

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Zadanie 3 - Evolučný algoritmus

Problém obchodného cestujúceho

Dokumentácia

Peter Smreček

AIS ID: 103130

E-mail: xsmrecek@stuba.sk

Predmet: UI

Deň a čas cvičenia: Utorok 16:00

Semester: ZS 20/21

Ročník: 2.

Obsah

1.	Zadanie	2
2.	Reprezentácia údajov	2
2.1	Gén.....	2
2.2	Chromozóm.....	2
2.3	Populácia.....	3
2.4	Fitnes.....	3
3.	Opis algoritmu.....	3
3.1	Tvorba populácie	4
3.2	Výber rodičov.....	4
3.2.1	Ruleta.....	4
3.2.2	Turnaj	5
3.3	Kríženie rodičov	5
3.4	Návratové hodnoty.....	6
4.	Testovanie	6
4.1.1	Testovací scenár zo zadania.....	6
4.1.2	Testovací scenár A	8
4.1.3	Testovací scenár B	9
4.1.4	Testovací scenár C	11
5.	Používateľská príručka	13
5.1	Nastavenia vzorových vstupov	15
5.2	Knižnice použité v riešení a prevzatý kód	18
6.	Zhodnotenie	18

1. Zadanie

Zadaním úlohy je nájsť čo najkratšiu cestu medzi mestami zadanými na mape vo forme mriežky. Každé mesto je nutné cestou navštíviť práve raz a nakoniec sa vrátiť do počiatočného mesta.

Zadaných je 20 – 40 miest, každé má 2 celočíselné súradnice X a Y. Súradnice sú náhodne generované pre mapu 200 * 200. Cena cesty medzi dvoma mestami zodpovedá Euklidovej vzdialenosti – vypočíta sa pomocou Pytagorovej vety. Celková dĺžka trasy je daná permutáciou miest. Cieľom je nájsť takú permutáciu miest, ktorá bude mať celkovú vzdialenosť čo najmenšiu.

Výstupom je poradie miest a dĺžka zodpovedajúcej cesty.

Riešenie je navrhnuté použitím genetického algoritmu.

2. Reprezentácia údajov

Keďže je problém riešený genetickým algoritmom, reprezentácia údajov pripomína (aj keď vzdialene) biologické štruktúry a ich DNA.

GÉN – v tomto prípade je to dvojica čísel X a Y reprezentujúcich súradnice mesta

CHROMOZÓM – v komentároch v kóde používam pojem jedinec – je v tomto prípade permutácia všetkých génov, tj. permutácia všetkých súradníc miest

POPULÁCIA – skupina chromozómov, jedincov, udržiavaná v zozname

FITNES – funkcia na ohodnotenie chromozómy vypočítaná ako $1/(\text{dĺžka cesty})$

2.1 Gén

Gény, súradnice miest, sú náhodne vygenerované funkciou `generuj_suradnice`, ktorá vygeneruje počet miest na mape v rozsahu od 20 do 40 miest a potom pre každé mesto vygeneruje 2 súradnice ako dvojprvkový zoznam. Ak sa v zozname miest mesto s takýmito súradnicami ešte nenachádza, zapíše ho do zoznamu miest. Ak sa tam nachádza, nezapíše ho do zoznamu miest a vygeneruje ďalšie súradnice. Funkcia vráti zoznam rôznych miest, čo je vlastne dvojrozmerný zoznam, kde sa na každej z pozícií 0 až (počet miest -1) nachádzajú súradnice mesta.

2.2 Chromozóm

Surové súradnice všetkých miest vstupujú do genetického algoritmu reprezentovaného funkciou `geneticky_algoritmus`, kde sa na začiatku z pôvodných súradníc vytvorí inštancia triedy `Graf`. Inštancie tejto triedy reprezentujú jedincov v populácii.

