

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

Umelá inteligencia

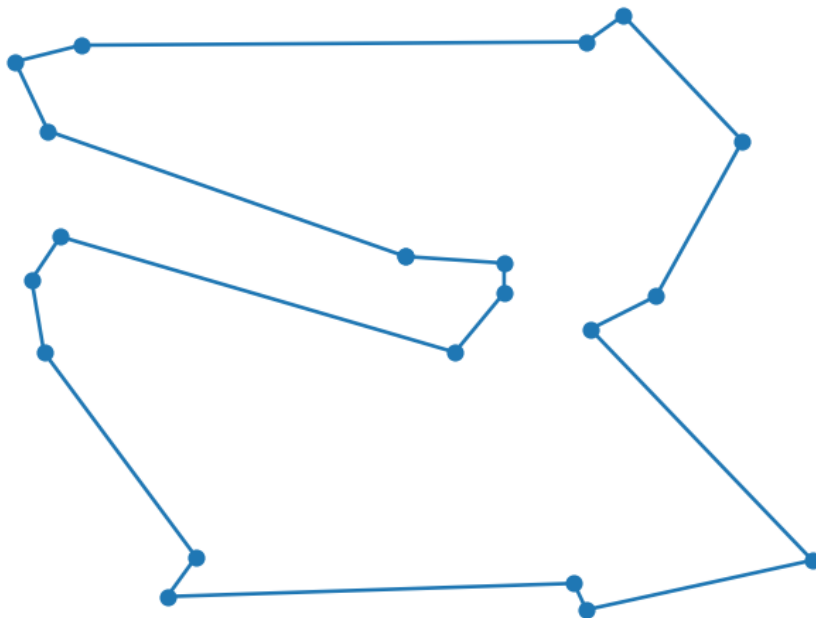
Zadanie 2: Problém obchodného cestujúceho

Obsah

Zadanie	3
Riešenie	3
Uchovávanie permutácií	3
Genetický algoritmus	3
Zvolenie rodičov	4
Crossover	4
Mutácie	4
Simulované žíhanie	4
Elitizmus	4
Výpis	4
Porovnanie výsledkov	5
Genetický algoritmus	5
Simulované žíhanie	6
Možné vylepšovanie	6

Zadanie

Navrhnete riešenie pre zistenie najkratšej cesty cez všetky mestá z permutácie. Cena medzi dvomi mestami zodpovedá Pytagorovej vete. Pre zistenie najlepšej permutácie použite genetický algoritmus a simulované žihanie. Na obrázku je najkratšia cesta pre 20 miest generovaná genetickým algoritmom.



Riešenie

Uchovávanie permutácií

Vytvoril som si triedu Successor v ktorej som uchovával postupnosť a dĺžku cesty. Generácia tvorila X týchto objektov.

```
class Successor:
    def __init__(self, chromosome, fitness):
        self.chromosome = chromosome # chromosome of cities
        self.fitness = fitness # Total distance of the path (Euclidean distance)
```

Genetický algoritmus

Vytvorím si prvú permutáciu miest z ktorej sa vytvorí generácia o zadanej veľkosti, tú pošlem do funkcie s genetickým algoritmom spolu s veľkosťou populácie, počtom generácií a typom výberu rodičov. Funkcia s genetickým algoritmom funguje nasledovne: uloží si najlepšiu

cestu z predošlej pridám ju do novej a generujem novú populáciu tak, že vyberiem rodičov z predošlej generácie spravím s nimi crossover a mutáciu a pridám ich do novej generácie. Toto opakujem dokým ich nie je dosť v populácii a toto celé opakujem toľko krát koľko je zadaných generácií. Ak sa generácia nezlepšila po 50 opakovaníach tak zavolám 2-opt heuristiku na jednu iteráciu, ktorá mi otočí časť trasy.

Zvolenie rodičov

Použil som metódu turnaja a ruletu. Turnaj je založený na vybratí niekoľkých náhodných permutácií a z nich sa vyberie tá najlepšia. Ruleta zistí celkový fitness populácie a každému priradí pravdepodobnosť zvolenia a následne sa jedna permutácia vyberie.

Crossover

Zobral som si dve náhodné čísla, ktoré slúžili na ohraňenie. Do tejto časti permutácie sa kopíruje časť z jedného rodiča a zvyšok sa kopíruje z druhého rodiča a zároveň tá lokácia nemôže byť už v childe. Toto sa zopakuje 2x a vrátia sa dva childy.

Mutácie

V mutácii sa menia iba dve náhodné súradnice v permutácii.

