

**Vedecká rada Fakulty elektrotechniky a informatiky
Slovenskej technickej univerzity v Bratislave**

Ing. Matej Cíba

**Návrh optimalizačných algoritmov na báze simulovania
kolónie mravcov**

Autoreferát dizertačnej práce

Na získanie vedecko-akademickej hodnosti
philosophiae doctor

v odbore doktorandského štúdia
9.2.7 Kybernetika

Bratislava, 2015

**Vedecká rada Fakulty elektrotechniky a informatiky
Slovenskej technickej univerzity v Bratislave**

Ing. Matej Cíba

**Návrh optimalizačných algoritmov na báze simulovania
kolónie mravcov**

Autoreferát dizertačnej práce

Na získanie vedecko-akademickej hodnosti
philosophiae doctor

v odbore doktorandského štúdia
9.2.7 Kybernetika

Bratislava, 2015

Dizertačná práca bola vypracovaná v externej forme doktorandského štúdia na Ústave robotiky a kybernetiky Fakulty elektrotechniky a informatiky Slovenskej technickej univerzity v Bratislave.

Predkladateľ: Ing. Mate Cíba
Ústav robotiky a kybernetiky
Fakulty elektrotechniky a informatiky, STU v Bratislave
Ilkovičova 3, 812 19 Bratislava

Školiteľ: doc. Ing. Ivan Sekaj, PhD.
Ústav robotiky a kybernetiky
Fakulty elektrotechniky a informatiky, STU v Bratislave
Ilkovičova 3, 812 19 Bratislava

Oponenti: prof. Ing. Vladimír Kvasnička, DrSc.
Ústav aplikovanej informatiky
Fakulty informatiky a informačných technológií, STU v Bratislave
Ilkovičova 2, 842 16 Bratislava

prof. Ing. Pavel Ošmera, CSc.
Fakulta strojního inženýrství
Vysoké učení technické v Brně
Antonínská 1, 601 90 Brno

Autoreferát bol rozoslaný dňa:

Obhajoba dizertačnej práce sa koná: pred komisiou pre obhajobu dizertačnej práce v odbore doktorandského štúdia, vymenovanou predsedom spoločnej odborovej komisie dňa číslo odboru: 9.2.7, odbor doktorandského štúdia: Kybernetika na Ústave robotiky a kybernetiky Fakulty elektrotechniky a informatiky Slovenskej technickej univerzity v Bratislave, Ilkovičova 3, 812 19 Bratislava

Prof. RNDr. Gabriel Juhás, PhD.
dekan Fakulty elektrotechniky a informatiky
Slovenskej technickej univerzity v Bratislave

Obsah

1 Úvod	1
2 Tézy dizertačnej práce	1
3 Úvod do ACO	2
3.1 Spávanie mravcov pri hľadaní potravy	2
3.2 Ant system	3
4 Reinicializácia feromónu	4
4.1 Akumulácia feromónu, minimálna a atraktívna cesta	4
4.2 Reinicializácia feromónu	6
4.3 ACO algoritmus s reinicializáciou	9
4.3.2 ACO s variabilným makro cyklom	9
4.3.1 Prípadová štúdia - hľadanie minimálnej cesty	10
4.3.3 Prípadová štúdia - hľadanie maximálnej cesty	12
4.3.4 Vplyv dĺžky makro cyklu	13
4.3.5 Vplyv démona	14
4.3.6 Porovnanie algoritmov	15
4.4 Záver pre ACO s reinicializáciou	15
5 Hybridné ACO algoritmy	16
5.1 Adaptácia GO pre ACO	16
5.2 ACO algoritmus s GO	18
5.3 Prípadová štúdia	19
5.3.1 Výsledky pre 10 mravcov	19
5.3.2 Vykonanie GO v opačnom poradí	21
5.3.1 Výsledky pre 50 mravcov	22
5.4 Genetické operácie a časové nároky	24
5.4.1 Oprávnenie genetických operácií	25
5.5 Záver pre ACO s GO	25
6 Záver	26
6.1 Reinicializácia v ACO	26
6.2 Genetické operácie v ACO	26
7 Literatúra	27
7.1 Okrem prác autora	27
7.2 Publikované práce autora	28

1 Úvod

Práca sa venuje zvýšeniu výkonnosti a efektivity ACO (Ant Colony Optimization) algoritmov. Tie sú založené na pozorovaní prírodných kolónii mravcov. Mravce sú spoločenské tvory. Kolónia aj napriek jednoduchosti jedincov dokáže vykonať komplexné úlohy ďaleko presahujúce možnosti jednotlivého mravca. Príkladom sú hľadanie potravy, deľba práce, zaraďovanie znášky, koordinovaný transport a spoločná obrana. To je možné dosiahnuť iba pri vysoko organizovaných jedincoch. Koordinované správanie v mravčích kolóniách upútalo pozornosť mnohých výskumníkov a viedlo k vytvoreniu novej triedy algoritmov. Spomeňme práce Goss s kolegami (1989) a experiment na dvojitom moste, Deneubourg s kolegami (1990) a stochastický opis dynamiky mravčej kolónie pri hľadaní potravy, a nakoniec Dorigo s kolegami (1991) a diskrétny model ako aj prvý funkčný mravčí algoritmus Ant system (AS).

K nevýhodám ACO algoritmu patrí množstvo užívateľských parametrov, nízka diverzita populácie a vysoká citlivosť stagnácie v lokálnom optime od hodnôt vstupných parametrov. Zatiaľ čo prvá nevýhoda je vlastnosťou algoritmu, druhej a tretej bolo venovaných mnoho prác. Napríklad Stützle a Hoos (2000) obmedzili feromónové hodnoty v algoritme Max-Min ant system (MMAS), Nakamichi a Arita (2001) implementovali náhodný výber, Kumar s kolegami (2003) vytvorili mechanizmy zabráňujúce stagnácii a rýchlej konvergencii, alebo Dorigo a Stützle (2004) s *pseudonáhodným proporcionálnym* pravidlom výberu v Ant Colony systéme (ACS). Problematike vplyvu rôznych ACO parametrov na správanie algoritmu sa venujú Becker a Szczerbicka (2005).

Jadro práce je rozdelené do štyroch hlavných častí, ktoré tvoria kapitoly dva až päť. V druhej kapitole v niekoľkých bodoch uvádzame, čo je cieľom dizertačnej práce. Tretia časť opisuje prírodné princípy ACO algoritmu, jeho hlavné mechanizmy a ich aplikáciu vo výpočtovej praxi. Štvrtá kapitola sa stručne venuje reinicializácii. Nechýba porovnanie so základným algoritmom na jednoduchom príklade. Piata časť obsahuje možnosti aplikácie evolučných techník na ACO algoritmus. Prináša rozšírenia a modifikácie, ako aj ich porovnanie s jedným z najvýkonnejších ACO algoritmov na konkrétnom príklade.

2 Témy dizertačnej práce

Prác, ktoré sa venujú tejto problematike je viacero a v súčasnosti existuje niekoľko verzií ACO algoritmov, ktoré majú svoje výhody aj nevýhody. My sme sa rozhodli venovať tejto problematike, analyzovať ich nedostatky a navrhnúť nové možnosti zdokonalenia týchto metód, respektíve navrhnúť možnosti kombinácie prístupov ACO s evolučnými algoritmami.

Na základe analýzy súčasného stavu v oblasti ACO algoritmov a evolučných algoritmov resp. iných bio-inšpirovaných výpočtových metód boli definované nasledovné tézy dizertačnej práce:

- 1) Analýza možností nárastu výkonnosti ACO algoritmov.
- 2) Návrh metód zvyšujúcich efektivitu ACO na báze zvyšovania diverzity pri prehľadávaní priestoru potenciálnych riešení.
- 3) Návrh nových metód a algoritmov, ktoré kombinujú prednosti ACO algoritmov s inými typmi prehľadávacích a optimalizačných algoritmov, ako sú napríklad evolučné algoritmy resp. iné bio-inšpirované prístupy.

- 4) Overenie navrhnutých metód na vhodných praktických príkladoch.

3 Úvod do ACO

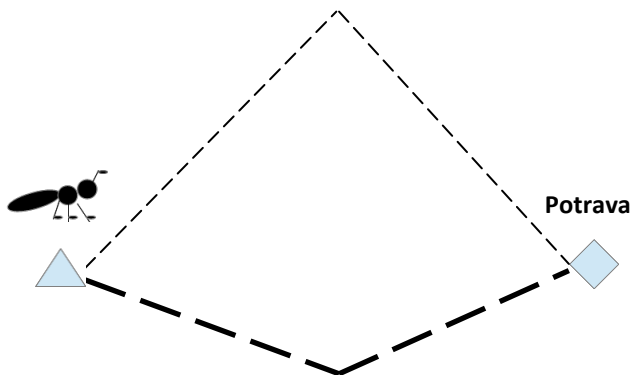
Mravčí algoritmus (Ant algorithm) ktorý prvý krát predstavil Marco Dorigo vo svojej PhD práci (1992) bol Single ACO (S-ACO). Základ práce tvorí Deneubourgov stochastický model (Deneubourg et al., 1990) popisujúci dynamiku kolónie mravcov. Jedná sa o multiagentový prístup pre riešenie obtiažnych kombinatorických problémov, ktoré sú reprezentované jednoduchým matematickým grafom.

Koordinované konanie mravcov je z pohľadu umelej inteligencie založené na kolektívnom správaní decentralizovaného, samo organizovaného systému (Beni, Wang, 1989). Takýto systém pozostáva z populácie jednoduchých agentov, ktorý vzájomne pôsobia sami na seba a na svoje okolie. Vzájomné ovplyvňovanie vedie ku vzniku spoločného správania.

Interakcia s prostredím sa vykonáva nepriamou (tzv. stigmergickou) formou komunikácie, pri ktorej mravce do prostredia vylučujú pachovú stopu nazývanú feromón. Pachová stopa predstavuje parameter priradený hranám grafu, ktorého hodnoty menia mravce počas riešenia. Proces riešenia je stochastický a je ovplyvnený feromónovým modelom.

3.1 Spávanie mravcov pri hľadaní potravy

Pri hľadaní potravy mravce používajú stopovací feromón na označenie prejdenej cesty. Mravce vedú tento feromón vnímať a nasledovať. Pri výbere cesty sú náchylné pravdepodobnostne zvoliť cestu s vyššou koncentráciou feromónu.



Obrázok 3.1: Experimentálna zostava mostu s dvoma vetvami (Goss et al., 1989).