Trieda `Graf` ďalej obsahuje aj funkciu `permutuj`, ktorá vytvorí náhodnú permutáciu miest z tohto jedinca a vytvorí a vráti nového jedinca, teda novú inštanciu triedy `Graf`. Toto sa používa pri úvodnom vytváraní prvej generácie v genetickom algoritme, kedy je z pôvodného grafu vygenerovaných toľko nových, permutovaných grafov, koľko je zadaný počet členov populácie.

2.3 Populácia

Populácia je v programe reprezentovaná v zozname jedincov, ktorý je vytvorený a modifikovaný vo funkcii `geneticky_algoritmus`.

2.4 Fitnes

Trieda `Graf` pri vytváraní novej inštancie, teda nového jedinca, vezme zoznam súradníc miest, uloží si ho a vypočíta z neho počet miest, dĺžku cesty medzi mestami v aktuálnom poradí a fitnes jedinca. Klasicky obsahuje metódy `get` pre tieto hodnoty.

Vzdialenosť 2 miest je počítaná euklidovsky, pomocou Pytagorovej vety. Celková dĺžka jedinca je počítaná ako súčet vzdialeností jednotlivých miest na mape v poradí v akom boli zadané. Tento výpočet realizujú funkcie triedy `Graf` - `dlzka_cesty` a `euklidova_vzdialenost`. Výpočet fitnes funkcie realizuje funkcia `fitnes_vypocet` triedy `Graf`. Výpočet fitnes je realizovaný podľa vzorca $1/(\text{dĺžka cesty})$.

Mutácie jedincov sú realizované funkciami `pravdepodobnost`, ktorá vráti `True` so zadanou pravdepodobnosťou a funkciou `mutacia`, ktorá v prípade návratu `True` z funkcie `pravdepodobnost` urobí mutáciu súčasného jedinca. Mutácia sa vykoná takým spôsobom, že sa vyberie jedno mesto, jeden gén súčasného jedinca a vymení sa s vedľajším génom.

3. Opis algoritmu

Genetický algoritmus je reprezentovaný funkciou `geneticky_algoritmus`. Parametrami tejto funkcie sú:

- pole pôvodných súradníc
- funkcia výberu rodičov
- počet generácií, predvolene nastavený na 2000
- pravdepodobnosť mutácií detí, predvolene nastavená na 0,1
- počet členov populácie, predvolene nastavený na 40
- boolean ponechania najlepšieho jedinca, predvolene nastavený na `False`
- boolean použitia metódy novej krvi pri tvorbe novej generácie, predvolene nastavený na `False`
- boolean vypisovania čiastkových výpisov pre každú v poradí 1000. generáciu, predvolene nastavený na `True`

Funkcia po svojom spustení vypíše, s akými nastaveniami bola spustená.

3.1 Tvorba populácie

Z pôvodne zadaného zoznamu súradníc sa vytvorí inštancia triedy **Graf**. Vytváraním náhodných permutácií pôvodného grafu sa vytvorí prvá populácia o zadanom počte jedincov. Každému z týchto jedincov pri jeho vytváraní trieda **Graf** vypočíta počet miest, dĺžku cesty a fitness.

Takto vytvorená populácia vstupuje do evolučného cyklu ktorý prebehne zadaný počet generácií krát. V tomto cykle je vytvorené pole pravdepodobností, v ktorom sú uložené číselné hodnoty fitness všetkých jedincov v populácii. Index do poľa pravdepodobností, na ktorom je uložený fitness jedinca zodpovedá indexu do poľa populácie, kde je uložený konkrétny jedinec. Takto vytvorené pole pravdepodobností vstupuje do funkcie na výber rodičov. Táto funkcia je jedným zo vstupných argumentov tohto algoritmu. Výbery rodičov budú samostatne popísané v nasledujúcej podkapitole. Funkcia na výber rodičov vráti dvojicu rodičov. Dvojíc rodičov je polovičný počet oproti členom populácie, pretože každá dvojica rodičov splodí 2 potomkov. Z toho vyplýva, že v mojom programe musí byť počet členov populácie párne číslo.