Simulované žihanie

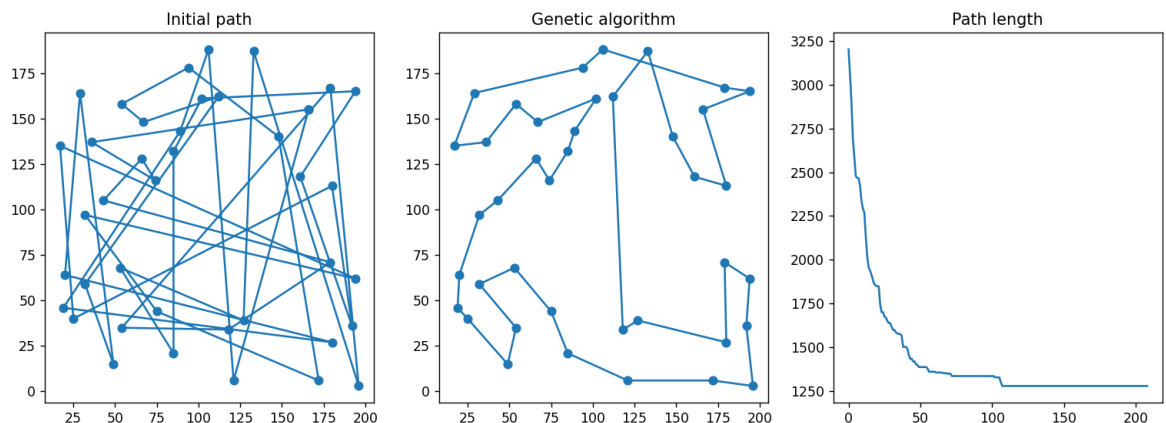
Pri simulovanom žihaní si určím počiatočnú pravdepodobnosť a hodnotu, ktorou ju budem znižovať. Vytvoríme si jeden objekt, ktorý budeme mutovať v každej iterácii a vždy k nemu vypočítame fitness hodnotu a pravdepodobnosť prejdenia do tohto stavu. Ak ma stav lepšiu fitness prejdeme doňho ak nie tak doň prejdeme s vypočítanou pravdepodobnosťou. Na konci sa znižuje počiatočná hodnota.

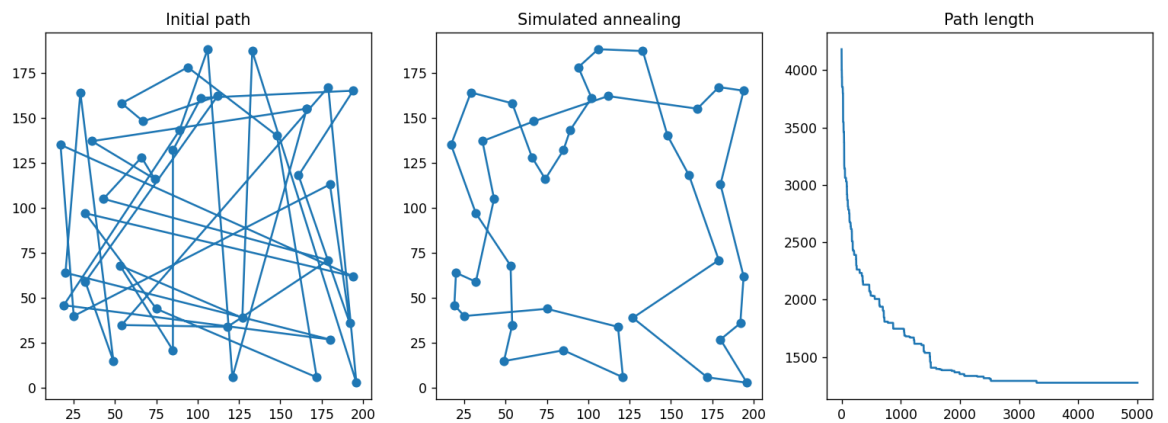
Elitizmus

Do novej generácie kopírujem najlepšiu cestu z predošlej generácie.

Výpis

Do konzole vypisujem dĺžku finálnej cesty a finálnu permutáciu miest. V samostatnom okne sa zobrazí počiatočná permutácia, finálna permutácia a graf ako sa menila dĺžka cesty. Pri path length je X os počet generácií/iterácií a Y je dĺžka cesty.



