

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4
15. 11. 2022

ZADANIE 2 – Obchodný cestujúci

Filip Remšík

Zadanie

Zadaním úlohy je nájsť čo najkratšiu cestu medzi mestami. Na vstupe dostaneme 20 až 30 miest, ktoré sú rozmiestnené po mape. Cestovateľ musí prejsť všetkými týmito mestami práve raz a vrátiť sa do mesta z ktorého vychádzal.

Typ úlohy:

Použite na riešenie úlohy algoritmus zakázaného prehľadávania a simulovaného žihania.

Opis riešenia

Fitness funkcia

Slúži nám na ohodnotenie populácie (permutácia miest), v tomto prípade bude vyjadrovať celkovú vzdialenosť, ktorá bude prejdená medzi danou kombináciou miest. Na výpočet vzdialenosti použijeme Euklidovu vzdialenosť dvoch bodov:

$$A=[x_1,y_1]$$

$$B=[x_2,y_2]$$

$$\text{Vzdialenosť}=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$$

Zakázané prehľadávanie

1. Zo vstupnej permutácie si vytvorím potomkov (novú generáciu), pričom tých, ktorý sa nachádzajú v zakázanom zozname hneď odstránim.
2. Z generácie vyberiem najlepšieho potomka (pomocou fitness ohodnotenia).
3. Ak je potomok lepší ako rodič označím ho za rodiča a tvorím jeho potomkov.
 - a. Ak je potomok horší tak pridám rodiča do zakázaného zoznamu a pokračujem v prehľadávaní potomka, pričom hodnotu rodiča si uložíť keďže sa jedná o lokálne maximum a teoreticky sa môže jednať o optimálne riešenie.
4. Po vytvorení určitého počtu generácii hľadanie končí.

Simulované žihanie

1. Zo vstupnej permutácie si tvorím potomkov (novú generáciu).
2. Pri každom novom ho porovnávam s rodičom
3. Ak je potomok lepší ako rodič označím ho za rodiča a tvorím jeho potomkov.
 - a. Ak je potomok horší tak sa doň môžem presunúť ale iba ak spĺňa určitú pravdepodobnosť. Táto závisí od toho o koľko je potomok horší a od aktuálnej teploty.
4. Pred tvorbou novej generácie znížim teplotu.
5. Po vychladnutí (klesnutí vstupnej teploty pod zadefinovanú hodnotu) hľadanie končí.

Reprezentácia údajov

Vytváranie nových permutácií (génov)

Údaje (poradie navštívenia jednotlivých miest) sú uchovávané v jednorozmernom poli. Nový potomkovia vznikajú výmenou dvoch susedných miest, napr. (12345) → (12435)

Polia používam na ukladanie rozmiestnenia čísel, taktiež mi slúžia na ukladanie úrovne v strome, ktorú práve prehľadávam.

Súradnice miest

Súradnice miest sa nachádzajú v dvojrozmernom poli, súradnice sú náhodne vygenerované zo zvoleného intervalu.

