

Optimizacija kolonijom mrava za problem trgovačkog putnika

Gorana Vučić, Vojkan Cvijović

Naučno izračunavanje
Matematički fakultet, Beograd

13. septembar 2019

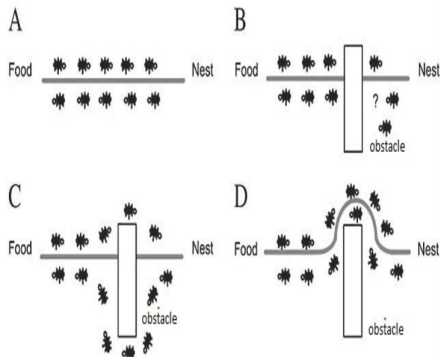
Pregled

- 1 Uvod
 - Uopšteno
 - Ponašanje mrava
 - Ponašanje mrava u slučaju prepreke na putu
- 2 Primena ACO na TSP
 - ACO i parametri
 - ACO algoritam
- 3 Implementacija
 - Klase
 - Klasa ACS i njene metode
 - Klasa Ant i njene metode
- 4 Pokretanje i izgled rešenja
- 5 Rezultati pokretanja
 - Promena α
 - Promena β

- **ACO** (eng. *ant colony optimization*)
 - Algoritam zasnovan na ponašanju mrava i mravljih kolonija
 - Pojam je uveo Marco Dorigo 1992. godine
 - ACO pripada klasi algoritama inteligencije roja
- Primena algoritma je prikazana na rešavanju problema trgovačkog putnika
- Optimizacija kolonijom mrava predstavlja algoritam koji se koristi za nalaženje optimalnih putanja u potpuno povezanom grafu

- Ukoliko postavimo mrave u nepoznatu okolinu kako bi došli do izvora hrane u početku će se nasumično kretati
- Kada pronadu hranu pri povratku u koloniju ostavljace trag feromona koji će privlačiti ostale mrave da se kreću u tom smeru
- Ostali mravi će početi da se kreću u smeru traga feromona, pojačavace trag feromona i na taj način će privlačiti još mrava
- Nakon određenog vremena počinje da se odvija proces evaporacije odnosno isparavanje feromona, što je od velike koristi s obzirom da se pozicija hrane stalno menja

- U početku verovatnoća da odaberu levi ili desni put je jednaka, polovina mrava će ići levim a druga polovina desnim putem
- Levi put je kraći \implies ostaće jači trag feromona
- Kako vreme odmiče veći broj mrava će prolaziti tim putem, jačina feromona će biti veća i na kraju će se čitava kolonija kretati tim putem



Problem trgovačkog putnika

- Jedan od najpozantijih problema iz grupe NP teških problema
- Trgovački putnik:
 - Zna koje gradove treba da poseti
 - Zna udaljenost među gradovima
 - U obavezi je da svaki grad poseti tačno jednom i da se vrati u grad iz kog je pošao
- Rešenje se ogleda u tome da trgovački putnik odredi redosled gradova i da pritom putuje najoptimalnijom mogućom rutom

ACO algoritam za TSP i parametri

- Algoritam se odvija u četiri koraka:
 - Na slučajan način odabere se m od ukupno n gradova ($m \leq n$) u koje se rasporedi po jedan mrav
 - Svaki mrav posećuje svaki grad tačno jedanput i u svakoj iteraciji ažurira listu gradova J_k koje treba da poseti
 - Mrav k koji se nalazi u gradu i prelazi u grad j gde je $j \in J_k$ sa verovatnoćom:

$$p_{i,j}^k = \frac{t_{i,j}^\alpha * distance_{i,j}^{-\beta}}{\sum_{l \in J_k} t_{i,l}^\alpha * distance_{i,l}^{-\beta}} \quad (1)$$

- ...

ACO algoritam za TSP i parametri

- Algoritam se odvija u četiri koraka:
 - Neka je L_k dužina puta T_k koji je prešao mrav k i Q pogodno izabran pozitivan parametar. Neka važi:

$$\Delta t_{ij}^k = \begin{cases} Q/L_k & \text{ako } (i,j) \in T_k, \\ 0 & \text{inače} \end{cases}$$

$$\Delta t_{i,j} = \sum_{k=1}^m \Delta t_{i,j}^k \quad (2)$$

Za pogodno izabran parametar $\rho \in (0,1)$, feromonski tragovi se ažuriraju na osnovu formule

$$t_{i,j} = (1 - \rho) * t_{i,j} + \Delta t_{i,j} \quad (3)$$

Parametar ρ predstavlja stepen isparavanja feromona, a $\Delta t_{i,j}$ pojačanje količine feromona na deonici (i,j) .

