



Demonstrace činnosti ACO

Dokumentace k projektu do předmětu SFC

3. listopadu 2014

Autor:

Lenka Turoňová, xturon02@stud.fit.vutbr.cz

1 Úvod

Pro řešení problému obchodního cestujícího existuje řada algoritmů, ať už deterministických či heuristických. Jedná se o velmi komplikovaný problém, jehož řešení pomocí deterministických algoritmů je velmi výpočetně náročné. Z tohoto důvodu se upřednostňují algoritmy heuristické, jako je například i algoritmus ACO (Ant Colony Optimization), které jsou poměrně rychlé. Jejich jedinou nevýhodou je však to, že nalezené řešení nemusí představovat vždy optimální řešení. Poměr rychlosti a kvality řešení je ale dostatečný.

Mravenčí algoritmus, jak se v českém překladu nazývá algoritmus Ant Colony Optimization, představuje zastupitele z oblasti počítačové vědy zvané rojová inteligence. Rojová inteligence je vlastnost rozsáhlých systémů, které jsou tvořeny skupinou agentů, jenž spolu vzájemně interagují, což vede v konečném důsledku ke vzniku globálního chování.

Implementovaný algoritmus je inspirován chováním reálných mravenců v přírodě při hledání potravy a materiálu. Komunikace mezi mravenci je nepřímá. Je založena na modifikaci prostředí způsobenou mravenci prostřednictvím zanechání feromonové stopy, která ovlivní chování ostatní mravenců, kteří se budou pohybovat po stejné cestě za ním. Proces optimalizace končí po jistém počtu iterací.

V rámci projektu byl implementován základní algoritmus mravenčího algoritmu a jeho modifikace. Algoritmus je prezentován na problému obchodního cestujícího. Součástí dokumentace je teoretický rozbor včetně popisu algoritmu mravenčí kolonie. Dále je popsána implementace společně s návodem na ovládání programu. V závěru jsou shrnuty dosažené výsledky.

2 Teoretický rozbor

2.1 Problém obchodního cestujícího

Problém obchodního cestujícího patří mezi diskrétní optimalizační problém. V matematice je problém znám jako nalezení nejkratší hamiltonovské kružnice v ohodnoceném úplném grafu. Obecně lze problém popsat jako nalezení nejkratší možné cesty takové, že prochází všemi městy a vede z počátečního města zase nazpět.

2.2 Ant Colony Optimization

Předlohou pro mravenčí algoritmus je reálné chování mravenců v přírodě, kteří při cestě za potravou či materiálem, nechávají za sebou feromonovou cestu. Tato stopa slouží k nepřímé komunikaci, prostřednictvím které se mravenci informují o optimální trajektorii. Pohyb mravenců je obecně náhodný ovlivněný právě informacemi o množství feromonu na jednotlivých cestách, mezi kterými si vybírá.

Algoritmus mravenčí kolonie je obecně velmi dobře znám, proto uvedu jen ve zkratce jeho algoritmus:

- Na počátku (v případě problému obchodního cestujícího) jsou mravenci náhodně umístěni do měst. Čas $t = 0$ a čítač iterací $c = 0$. Intenzita feromonu všech cest je nastavena na malé kladné číslo a .
- Ve struktuře se seznamem projitých měst $tabu[0] = \text{indexy počátečních měst jednotlivého mravence}$. Index struktury $tabu$ je inkrementován.
- V dalším kroku si každý mravenec zvolí cestu na základě pravděpodobnosti dané vzorcem:

$$p_{ij}^k(t) = \left\{ \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l=1, l \notin tabu_k}^n ([\tau_{il}(t)]^\alpha [\eta_{il}]^\beta)} \right\}, \text{ pro } j \notin tabu_k,$$

$$p_{ij}^k(t) = 0, \text{ jinak.}$$

- Index vybraného (dosud nenavštíveného) města je uložen.
- Cyklus se opakuje, dokud každý z mravenců neprojde všemi městy.
- Je vypočtena délka cesty každého mravence.
- Pokud je nejkratší cesta kratší než dosud nalezena, je uložena.
- V dalším kroku je vypočten přírůstek intenzit feromonových stop:

$$\Delta\tau_{ij}^k = \frac{Q}{L_k}, \text{ pokud je hrana } ij \text{ v } tabu_k,$$

$$\Delta\tau_{ij}^k = 0, \text{ jinak.}$$

$$\Delta\tau_{ij} = \Delta\tau_{ij} + \Delta\tau_{ij}^k,$$

kde k je index mravence a Q je konstanta celkového množství feromonu vyloučeného jedním mravencem během jednoho cyklu.

