 + (y_i - y_j)^2 \right]^{1/2}$$

kde x_i a y_i jsou souřadnice města i , nebo může být obecně definován na grafu (V, E) , kde města jsou vrcholy V a spojení mezi městy jsou hrany grafu E . Matice vzdáleností nemusí být symetrická, pokud je asymetrická, délka hrany spojující dvě města i a j závisí na tom, zdali jde z i do j , nebo z j do i . Ale zdali je problém symetrický nebo asymetrický, nemění nic na tom, jak je řešen algoritmem mravenčích kolonií.

Pohybem po grafu problému z jednoho města do druhého budují mravenci řešení, dokud nedokončí svou kompletní cestu. Během iterace ACO algoritmu každý mravenec k , $k = 1, \dots, m$, buduje cestu provedením $n = |M|$ kroků. Iterace jsou indexovány pomocí t , t_{max} je uživatelem definovaný maximální počet povolených iterací.

Pro každého mravence, přechod z města i do města j v iteraci t algoritmu závisí:

- Na tom, zda-li město již bylo navštíveno. Pro každého mravence je udržována paměť, která roste během cesty a která je vyprázdněna mezi cestami. Paměť je užita pro definování: pro každého mravence k , množina J_i^k měst, které mravenec stále má navštívit, když je v městě i (na začátku J_i^k obsahuje všechna města s výjimkou i). Využitím J_i^k se mravenec k může vyhnout návštěvě města více než jednou.
- Na převrácené hodnotě vzdálenosti $h_{ij} = 1/d_{ij}$, zvané *dohlednost*. Dohlednost je založena na striktně lokální informaci a reprezentuje *vhodnou heuristiku* výběru města j z města i . Dohlednost může být použita k přímému hledání, ačkoli konstruktivní metoda založená na tomto základu může produkovat velmi nekvalitní řešení. Heuristická informace je statická, nemění se tedy v průběhu řešení problému.
- Na množství virtuálních *feromonových stop* $\tau_{ij}(t)$ na hraně spojující město i s městem j . Feromonové stopy jsou aktualizovány on-line a jsou reprezentací *naučené vhodnosti*

výběru města j z města i . Oproti vzdálenosti jsou feromonové stopy více globální druh informace. Tato informace se během řešení problému mění a reflektuje zkušenosti získané mravenci. [13]

2.4.1. Pravidlo přechodu

Mravenec k ve městě i vybírá město j podle následujícího pravidla:

$$j = \begin{cases} \arg \max_{j \in J_i^k} \{ [t_{ij}(t)] \times [h_{ij}]^b \} & \text{pro } q \leq q_0 \\ J & \text{pro } q > q_0 \end{cases}$$

kde q je náhodná veličina s rovnoměrným rozložením $[0,1]$, q_0 je nastavitelný parametr z intervalu $[0,1]$ a J z množiny J_i^k je město, které bylo náhodně vybráno podle pravděpodobnosti

$$p_{ij}^k(t) = \frac{[t_{ij}(t)] \times [h_{ij}]^b}{\sum_{j \in J_i^k} [t_{ij}(t)] \times [h_{ij}]^b}$$

b je nastavitelný parametr, který reguluje relativní váhu dohlednosti h_{ij} . Za zmínku stojí, že hodnota pravděpodobnosti $p_{ij}^k(t)$ může být rozdílná pro dva mravence ve stejném městě i , poněvadž $p_{ij}^k(t)$ je funkcí J_i^k , to jest dílčího řešení stavěného mravencem k . $q \leq q_0$ odpovídá využití známých znalostí o problému, to jest heuristických znalostí o vzdálenostech mezi městy a naučených znalostí uložených ve formě feromonových stop, kdežto $q > q_0$ upřednostňuje více výzkumu. Omezení výzkumu naladěním q_0 dovoluje koncentrovat aktivitu systému na nejlepší řešení, namísto stálého prohledávání. Když q_0 je blízko 1, je vybráno jen lokálně optimální řešení (ale kombinace lokálně optimálních řešení nemusí vyústit v globálně optimální řešení), zatímco když q_0 je blízko 0, zkoumána jsou všechna lokální řešení, ačkoli značná váha je dána lokálně optimálnímu řešení. [13]

2.4.2. Globální aktualizace feromonových stop

Globálně aktualizovat koncentraci feromonu může pouze takový mravenec, který generuje nejlepší cestu od začátku. Proto jsou mravenci povzbuzováni k hledání cest v nejbližším okolí doposud nejlepší nalezené cesty. Globální aktualizace feromonových stop je aplikována jen na hrany náležící nejlepší trase od začátku cesty. Aktualizační pravidlo je:

$$\tau_{ij}(t) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta \tau_{ij}(t),$$

kde (i, j) jsou hrany náležící T^+ , nejlepší trasy od začátku, ρ je parametr řídící rozklad feromonu a

$$\Delta \tau_{ij}(t) = 1/L^+,$$

kde L^+ je délka T^+ . Procedura dovoluje pouze té nejlepší cestě, aby byla zpevněna globální aktualizací. Lokální aktualizace jsou ovšem také prováděny, proto další řešení mohou vyjít najevo. [13]

2.4.3. Lokální aktualizace feromonových stop

Lokální aktualizace jsou prováděny následovně. Když je mravenec k ve městě i a vybírá město $j \in J_i^k$, koncentrace feromonu na (i, j) je aktualizována podle následujícího pravidla:

$$\tau_{ij}(t) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_0.$$

Hodnota τ_0 je stejná jako počáteční hodnota feromonových stop a bylo experimentálně zjištěno, že nastavením $\tau_0 = (n \cdot L_{mn})^{-1}$, kde n je počet měst a L_{mn} je délka cesty produkovaná heuristikou nejbližšího souseda, dává dobré výsledky. [13]

2.5. Experimenty se skutečnými mravenci

V 50. a 60. letech 20. století francouzský entomolog Pierre-Paul Grassé pozoroval, že některé druhy termitů reagují na něco, co nazval „významným podnětem (stimulem)“. Pozoroval, že efekty těchto reakcí mohou fungovat jako významné stimuly pro oba hmyzy, kteří je produkují a i pro jiné hmyzy v kolonii. Pierre-Paul Grassé používal výraz stigmergie k popsání tohoto speciálního typu komunikace, ve které jsou pracující stimulovali výkony, jež mají dosáhnout.

Dvě hlavní charakteristiky stigmergie, které ji odlišují od jiných forem komunikace jsou následující:

- stigmergie je nějaká nepřímá, nesymbolická forma komunikace zprostředkovaná u životního prostředí – hmyz vyměňuje informace u svého modifikovaného prostředí;
- stigmergická informace je místní – může být zpřístupněná jen u toho hmyzu, který navštíví místo, ve kterém byla uvolněna (nebo v jeho bezprostředním sousedství).