Tieto dvojice rodičov ďalej vstupujú do funkcie **porod**, ktorá vytvorí deti týchto zadaných dvojíc rodičov. Tvorba detí rodičov bude popísaná v samostatnej podkapitole. Počet detí vrátený touto funkciou je zhodný s počtom členov pôvodnej populácie.

V prípade, že je zvolená možnosť na použitie metódy „Nová Krv“, 2 náhodné deti z novej populácie budú zabité a nahradené 2 novými jedincami, náhodnými permutáciami pôvodného grafu.

V prípade zvolenia možnosti „Ponechanie najlepšieho jedinca“ bude najhoršie dieťa v novej populácii nahradené najlepším z rodičov.

Populácia rodičov je na konci cyklu nahradená populáciou detí a cyklus evolúcie sa vracia na začiatok, kedy sa opäť vyberajú rodičia, tvoria deti, nahrádza populácia rodičov populáciou detí. Cyklus prebehne zadaný počet generácií krát.

3.2 Výber rodičov

Do funkcie evolučného algoritmu vstupuje ako argument funkcia pre výber rodičov. Implementoval som 2 rôzne metódy na výber rodičov.

3.2.1 Ruleta

Výber rodičov pomocou rulety je reprezentovaný funkciou **ruleta**. Táto funkcia dostane ako vstupné argumenty počet členov populácie, indexy týchto členov a pravdepodobnosti jedincov namapované na tieto indexy.

Kvôli použitiu prevzatej funkcie váženého výberu z poľa z balíka **numpy**, je nutné každý prvok poľa vydeliť celkovým súčtom všetkým prvkov tohto poľa. Tým sa dosiahne, že súčet všetkých prvkov poľa je cca 1. Až takto upravené pole pravdepodobností môže vstúpiť do funkcie **choice** z balíka **numpy**.

Funkcia **choice** vráti indexy 2 rodičov vybraných náhodne, avšak váženým výberom z celej populácie. Indexy rodičov sú zapísané do poľa rodičov. Takýto vážený výber sa udeje

polovičný počet členov populácie krát. Týmto sa dosiahne, že najlepší jedinci majú najväčšiu šancu sa rozmnožiť a najhorší jedinci majú najmenšiu šancu sa rozmnožiť, ale stále môžu byť vybratí. To, že môžu byť vybratí aj slabší jedinci je dôležité, pretože aj slabší jedinci môžu niesť dôležitú genetickú informáciu.

Funkcia `ruleta` po váženom výbere všetkých dvojíc rodičov vráti pole obsahujúce indexy rodičov.

3.2.2 Turnaj

Turnajový výber rodičov prebieha tak, že z pôvodnej populácie vyberiem náhodne 2 dvojice jedincov. Z každej takejto dvojice jedincov vyberiem jedinca s väčšou hodnotou fitness. Týchto 2 lepších jedincov pridám do poľa ako dvojicu rodičov, ak sú navzájom odlišní.

Turnajový výber zabezpečuje, že sa môžu za rodiča vybrať aj horší jedinci, nesúci dôležitú genetickú informáciu. Pri výbere viacerých jedincov vstupujúcich do turnaja by hrozilo, že vyberaní budú iba tí najlepší, čo by pripomínalo riešenie elitarizmom. Preto som zvolil, že do turnaja vstupujú vždy len 2 jedinci.

Funkcia `turnaj` má rovnaké argumenty aj návratovú hodnotu ako funkcia `ruleta`.

3.3 Kríženie rodičov

Po tom, ako je konkrétnou funkciou vytvorené pole rodičov, začína kríženie. Kríženie zabezpečuje funkcia `porod`, ktorá vráti 2 deti jednej dvojici rodičov. Táto funkcia je volaná v cykle pre všetky dvojice rodičov. Tým sa zabezpečí konštantný počet členov populácie počas všetkých generácií.