- Dále je upravena podle přírůstku intenzit feromonových stop celková intenzita:

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij},$$

kde $\rho \in \langle 0, 1 \rangle$ je koeficient vypařování feromonů v intervalu $\langle t, t+n \rangle$.

- Čítač c je inkrementován. Čas je zvýšen o n .
- Pokud nebyl proveden daný počet iterací, struktura $tabu$ je inicializován a algoritmus pokračuje od kroku, kdy jsou mravenci rozmístěni do počátečních měst.
- Výsledkem je nejkratší cesta.

2.3 Optimalizace

Mravenčí algoritmus má i své modifikace. Jednou z nich je *Ant-density* algoritmus. Ten spočívá v odlišném počítání přírůstku intenzit feromonových stop $\Delta\tau_{ij}^k = Q$, pokud je hrana ij v $tabu_k$, $\Delta\tau_{ij}^k = 0$, jinak. Další modifikací je algoritmus *Ant-quantity*, který modifikuje vzorec pro výpočet přírůstku feromonových stop následovně:

$$\Delta\tau_{ij}^k = \frac{Q}{d_{ij}}, \text{ pokud je hrana } ij \text{ v } tabu_k,$$

$$\Delta\tau_{ij}^k = 0, \text{ jinak.}$$

Více odlišná je pak modifikace *Elitist strategy*, která posiluje intenzitu feromonů na nejkratší aktuální cestě. Nechť L^* je nejkratší ceta a e je počet mravenců, kteří prošli částí $i - j$ nejkratší cesty, pak:

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} + e\frac{Q}{L^*}.$$

Jeden z problémů elitismus je však možnost předčasné konvergence problému.

Na principu mravenčího algoritmu pracuje několik systému, k nimž patří například *Ant Colony System*, který rozšiřuje možnosti výběru cesty mravence. Díky tomu se s jistou pravděpodobností rozhodne mravenec nedbat na feromonové cesty a vydat se sám svou cestou objevovat dosud neobjevené cesty, či se vypadat cestou, která nemusí být potencionálně upřednostňován.

Existuje náhodná veličina $q \in \langle 0, 1 \rangle$, která se porovná s pevně vhodně předem zvolenou hodnotou parametru q_0 . V případě, že hodnota náhodné veličiny bude větší než daná hodnota parametru q , mravenec zvolí tuto cestu. Jinak vybere cestu na základě feromonové stopy:

$$j = \arg \max_{l \in \{1, n\}, l \notin tabu_k} \left([\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta \right).$$

Tomuto způsobu rozhodování říkáme *exploitation*, tedy využívání a poskytuje potencionálně lepší výsledky oproti předešlým variantám, které obvykle vedou k uváznutí v lokálních extrémech.

MMAS poskytuje optimalizace klasického mravenčího algoritmu z toho hlediska, že na vylepšení feromonové cesty má vliv jen nejlepší mravenec, tedy ten, který našel nejkratší cestu. Úprava cesty bude probíhat na základě následujícího vzorce:

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best}.$$

Navíc jsou algoritmus dány k dispozici maximální a minimální mez hodnoty feromonu, mezi kterými se hodnota feromonu na jednotlivých hranách má pohybovat. Tímto způsob se může docílit toho, že algoritmus nebude mít tendenci uváznout v lokálních extrémech.

Poslední modifikací mravenčího algoritmu je *Rank-base Ant System*. Je založen na aktualizaci feromonové cesty ω nejlepšími mravenci. Modifikovaný vzorec pro výpočet feromonové stopy tak bude následující:

$$\tau_{ij}(t+n) = (1-\rho)\tau_{ij}(t) + \sum_{r=1}^{\omega-1} (\omega-r)\Delta\tau_{ij}^r + \omega\Delta\tau_{ij}^{best}.$$

3 Implementace

V rámci projektu byl naprogramován základní mraveční algoritmus včetně všech jeho modifikací, které byly v průběhu přednášek prezentovány. Uživatel si může sám zvolit v úvodním okně aplikace *Ant Colony Optimalization*, jakou variantu algoritmu chce demonstrovat.

Po nastavení všech potřebných parametrů, musí uživatel zadat pomocí kliknutí myši do panelu s mapou pozice jednotlivých měst. Jednotlivá města jsou označena indexem a spojena hranou se všemi ostatními tak, aby se vytvořil úplný graf.