Testovanie

Výstup do konzole

Tabu

Čas 0.234358549118042

Počiatočná trasa 7806.866318528605

Nájdene tabu 4834.661249338973

Zlepšenie 38.07167880069217 %

Žíhanie

Čas 0.0468745231628418

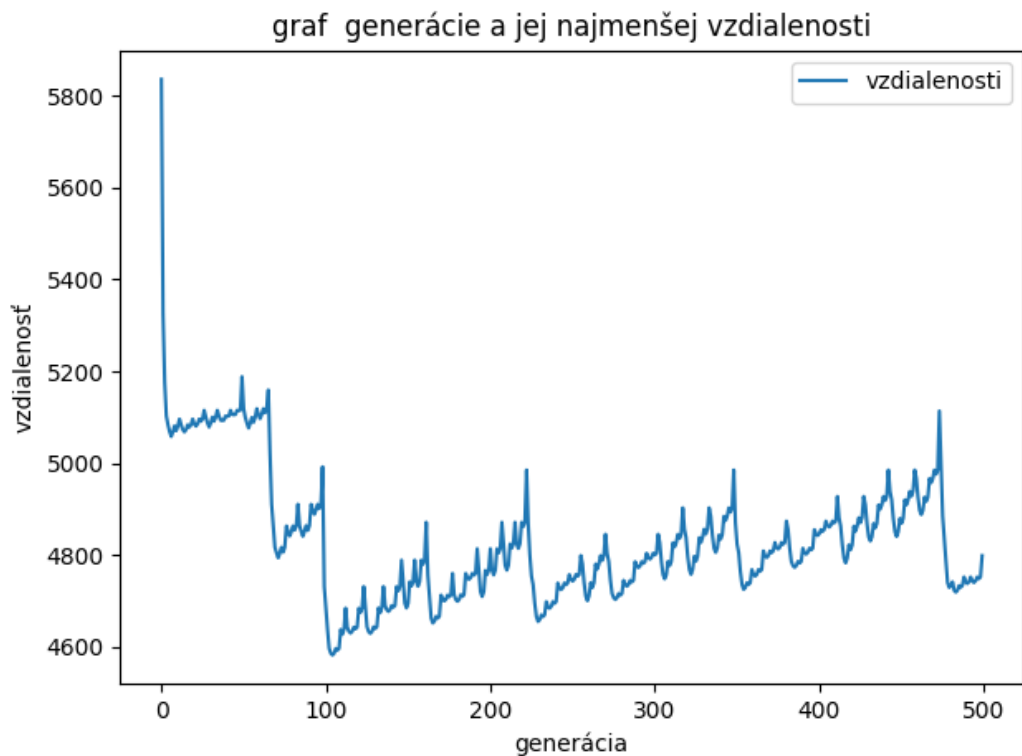
Počiatočná trasa 7806.866318528605

Nájdene simulované žíhanie 5258.3617118446

Zlepšenie 32.64439920836678 %

Zakázané prehľadávanie

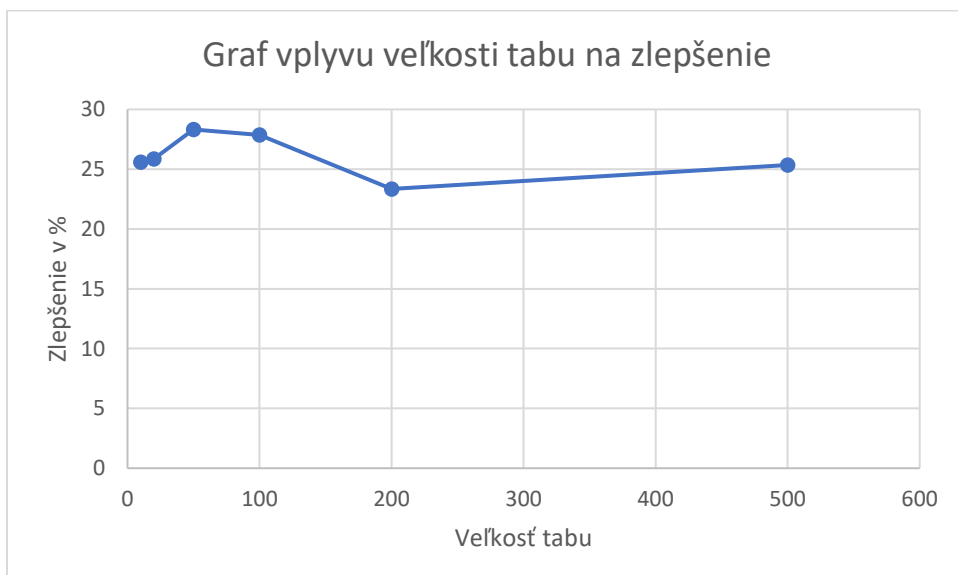
Ukážka dĺžok nájdenej cesty



Z grafu môžeme vidieť že po dosiahnutí lokálneho maxima (v našom prípade čo najmensej hodnoty) pokračovalo hľadanie v horších stavoch. (Počet uzlov 20, veľkosť tabu 1000)