Majme príklad kolónie a zdroja potravy spojených dvoma cestami (Obrázok 3.1). Na začiatku cesty neobsahujú feromón, takže sa obe zdajú mravcom rovnaké. Tie si náhodne volia jednu z ciest, ktorou sa vydajú. Mravce, ktoré si zvolili kratšiu cestu, sa začnú po nej skôr vracaf. V čase ich návratu do mraveniska bude kratšia cesta obsahovať až dvojnásobné množstvo feromónu. Vďaka tomu bude cesta atraktívnejšia. Mravce, ktoré sa po nej znova vydajú za potravou alebo späť vylúčia viac feromónu a kratšia cesta ich bude ešte viac lákať. Z experimentu (Goss et al., 1989) je

vidieť, že samoorganizácia pri hľadaní potravy je výsledkom autokatalytického procesu (kladnej spätnej väzby), v ktorej prospech pôsobí rozdielna dĺžka ciest.

3.2 Ant system

Ak uvažujeme zložitejší graf, mravce budú vytvárať slučky v ktorých sa hromadí feromón. Ten bude priťahovať ďalšie mravce, ktoré v nich ostanú uväznené. Pri takomto rozložení feromónu nebudú kratšie cesty preferované. Je to dôsledok príliš jednoduchého agenta bez priestorovej orientácie. Dorigo (1992) vyriešil problém pomocou umelého mravca, ktorý si pamätá cestu a dĺžku prejdenej hrán. Umelý mravec má dva režimy, medzi ktorými sa prepína pri nájdení zdroja potravy. V doprednom (stochastickom) režime mravec vytvára riešenie pomocou náhodnej voľby ďalšieho vrcholu. Pred spätným (deterministickým) pohybom sa z cesty eliminujú slučky čím sa zabráni ich tvorbe. Mravec na spätnočnej ceste vylúči feromón. Jeho množstvo závisí od dĺžky nájdenej cesty, čím sa mravce nasmerujú k lepším riešeniam a urýchli sa konvergencia.

Súvislý graf G je dvojprvková množina $G = (N, A)$ vrcholov $n = |N|$ a hrán $a = |A|$, ktoré spájajú vrcholy. Vrcholy, medzi ktorými hľadáme minimálnu cestu nazývame zdrojový a cieľový. Ku každej hrane a_{ij} priradíme premennú τ_{ij} , ktorú nazveme *umelá feromónová stopa*. Na začiatku má každá hrana rovnaké nenulové množstvo feromónu τ_0

$$\tau_{ij}(0) = \tau_0 ; \tau_0 > 0 \text{ pre } \forall a_{ij} \in |A| \quad (3.1)$$

S hranami grafu sú spojené aj heuristické hodnoty η_{ij} , ktoré predstavujú užívateľsky definovaný parameter. Heuristické hodnoty slúžia na nasmerovanie algoritmu do perspektívnych oblastí na začiatku vyhľadávacieho procesu. Ich veľkosť zväčša zodpovedá prevrátenej hodnote dĺžky hrán a_{ij}

$$\eta_{ij} = 1/d_{ij} \quad (3.2).$$

Mravec v i -tom uzle číta feromónovú stopu $\tau_{ij}(t)$ a heuristickú informáciu η_{ij} z ktorých určí mieru pravdepodobnosti voľby ďalšieho vrcholu:

$$a_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} \quad \forall j \in N_i^k \quad (3.3)$$

kde N_i^k sú susedné vrcholy vrcholu i , okrem predchádzajúceho vrcholu. Takto sa mravce vyhnú návratu do predchádzajúceho vrcholu okrem prípadu, keď je množina N_i^k prázdna. Parametre α a β určujú váhu množstva feromónu a heuristickej hodnoty. Pravdepodobnosť, s akou si k -ty mravec z uzla i vyberie cestu do uzla j ($j \in N_i^k$) v iterácii t bude

$$p_{ij}^k(t) = \frac{a_{ij}^k(t)}{\sum_{l \in N_i^k} a_{il}^k(t)} \quad (3.4).$$

Mravec opakovane prechádza z vrcholu do vrcholu až pokiaľ nedosiahne cieľ (potravu). Pred návratom eliminuje z cesty všetky slučky. Pri spätnom pohybe k -ty mravec prechodom cez hranu a_{ij} zmení množstvo stopovacieho feromónu τ_{ij} medzi vrcholmi i a j nasledovne:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta\tau^k(t) \quad (3.5)$$

Vo všeobecnosti požadujeme, aby množstvo vylúčeného feromónu $\Delta\tau^k$ bolo nerastúcou funkciou dĺžky cesty

$$\Delta\tau_{ij}^k = \begin{cases} \frac{1}{L^k(t)} & \text{ak } (i, j) \in T^k(t) \\ 0 & \text{ak } (i, j) \notin T^k(t) \end{cases} \quad (3.6)$$

kde $T^k(t)$ je cesta k -teho mravca a $L^k(t)$ je jej dĺžka. Potom, čo každý mravec dokončí svoju cestu sa časť feromónu vypari

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t); \quad \rho \in (0,1) \quad (3.7)$$

kde $\rho \in (0,1)$ je koeficient životnosti feromónu ($1 - \rho$ je koeficient rozpadu feromónu). Vyparovanie redukuje vplyv stôp zanechaných v počiatočných fázach, kedy mravce nájdu menej kvalitné riešenia, čím sa umožní prieskum nových ciest. V praxi sa vyparovanie a aktualizácia feromónu vykonáva súčasne:

$$\tau_{ij}(t+1) = (1 - \rho) \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (3.8)$$

pre všetky mravce $\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$, kde m je počet mravcov (konštanta).

ACO algoritmus bol prispôsobený pre riešenie širokého záberu úloh, ktoré sa dajú transformovať do podoby grafu. Súčasný aplikácie zahŕňajú rôzne diskretné optimalizačné problémy, medzi inými kvadratický priradovací problém (QAP, Maniezzo, Colorni a Dorigo, 1994), problém obchodného cestujúceho (TSP, Dorigo, Maniezzo a Colorni, 1991a,b), problém rozvozu delenej dodávky (VRP, Hartl a Strauss, 1999a,b), problém poradia sekvencií (SOP, Gambardella a Dorigo, 1997), problém ofarbenia grafu (GCP, Costa a Hertz, 1997), trasovanie v komunikačných sieťach (Schoonderwoerd, Holland, Bruten a Rothkrantz, 1996).

4 Reinicializácia feromónu

4.1 Akumulácia feromónu, minimálna a atraktívna cesta

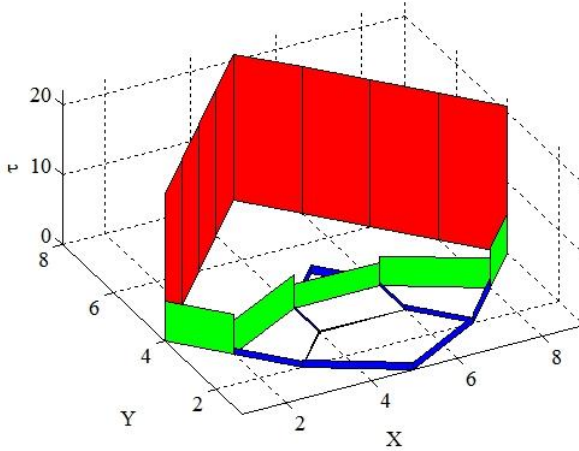
Myšlienkou reinicializácie sa už zaoberali autori Stützle a Hoos (1997a, 1997b). Vo svojej práci *MAX-MIN Ant system (MMAS)* obmedzili feromónové hodnoty na interval $[\tau_{\min}, \tau_{\max}]$. Avšak feromón má svoje prirodzené limitné hodnoty, ktorých teoretické maximum bude

$$\tau_{ij}(\infty) = m \frac{Q}{L^{opt}} \frac{1}{\rho} \quad (4.1),$$

kde L^{opt} je globálne optimálna cesta. Praktické maximum však bude závisieť od cesty L^+ s najvyšším množstvom feromónu a priemerného počtu mravcov m^+ , ktorý ju používajú (Cíba, 2011)

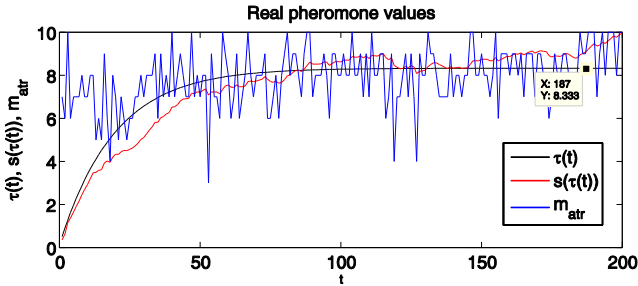
$$\tau_{ij}(\infty) = m^+ \frac{Q}{L^+} \frac{1}{\rho} \quad (4.2).$$

Pozrime sa na skutočné správanie mravcov pri simulácii algoritmom ACO_{KTS} (Kumar et al., 2003) na príklade dvojitého mosta (Obrázok 4.1).



Obrázok 4.1: Graf rozšíreného dvojitého mosta (Dorigo and Stützle, 2004) so znázorneným množstvom feromónu na jednotlivých hranách po skončení simulácie.

Počas simulácie bola nájdená najkratšia cesta (zelená), avšak cesta, ktorá má najviac feromónu (najatraktívnejšia cesta), je odlišná (červená).



Obrázok 4.2: Vývoj reálnych feromónových hodnôt pri simulácii ACO_{KTS} algoritmu.

Pre opis vývoja reálnych feromónových hodnôt boli zvolené nasledovné metriky:

$$\bar{\tau}(t) = \frac{\sum_{k=1}^m \tau_{ij}^k(t)}{m}$$

– Aritmetický priemer feromónových hodnôt

$$- \text{Štandardná odchýlka feromónových hodnôt } \sigma(t) = \sqrt{\frac{1}{m-1} \sum_{k=1}^m (\tau_{ij}^k(t) - \bar{\tau}(t))^2}$$

- Počet mravcov generujúcich najatraktívnejšiu cestu m^+

Z časového priebehu simulácie (Obrázok 4.2) možno rozoznať tri vývojové fázy prehľadávacieho procesu: úvodnú, saturačnú a stagnačnú (Cíba, 2012).

Úvodná fáza je charakteristická rýchlym nárastom feromónových hodnôt $\bar{\tau}(t)$ a štandardnej odchýlky $\sigma(t)$ (Obrázok 4.2, $0 < t < 40$). Rýchly nárast štandardnej odchýlky $\sigma(t)$ prezrádza nerovnomernú distribúciu feromónu na hranách grafu.