Dále musí uživatel nastavit parametry. Po kliknutí na tlačítko *Set parameters* se mu otevře nové okno, ve kterém může nastavit následující parametry i pro jednotlivé modifikace:

- *Ants* - počet mravenců,
- *Cycles* - počet cyklů algoritmu,
- *Alpha* - relativní důležitost feromonu,
- *Beta* - relativní důležitost viditelnosti a
- ρ - faktor vypařování,
- *Probability* - pravděpodobnost při výběru cesty u *Ant Colony System*,
- *Minimum* - minimální hodnota hodnoty feromonu u *MMAS*,
- *Maximum* - minimální hodnota hodnoty feromonu u *MMAS* a
- *Omega* - počet mravenců ovlivňující feromonovou stopu u *Rank-base AS*.

Po uložení parametrů může uživatel nastavit rychlost vykreslování průběhu algoritmu pomocí posuvníku v dolní části hlavního okna. *Speed* udává počet sekund mezi jednotlivými iteracemi vykreslení.

Program načte vstupní parametry a vypočte vzdálenosti mezi městy. Po vypočtení zvoleného algoritmu je otevřeno automaticky okno s grafem znázorňující vývoj nejkratší nalezené cesty v průběhu cyklů. Pokud v hlavním okně uživatel zvolí jiný algoritmus a spustí výpočet znovu, otevře se mu další okno s příslušným grafem, takže má možnost jednotlivé modifikace srovnat. Je však nutné upozornit na fakt, že v případě zavření některého z grafu se zavře celý program.

Tlačítkem *Clear* může uživatel kompletně odstranit všechna města a zvolit nové rozmístění měst na ploše. Tlačítkem *Abort* je možné přerušit probíhající výpočet. S přerušením se ovšem odstraní automaticky i města.

Na dolní liště je uživatel informován o aktuálně nejkratší nalezené cestě, popř. informován o chybách během výpočtu či o skončení výpočtu.

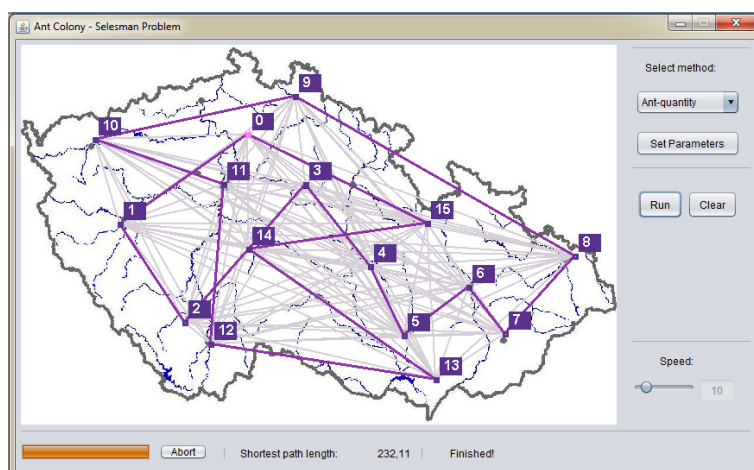


Figure 1: Hlavní okno aplikace

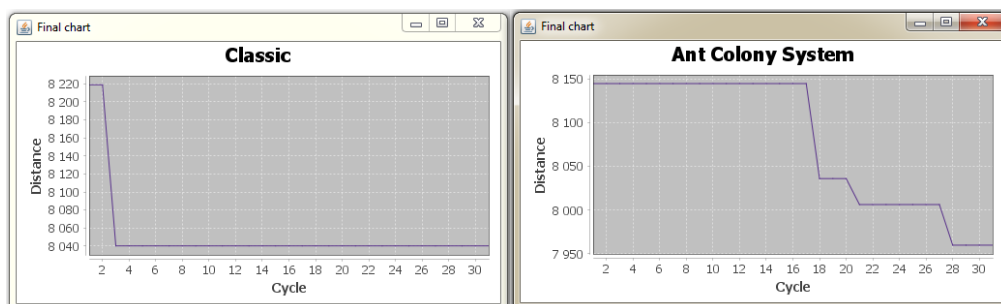


Figure 2: Ukázky grafů: a) *Ant Colony Optimization* a b) *Ant Colony System*

4 Závěr

V rámci projektu byl naimplementován s použitím jazyka *Java* algoritmus *Ant Colony Optimization*. Program slouží především k demonstraci algoritmu, a to včetně všech jeho variant, které byly v rámci přednášek předmětu *Softcomputing* prezentovány. Program klade důraz především na uživatelsky přívětivé ovládání. Nastavení parametrů i intuitivní zadávání měst. Uživatel je o všech chybách a průběhu algoritmu informován pomocí výpisu na dolní liště. Vývoj výběru nejkratší cesty je graficky průběžně názorně vykreslován. Součástí implementace jsou i názorné grafy, díky nimž může uživatel srovnávat rozdíly v průběhu nejkratší délky cesty jednotlivých nastavení algoritmu, či jednotlivých variant.