Klase i metode koje su korišćene

- Korišćene su tri klase:
 - *Graph*
 - *ACS*
 - *Ant*
- *Graph* sadrži neke opšte informacije o gradovima kao što su:
 - *distances*: predstavlja matricu rastojanja između gradova
 - *rank*: predstavlja broj gradova, važi da između svaka dva grada postoji put
 - *pheromone*: predstavlja matricu nivoa feromona između gradova

Klase i metode koje su korišćene

- ACS - klasa koja se koristi za rešavanje problema putujućeg trgovca primenom ACO algoritma:
 - *generations*: predstavlja broj iteracija samog algoritma
 - *ant_count*: predstavlja broj mrava u svakoj iteraciji
 - *alpha*: parametar koji određuje uticaj feromona
 - *beta*: parametar koji određuje uticaj udaljenosti između gradova
 - *rho*: parametar koji određuje koja količina starog feromona se prenosi u narednu iteraciju algoritma
 - *Q*: pogodno izabran pozitivan parametar
- Klasa sadrži metode *update_pheromone* i *solve*.

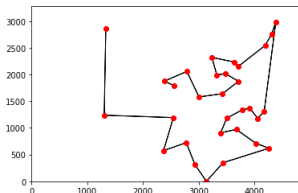
Klase i metode koje su korišćene

- *Ant* - klasa koja predstavlja jednog mrava u sistemu:
 - *colony*: instanca klase ACS
 - *graph*: instanca grafa koji mrav obilazi
 - *total_cost*: cena puta koju je mrav prešao
 - *visited_nodes*: niz čvorova koje je mrav obišao
 - *pheromone_delta*: veličina feromona koju je mrav proizveo
 - *unvisited_nodes*: neposećeni čvorovi u grafu
 - *start_node*: početni čvor iz kog mrav polazi
 - *visited_nodes*: niz čvorova koje je mrav obišao
 - *current_node*: indeks čvora koji mrav trenutno obilazi
- Klasa sadrži metode *update_pheromone_delta* i *select_next_node*.

Pokretanje i izgled rešenja

- Metodom *find_optimal_path* se pokreće izvršavanje programa.
- Rešenje je dato u sledećem obliku:

COST: 15888.860695131287,
PATH: [11, 13, 12, 10, 22, 15, 4, 5, 6, 1, 3, 7, 8, 9, 2, 17, 16, 18, 23, 24, 19, 20, 21, 25, 27, 26, 29, 30, 28, 0, 14]
TIME EXECUTION: 1.7936382293701172

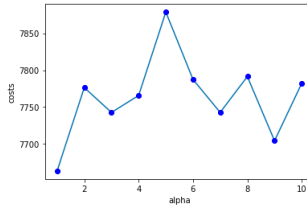


Rezultati pokretanja

- Najbolje rešenje je dobijeno za sledeće parametre:
 - *ant_count* = 10
 - *generations* = 100
 - *alpha* = 1.0
 - *beta* = 10.0
 - *rho* = 0.5
 - *q* = 10
- Prosečno vreme izvršavanja za zadate parametre je 5.32s, najbolje rešenje iznosi 7663.58 sa vremenom izvršavanja 5.56s.

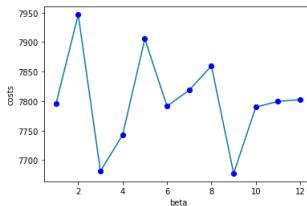
Promena α

- Najbolje rešenje je dobijeno za $\alpha = 1$:



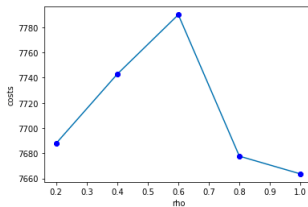
Promena β

- Najbolje rešenje je dobijeno za $beta = 9$:



Promena ρ

- Najbolje rešenje je dobijeno za $\rho = 1.0$:



Literatura

- Stefan Mišković. Optimizacija kolonijom mrava.
<http://poincare.matf.bg.ac.rs/~stefan/ri/aco.pdf>.
- Dorian Gaertner and Keith Clark. On optimal parameters for ant colony optimization algorithms. In Proceedings of the International Conference on Artificial Intelligence 2005, pages 83–89. CSREA Press, 2005.
- Ivan Brezina Jr. and Zuzana Čičková. In Solving the Travelling Salesman Problem Using the Ant Colony Optimization, 2011.
- K. M. Shweta and A. Singh. In An effect and analysis of parameter on ant colony optimization for solving travelling salesman problem, page 222–229. IJCSMC, 2013.
- Matplotlib library. on-line at: <https://matplotlib.org/>.

Hvala na pažnji!