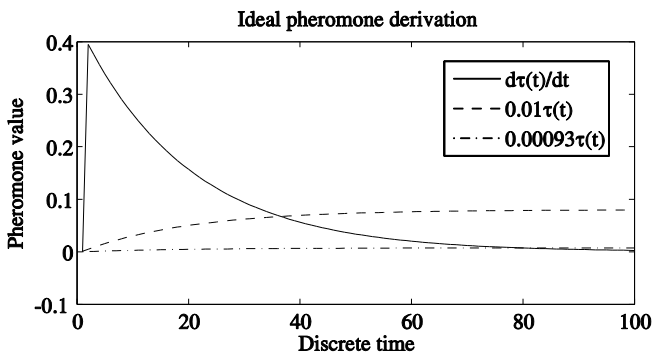
Saturačná fáza je charakteristická pozvoľným nárastom feromónových hodnôt $\tau_{ij}(t)$ aj ich štandardnej odchýlky $\sigma(t)$, čo súvisí s postupným nasycovaním voľnej kapacity hrán preferovanej cesty až do hodnoty τ_{max} podľa (4.2) (Obrázok 4.2, $40 < t < 80$).

V *stagnačnej* fáze je vyparovanie feromónu v rovnováhe s jeho akumuláciou, priemerná hodnota feromónu $\bar{\tau}(t)$ už nerastie (Obrázok 4.2, $t > 80$). Avšak štandardná odchýlka feromónových hodnôt $\sigma(t)$ a počet mravcov m^+ vytvárajúcich najatraktívnejšiu cestu narastajú. To je spôsobené ďalšou konsolidáciou, pri ktorej značná časť mravcov generuje to isté riešenie. Počet mravcov generujúcich najatraktívnejšiu cestu je nižší ako počet všetkých mravcov. Je to spôsobené mechanizmom zabraňujúcim stagnácii, ktorý posilňuje vplyv náhodného výberu pri voľbe nasledujúceho vrcholu v neskorších štádiách prehľadávacieho procesu (Obrázok 4.2, $t > 80$).

4.2 Reinicializácia feromónu

Vysoké feromónové hodnoty na niektorých hranách grafu prilákajú ďalšiu generáciu mravcov a algoritmus je uväznený v lokálnom optime. Algoritmus sa môže vyhnúť stagnačnej fáze reinicializáciou feromónových hodnôt. Spúšťač reinicializácie, ktorý musí rozpoznať stagnáciu, nemôže byť založený na vzťahu 4.2, nakoľko nevieme dopredu určiť počet mravcov tvoriacich najatraktívnejšiu cestu a ani jej dĺžku. V práci Cíba (2012) je navrhnuté opätovné spustenie prehľadávacieho procesu v užívateľsky nastaviteľnom bode x , v ktorom sa hodnota časovej derivácie feromónu zhoduje s aritmetickým priemerom feromónových hodnôt (Obrázok 4.3):

$$x\bar{\tau}(t) = \frac{d\bar{\tau}(t)}{dt} \quad (4.3).$$

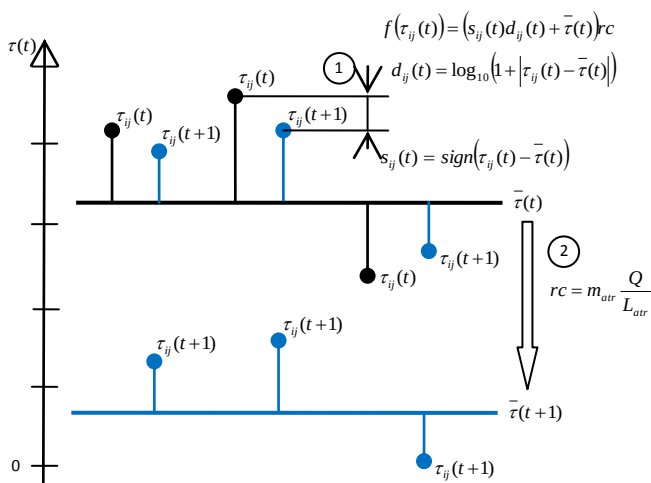


Obrázok 4.3: Načasovanie reinitializácie pre rôzne hodnoty užívateľského parametra x (Cíba, 2012).

Pri reinitializácii sa vykonajú dve operácie. Najskôr sa znížia rozdiely feromónových hodnôt medzi hranami. Najatraktívnejšie riešenie bude menej príťažlivé a mravce ho nebudú nasledovať, tým sa podporí prieskum nových ciest. Požadujeme, aby zníženie bolo priamo úmerné veľkosti rozdielu a zároveň sa musí zachovať poradie jednotlivých feromónových hodnôt. Druhá operácia spočíva v znížení celkového množstva feromónu. Prehľadávací proces tak opustí saturačnú alebo stagnačnú fázu a bude pokračovať v novej počiatkovej fáze. Odobraté množstvo feromónu zároveň uvoľní kapacitu pre nový feromón. Nové feromónové hodnoty sú získané pomocou nelineárnej transformácie

$$\begin{aligned}\tau_{ij}(t+1) &= (s_{ij}(t)d_{ij}(t) + \bar{\tau}(t))rc \\ s_{ij}(t) &= \text{sign}(\tau_{ij}(t) - \bar{\tau}(t)) \\ d_{ij}(t) &= \log_{10}\left(1 + \left|\tau_{ij}(t) - \bar{\tau}(t)\right|\right)\end{aligned}\tag{4.4}.$$

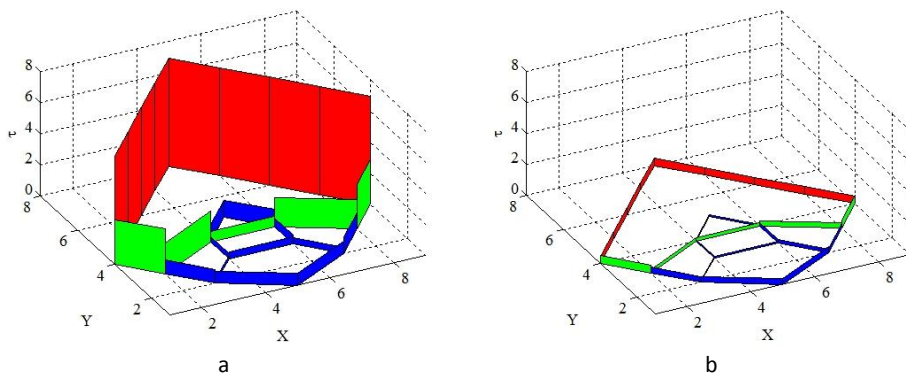
Oba kroky je možné názorne vidieť na Obrázku 4.4.



Obrázok 4.4: Reinicializácia feromónu pozostáva z dvoch krokov: (1) zníženie rozdielov feromónových hodnôt medzi jednotlivými hranami a (2) zníženie celkového množstva feromónu (Cíba, 2012).

Pre zníženie množstva feromónu používame redukčný koeficient rc . Pri stanovení tejto hodnoty je dôležité nájsť rovnováhu medzi uvoľnením kapacity pre ďalšiu akumuláciu a stratou už nadobudnutých vedomostí.

Reinicializáciou je vyhľadávací proces rozdelený na menší počet makro cyklov, pričom tieto nie sú úplne nezávislé. Nasledujúci makro cyklus začína s malým množstvom feromónu, ktorý predstavuje nadobudnuté vedomosti z predošlého makro cyklu.



Obrázok 4.5 : Príklad rozloženia feromónu pred (a) a po (b) reinicializácii.

Príklad rozloženia feromónových hodnôt pred a po reinicializácii je vidieť na Obrázku 4.5 a,b.

4.3 ACO algoritmus s reinicializáciou

Uvedený mechanizmus reinicializácie bol aplikovaný na ACO_{KTS} (Kumar et al., 2003) a nesie názov po makro cykloch ako ACO_{MC} (Cíba, 2012). Prehľadavací proces je reinicializovaný medzi úvodnou a saturačnou fázou. Hodnota užívateľského parametra x (4.3) bola empiricky určená $x = 0.01$. Redukčný koeficient rc (4.4) je priamo úmerný rýchlosti akumulácie feromónu (4.5)

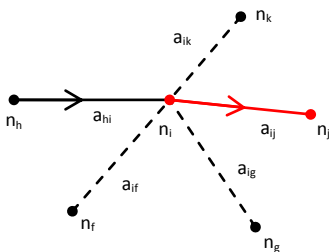
$$rc = m^+(t_r) \frac{Q}{L^+(t_r)}; \quad Q = 1 \quad (4.5)$$

kde t_r je čas (,resp. iterácia) reinicializácie. Hodnoty parametra q_0 v mechanizme zabraňujúcom rýchlej konvergencii sú rovnomerne rozdelené na intervale pravdepodobnosti $\langle p_{\min}, p_{\max} \rangle$, pričom tieto hodnoty sú rastúce a každá z nich pripadá na iný makro cyklus (4.6)

$$q_0 \in \left(q_{01}, q_{02}, q_{03}, \dots | q_{0i+1} - q_{0i} = \Delta; \quad \Delta = \frac{P_{\max} - P_{\min}}{N_{mc} - 1} \right) \quad (4.6)$$

Kde N_{mc} je počet makro cyklov. Po každej iterácii, keď všetky mravce aktualizujú svoju cestu, démon prechádza doteraz najlepším nájdeným riešením a v každom uzle n_i kontroluje, či na jeho hranách je aspoň o 20% feromónu viac v porovnaní s ostatnými odchádzajúcimi hranami (Obrázok 4.6). Ak nie, pridá dodatočné množstvo feromónu $\Delta\tau_{ij}^{bs}(t)$ na nasledujúcu hranu doteraz najlepšieho nájdeného riešenia (4.7):

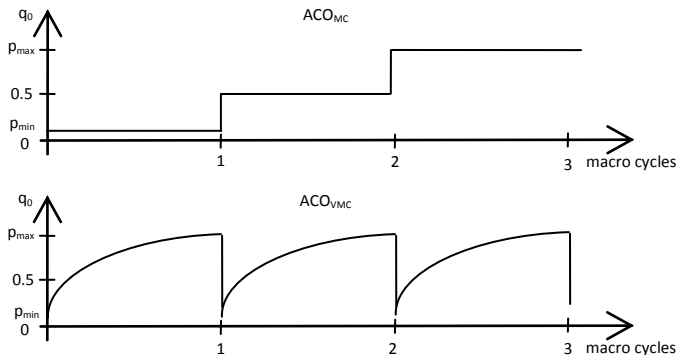
$$\Delta\tau_{ij}^{bs}(t) = \begin{cases} 1.2 \max(\tau_{ik}(t)) - \tau_{ij}^{bs}(t) & \text{if } \tau_{ij}^{bs}(t) < 1.2 \max(\tau_{ik}(t)) \\ 0 & \text{otherwise} \end{cases} \quad (4.7).$$



Obrázok 4.6: Okolie vrcholu n_i s hranami doteraz najlepšieho nájdeného riešenia (plná čiara).