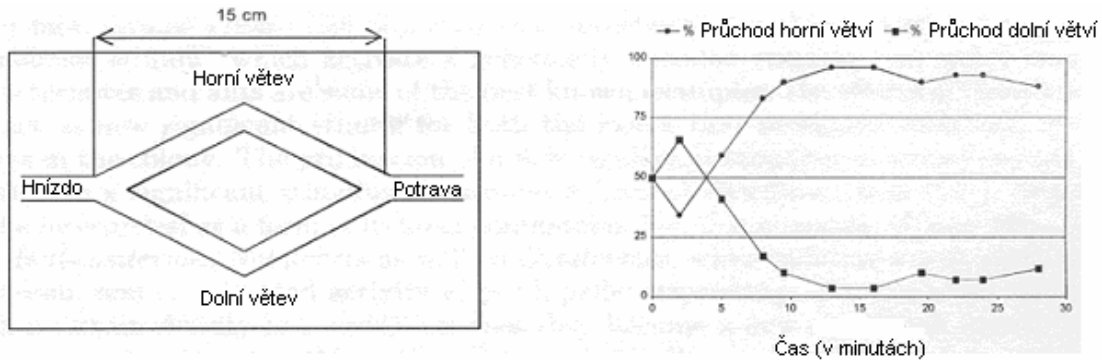
Příklady stigmergie mohou být pozorovány v koloniích mravenců. V mnoha mravenčích druzích mravenci chodí cestou, ke a ze zdroje potravy, uloženou na základě substance nazývané feromon. Jiní mravenci vnímají přítomnost feromonů a přiklánějí se k následující cestě, ve které je vyšší koncentrace feromonu. Prostřednictvím tohoto mechanismu jsou mravenci schopni přepravit potravu do jejich hnízda pozoruhodně účinným způsobem.

Deneubourg a spol.⁵ důkladně zkoumali feromonové ukládání a sledovali chování mravenců v experimentu známém jako „experiment s dvojitým mostem“.

a) Mraveniště oddělené od zdroje potravy dvojitým mostem

⁵ DENEUBOURG, J.L. - ARON, S. - GOSS, S. - PASTEELS, J. M. "The selforganizing exploratory pattern of the Argentine ant." *Journal of Insect Behavio.* vol. 3. p. 159. 1990.

Obě cesty vedoucí z mraveniště ke zdroji potravy jsou naprosto stejně dlouhé a na začátku není žádný feromon na žádné z cest. Přesto po čase jedna cesta, díky náhodným fluktuacím převáží. Mravenci při chůzi totiž zanechávají feromon, a tak větší množství mravenců zanechalo větší množství feromonu, který stimuloval větší množství mravenců k výběru dané cesty.

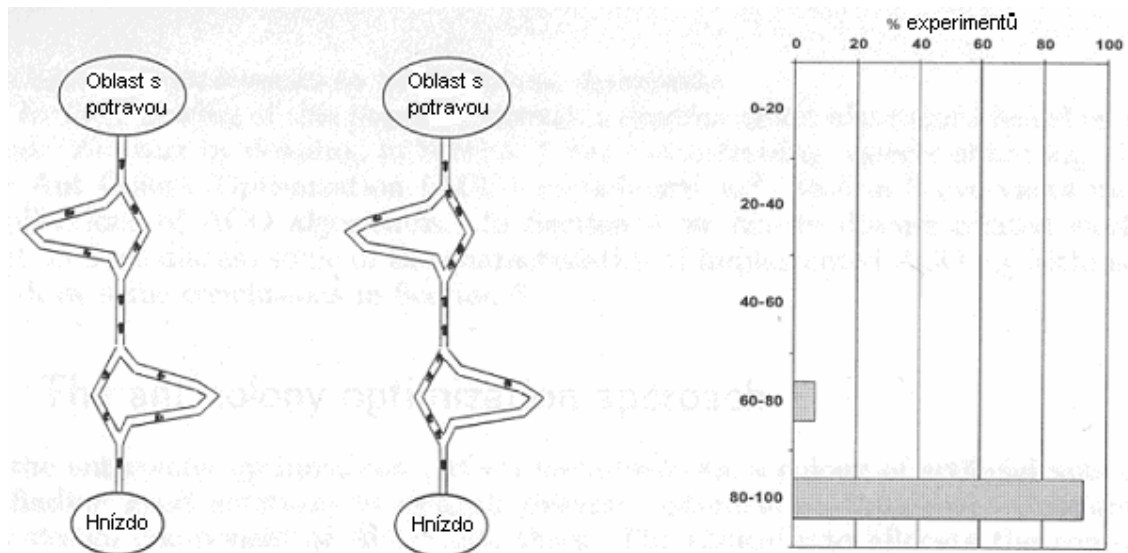


Obrázek 6 - Dvojité mosty od mraveniště ke zdroji potravy (zdroj: [23])

b) Mosty s různě dlouhými variantami

Vliv náhodných fluktuací je značně redukován. Většina mravenců nakonec volí tu nejkratší cestu. Experimenty prokázaly, že šance vybrání kratšího ramene se zvyšuje s poměrem r délek obou ramen. Ovšem v některých případech více mravenců zpočátku vybere delší rameno a to je poté silněji označeno feromonem. Toto indikuje, že systém založený jen na feromonu, nemusí být vždy dostatečně flexibilní. Zvláště v případě, kdy kratší rameno je do systému dodáno až nějaký čas po začátku pokusu. V takovém případě zůstává kolonie stále v delším rameni (je třeba připomenout, že vypařování feromonu je mnohem pomalejší, než byla obvyklá doba pokusů).

Určitý druh mravenců však vykazuje pozoruhodnou vlastnost, která umožňuje výběr kratší cesty, i když je do systému přidána až po určitém čase. Když se mravenec nachází v polovině dlouhého ramene, často si uvědomí, že směřuje téměř kolmo k potřebnému směru, otočí se a zamíří zpět. V tomto případě, kombinace individuální paměti směru a kolektivního sledování označené cesty, umožňuje systematický výběr kratšího ramene. Jinými slovy, kolonie je více flexibilní.



Obrázek 7 - Mosty s různě dlouhými variantami (zdroj: [23])

2.6. ACO algoritmus pro řešení TSP

/ Inicializace */*

For každá hrana (i,j) **do**

$$\tau_{ij}(0) = \tau_0$$

End For

For $k = 1$ **do** m **do**

Umístit mravence k do náhodně vybraného města

End For

Let T^+ je nejkratší od začátku nalezená cesta a L^+ její délka

/ Hlavní smyčka */*

For $t = 1$ **do** t_{\max} **do**

For $k = 1$ **do** m **do**

Postavit cestu $T^k(t)$ aplikací následujících kroků $n - 1$ krát:

If existuje alespoň jedno město $j \in$ seznam kandidátů **potom**

Vybrat příští město $j, j \in J_i^k$, mezi cl městy na seznamu kandidátů následujícím způsobem

$$j = \begin{cases} \arg \max_{j \in J_i^k} \{ [t_{ij}(t)] \times [h_{ij}]^b \} & \text{pro } q \leq q_0 \\ J & \text{pro } q > q_0 \end{cases}$$

kde $J \in J_i^k$ je vybráno podle pravděpodobnosti

$$p_{ij}^k(t) = \frac{[t_{ij}(t)] \times [h_{ij}]^b}{\sum_{J \in J_i^k} [t_{ij}(t)] \times [h_{ij}]^b}$$

a kde i je aktuální město

Else

vybrat nejbližší $j \in J_i^k$

End If

Po každém přesunu mravenec k aplikuje lokální aktualizací pravidlo:

$$\tau_{ij}(t) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_0$$

End For

For $k = 1$ do m **do**

Vypočítat délku $L^k(t)$ cesty $T^k(t)$ produkované mravencem k

End For

If byla nalezena lepší cesta **then**

aktualizovat T^+ a L^+

End If

For každou hranu $(i, j) \in T^+$ **do**

Aktualizovat feromonové stopy aplikací následujícího pravidla:

$$\tau_{ij}(t) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta \tau_{ij}(t) \text{ kde } \Delta \tau_{ij}(t) = 1/L^+$$

End For

For Každou hranu (i, j) **do**

$$\tau_{ij}(t+1) = \tau_{ij}(t)$$

End For

End For

Print nejkratší cestu T^+ a její délku L^+

Stop.