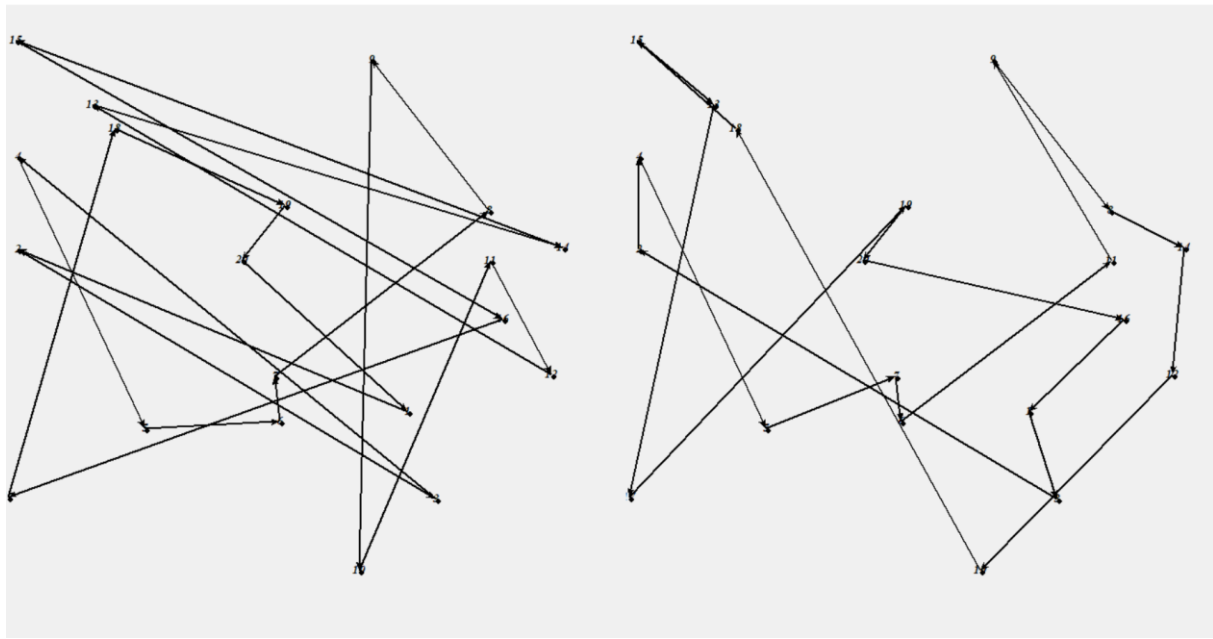
Vplyv veľkosti tabu na nájdenú vzdialenosť

Testované s 500 generáciami a 30 miestami, pre každú veľkosť bolo vykonaných 20 meraní, z ktorých bol zostavený priemer.

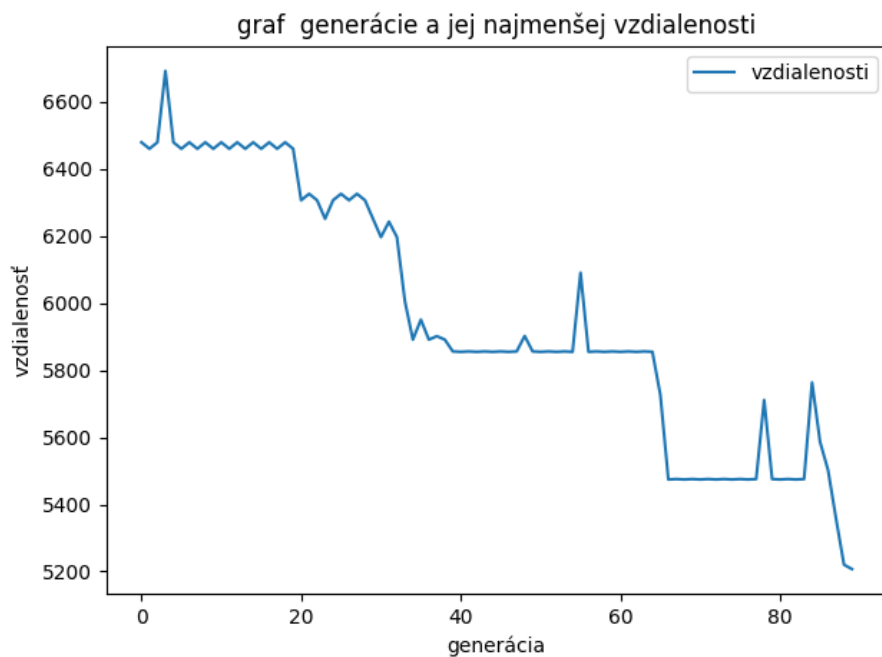


Môžeme vidieť že malá veľkosť tabu a veľkosť približujúca sa k maximálnemu počtu majú vplyv na zlepšenie cesty.