Funkcia `porod` dostane ako argumenty 2 rodičov a pravdepodobnosť mutácie. Náhodne vyberie 2 rôzne čísla z rozsahu 0 až počet miest tak, že prvé číslo je menšie ako druhé. Tieto čísla predstavujú súvislý úsek génov, ktoré sa premiestnia do dieťaťa.

Následne je volaná funkcia `mixuj`, ktorá dostane v argumentoch počet miest, polia súradníc miest oboch rodičov a obe hranice. Následne z rodiča 1 vyberie úsek miest medzi hranicami. Z rodiča 2 vyberie všetky ostatné mestá v poradí, v akom sa nachádzajú v rodičovi 2. Tieto úseky súradníc spojí tak, že v dieťati sa nachádza súvislý úsek génov z rodiča 1 na tom istom mieste ako v rodičovi 1 a pred ním a za ním sa nachádzajú zvyšné gény z rodiča 2. Funkcia vracia pole súradníc dieťaťa. Táto funkcia sa spúšťa pre každú dvojicu rodičov dvakrát, tak, že druhýkrát sa vymení poradie rodičov.

Po dvojnásobnom spustení funkcie `mixuj` s rozdielnym poradím rodičov sa vytvorili 2 polia súradníc pre 2 deti. Tieto deti – noví jedinci sú vytvorené ako inštancie triedy `Graf`. Nad každým dieťaťom je potom zavolaná metóda triedy `Graf` – `mutacia`. Táto metóda so zadanou pravdepodobnosťou vykoná vyššie opísanú mutáciu na nových jedincoch.

Funkcia `porod` vracia pole týchto 2 nových jedincov.

3.4 Návratové hodnoty

Funkcia `geneticky_algoritmus` počas evolučného cyklu počíta a zapisuje do poľa priemerný fitness každej generácie a fitness najlepšieho jedinca v každej generácii. Po dokončení evolučného cyklu funkcia vypíše najlepšieho jedinca, jeho cestu, dĺžku cesty a jeho fitness. V prípade, ak bola zvolená možnosť, že najlepší jedinec nepostupuje automaticky do ďalšej generácie, bude vypísaný aj najlepší jedinec z poslednej generácie, pretože ten je v skutočnosti výstupom takto nastaveného programu.

Funkcia vracia pole priemerných hodnôt fitness generácie, pole maximálnych hodnôt fitness generácie a samotného najlepšieho jedinca. Z týchto hodnôt sa dá pomocou funkcie `generuj_excel` vytvoriť tabuľka.