(převzato zdroj: [22])

2.6.1. Vysvětlivky k algoritmu

i, j – jsou dvě různá města

$t_{ij}(0)$ – je počáteční úroveň cesty

k – je mravenec

m – je náhodně vybrané město

T^+ – je nejkratší nalezená cesta (od začátku)

L^+ – je délka T^+

t – je počet iterací

t_{max} – je uživatelem definovaný maximální počet povolených iterací

J_i^k – je množina měst, které má mravenec k navštívit, když je ve městě i

cl – označuje parametr algoritmu

$t_{ij}(t)$ – označuje množství virtuálních feromonových stop na hraně spojující město i s j

h_{ij} – je dohlednost

q – je náhodná veličina s rovnoměrným rozložením $[0,1]$

q_0 – je nastavitelný parametr z intervalu $[0,1]$

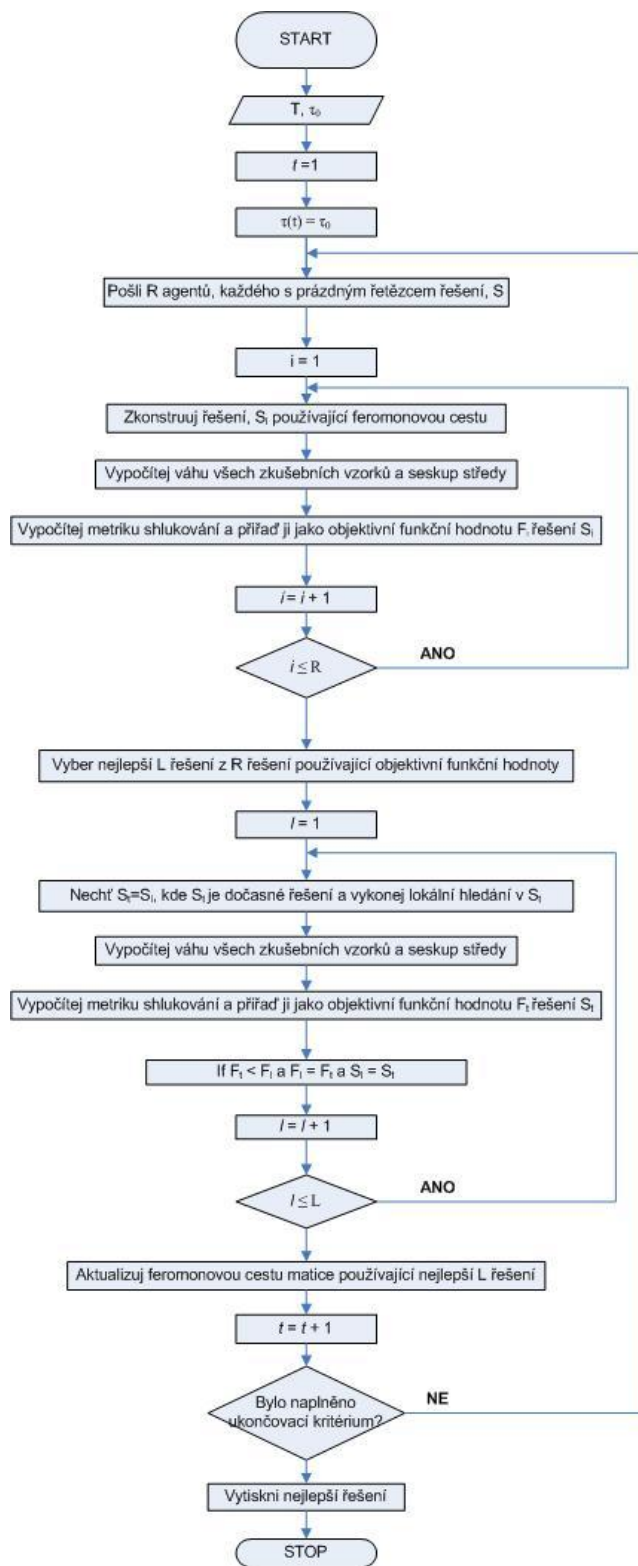
J – je město, které bylo náhodně vybráno podle pravděpodobnosti $p_{ij}^k(t)$

β – je nastavitelný parametr, který reguluje relativní váhu dohlednosti h_{ij}

t_0 – je hodnota stejná jako počáteční hodnota feromonových stop

ρ – je parametr řídící rozklad feromonu

3. Vývojový diagram ACO



Obrázek 8 - Vývojový diagram ACO algoritmu (zdroj: [11])

4. Praktická aplikace

V této části své bakalářské práce uvádím dva různé modely pro řešení úlohy TSP pomocí ACO. Nejedná se však o mou vlastní praktickou aplikaci, ale o přejaté výsledky od dvou různých autorů. Mým cílem je představení těchto výsledků.

První model obsahuje výsledky řešení úlohy TSP pomocí ACO, ale i jiných algoritmů (genetický algoritmus, simulované žíhání, včelí kolonie). Tím nám poskytuje srovnání různých algoritmů a jejich vhodnost k použití na úlohu TSP. Druhý model obsahuje řešení pouze pomocí ACO, ale zahrnuje v sobě použití či nepoužití optimalizace. Oba modely vycházejí z příkladů TSPLIB knihovny.

TSPLIB knihovna je knihovna vzorových příkladů pro různé typy úloh obchodního cestujícího od různých autorů. Pro lidi, kteří se zabývají úlohami spjatými s problémem obchodního cestujícího, je tato knihovna velmi oblíbená, protože lze díky ní snadno testovat různé algoritmy. V současnosti je v TSPLIB knihovně 111 úloh obchodního cestujícího, z toho u 33 je publikované optimální řešení. Pro všechny úlohy lze zjistit na serveru [30] nejlepší dosažené výsledky.