4.3.2 ACO s variabilným makro cyklom

Ďalšia modifikácia ACO_{MC} spočíva v hodnote q_0 parametra, ktorý nebude konštantný, ale bude sa plynule meniť v nastavenom rozsahu pravdepodobnosti $\langle p_{\min}, p_{\max} \rangle$ počas každého makro cyklu (Obrázok 4.7).



Obrázok 4.7: Rozdiel medzi ACO_{MC} and ACO_{VMC} je premenlivá hodnota q_0 parametra (Cíba, Sekaj, 2012a).

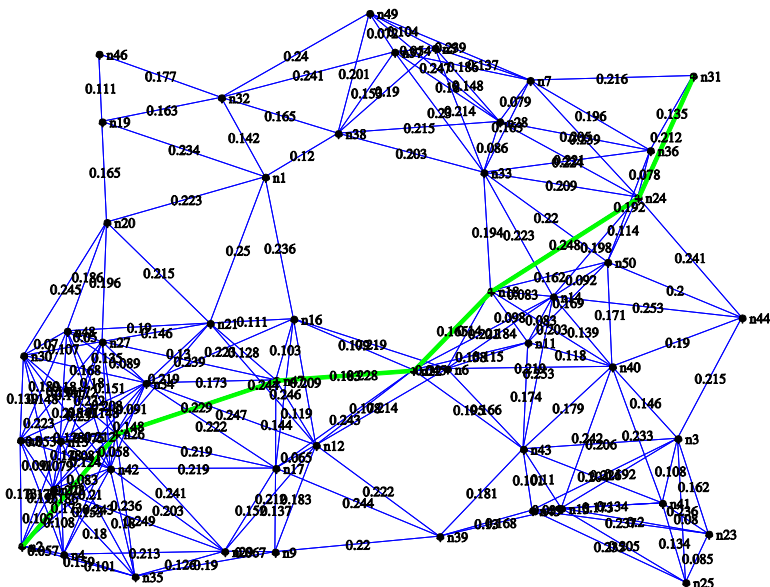
Názov algoritmu je odvodený od premenlivej hodnoty parametra q_0 ako ACO s variabilným makro cyklom (ACO_{VMC} ; Cíba, Sekaj, 2012a). Riešenie je založené na diferencií (4.3) a bude blízke nule na začiatku a blízko jednej na konci každého makro cyklu (4.8)

$$q_0(t) = 1 - k \left(\frac{d\bar{\tau}(t)}{dt} - 0.01\bar{\tau}(t) \right); \quad k = \frac{1}{d\bar{\tau}(t_r + 1)/dt} \quad (4.8)$$

kde t_r je iterácia reinitializácie. Mechanizmus imituje správanie pôvodného parametra q_0 (Kumar et al., 2003).

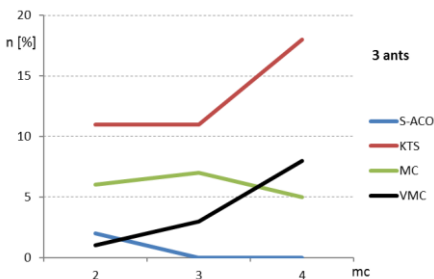
4.3.1 Prípadová štúdia - hľadanie minimálnej cesty

Algoritmus s reinitializáciou bol porovnaný s S-ACO (Dorigo, 1992) a ACO_{KTS} (Kumar et al., 2003) na náhodne vygenerovanom symetrickom multigrafe o 50 uzloch a 200 hranách (Obrázok 4.8). Hodnoty hrán c_{ij} sú Euklidovské vzdialenosti medzi príslušnými uzlami. Heuristicke hodnoty η_{ij} majú veľkosť 1. Úlohou je nájsť najkratšiu cestu medzi počiatočným $n_s = 2$ a koncovým $n_e = 31$ vrcholom. ACO parametre sú nastavené podľa práce Becker a Szczerbicka (2005). Interval pravdepodobnosti $\langle p_{min}, p_{max} \rangle$ pre hodnoty q_0 je nastavený v rozmedzí $\langle 0.1, 0.9 \rangle$.

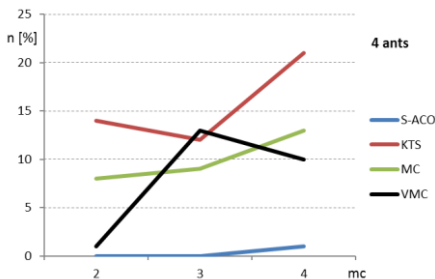


Obrázok 4.8: Náhodne generovaný graf so zobrazením najkratšej nájdenej cesty (zelená).

Test bol vykonaný pre rôzne počty mravcov (od 3 do 6) a rôzne počty reinicializácií (od 2 do 4), pričom pre každé nastavenie bolo vykonaných 100 opakovaní. Najskôr bol spustený ACO_{MC} (pre príslušný počet makro cyklov) a podľa priemerného počtu iterácií bola nastavená ukončovacia podmienka pre S-ACO a ACO_{KTS} . Pre účely porovnania bola použitá pravdepodobnosť nájdenia najkratšej známej cesty n [%]. Výsledky sú uvedené na Obrázku 4.9.



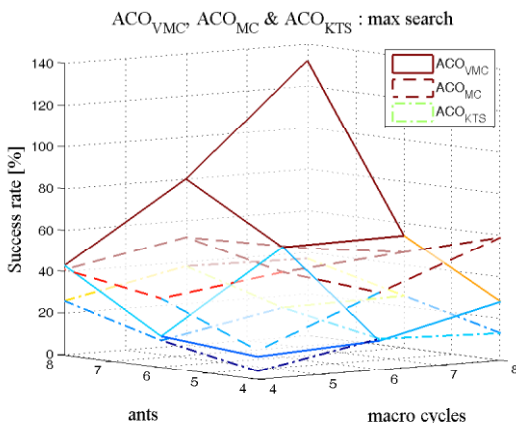
a



b

ACO zdroje boli nastavené v rozmedzí 4 až 8 mravcov a 4 až 8 makro cyklov. Hodnoty ostatných ACO parametrov sú podľa Becker a Szczerbicka (2005). Heuristicke hodnoty boli zhodné s váhami hrán.

Výsledky ukazujú, že obe verzie s reinicializáciou prekonalí ACO_{KTS} v pravdepodobnosti nájdenia globálneho optima. Ukázalo sa, že verzia s variabilným makro cyklom bola schopná lepšie zúžitkovať rastúce zdroje v porovnaní s ACO_{MC} , a pre viac zdrojov (mravcov alebo makro cyklov) sa nárast zvyšoval až na dvojnásobok (Cíba, Sekaj, 2012b).



Obrázok 4.11: Porovnanie algoritmov pri hľadaní maximálnej cesty.

Tieto povzbudivé výsledky viedli k analýze a hľadaniu možných rezerv algoritmu s reinicializáciou.

V poslednej práci (Cíba, Sekaj, 2013a) boli vykonané ďalšie úpravy a porovnania. Metodika, graf a parametre sú rovnaké ako v 4.3.1 len s tým rozdielom, že heuristicke hodnoty sú rovné prevrátenej hodnote Euklidovských vzdialeností

$$\eta_{ij} = \frac{1}{c_{ij}} \quad (4.9)$$

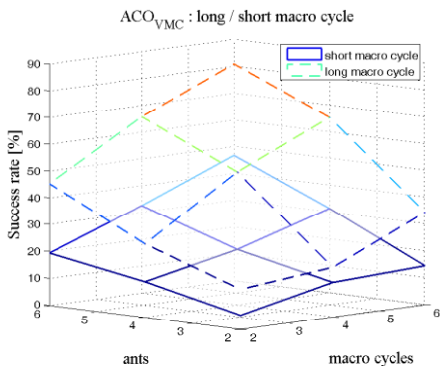
a pre každé nastavenie bolo vykonaných 500 opakovaní. Pre vyhodnotenie boli použité okrem (i) pravdepodobnosti nájdenia globálneho optima ($T^* = [2 \ 26 \ 47 \ 22 \ 18 \ 24 \ 31]$) n [%] aj (ii) priemerná hodnota ukončovacej iterácie $\bar{c}_t$. Premenné pre počet mravcov a makro cyklov sú uvedené na príslušných osiach grafov (4.12 až 4.14).

4.3.4 Vplyv dĺžky makro cyklu

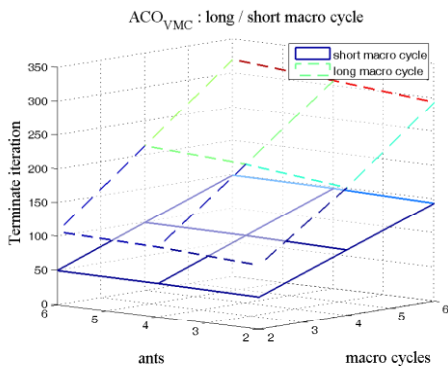
Makro cyklus pôvodného algoritmu zahŕňa iba úvodnú fázu prehľadacieho procesu (Obrázok 4.3). V tomto prípade bude rozšírený o saturačnú fázu zmenou hodnoty parametra x vo vzťahu 4.3 na

$$0.00093\bar{r}(t) = \frac{d\bar{r}(t)}{dt} \quad (4.10),$$

kde hodnota 0.00093 bola získaná empiricky pre ideálnu akumuláciu feromónu.



a: Pravdepodobnosť nájdenia globálneho optima.



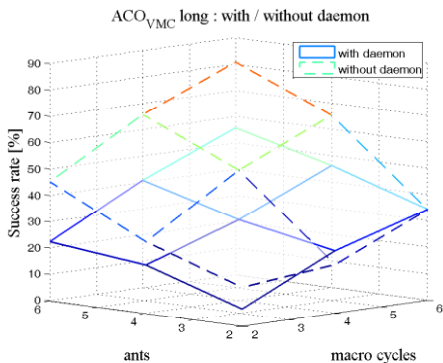
b: Dĺžka prehľadávacieho procesu.

Obrázok 4.12: Porovnanie verzii ACO s reinicializáciou pre rôzne dĺžky makro cyklu.

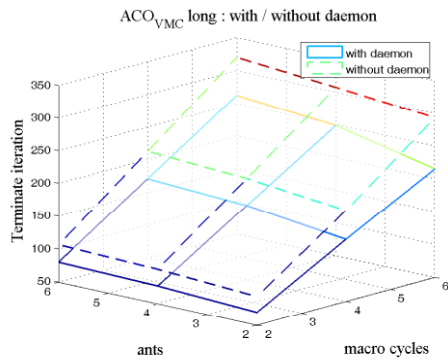
Dĺžka prehľadávacieho procesu s makro cyklom rozšíreným o saturačnú fázu (4.10) je dvojnásobná (Obrázok 4.12 b), avšak algoritmus má viac než dvojnásobnú pravdepodobnosť nájdenia globálneho optima vo väčšine prípadov (Obrázok 4.12 a). Keďže ACO_{VMC} s dlhým makro cyklom je výkonnejší, v ďalšom porovnaní bude použitý tento variant.