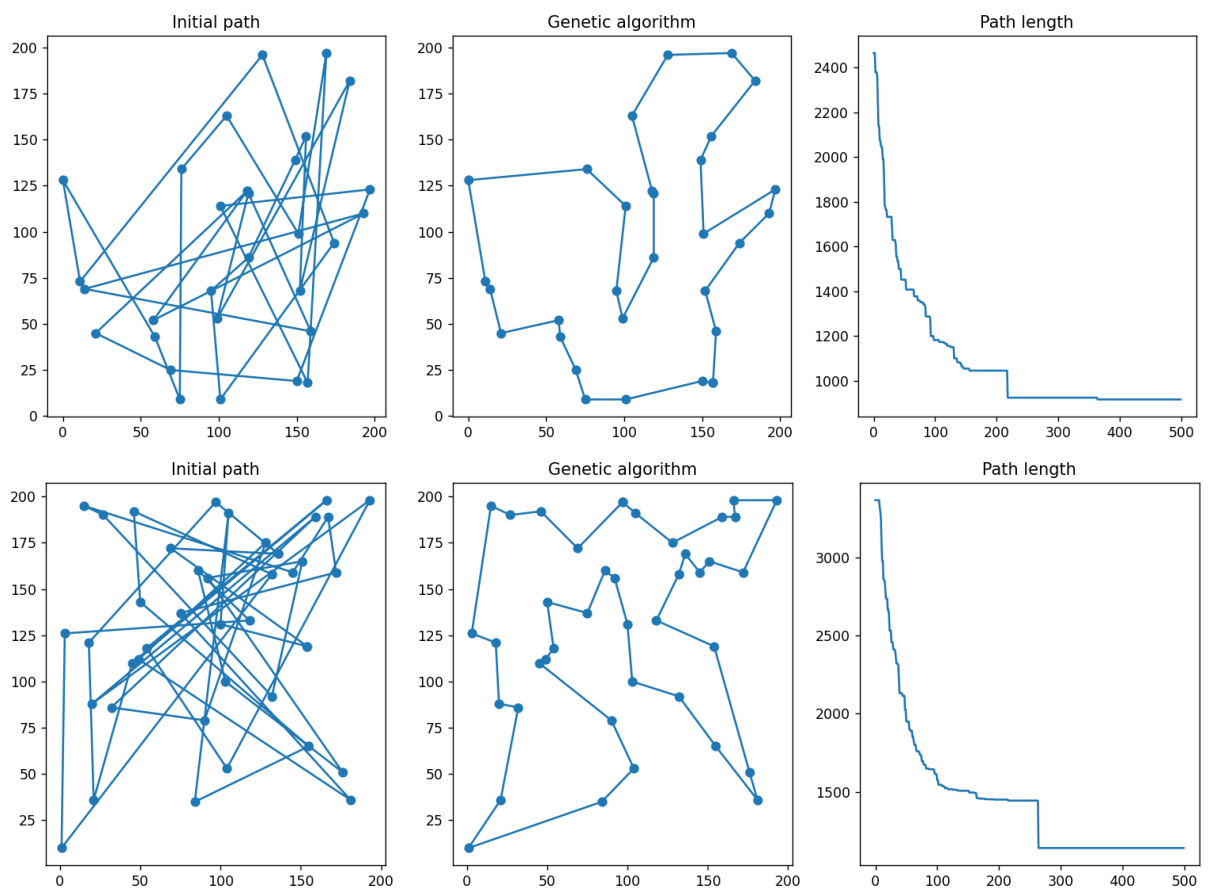


Porovnanie výsledkov

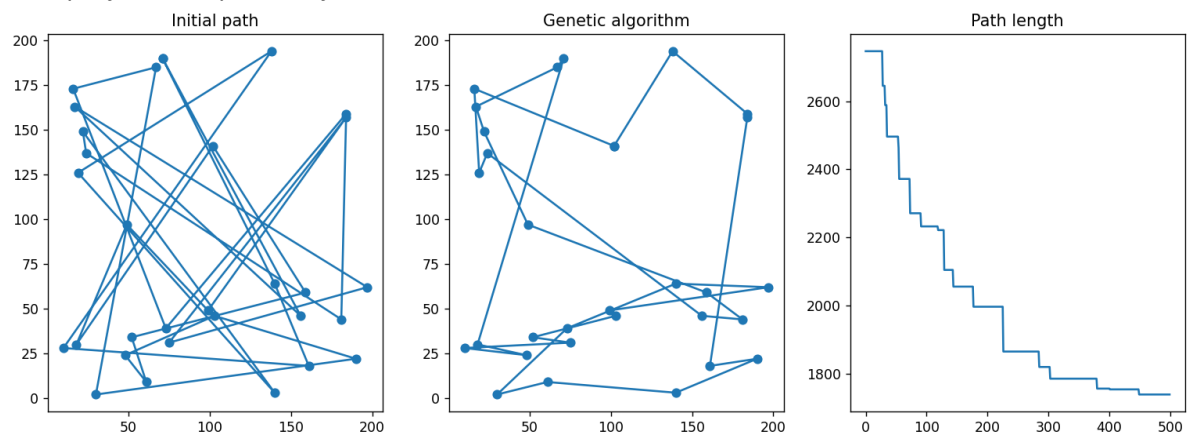
GA trvá dlhšie ale nájde optimálnu cestu pri použití turnaju. Pri rulete mi algoritmus nájde o niečo horšiu optimálnu trasu než turnaj. Simulované žíhanie beží v porovnaní s GA nájde optimálnu cestu rýchlejšie.