4.1. Modely pro řešení úlohy TSP pomocí ACO i jiných algoritmů

Bylo testováno 7 úloh na počítači DELL Latitude C540 s procesorem INTEL Celeron (1,70 GHz, 256 MB RAM). Pět úloh bylo z knihovny TSPLIB (v souborech swiss42.tsp, eil51.tsp, berlin52.tsp, pr76.tsp, eil101.tsp). Jedna úloha byla zkonstruována na malém počtu měst (v souborech testik_bk5.tsp). Dále byla ještě zkonstruována pomocná úloha do souboru grb17.tsp, která se liší od souboru gr17.tsp z knihovny TSPLIB jen formátem dat. Z původní dolní trojúhelníkové matice (LOW_DIAG_ROW) je předělán na celou matici (FULL_MATRIX). Optimální hodnota uvedená v knihovně TSPLIB se tudíž pro tento problém nemění. Programy byly napsány v jazyce C a v prostředí Visual Studia. [3]

Tabulka 3 - Tabulka úloh obsažených v archivu TSPLIB (převzato zdroj: [3])

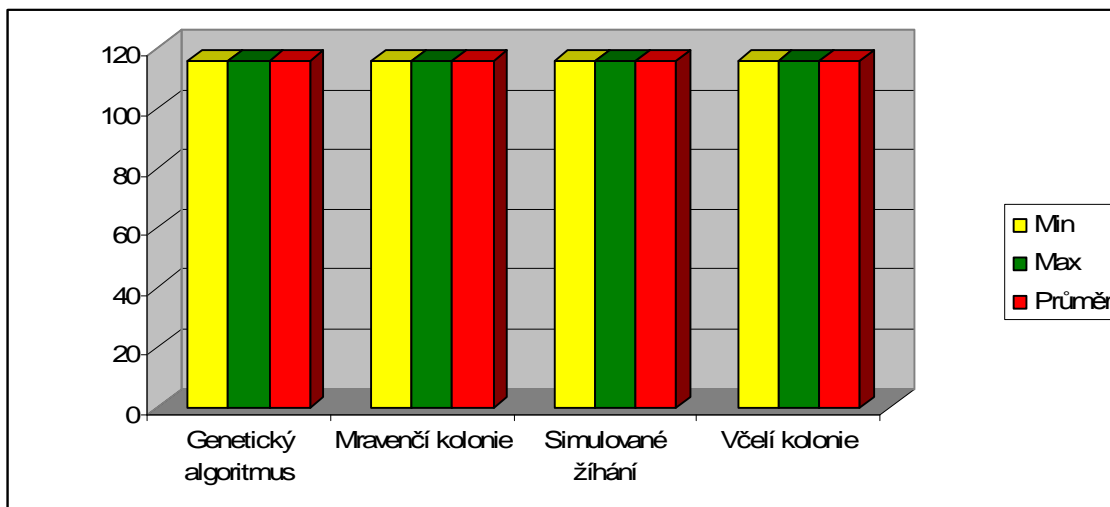
Pořadí	Jméno	Počet měst	Ohraničení výsledku
1	swiss42	42	1273
2	eil51	51	426
3	berlin52	52	7542
4	pr76	76	108159
5	eil101	101	629
6	grb17	17	2085
7	testik_bk5	5	116

§ Optimální délka cesty tohoto problému je 116 (testik_bk5).

Všechny testované algoritmy našly v každém cyklu optimální cestu.

Tabulka 4 - Výsledky algoritmů pro úlohu 116 (převzato zdroj: [3])

Průběh	Genetický algoritmus	Mravenčí kolonie	Simulované žhání	Včelí kolonie
1	116	116	116	116
2	116	116	116	116
3	116	116	116	116
4	116	116	116	116
5	116	116	116	116
6	116	116	116	116
7	116	116	116	116
8	116	116	116	116
9	116	116	116	116
10	116	116	116	116



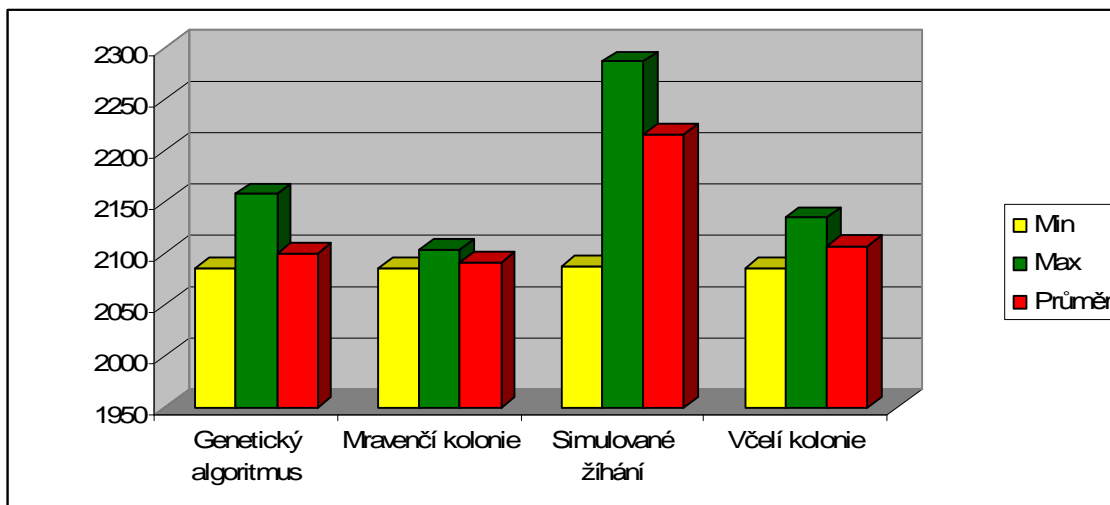
Graf 1 – Porovnání metod pro úlohu 116

§ Optimální délka cesty tohoto problému je 2085 (grb17).

Všechny algoritmy (kromě simulovaného žhání) našly alespoň jednou optimální řešení.

Tabulka 5 - Výsledky algoritmů pro úlohu 2085 (převzato zdroj: [3])

Průběh	Genetický algoritmus	Mravenčí kolonie	Simulované žhání	Včelí kolonie
1	2149	2094	2153	2104
2	2094	2088	2245	2100
3	2158	2096	2088	2094
4	2085	2088	2287	2085
5	2085	2085	2270	2106
6	2085	2104	2210	2098
7	2085	2088	2164	2104
8	2085	2096	2225	2115
9	2085	2085	2286	2135
10	2085	2088	2235	2131



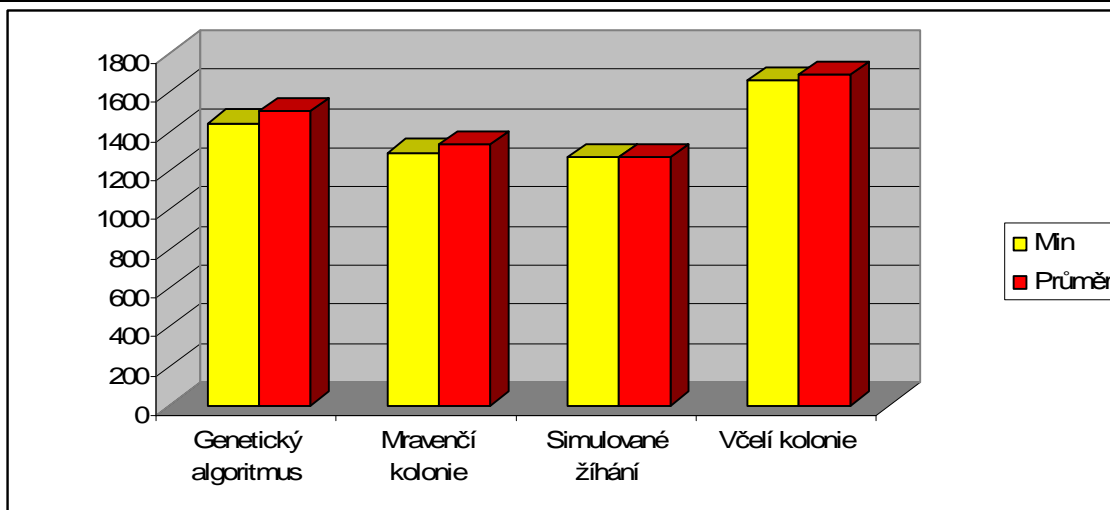
Graf 2 – Porovnání metod pro úlohu 2085

§ Optimální délka cesty tohoto problému je 1273 (swiss42).