4.3.5 Vplyv démona

Démon založený na elitárskej stratégii aktualizuje feromón na hranách doteraz najlepšieho nájdeného riešenia. V tomto teste bude démon potlačený, aby sme mohli vyhodnotiť jeho dopad na výkon algoritmu.



a: Pravdepodobnosť nájdenia globálneho optima



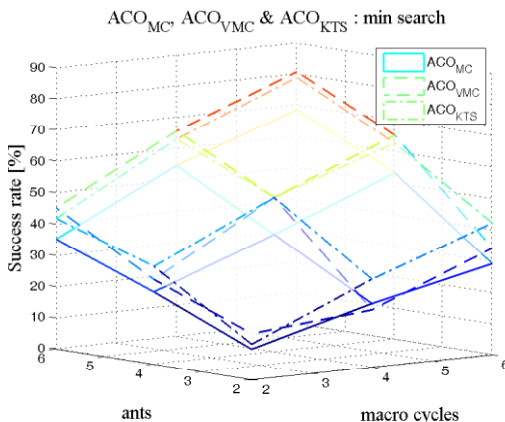
b: Dĺžka prehľadávacieho procesu

Obrázok 4.13: Porovnanie verzii ACO s reinicializáciou s a bez démona.

Démon navyšovaním hodnôt feromónu spôsobuje skoršiu saturáciu a tým skracuje prehľadávací proces (Obrázok 4.13 b). Naviac elitárska stratégia zvyšuje selekčný tlak a algoritmus častejšie uviazne v lokálnom optime (Obrázok 4.13 a). Keďže démon má negatívny dopad na výkon algoritmu, v ďalšom teste bude použitá verzia ACO_{VMC} bez démona a s dlhým makro cyklom.

4.3.6 Porovnanie algoritmov

ACO_{VMC} and ACO_{MC} s dlhým makro cyklom bez démona sú porovnané s jedným z najvýkonnejších ACO algoritmov ACO_{KTS} (Kumar et al., 2003).



Tabuľka 4.14: ACO_{KTS} , ACO_{MC} a ACO_{VMC} s dlhým makro cyklom bez démona.

ACO s reinicializáciou vykazuje vo väčšine prípadov vyššiu pravdepodobnosť nájdenia globálneho optima oproti ACO_{KTS} , rozdiel však nie je vysoký (Obrázok 4.14).

4.4 Záver pre ACO s reinicializáciou

Na vzorovom príklade bolo ukázané, že algoritmus s reinicializáciou je výkonovo porovnateľný s jedným z najvýkonnejších ACO algoritmov od Kumar et al. (2003). Algoritmus s reinicializáciou prenáša už nadobudnuté vedomosti do nového makro cyklu, ale zároveň umožňuje aj ďalšiu akumuláciu feromónových hodnôt.

Výhoda nižšieho selekčného tlaku ACO_{VMC} sa ukázala na príklade s grafom so 62 uzlami (Obrázok 4.9), ktorý vedie chamtivý algoritmus preč z globálneho optima. V tomto prípade bol výkon algoritmov s reinicializáciou oveľa lepší.

Tým, že vstupný parameter algoritmu nie je počet iterácií ale počet makro cyklov, pri jeho odhade nie je potrebné zohľadniť rýchlosť akumulácie (závislej na počte mravcov a veľkosti problému). Zvyčajne postačia 3-4 reinicializácie.

Vďaka preddefinovanej ukončovacej podmienke je algoritmus vhodný pre optimalizáciu rôzne veľkých problémov v automatickom režime. Vďaka tomu, že algoritmus nepracuje v saturačnej zóne, má nižšie nároky na procesorový čas.

5 Hybridné ACO algoritmy

Aktuálny výskum sa orientuje na hybridné algoritmy kombinujúce rôzne prístupy. Ich účelom je návrh vhodnej kombinácie, ktorá potlačí nevýhody samostatných algoritmov. Príklad hybridného algoritmu je kombinácia simulovaného žihania a zakázaného hľadania od Chiang a Russell (1995) alebo ACO a Genetický algoritmus (GA) od Reimann et al. (2001). Zameriame sa práve na možnosť kombinácie ACO s GA, ktorú skúmalo niekoľko autorov, avšak každý s odlišným prístupom.

Reimann et al. (2001) využili paralelný ACO-GA prístup, kde po preddefinovanom počte iterácií môže nastať výmena. V prípade, že bolo prekonané globálne najlepšie riešenie, sa toto prenesie do druhého algoritmu. Porovnanie na VRP problému však ukázalo, že nebol veľký rozdiel medzi ACO a hybridným prístupom.

Hajimirsadeghi et al. (2008) čiastočne náhodne reinitializovali upravený *MMAS*. Počas reinitializácie *GO* vytvoria nové počiatočné riešenia z dvoch rodičov (globálne a iteračne najlepšia cesta). Test na MS7 (Multiplicative Square, súčinnové magické štvorce) s 50 a na MS8 so 68 mravcami ukázal, že variant s genetickou reinitializáciou bol výkonnejší ako pôvodný *MMAS* algoritmus.

Li et al. (2011) navrhli hybridný GA-ACO algoritmus ktorý má eliminovať slabšie lokálne prehľadávanie GA pomocou ACO. Každý z algoritmov pracuje po preddefinovaný počet iterácií. Ako prvý sa spúšťa GA. Využíva turnajový výber, jednobodové kríženie s testom uskutočniteľnosti (na 20% populácie) a inverznú mutáciu (na 10% populácie). Výsledok GA je vstupom pre ACO algoritmus, ktorý je variant *MMAS* s obmedzením feromónových hodnôt na intervale $T_{\min}$ a $T_{\max}$. GA-ACO prístup bol porovnaný so samostatnými GA a ACO algoritmami na VRP s 20 zákazníkmi. Autori uvádzajú, že hybridný prístup bol schopný nájsť kratšie riešenie s nižším počtom iterácií, avšak počet mravcov ani veľkosť populácie či už pri hybridnom, alebo pri samostatných algoritmoch nie je uvedený.

Aravindh and Michael (2012) navrhli sériový ACO-GA variant pre hľadanie minimálnej cesty v meniacom sa prostredí komunikačných sietí. Každý z algoritmov sa spúšťa iba jedenkrát. Najskôr ACO algoritmus generuje uskutočniteľné riešenia, ktoré potom slúžia ako počiatočná populácia pre GA. Autori v práci neuvádzajú žiadny konkrétny príklad ani porovnanie s iným algoritmom.

Na rozdiel od predošlých prác, ktoré pristupujú k tvorbe hybridného algoritmu pomocou kombinácie samostatných algoritmov sme zvolili prístup, v ktorom využívame len kombináciu vybraných operácií. V našej práci (Cíba, Sekaj, 2013b) aplikujeme dodatočné operácie na mravcami generovaných riešeniach na konci každej iterácie. Pridané operácie sú adaptované z Genetických algoritmov. Týmto prístupom sa snažíme prekonať problém stagnácie ACO algoritmov spôsobený príliš vysokým selekčným tlakom.

5.1 Adaptácia *GO* pre ACO

Genetický algoritmus predstavil Holland (1975) známy ako jednoduchý genetický algoritmus (SGA). Štyri hlavné komponenty GA sú: (i) reprezentácia, (ii) mutácia, (iii) kríženie a (iv) výber (Sekaj, 2005). Všetky komponenty boli prispôsobené pre ACO algoritmus tak, aby produkovali iba uskutočniteľné riešenia.

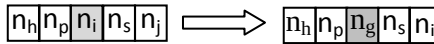
V ACO algoritme je *reprezentácia genotypu* (i) také poradie vrcholov grafu, ktoré definuje

uskutočniteľnú cestu medzi dvoma danými vrcholmi:

$$n^L; \quad n \in N \quad (5.1)$$

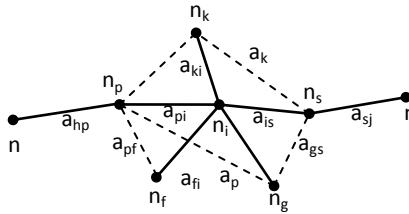
kde *gény* n predstavujú vrcholy grafu a L dĺžku cesty. Pôvodný GA pracuje s konštantnou dĺžkou genómu (reťazca). Pre potreby ACO algoritmu bude mať genotyp premenlivú dĺžku. Veľkosť populácie je daná počtom *genómov*, v tomto prípade počtom mravcov m .

Mutácia (ii) vykonáva náhodné zmeny génov, pričom ľubovoľný gén v hociktorom reťazci môže byť zmenený s rovnakou pravdepodobnosťou. Najjednoduchšia forma je jednobodová mutácia (Obrázok 5.1), pri ktorej je zmenený len jeden gén.



Obrázok 5.1: Jednobodová mutácia.

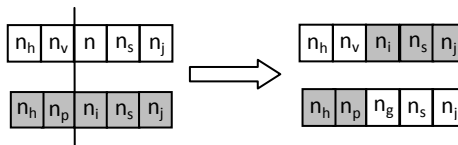
Pre potreby ACO algoritmu z mutácie vylúčime prvý a posledný vrchol reťazca, nakoľko sú pevne dané. Z dôvodu uskutočniteľnosti je nový gén taký vrchol z okolia vrcholu n_i do ktorého vedie hrana z predchodcu n_p vrcholu n_i a z ktorého vedie hrana do nasledovníka n_n vrcholu n_i (Obrázok 5.2).



Obrázok 5.2: Kandidáti, ktorý môžu nahradiť vrchol n_i sú len n_g a n_k .

Ak existuje viac takýchto vrcholov, jeden sa náhodne zvolí. Ak žiaden neexistuje, pre mutáciu sa vyberie iný gén z daného reťazca.

Kríženie (iii) napodobňuje biologickú operáciu. Oba rodičovské reťazce (cesty) sa v ACO algoritme volia náhodne. Najjednoduchšia forma je jednobodové kríženie (Obrázok 5.3). Aby sa zabezpečila uskutočniteľnosť potomkovských ciest, kríženie je možné len v spoločných vrchoch, alebo vo vrchoch, ktorých okolie má spoločný aspoň jeden vrchol. Opäť neuvažujeme prvú a poslednú pozíciu. Ak existuje viac takých vrcholov, pozícia kríženia sa zvolí náhodne. Operácia kríženia je platná, ak sú potomkovské cesty odlišné od rodičovských.