Genetický algoritmus

Na obrázkoch sú dve rôzne postupnosti 30 a 40 miest obe majú veľkosť populácie 100 a počet populácií 500. Metóda na výber rodičov je turnaj z 2 permutácií.

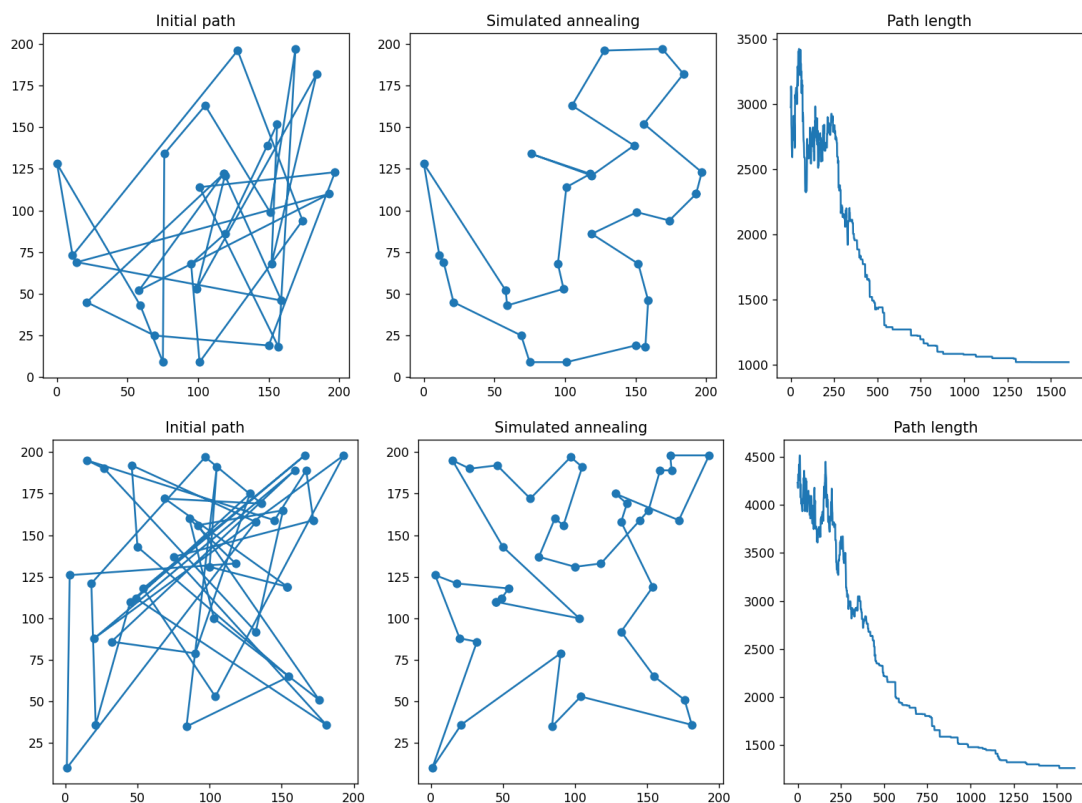


Pri rulete mi vo väčšine prípadov šťastie nepraje, zrejme kvôli tomu, že sa výsledky líšia a nezlepšujú sa ako pri turnaji.



Simulované žihanie

Pre oba obrázky som použil počiatočnú teplotu 1000 a chladenie 0.99. Na obrázku je 30 a 40 miest.



Možné vylepšovanie

Mohol som implementovať dynamickú pravdepodobnosť pre crossover a mutácie pri genetickom algoritme, ktoré by sa zvyšovali ak sa najkratšia cesta dlho nezmenila. Ďalej sa mohla použiť iná metóda na crossover alebo mutácie. Pri simulovanom žihaní sa môže vyberať z viacerých nasledovníkov. Ešte sa mohla použiť lepšia select parent metóda.