Optimální řešení ve všech cyklech našel algoritmus simulovaného žihání a velice úspěšný algoritmus na tuto úlohu jsou mravenčí kolonie.

Tabulka 6 - Min a průměrná hodnota řešení metod na úloze 1273 (převzato zdroj: [3])

	Genetický algoritmus	Mravenčí kolonie	Simulované žihání	Včelí kolonie
Min	1445	1294	1273	1665
Průměr	1506	1335	1273	1694,571



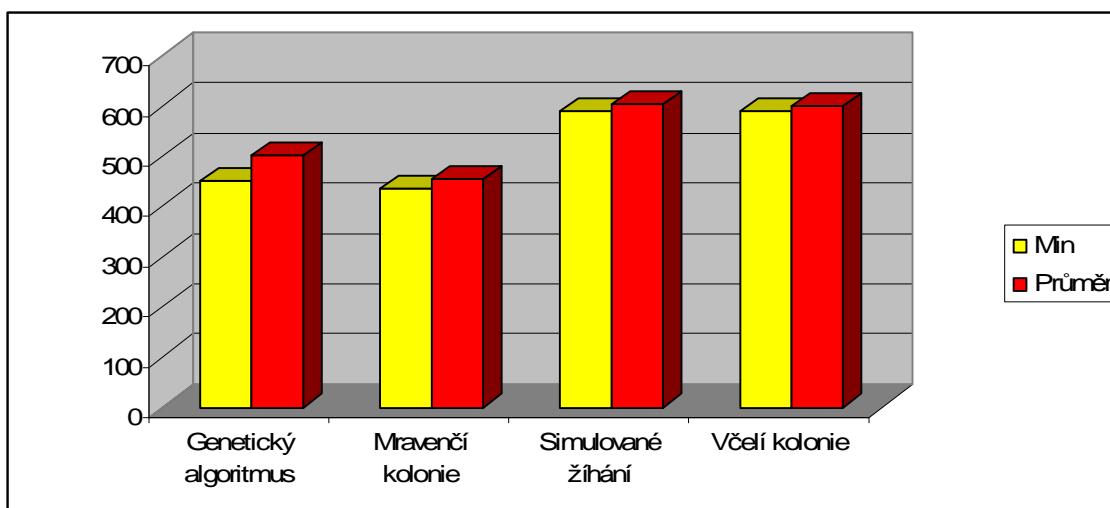
Graf 3 – Porovnání metod pro úlohu 1273

§ Optimální délka cesty tohoto problému je 426 (eil51).

Mravenčí kolonie dosáhly nejlepšího výsledku, poměrně slušné výsledky vykazuje také genetický algoritmus. Algoritmy simulované žihání a včelí kolonie vykazují téměř shodné hodnoty.

Tabulka 7 - Min a průměrná hodnota řešení metod na úloze 426 (převzato zdroj: [3])

	Genetický algoritmus	Mravenčí kolonie	Simulované žihání	Včelí kolonie
Min	450	435	588	588
Průměr	502,4	454,2222	603,5	601,1111



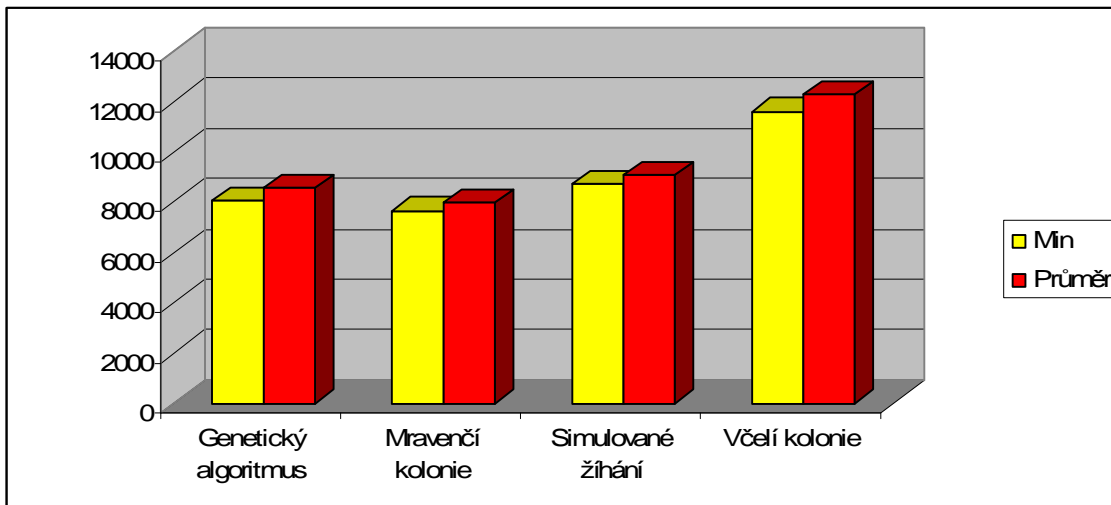
Graf 4 – Porovnání metod pro úlohu 426

§ Optimální délka cesty tohoto problému je 7542 (berlin52).

Je patrné, že mravenčí kolonie jsou nejúspěšnějším algoritmem jak v minimu, tak i v průměru.

Tabulka 8 - Min a průměrná hodnota řešení metod na úloze 7542 (převzato zdroj: [3])

	Genetický algoritmus	Mravenčí kolonie	Simulované žihání	Včelí kolonie
Min	8043	7662	8708	11589
Průměr	8593,8	7962	9064,8	12272,67



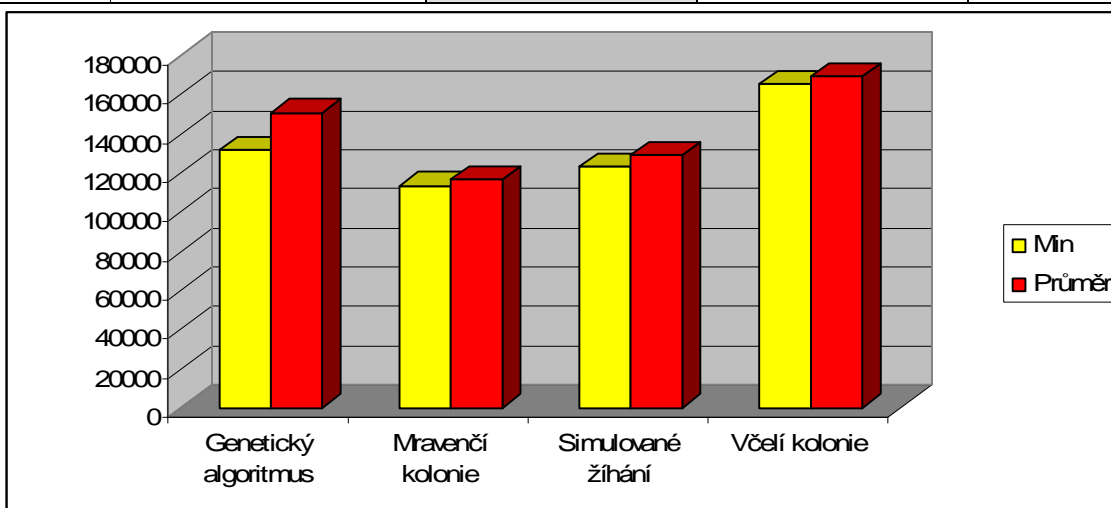
Graf 5 – Porovnání metod pro úlohu 7542

§ Optimální délka cesty tohoto problému je 108159 (pr76).