Obrázok 5.3: Jednobodové križenie.

Výber (v_i) je zodpovedný za voľbu reťazcov pre genetické operácie (GO). Z GA sú známe mnohé mechanizmy výberu, z ktorých väčšina zohľadňuje úspešnosť jedince v populácii. Keďže v ACO algoritmoch je optimalizácia výsledkom kooperatívneho správania mravcov, koncept výberu reťazcov pre je čisto náhodný. Tým sa zároveň vyhneme časovo náročnému viacnásobnému vyhodnoteniu účelovej funkcie. Na obrázkoch 5.1 a 5.3 je znázornená *stratégia prežitia*, pri ktorej rodičov nahradia potomkovia.

5.2 ACO algoritmus s GO

Uvedené GO boli aplikované na ACO algoritmus od Kumar et al. (2003), ACO_{KTS} . Na konci každej iterácie, keď všetky mravce dorazia do cieľa, sa vykonajú GO. Ešte pred tým sú z ciest odstránené všetky slučky.

Najskôr sa uplatní mutácia na náhodne zvolenej pozícii. Ak sa mutácia na danom vrchole nedá vykonať, náhodne sa vyberie iný vrchol z tej istej cesty. Ak je naopak viacero vhodných kandidátov, ktorými je možné zvolený vrchol nahradiť, opäť sa medzi nimi vyberá náhodne. Ak sa mutácia nedá vykonať v žiadnom vrchole danej cesty, náhodne sa zvolí iná cesta.

Po vykonaní zadaného počtu mutácií príde na rad križenie. Náhodne sa zvolí prvý rodič a k nemu ďalšia cesta. Ak operáciu križenia nie je možné uskutočniť medzi dvoma zvolenými cestami, vyberie sa iný druhý rodič. Ak zvolený prvý rodič nemá spoločný vrchol (alebo okolie vrcholu) so žiadnou inou cestou, zvolí sa iný prvý rodič a pokračuje náhodná voľba druhého rodiča.

Akonáhle bola určitá cesta modifikovaná, už nebude opäť vybratá pre rovnakú GO v tej istej generácii.

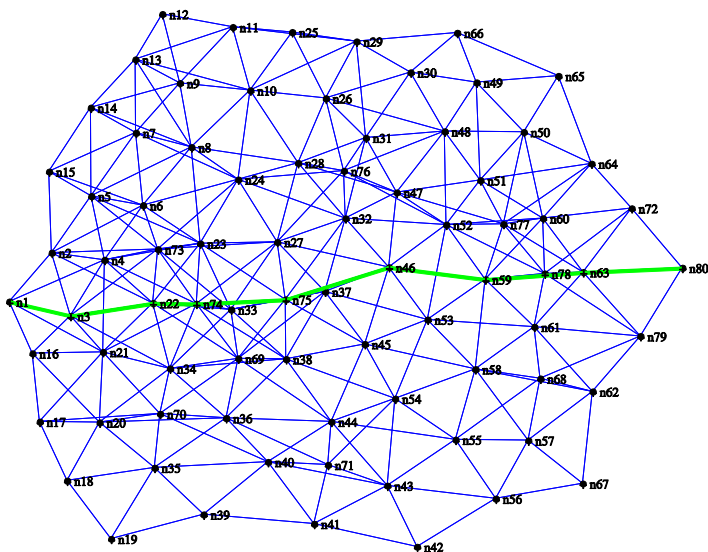
Keďže genetické operácie môžu produkovať slučky, v ACO algoritme sa pred a aj po vykonaní každej GO kontrolujú potomkovské cesty na prítomnosť slučiek. Až po vykonaní všetkých GO sa vyhodnotí dĺžka ciest.

Ako bolo uvedené, genetické operácie aplikované na ACO algoritmus musia pracovať s variabilnou dĺžkou genómu. Z toho dôvodu GO kvantifikujeme pomocou štyroch parametrov, dva pre každú operáciu. Sú to *počet mutovaných ciest* v rámci generácie a *počet mutovaných vrcholov* v rámci cesty. Analogicky dostávame počet križených dvojíc v populácii a počet bodov križenia v rámci dvojice.

Z dôvodu obmedzenia vykonateľnosti zvolený počet GO nemusí byť realizovateľný. Preto sme sa rozhodli ACO algoritmus s GO (ACO_{GO}) obohatiť o užívateľský výstup vo forme dvoch parametrov, ktoré vyjadrujú pomer medzi želaným a uskutočneným počtom danej operácie, v_c a v_M .

5.3 Prípadová štúdia

ACO_{GO} (Cíba, Sekaj, 2013b) algoritmus bol testovaný na náhodne vygenerovanom symetrickom grafe s 80 vrcholmi a 300 hranami (Obrázok 5.4). Súradnice vrcholov x, y spadajú do intervalu $<0,1>$. Hodnoty hrán c_{ij} predstavujú Euklidovské vzdialenosti koncových vrcholov. Úloha je nájsť najkratšiu cestu medzi vrcholmi 1 a 80. Pre vyhodnotenie testu je použitá pravdepodobnosť nájdenia globálneho optima n [1 3 22 74 75 46 59 63 80] v [%].



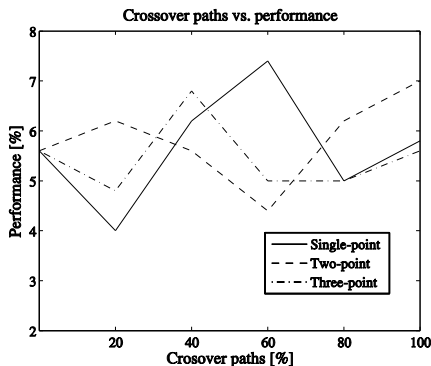
Obrázok 5.4: Náhodne vygenerovaný graf s 80 vrcholmi a vyznačenou minimálnou cestou (zelená).

ACO parametre boli nastavené podľa Becker a Szczerbicka (2005) okrem heuristických hodnôt η_{ij} , ktoré majú hodnotu 0.1. Počet cyklov bol obmedzený podľa priemerného počtu iterácií potrebných pre ACO_{MC} (Cíba, 2012) s tromi makro cyklami na rovnakom grafe a s rovnakými parametrami. Pre určenie optimálneho počtu GO boli vykonané testy, a to pre $m = 10$ a $m = 50$ mravcov po 500, resp. 100 opakovaní pre každé nastavenie.

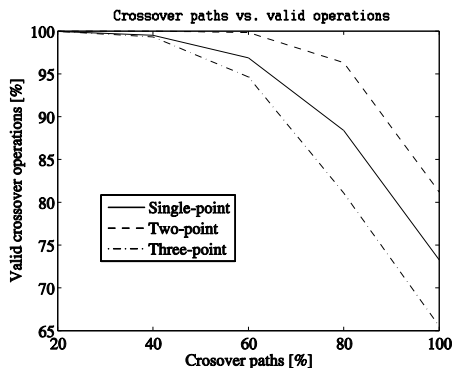
5.3.1 Výsledky pre 10 mravcov

a) kríženie

Pre určenie vplyvu kríženia sme zvyšovali počet krížených dvojíc až na 100% a počet bodov kríženia sme menili od jedna po tri. Operácia mutácie nebola použitá. Výsledky sú znázornené graficky (Obrázok 5.5. a 5.6).



Obrázok 5.5: Vplyv operácie kríženia.

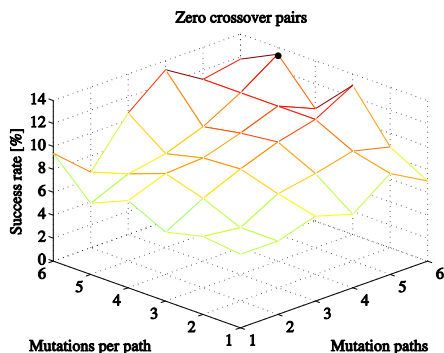


Obrázok 5.6: Počet operácií kríženia a platných operácií (v_c).

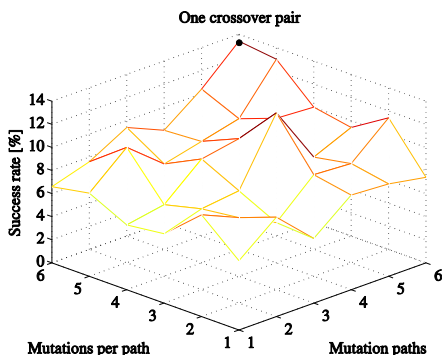
Najvyšší prírastok bol dosiahnutý pre jednobodové kríženie pre 60% krížených dvojiciach (Obrázok 5.5), a to 7.4% oproti 5.6% bez GO. Viacbodové kríženie má menší dopad než jednobodové. S nárastom počtu krížených dvojíc klesá pomer požadovaných a vykonaných operácií kríženia v_c [%] (Obrázok 5.6).

b) mutácia

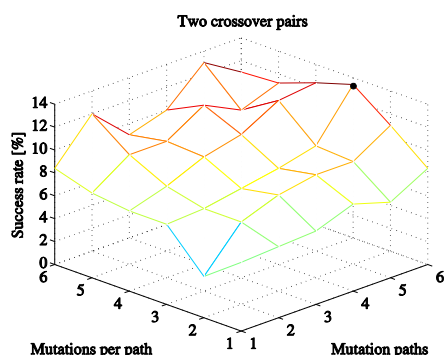
Pre určenie vplyvu mutácie boli vykonané tri pokusy, a to bez kríženia, s jedno- a s dvojbodovým krížením. Výsledky sú uvedené opäť graficky (Obrázky 5.7 až 5.10), pričom najlepší výsledok je označený symbolom “•”. Pri porovnávaní vychádzame z referenčnej hodnoty $n = 5.6\%$ získanej bez použitia GO. Výsledky s použitím GO sú takmer vždy lepšie. Z výsledkov je vidieť, že čím vyšší počet operácií mutácie, tým lepší výsledok. Maximálna dosiahnutá hodnota $n = 13.4\%$ bola dosiahnutá bez kríženia, ale aj pri jedno- a aj dvojbodovom krížení. Vplyv kríženia nie je významný. Podobné hodnoty boli získané pre všetky tri nastavenia.



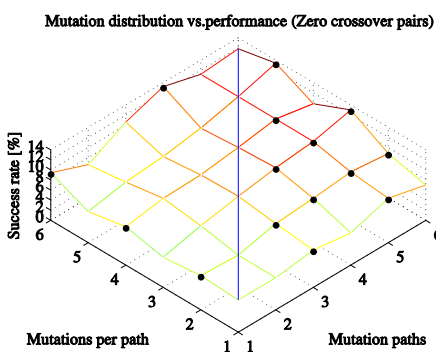
Obrázok 5.7: Vplyv operácie mutácie.



