

Optimalizácia kolóniou mravcov

Náš problém je Vehicle Routing Problem, ide o zovšeobecnenie problému obchodného cestujúceho. My máme trochu jednoduchšiu verziu. Máme depo a mestá kde do každého jedného mesta potrebujeme zaviesť práve jednu zásielku, ktorá má zadanú hmotnosť. Na rozvoz môžeme použiť ľubovoľne veľa vozidiel, a všetky majú rovnakú nosnosť. Vozidlo musí začínať a takisto aj končiť v depe. Našou úlohou je nájsť čo najnižší počet vozidiel a čo najnižší počet najazdenej vzdialenosti. Zároveň ale chceme rozviesť všetky zásielky.

Riešenie problému

Tento náš problém sa dá previesť na grafový problém, kde hľadáme čo najnižší počet skoro disjunktných kružníc, ktoré všetky obsahujú jeden konkrétny vrchol (depo) a majú čo najnižšiu váhu použitých hrán. Zároveň ale vrcholy majú tiež svoje váhy, a vrcholy na každej kružnici nesmú prekročiť určený threshold.

Idea riešenia:

Každý náš mravec vždy vygeneruje nejaké kompletne riešenie. To robí tak, že vždy začne v depe, a vyberá si náhodnú hranu pomocou distribúcie, ktorá je určená niekoľkými faktormi, ako napr. počet feromónov na hrane, alebo dĺžka hrany. Pri výbere hrany zo sebou zoberie aj zásielku určenú mestu, do ktorého vybraná hrana vedie. Toto opakujeme dovtedy, kým už nemôžeme pridať do vozidla zásielku pre mesto, kde sme si vybrali hranu. V takom prípade sa vrátíme do depe a pokračujeme znovu s prázdny vozidlom, s tým že vynechávame mestá, ktoré sme už navštívili. Takto pokračujeme kým neobídeme všetky mestá.

Popis algoritmu:

Algoritmus je upravený algoritmus z cvičenia, ktorý bol určený pre problém obchodného cestujúceho. Úpravy:

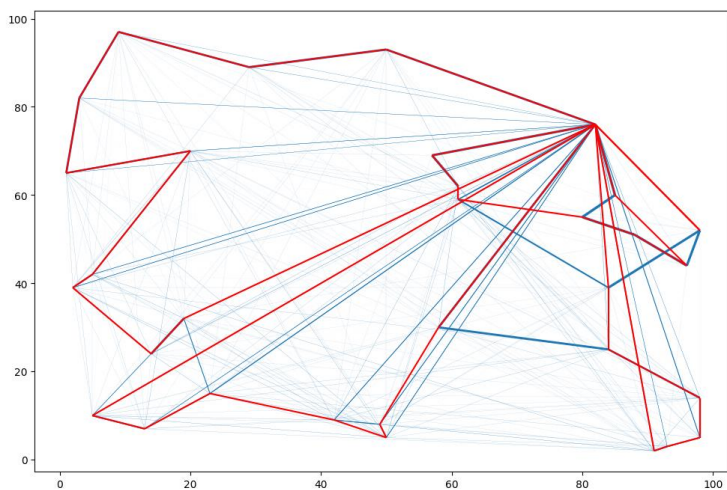
- Načítanie vstupu: vstup sa nenačítaval z .csv súboru ako na cvičení, ale z .xml. A samozrejme formát vstupu bol rozdielny, takže boli nutné úpravy
- Fitness funkcia: tá funguje podobne ako v pôvodnom algoritme, ale jej výstupom je vzdialenosť všetkých kružníc riešenia. Mal som neimplementovanú aj verziu, kde sa vracala dvojica, (vzdialenosť, počet kružníc) alebo verziu kde som počet kružníc navážil novou konštantou omega a sčítal so vzdialenosťou a to vrátil. Avšak som sa rozhodol že výstup fitness funkcie bude iba vzdialenosť, v závere vysvetlím prečo.
- Distance funkcia: ostala rovnaká
- Feromónová funkcia: ostala rovnaká
- Generate solutions: táto funkcia bola zmenená, pretože potrebujeme generovať rozdielne solutions ako v TSP. Úprava funkcie je popísaná v idei. (za zmienku možno stojí, že počítanie pravdepodobnosti ostalo rovnaké)
- Ant solver: ostal rovnaký

Parametre som ziniclizoval na rovnaké hodnoty, a od nich som sa malými zmenami posúval a skúšal, čo spraví s algoritmom. Väčšinou som si nastavil menej epóch, a sledoval som ako sa vyvíjal algoritmus,