Nejúspěšnější metodou jsou mravenčí kolonie, nejhůře naopak dopadl algoritmus včelí kolonie.

Tabulka 9 - Min a průměrná hodnota řešení metod na úloze 108159 (převzato zdroj: [3])

	Genetický algoritmus	Mravenčí kolonie	Simulované žihání	Včelí kolonie
Min	131746	113418	123242	165295
Průměr	150937,6	117072,8	129009,2	169155,8



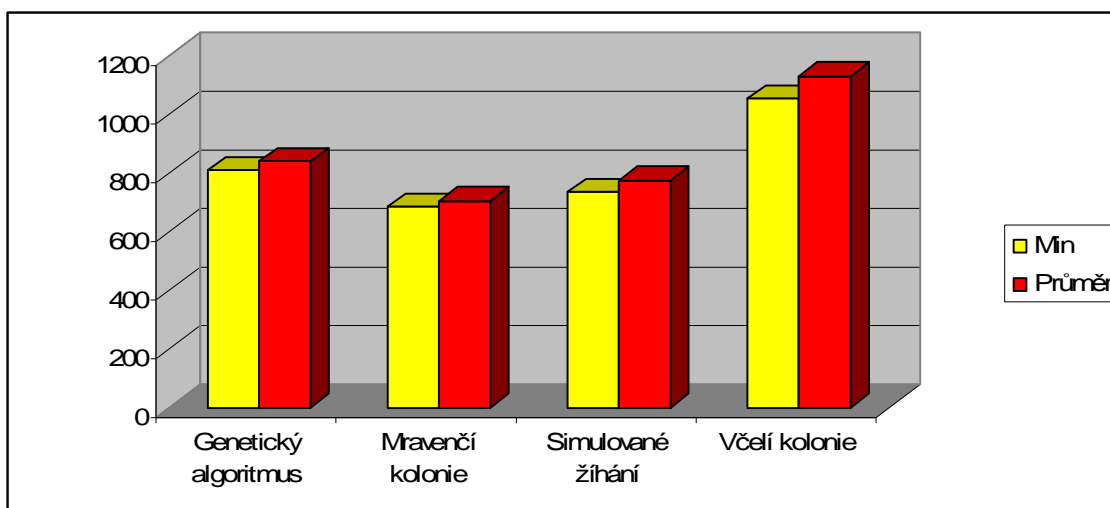
Graf 6 – Porovnání metod pro úlohu 108159

§ Optimální délka cesty tohoto problému je 629 (eil101).

Nejlepších výsledků opět dosahují mravenčí kolonie. Relativně slušné výsledky byly získány simulovaným žiháním a genetickým algoritmem. Včelí kolonie dosahují dvojnásobku optimální délky.

Tabulka 10 - Min a průměrná hodnota řešení metod na úloze 629 (převzato zdroj: [3])

	Genetický algoritmus	Mravenčí kolonie	Simulované žihání	Včelí kolonie
Min	811	684	734	1053
Průměr	839,9	707	774,8	1130,857

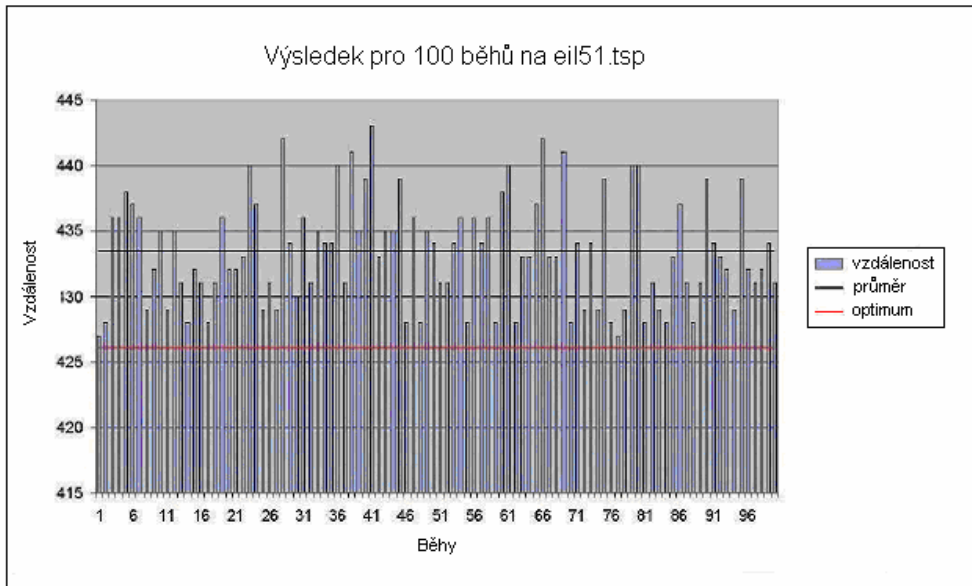


Graf 7 – Porovnání metod pro úlohu 629

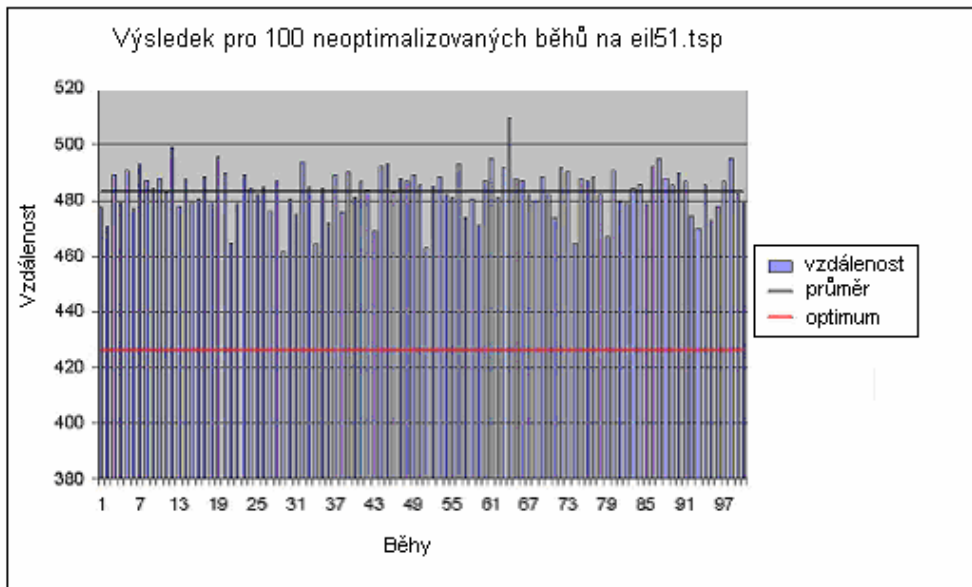
4.2. Model pro řešení TSP pomocí ACO

Úlohy byly testovány na Pentium 4 3,0 GHz s 512 MB RAM a Windows XP. Program byl vytvořen v programovacím jazyce Java.