Obrázok 5.8: Vplyv operácie mutácie pri jednobodovom krížení.



Obrázok 5.9: Vplyv operácie mutácie pri dvojbodovom krížení.



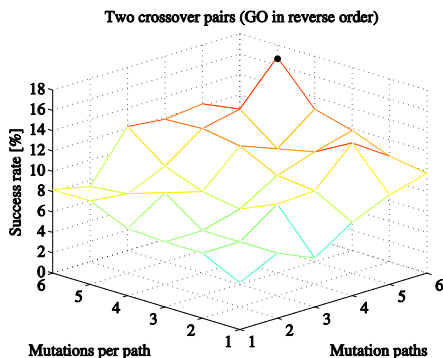
Obrázok 5.10: Vplyv distribúcie operácie mutácie.

Test hodnotil aj vplyv distribúcie mutácie (Obrázok 5.10), v ktorom modrá čiara spája body so symetrickým rozdelením. To sú tie, kde počet mutovaných ciest v generácii je v zhodný s počtom mutovaných vrcholov na ceste. Zrkadlivo obrátené body vzhľadom na čiaru predstavujú rovnaký počet operácií mutácie, avšak s rôznym prerozdelením medzi jednotlivé cesty. Ľavá strana predstavuje koncentrované rozdelenie (viacbodová mutácia na menej cestách) a pravá strana distribuované rozdelenie (menej bodová mutácia na viacero cestách). Prerozdelenie vykazuje 4 ku 11 preferenciu pre distribuovanú mutáciu.

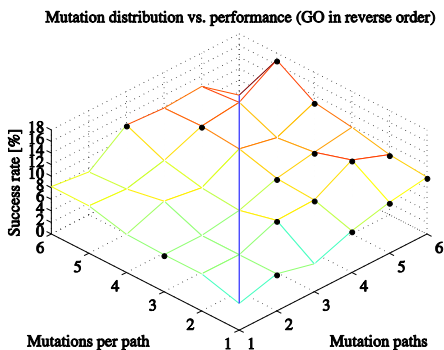
5.3.2 Vykonalie GO v opačnom poradí

Pri tomto experimente sme vykonali GO v opačnom poradí: najskôr kríženie a potom mutáciu, čím sa zabráni prerozdeleniu mutovaných génov krížením. Výsledky boli získané pre dve krížené dvojice (40% krížených ciest) s jednobodovým krížením, čo umožnilo priame porovnanie s predošlým prípadom (Obrázok 5.9).

Výsledky sú odlišné. Pravdepodobnosť nájdenia globálneho optima je vyššia. Nové maximum je 16.8%, čo predstavuje až troj násobok oproti prípadu bez GO (Obrázok 5.11). Vplyv distribúcie mutácie je zhodný s pozorovaním bez kríženia, 12 ku 3 je preferencia na strane distribúcie mutácie medzi viacerou ciest (Obrázok 5.12).



Obrázok 5.11: Výsledky pri opačnom poradí GO.



Obrázok 5.12: Vplyv distribúcie operácie mutácie pri opačnom poradí GO.

GO neovplyvnili priemernú hodnotu iterácie, kedy bolo nájdené najlepšie riešenie (109.08 so smerodajnou odchýlkou 2.76) pri prednastavenom počte iterácií.

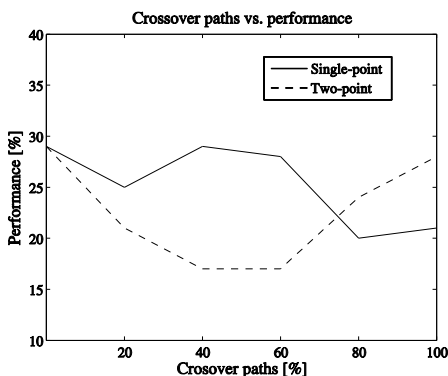
Keďže reverzné poradie GO sa ukazuje ako výkonnejšie, v ďalších experimentoch budeme uvažovať práve túto konfiguráciu.

5.3.1 Výsledky pre 50 mravcov

Pri tomto teste pracuje 50 mravcov na rovnakom grafe. Pre každé nastavenie bolo vykonaných 100 opakovaní.

a) kríženie

Pre vyhodnotenie dopadu bolo použité jedno- a dvojbodové kríženie v rozmedzí nastavenia od 20 až do 100% krížených dvojíc. Referenčná hodnota získaná bez GO je $n = 29\%$.

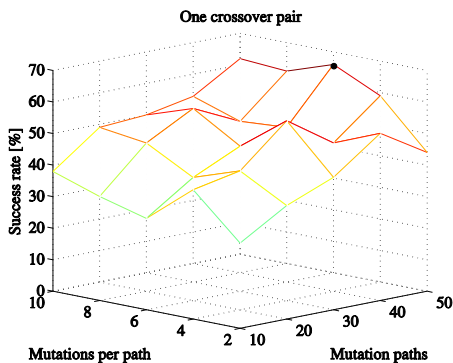


Obrázok 5.13: Dopad kríženia pri 50 mravcoch.

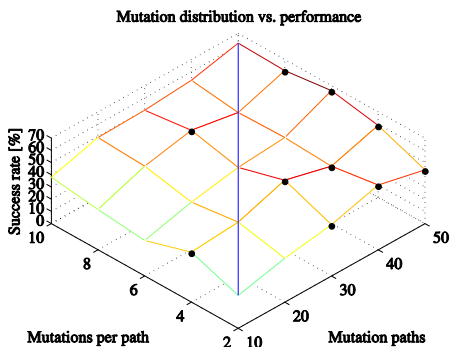
Výsledky ukázali, že pri dostatočnom počte mravcov operácia kríženia nemá žiaden pozitívny efekt.

b) mutácia

Test bol vykonaný s jednobodovým krížením pre 60% krížených dvojíc. Pre vyhodnotenie vplyvu sme použili dvoj až 10 bodovú mutáciu pre 10 až 50 mutovaných ciest. Výsledky sú zobrazené na obrázkoch 5.14 a 5.15.



Obrázok 5.14: Vplyv mutácie pri jednobodovom krížení pri 60% krížených dvojíc.



Obrázok 5.15: Vplyv distribúcie mutácie.

Takmer všetky výsledky dosiahnuté s GO vykazujú vyššiu pravdepodobnosť nájdenia globálneho optima oproti referenčnej hodnote získanej bez GO. Maximum bolo získané pri 6 bodovej mutácii pre 50% mutovaných ciest a dosahuje 66%, čo predstavuje viac než 2 násobok oproti referenčnej hodnote 29%.

Vplyv distribúcie mutácie kopíruje výsledky pre 10 mravcov. V 80% prípadov vykazuje lepšie výsledky mutácia distribuovaná medzi viacero ciest.

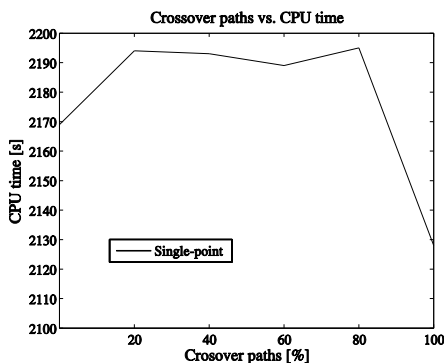
Ani v tomto prípade GO neovplyvnili priemernú hodnotu iterácie, kedy bolo nájdené najlepšie riešenie (91.36 so smerodajnou odchýlkou 5.15) pri prednastavenom počte iterácií.

5.4 Genetické operácie a časové nároky

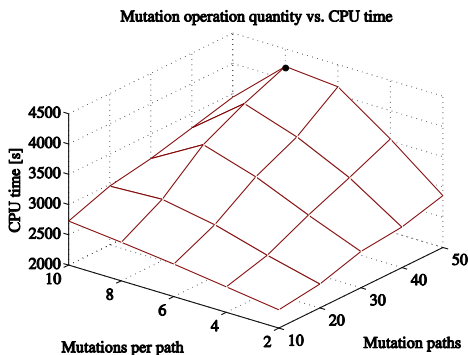
V predošlých experimentoch bolo ukázané, že GO zvyšujú pravdepodobnosť nájdenia globálneho optima. Tiež sa ukázalo, že nevzrástla priemerná hodnota iterácie, pri ktorej bolo nájdené najlepšie riešenie. Avšak dodatočné operácie na konci každej iterácie zvyšujú časové nároky na spracovanie. Z toho dôvodu sme vykonali experiment zameraný na určenie vzťahu medzi spotrebou výpočtového času a nárastom výkonu ACO algoritmu. Nastavenie experimentu je zhodné s prípadom pre 50 mravcov len s nižším počtom opakovaní (10).

a) kríženie

Referenčná hodnota pre časové nároky je 2169 s a bola získaná bez GO. Pre zistenie závislosti bolo použité jednobodové kríženie až do 100% krížených ciest. Ako vidno na Obrázku 5.16, hodnota nie je citlivá na operáciu kríženia ($< \pm 2\%$).



Obrázok 5.16: Vplyv jednobodového kríženia na časové nároky.



Obrázok 5.17: Vplyv mutácie na časové nároky.

b) mutácia

Výsledky boli získané v kombinácii s jednobodovým krížením so 60% krížených ciest spolu s dvoj až desať bodovou mutáciou v rozmedzí 10 až 50% mutovaných ciest. Ako vidno z Obrázku 5.17, mutácia výrazne ovplyvňuje časové nároky algoritmu. So vzrastajúcim počtom mutácií narastá spotreba času, a to až na takmer dvojnásobok času (+6 až +94%).

5.4.1 Oprávnenie genetických operácii

Z predošlých výsledkov vidno, že aplikácia GO prináša zvýšenú pravdepodobnosť nájdenia globálneho optima, avšak aj zvýšený výpočtový čas. Z doteraz získaných výsledkov pre 50 mravcov možno porovnať ACO_{GO} s referenčnými hodnotami od ACO_{KTS} :

Tabuľka 5.1: Porovnanie ACO_{GO} s referenčnými hodnotami.

	100 opakovaní	10 opakovaní
ACO_{GO}	n = 66 [%]	t = 4135 [s]
ACO_{KTS}	n = 29 [%]	t = 2169 [s]
Δ	56%	47.5%

Možno pozorovať, že dvojnásobná pravdepodobnosť nájdenia globálneho optima bola dosiahnutá dvojnásobným výpočtovým časom. Preto sme sa rozhodli porovnať ACO_{GO} s pôvodným ACO_{KTS} , ktorý bude pracovať s dvojnásobným počtom mravcov ($m = 100$) alebo s dvojnásobným počtom iterácií ($c_{ter} = 400$). V oboch týchto prípadoch existuje predpoklad na zvýšenie pravdepodobnosti nájdenia globálneho optima, keďže dvojnásobná zdroje prinesú dvojnásobný počet vyhodnotení účelovej funkcie. V Tabuľke 5.2 sú uvedené priemerné hodnoty získané pri počte opakovaní 100.