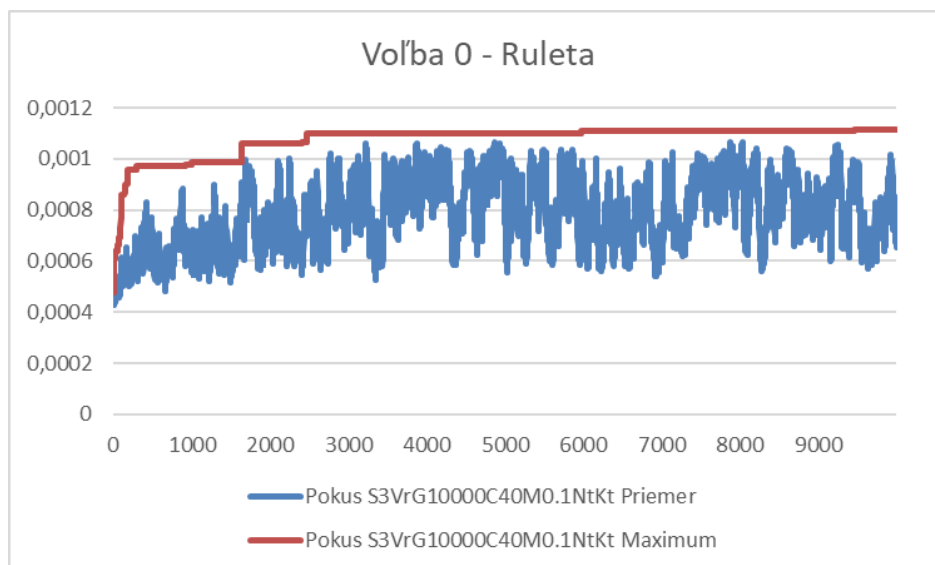
4. Testovanie

Pre potreby testovania som si zvolil okrem zadaného testovacieho scenáru aj 3 vlastné. Funkcia `riadic` manažuje ovládanie programu a dáva používateľovi na výber z možností spustenia testov a zadania vlastných parametrov. V tejto časti rozoberiem celkovo 4 testovacie scenáre. Keďže som zistil, že nastavenie 40 jedincov v populácii, 10000 generácií, pravdepodobnosť mutácií 0,1 a ponechávanie najlepšieho jedinca spolu s aktívnou Novou Krvou dosahuje najlepšie výsledky, tieto nastavenia v testovacích scenároch nemením a jediné čo mením, je seed pre random a spôsob výberu rodičov.

Testovacie výstupy sú výsledkom funkcie `automaticky_testovac`, ktorá vytvorila excel súbory pre všetky testovacie scenáre a zachytíla výstup z konzoly do textového súboru. Všetky súbory sú označené kódom a obsahujú príslušné hodnoty a grafy.

4.1.1 Testovací scenár zo zadania

Tento testovací scenár obsahuje počiatočnú postupnosť súradníc zo zadania. Viacerými testami som zistil, že najlepší možný dosiahnutý výsledok dĺžky cesty je 895,7.



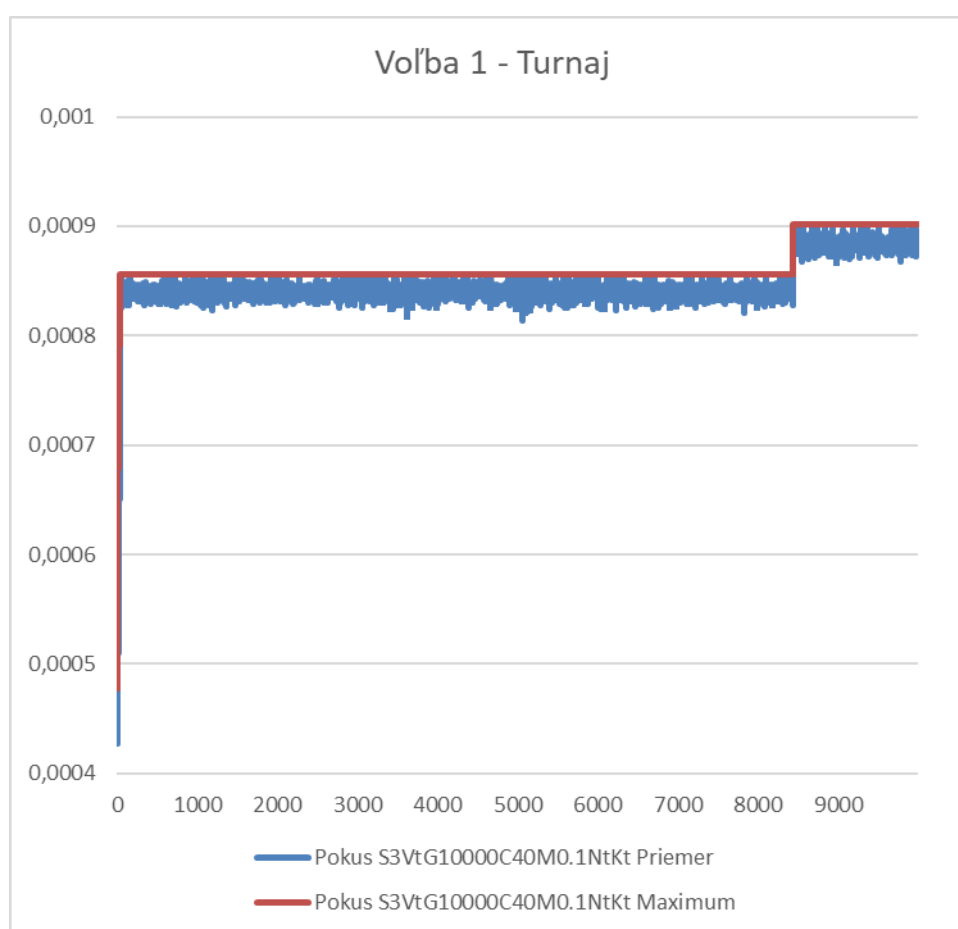
Použitie rulety prinieslo postupné zlepšovanie riešenia s narastajúcim počtom generácií. Program sa dokázal efektívne dostať z lokálnych optím. Z krivky priemerného fitness generácie je vidno, že v generácií sa nachádzajú lepší, aj horší jedinci.