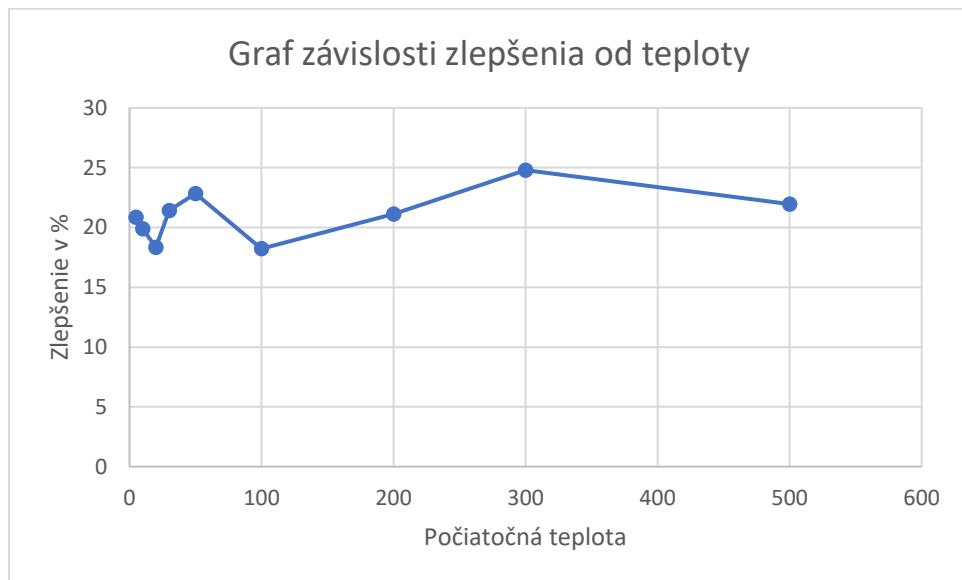
Grafické znázornenie ciest



Simulované žíhanie

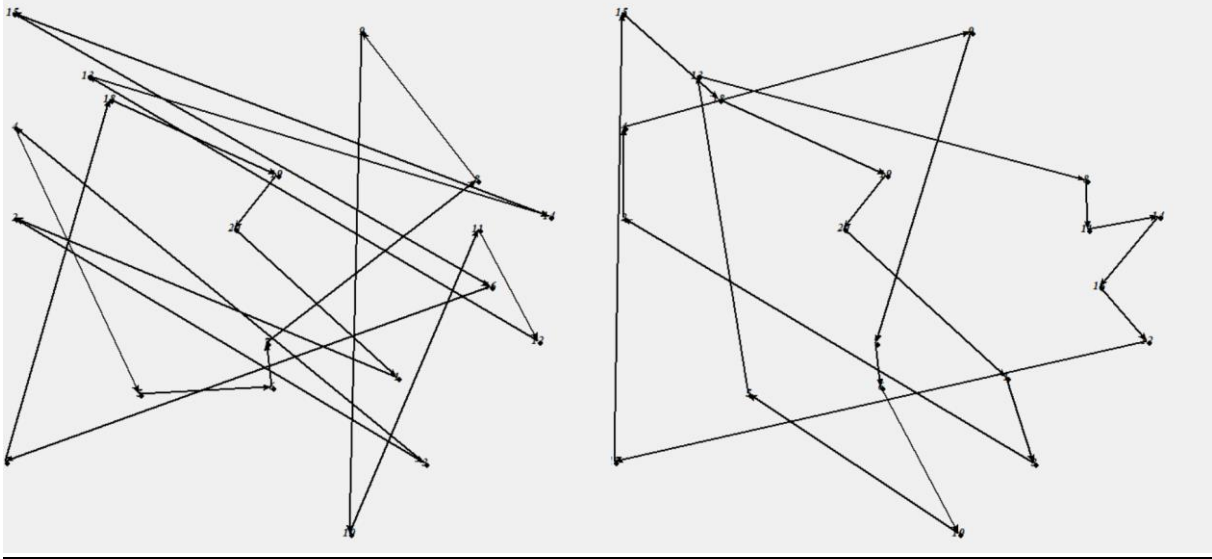


Graf nám znázorňuje ako sa postupne hľadajú nový „lepší“ jedinci. Ak bol jednoznačne nájdený lepší tak nám nastane pokles najkratšej vzdialenosti. Ak si vyberie horšieho jedinca a presune sa doň tak „cúvneme“ od lokálneho maxima a hľadáme lepšie riešenie.



Napriek tomu že spočiatku sme dosiahli lepšie výsledky aj s malou počiatočnou teplotou a pri prvotnom zvýšení nám zlepšenie kleslo, tak pri ďalšom zvyšovaní začalo zlepšenie zase narastať.

Grafické znázornenie ciest



Zhodnotenie

Na základe predchádzajúcich testov môžeme zhodnotiť že pri oboch typoch algoritmov závisí zlepšenie oproti vstupnej ceste od toho aké vstupné údaje zadám. Ak sa algoritmus slepého hľadania dostane do lokálneho maxima tak sa stáva že začne nachádzať menšie lokálne maximá. Oproti nemu simulované žihanie sa z lokálneho maxima vráti aj k horším výsledkom.

Používateľská príručka

Program je písaný v jazyku Python vo verzii 3.10.2 Po spustení je možné si navoliť vstupné údaje (počet miest, rozmer mapy, veľkosť tabu listu (tabu), maximálny počet generácii (tabu), počiatočnú teplotu (žihanie)).

Ukážka zadaných vstupných údajov

```
Zadaj počet uzlov 20
Zadaj rozmer mapy 300
Zadaj počet generácii 20
Zadaj veľkosť tabu listu 100
Zadaj počiatočnú teplotu 80
```