Tabuľka 5.2: Výsledky ACO_{KTS} pre dvojnásobok mravcov a / alebo dvojnásobok iterácií.

2 x mravcov	n = 28[%]	t = 4300 [s]
2 x iterácii	n = 28 [%]	t = 4281 [s]
2 x mravcov aj iterácii	n = 54 [%]	t = 8414 [s]

Priame porovnanie pôvodného algoritmu s dvojnásobnou kolóniou alebo s dvojnásobným počtom iterácií potvrdilo dvojnásobné časové nároky, ale nie úmerne vysokú pravdepodobnosť nájdenia globálneho optima. Dokonca ani ACO_{KTS} so štvornásobnými zdrojmi (dvojnásobná kolónia a dvojnásobný počet iterácií) nebol schopný prekonať algoritmus ACO_{GO} .

5.5 Záver pre ACO s GO

Bolo preukázané, že aplikovanie GO podstatne zvýšilo pravdepodobnosť nájdenia globálneho optima, pritom však neovplyvňuje iteráciu, kedy bolo nájdené najlepšie riešenie. Napriek tomu, že dodatočné operácie v každej iterácii zvyšujú čas výpočtu až na dvojnásobok, pôvodný algoritmus ACO_{KTS} nebol schopný výkonnostne prekonať modifikáciu s GO pri dvojnásobku iterácii, ani pri dvojnásobnej veľkosti kolónie.

Prínos kríženia bol zanedbateľný. To môže byť vysvetlené povahou tejto operácie, ktorá iba kombinuje existujúce riešenia. Pri dostatočnom počte mravcov operácia kríženia nemala žiadny pozitívny efekt; vďaka vysokému selektívnemu tlaku ACO algoritmu budú mravce kombinovať

potenciálne riešenia aj sami. Podobná situácia je aj pri malých kolóniách na veľkom prehľadanom priestore. Vtedy klesá šanca, že sa jednotlivé cesty budú krížiť.

Operácia mutácie nie je ovplyvnená feromónovou ani heuristickou informáciou, preto je náchylná meniť existujúce cesty spôsobom, ktorý nie je pre mravce pravdepodobný. Pri experimente s 50 mravcami nebol zistený žiadny limit pre počet mutovaných ciest, výkonnosť algoritmu stúpala až do 100% mutovaných ciest. Čo sa týka počtu náhodne zmenených vrcholov, najlepšia hodnota bola dosiahnutá až pri 60% modifikovaných génoch dĺžky optimálneho riešenia. Experimentálne bolo zistené, že pri danom počte mutácií sa dosiahnu lepšie výsledky, ak sú body mutácie rozdelené medzi viacero ciest.

6 Záver

Táto práca je venovaná nárastu výkonnosti ACO algoritmov. Pojednáva o efektívite prehľadávania a zvýšení diverzity populácie. Boli navrhnuté a otestované rôzne prístupy. Výsledkom je návrh dvoch nových verzií algoritmu. ACO s reinicializáciou je zameraný na efektívnejšie využitie času prehľadávania, ACO s genetickými operáciami na zvýšenie diverzity populácie.

6.1 Reinicializácia v ACO

Na základe porovnania akumulácie feromónu v ideálnom a reálnom prípade boli identifikované tri štádiá prehľadávacieho procesu. Výsledkom analýzy je reinicializačný prístup. Táto modifikácia je plne výkonnostne kompatibilná s pôvodným algoritmom. Jej hlavná výhoda spočíva vo vynechaní neefektívneho prehľadávania vo fáze stagnácie. Vďaka kratšiemu prehľadávaciemu procesu a jednoduchšiemu odhadu vstupných parametrov je táto verzia vhodná pre automatizovanú optimalizáciu rôzne veľkých aj dynamických problémov, kde výsledok optimalizácie potrebujeme poznať v reálnom čase.

6.2 Genetické operácie v ACO

Genetické operácie, ak sa priamo aplikujú na mravcami vygenerované reťazce, výrazne zvyšujú výkon ACO algoritmu. Pravdepodobnosť nájdenia globálneho optima je dva až tri krát vyššia v závislosti od veľkosti mravčej kolónie. Nevýhoda pridaných operácií pre každú iteráciu je nárast výpočtového času. Aj napriek tomu je využitie genetických operácií oprávnené, keďže zvyšujú výkon ACO algoritmu za hranicu obmedzení, kde dvojnásobná veľkosť kolónie ani dvojnásobný počet iterácií už nie sú nápomocné.

7 Literatura

7.1 Okrem prác autora

Aravindh, S., and Michael, G. (2012). Hybrid of Ant Colony Optimization and Genetic Algorithms for shortest path in wireless mesh networks, *Journal of Global Research in Computer Science*, Research paper, Vol. 3, Jan.

Becker, M., and Szczerbicka, H. (2005). Parameters influencing the performance of ant algorithms applied to optimisation of buffer size in manufacturing. *IEMS* Vol. 4, No. 2, December 2005, 184–191

Beni, G., and Wang, J. (1989) *Swarm Intelligence in Cellular Robotic Systems*, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30

Bullnheimer, B., Hartl, R. F., and Strauss, C. (1999a). Applying the Ant System to the vehicle routing problem. In S. Voss, S. Martello, I. H. Osman, & C. Roucairol (Eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization* (pp. 285–296). Dordrecht, Netherlands, Kluwer Academic Publishers.

Bullnheimer, B., Hartl, R. F., and Strauss, C. (1999b). An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, 89, 319–328.

Chiang, W., C., and Russell, R., A. (1995). Simulated annealing metaheuristics for the vehicle routing problems with time windows, *Annals of Operations Research*, 60, Forthcoming

Costa, D., and Hertz, A. (1997). Ants can colour graphs. *Journal of the Operational Research Society*, 48, 295–305.

Deneubourg, J. L., Aron, S., Goss, S., and J. M. Pasteels (1990). The self-organizing exploratory pattern of the Argentine ant. *Journal of Insect Behavior*, 3, 159–168

Dorigo, M., Maniezzo, V., and Colorni, A. (1991). The Ant System: An autocatalytic optimizing process, Technical report 91-016 revised, Dipartimento di Elettronica, Politecnico di Milano, Milan

Dorigo, M., Maniezzo, V., and Colorni, A. (1991a). Positive feedback as a search strategy. Technical report 91-016, Dipartimento di Elettronica, Politecnico di Milano, Milan.

Dorigo, M., Maniezzo, V., and Colorni, A. (1991b). The Ant System: An autocatalytic optimizing process. Technical report 91-016 revised, Dipartimento di Elettronica, Politecnico di Milano, Milano.

Dorigo, M., and Stützle, T. (2004), *Ant Colony Optimization*, MIT Press

Gambardella, L. M., and Dorigo, M. (1997). HAS-SOP: An hybrid Ant System for the sequential ordering problem. Technical report IDSIA-11-97, IDSIA, Lugano, Switzerland.

Goss, S., Aron, S., Deneubourg, J. L., and Pasteels, J. M. (1989). Self-organized shortcuts in the Argentine ant. *Naturwissenschaften*, 76, 579–581

Hajimirsadeghi, G., H., Nabae, M., and Arabi, B., N. (2008). Ant Colony Optimization with a Genetic Restart, *Communications in Computer and Information science* vol. 6, pp. 9-16, Iran

- Kumar, R., Tiwari, M., K., and Shankar, R.** (2003). Scheduling of flexible manufacturing systems: an ant colony optimization approach, *proc. Instn. Mech. Engrs Vol. 217 Part B: J. Engineering Manufacture*, 1443–1453
- Li, Na, Wang S., and Li, Y.** (2011). A Hybrid of GA and ACO for VRP, *Journal of Computational Information Systems* 7:13; Binary information Press, Dec.
- Maniezzo, V., Colorni, A., and Dorigo, M.** (1994). The Ant System applied to the quadratic assignment problem. Technical report IRIDIA/94-28, IRIDIA, Université Libre de Bruxelles, Brussels
- Nakamichi, Y., and Arita, T.** (2001). Diversity control in ant colony optimization, In Abbas HA (ed) *Proceedings of the Inaugural Workshop on Artificial Life (AL'01)*, Adelaide, Australia, Dec 11, 70-78
- Reimann, M., Shtovba, S., and Nepomuceno, E.** (2001). A hybrid ACO-GA approach to solve Vehicle Routing Problems, *Student Papers of the Complex Systems Summer School*, Budapest July 15 - August 9, Santa Fe Institute
- Schoonderwoerd, R., Holland, O., Bruten, J., and Rothkrantz, L.** (1996). Ant-based load balancing in telecommunications networks. *Adaptive Behavior*, 5(2), 169–207.
- Stützle, T., and Hoos, H., H.** (2000). MAX-MIN Ant System. *Future Generation Computer Systems*, 16(8), 889–914
- Stützle, T., and Hoos, H., H. (1997a).** The MAX-MIN ant system and local search for the travelling salesman problem. In T. Baeck, Z. Michalewicz, and X. Yao, editors, *Proceedings of IEEE-ICEE-EPS'97, IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference*, pages 309–314. IEEE Press, 1997.
- Stützle, T., and Hoos, H., H. (1997b).** Improvements on the ant system: Introducing MAX-MIN ant system. In *Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 245–249. Springer Verlag, Wien, 1997.

7.2 Publikované práce autora

- Cíba, M.** (2012), ACO algorithm with re-initialisation, 14th Conference of Doctoral Students ELITECH'12, Bratislava, Slovakia, May
- Cíba, M., and Sekaj, I.** (2012a), ACO algorithm with variable macro cycles, 18th International Conference on Soft Computing Mendel'12, Brno, Czech republic, June
- Cíba, M., and Sekaj, I.** (2012b), Ant colony optimization with re-initialization, 20th International Conference on Technical Computing, Bratislava, Slovakia, October
- Cíba, M., and Sekaj, I.** (2013a), Ant colony optimisation with re-initialization, *International Journal of Automation, Control and Intelligent Systems*, Science Publishing Group Vol. 3, USA
- Cíba, M. and Sekaj, I.** (2013b), Genetic Operations applied to Ant Colony Optimisation, 19th International Conference on Soft Computing Mendel'13, Brno, Czech republic, June
- Cíba, M. and Sekaj, I.** (2013c), Ant Colony Optimisation with genetic operations, *International Journal of Automation, Control and Intelligent Systems*, Science Publishing Group Vol. 3, USA