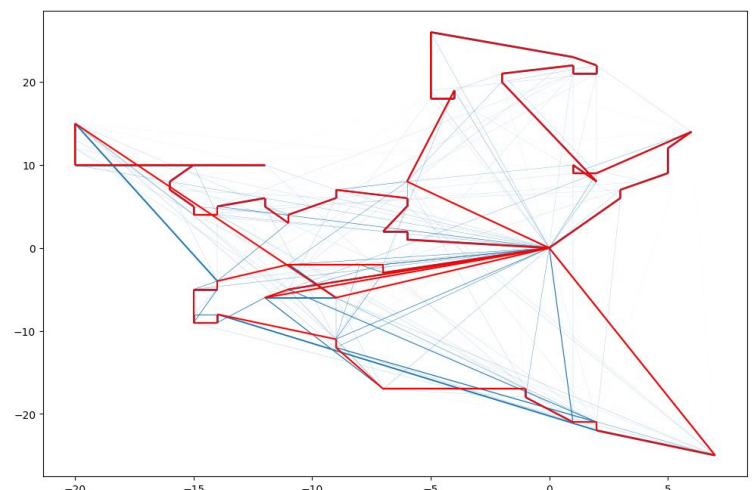
a keď sa hodnoty len náhodne opakoali, tak som vyskúšal inú konfiguráciu. Ale keď to vyzeralo nádejne, nechal som algoritmus bežať dlhšie. Popri tom som sledoval aj grafy, a snažil sa z nich niečo vyčítať (napr. či pravdepodobnostná distribúcia nepreferuje až príliš alebo primálo feromónovú stopu). Najlepšie výsledky a ich parametre som si ukladal do zložky ,results', ktorú prikladám ku riešeniu aj s obrázkami grafov. Najlepšie parametre pre rôzne testy boli celkom rozdielne.

Výsledky

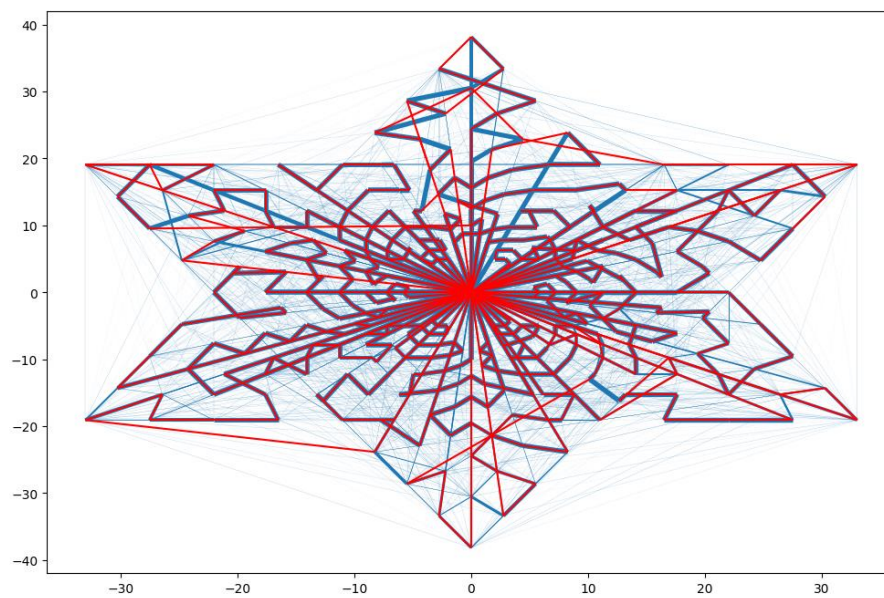
	Vzdialenosť (fitness)	Počet kružníc	Približná doba behu
data_32.xml	833.44	5	1 min
data_72.xml	277.98	4	1 min
data_422.xml	2349.22	39	1 hod



data_32.xml – best result graph



data_72.xml – best result graph



data_422.xml – best result graph

Záver

Neviem či sa moje výsledky aspoň trochu približujú k optimálnym riešeniam, ale po niekoľkých behoch algoritmu pri dobrej konfigurácii skoro vždy konvergoval práve k týmto hodnotám. Takže si myslím, že príliš ďaleko nebudem. Zároveň pri pohľade na graf to vyzerá tak, že kružnice neskáču zbytočne z jednej strany na druhú, ale pohybujú sa iba v nejakej menšej časti grafu, čo by mohlo znamenať, že riešenia sú celkom rozumné.

Nakoniec by som chcel osvetliť, prečo som nakoniec zahodil spomenuté verzie fitness funkcie, ktoré sa snažili minimalizovať aj počet kružníc. Prvú verziu, ktorej výstupom bola dvojica (vzdialenosť, počet kružníc) som zavrhol, pretože mi prišlo časovo náročné a zbytočné prepisovať algoritmus tak, aby bral do úvahy 2 parametre fitness funkcie, ktoré sa snažíme minimalizovať.

Druhú verziu so súčtom vzdialeností a počtom kružníc pre násobeným parametrom omega som zavrhol pretože behy algoritmu mi pri tejto verzii boli dosť nestabilné, hodnoty fitness funkcie v jednotlivých epochách dosť skákali a keď som menil parameter omega, tak sa to moc nezlepšovalo. Preto som si povedal, že keď nájdem riešenie blízke optimu ktoré berie do úvahy vzdialenosť, tak pravdepodobne nebude mať príliš veľký počet kružníc. Takže som sa rozhodol vymeniť vyšší počet kružníc za nižšiu vzdialenosť.