

Dokumentácia k programu astsp

Peter Zeman

7. augusta 2012

Problém obchodného cestujúceho

Formálne môžeme TSP (traveling salesman problem) reprezentovať úplným ohodnoteným grafom $G = (V, E)$, kde V je množina vrcholov, ktoré reprezentujú mestá a E množina hrán, ktoré reprezentujú cesty medzi mestami. Každá cesta $\{i, j\} \in E$ má priradenú dĺžku d_{ij} . Cieľom je nájsť hamiltonovskú kružnicu s minimálnou dĺžkou. Optimálne riešenie TSP je teda permutácia π vrcholov $\{1, \dots, n\}$ taká, že dĺžka $f(\pi)$ je minimálna, pričom

$$f(\pi) = \sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)} + d_{\pi(n)\pi(1)}.$$

Úlohou programu je priblížiť sa v rozumnom čase čo najviac k optimálnemu riešeniu TSP. Zavolá si na pomoc mravce!

Ant System

Dve hlavné fázy algoritmu Ant System (AS) sú budovanie ciest pomocou mravcov a aktualizácia feromónov. Základná kostra algoritmu vyzerá takto:

```
def aco_for_tsp():
    # Inicializacia
    initialize_data(...)

    while not terminate(...):
        construct_solutions(...) # Budovanie ciest
        best_tour_length, best_tour = update_statistics(...)
        update_pheromone_trails(...) # Aktualizacia feromonov
```

Pri budovaní ciest je najprv každému mravcovi náhodne priradené nejaké štartovné mesto. Potom každý mravec z tohoto mesta vybuduje cestu, ktorá spĺňa podmienky TSP (končí v štartovnom meste a súčasne prejde každým mestom práve raz). Budovanie ciest prebehne v n krokoch (n je počet miest) a v každom kroku je pre každého mravca vybrané

jedno mesto. Po skončení tejto procedúry je zaznamenaná najlepšia nájdená cesta a sú aktualizované feromóny. Je viac možností, kedy ukončiť celý algoritmus, napríklad keď sa dosiahne maximálneho počtu iterácií.

Výber mesta

V každom kroku budovania ciest si musí mravec k vybrať, ktoré bude jeho ďalšie navštívené mesto. Mravec sa rozhodne na základe pravdepodobnostného pravidla. Pravdepodobnosť, s ktorou si mravec k v meste i vyberie mesto j , je

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \quad \text{ak } j \in N_i^k,$$

kde $\eta_{ij} = 1/d_{ij}$ (d_{ij} je vzdialenosť medzi mestami i a j) je heuristická hodnota, τ_{ij} je množstvo feromónov uložených medzi mestami i a j . Konštanty α a β sú parametre, ktoré určujú vplyv feromónov a heuristickej informácie. Ak $\alpha = 0$, tak feromóny nemajú žiadny vplyv a dostávame obyčajný hladný algoritmus. Ak je $\beta = 0$, tak majú vplyv len feromóny. Konečne N_i^k značí množinu doposiaľ nenavštívených miest mravcom k v meste i . Podľa uvedeného vzorca, pravdepodobnosť výberu nejakej hrany (i, j) rastie s hodnotami τ_{ij} a η_{ij} .

Procedúra, ktorá vyberá nasledujúce mesto funguje nasledovne. Nech c značí mesto, v ktorom sa práve nachádza mravec k . Každá hodnota $choice_info[c][j]$ ($choice_info[i][j] = [\tau_{ij}]^\alpha [\eta_{ij}]^\beta$), kde j je nenavštívené mesto mravcom k , sa pripočíta k $sum_probabilities$. Položíme $selection_probability[j] = choice_info[c][j]$ ak mesto j ešte navštívené, inak kladieme $selection_probability[j] = 0$. Teraz zvolíme náhodne číslo r z intervalu $[0, sum_probabilities]$. Potom začneme prechádzať polom $selection_probability$ a jednotlivé prvky nasčítavame do premenenej p . V momente keď je $p \geq r$ prechádzanie zastavíme a mesto, pri ktorom sa tak stalo bude nasledujúce.

```
r = random.uniform(0.0, sum_probabilities)
p = 0.0
for i in range(len(selection_probability)):
    p += selection_probability[i] # Pripocitame selection probability
    if p >= r:
        # Vybrali sme mesto
        break
ants[ant].tour[step] = nn_list[city][i]
```

Aktualizácia feromónov

Posledným krokom je aktualizácia feromónov. Kostra tejto procedúry je takáto:

```
def update_pheromone_trails(ants, dist, pheromone, choice_info):
    # Najprv nechame feromony vyprchat
```

```

evaporate(pheromone)
# S kazdym mravcom aktualizujeme feromony na jeho ceste
for i in range(len(ants)):
    deposit_pheromone(i, ants, pheromone)
# Aktualizujeme choice_info
create_choice_info_matrix(choice_info, dist, pheromone)

```

Najprv necháme vyprchať feromóny, to znamená, že ρ -krát znížime hodnotu τ_{ij} , kde ρ je nejaká konštanta. Potom za každého mravca pridáme feromóny na hrany, ktorými prešiel. Nech tl značí dĺžku cesty, ktorú prešiel nejaký mravec k . Potom mravec k uloží na každú hranu, ktorou prešiel, $1/tl$ feromónov. Teda čím kratšiu cestu mravec prešiel, tým viac feromónov uloží. Nakoniec stačí už len aktualizovať *choice_info*.

Literatúra

- [1] *M. Dorigo, T. Stützle: Ant Colony Optimization*
The MIT Press, 2004