- § Pro parametry ξ , ρ , α , β , W , Q_0 byly zvoleny následující hodnoty: $\rho = 0.1$, $\alpha = 0.1$, $\beta = 2$, $Q_0 = 0.9$, $\xi = 0.9$, $W = 1$. Příklady použité pro testování byly všechny nalezeny v TSPLIB. Pro testování bylo vždy použito 10 mravenců a 10 iterací.



Graf 8 - Výsledek pro 100 běhů při použití optimalizace (převzato zdroj: [9])



Graf 9 - Výsledek pro 100 běhů bez použití optimalizace (převzato zdroj: [9])

§ Pro toto testování bylo použito 30 mravenců a 300 iterací. Žádný z programů neměl implementován optimalizační algoritmus.

Tabulka 11 - Srovnání výkonu (převzato zdroj: [9])

	Chirocovo řešení ⁶		ACS		Řešení našeho program	
	Čas	Výsledek	Čas	Výsledek	Čas	Výsledek
a280.tsp	423,8	3299	14,328	2811	535,609	3108
berlin52.tsp	26,688	8099	0,703	7748	39	8059
bier127.tsp	109,859	136408	3,313	125555	123,125	131570
ch130.tsp	138,437	7586	3,547	6364	133,36	7129
d1655.tsp			478,234	88119		
d198.tsp	308,984	18765	6,968	17337	280,172	16128
d2103.tsp						
d657.tsp	3434,282	74506	84,766	61329		
eil101.tsp	75,062	747	2,172	654	91,850	748
eil51.tsp	26,359	460	0,656	433	28,515	469
eil76.tsp	48,203	603	1,328	547	58,328	615
fl1400.tsp			332,609	24212		
gil262.tsp	368,766	3388	15,047	2640	460,188	2823
kroA100.tsp	75,969	25326	2,156	21585	104,672	25420
kroA150.tsp	160,281	34686	4,641	27602	208	31610
pr1002.tsp			164,375	335872		
pr2392.tsp						
u724.tsp			100,719	55783		
usual3509.tsp						

⁶ CHIRICO, Ugo. *A java framework for ant colony systems*. Technical report. Siemens Informatica S.p.A. 2004.

§ Pro tento test bylo použito jen 10 mravenců a iterace. Optimalizace by byly nahrazeny pro nízkou kvalitu počátečního výsledku z TSP algoritmu.

Tabulka 12 – Zlepšení, které vychází z používání a nepoužívání optimalizace (převzato zdroj: [9])

	Bez optimalizace		S optimalizací		Zlepšení	
	Čas	Výsledek	Čas	Výsledek	Čas	Výsledek
a280.tsp	535,609	3108	12,578	2675	97,65%	13,93%
berlin52.tsp	39	8059	0,422	7902	98,92%	1,95%
bier127.tsp	123,125	131570	2,125	122440	98,27%	6,94%
ch130.tsp	133,36	7129	2,203	6349	98,35%	10,94%
d198.tsp	280,172	16128	5,797	16017	97,93%	0,69%
d657.tsp			98,89	51678		
eil101.tsp	91,859	748	1,468	652	98,40%	12,83%
eil51.tsp	28,515	469	0,406	428	98,58%	8,74%
eil76.tsp	58,328	615	0,782	543	98,66%	11,71%
gil262.tsp	460,188	2823	9,266	2518	97,99%	10,80%
kroA100.tsp	104,672	25420	1,375	21959	98,69%	13,62%
kroA150.tsp	208	31610	3,11	28098	98,50%	11,11%

ZÁVĚR

Ve své bakalářské práci jsem se zabývala řešením problému obchodního cestujícího pomocí metody mravenčí kolonie. Cílem této práce bylo charakterizovat problém obchodního cestujícího a metodu mravenčí kolonie za pomoci poznatků a již známých výsledků konkrétních autorů.

V první části práce jsem popsala historii, různé varianty a modifikace problému obchodního cestujícího. V další fázi jsem přešla k relativně mladé metodě mravenčí kolonie. Tu jsem důkladně charakterizovala od jejího počátku až k nynějšímu stavu této metody. Především jsem svou pozornost zaměřila na biologickou inspiraci této metody, na živé mravence (z biologického hlediska) a experimenty s nimi. Ale také na vytvořené, umělé mravence.

V závěrečné části práce jsem uvedla příklad algoritmu a vývojového diagramu, který se zabývá problémem obchodního cestujícího řešeného pomocí metody mravenčí kolonie. Poté jsem představila dva různé modely řešení již zmiňovaných úloh. První model obsahuje řešení pomocí různých algoritmů (genetický algoritmus, mravenčí kolonie, simulované žíhání, včelí kolonie). Druhý model obsahuje řešení pouze pomocí metody mravenčí kolonie, ale byl modifikován použitím či nepoužitím optimalizace.

Na základě mé práce jsem dospěla k závěru, že k řešení problému obchodního cestujícího je použití metody mravenčí kolonie velice vhodné.

Věřím, že se mi povedlo vytvořit ucelený přehled o problému obchodního cestujícího a metody mravenčí kolonie a také nastínění možných řešení těchto úloh.

LITERATURA

- [1] DEMEL, Jiří. *Grafy a jejich aplikace*. Academia. Praha, 2002. ISBN 80-200-0990-6.
- [2] NEŠETŘIL, Jaroslav. *Teorie grafů*. 1. vyd. Praha: SNTL, 1979. 316 s.
- [3] KUDELOVÁ, Blanka. *Srovnání efektivnosti algoritmů pro řešení úloh obchodního cestujícího*. Praha, 2007. 79 s. Diplomová práce na Fakultě informatiky a statistiky Vysoké školy ekonomické hlavní specializace Ekonometrie a operační výzkum. Vedoucí diplomové práce Prof. Ing. Josef Jablonský, CSc.
- [4] MÁLEK, Richard. *Alternation of Meta-heuristic Algorithms with Runtime Analysis and Parameter Adaptation*. Prag, 2007. 50 s. Diploma thesis on Czech Technical University in Prague on Faculty of Electrical Engineering. Thesis Supervisor Ing. Pavel Kordík, Ph.D.
- [5] MANIEZZO, V. – GAMBARDELLA, M. L. – DE LUIGI, F. Ant Colony Optimization. *New Optimization Techniques in Engineering*, 2004, by Onwubolu, G. C., and B. V. Babu, Springer-Verlag Berlin Heidelberg, s. 101-117.
- [6] DOROTO, M. – GAMBARDELLA, M. L. Ant colonies for the traveling salesman problem. *Tech. Rep. No. IRIDIA 96-13*, 1996, IRIDIA, Université Libre de Bruxelles, Belgium.
- [7] DORIGO, M. – MANIEZZO, V. – COLONI, A. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics–Part B*, 1996, vol. 26, no.1, s. 1-13.
- [8] NĚMEC, Miloš. *Optimalizace pomocí mravenčích kolonií*. Praha, 2006. 53 s. Diplomová práce na Fakultě elektrotechnické ČVUT v Praze na katedře kybernetiky. Vedoucí diplomové práce Ing. Petr Pošík.
- [9] NGASSA, Jean-Luc – KIERKEGAARD, Jakob. *ACO and TSP*. Denmark, 2007. 116 s. Bachelor of Computer Science on Roskilde University. Supervisor Keld Helsing.
- [10] JIŘÍK, Přemysl. *Vizualizace běhu simulovaného ochlazování*. Praha, 2007. 47 s., 14 s. příloh. Diplomová práce na Fakultě elektrotechnické ČVUT na katedře počítačů. Vedoucí diplomové práce Ing. Petr Fišer.
- [11] SHELOKAR, V.K. – JAYARAMAN, V.K. – KULKARNI, B.D. *An Ant Colony Approach for Clustering*. *Analytica Chimica Acta* 509, 187–195, 2004.
- [12] WINSTON, W. L. – VENKATARAMANAN, M. *Introduction to Mathematical Programming: Operations Research*. Brooks/Cole-Thomson Learning. California 2003.

ELEKTRONICKÉ ZDROJE

- [13] *Řešení TSP pomocí ACO* [online]. cit. [26-11-2007]. Dostupné z: <<http://www.milosnemec.cz/clanek.php?75>>.
- [14] *TSP History Home* [online]. cit. [22-04-2008]. Dostupné z: <<http://www.tsp.gatech.edu/index.html>>.
- [15] *DNA Computing* [online]. cit. [22-04-2008]. Dostupné z: <<http://www.fit.vutbr.cz/~sekanina/ehw/kantor/index.html.en>>.
- [16] *a-tsp_approx.pdf* [online]. cit. [21-04-2008]. Dostupné z: <www.cs.vsb.cz/kot/soubory_animaci/a-tsp_approx.pdf>.
- [17] *Icosian_game.jpeg* [online]. cit. [21-04-2008]. Dostupné z: <http://www.groups.dcs.st-andrews.ac.uk/~history/Diagrams/Icosian_game.jpeg>.
- [18] *graf.jpg* [online]. cit. [21-04-2008]. Dostupné z: <<http://www.fit.vutbr.cz/~sekanina/ehw/kantor/pct/graf.jpg>>.
- [19] *Seznam Encyklopedie - Problém obchodního cestujícího* [online]. cit. [15-04-2008]. Dostupné z: <<http://encyklopedie.seznam.cz/heslo/145366-problem-obchodniho-cestujiciho>>.
- [20] *Seznam Encyklopedie - Eusocialita* [online]. cit. [12-04-2008]. Dostupné z: <<http://encyklopedie.seznam.cz/heslo/528342-socialni-hmyz>>.
- [21] *Seznam Encyklopedie - Mravencovité* [online]. cit. [08-04-2008]. Dostupné z: <<http://encyklopedie.seznam.cz/heslo/139764-mravencoviti>>.
- [22] *Mravenčí kolonie: Zevrubný popis ACO algoritmu pro řešení TSP* [online]. cit. [25-11-2007]. Dostupné z: <<http://www.milosnemec.cz/clanek.php?76>>.
- [23] *Jiří Kubalík: Optimalizační algoritmy inspirované chováním mravenců* [online]. [cit. 23-04-2008]. Dostupné z: <http://labe.felk.cvut.cz/~kubalik/kopr/kopr_ants.pdf>.
- [24] *Ant colonies* [online]. cit. [25-11-2007]. Dostupné z: <<http://www.antcolonies.net/>>.
- [25] *UNM Computer Science* [online]. cit. [25-11-2007]. Dostupné z: <<http://www.cs.unm.edu/kwiley/artificialLife/antColony.html>>.
- [26] *Genetic and Ant Colony Optimization Algorithms - The Code Project* [online]. cit. [25-11-2007]. Dostupné z: <<http://www.codeproject.com/cpp/GeneticandAntAlgorithms.asp>>.
- [27] *Salesman-119438019289930.PNG* [online]. cit. [15-04-2008]. Dostupné z: <<http://i.iinfo.cz/urs/Salesman-119438019289930.PNG>>.

- [28] *Mravenčí kolonie – Biologická inspirace* [online]. cit. [25-11-2007]. Dostupné z: <<http://www.milosnemecek.cz/clanek.php?69>>.
- [29] *ACO Publications* [online]. cit. [24-04-2008]. Dostupné z: <<http://iridia.ulb.ac.be/~mdorigo/ACO/publications.html>>.
- [30] *Travelling Salesman Problem* [online]. cit. [20-04-2007]. Dostupné z: <<http://www.iwr.uni-heidelberg.de/groups/comopt/software/TPSLIB95>>.

SEZNAM OBRÁZKŮ

Obrázek 1 - Problém obchodního cestujícího (zdroj: [16]).....	4
Obrázek 2 - Hierarchie tříd složitosti (zdroj: [10])	6
Obrázek 3 - Příklad hamiltonovské cesty (zdroj: [18]).....	8
Obrázek 4 - Zvyšování pravděpodobnosti kratší cesty u metody ACO (zdroj: [10]).....	12
Obrázek 5 - Chování živých mravenců (zdroj: [6])	17
Obrázek 6 - Dvojité mosty od mraveniště ke zdroji potravy (zdroj: [23]).....	24
Obrázek 7 - Mosty s různě dlouhými variantami (zdroj: [23]).....	25
Obrázek 8 - Vývojový diagram ACO algoritmu (zdroj: [11]).....	28

SEZNAM TABULEK

Tabulka 1 - Počet uzlů versus počet cest.....	5
Tabulka 2 - Neúplný seznam úspěšných ACO algoritmů (v chronologickém pořadí).....	12
Tabulka 3 - Tabulka úloh obsažených v archivu TSPLIB (převzato zdroj: [3])	30
Tabulka 4 - Výsledky algoritmů pro úlohu 116 (převzato zdroj: [3]).....	30
Tabulka 5 - Výsledky algoritmů pro úlohu 2085 (převzato zdroj: [3]).....	31
Tabulka 6 - Min a průměrná hodnota řešení metod na úloze 1273 (převzato zdroj: [3])	32
Tabulka 7 - Min a průměrná hodnota řešení metod na úloze 426 (převzato zdroj: [3])	33
Tabulka 8 - Min a průměrná hodnota řešení metod na úloze 7542 (převzato zdroj: [3])	33
Tabulka 9 - Min a průměrná hodnota řešení metod na úloze 108159 (převzato zdroj: [3]).....	34
Tabulka 10 - Min a průměrná hodnota řešení metod na úloze 629 (převzato zdroj: [3])	35
Tabulka 11 - Srovnání výkonu (převzato zdroj: [9])	37
Tabulka 12 – Zlepšení, které vychází z používání a nepoužívání optimalizace (převzato zdroj: [9])	38

SEZNAM GRAFŮ

Graf 1 – Porovnání metod pro úlohu 116.....	31
Graf 2 – Porovnání metod pro úlohu 2085.....	32
Graf 3 – Porovnání metod pro úlohu 1273.....	32
Graf 4 – Porovnání metod pro úlohu 426.....	33
Graf 5 – Porovnání metod pro úlohu 7542.....	34
Graf 6 – Porovnání metod pro úlohu 108159.....	34
Graf 7 – Porovnání metod pro úlohu 629.....	35
Graf 8 - Výsledek pro 100 běhů při použití optimalizace (převzato zdroj: [9]).....	36
Graf 9 - Výsledek pro 100 běhů bez použití optimalizace (převzato zdroj: [9]).....	36