Výstup programu je:

S dĺžkou cesty 895.706325022881 je najlepší najdený jedinec z poslednej populácie:

[200, 160] [180, 100] [180, 60] [200, 40] [160, 20] [100, 80] [100, 40] [60, 20] [20, 20] [20, 40] [60, 80] [40, 120] [20, 160] [60, 200] [100, 180] [80, 160] [120, 120] [140, 140] [180, 200] [140, 180]

Jeho fitness je 0.001116437354592148



Riešenie turnajom na dlhú dobu uviazlo v lokálnom optime a v porovnaní s ruletou našlo horšie riešenie. Jediný posun k lepšiemu nastal skokovo až zhruba v 8000. generácií a pravdepodobne bol spôsobený mutáciou. Z krivky priemerného fitness generácie je jasné, že generácia obsahovala len jedincov s veľmi podobnými hodnotami fitness.

Výstup programu je:

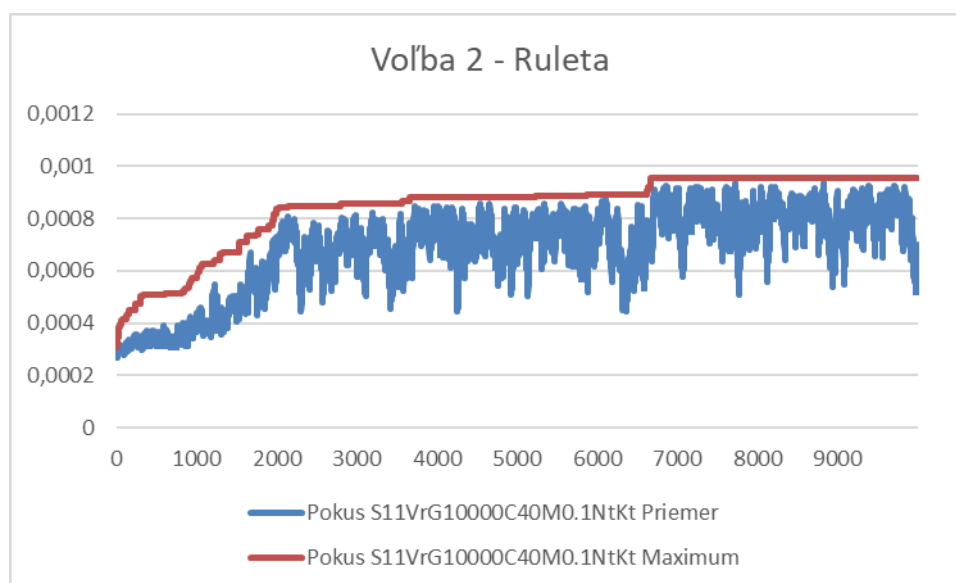
S dĺžkou cesty 1109.527054478831 je najlepší najdený jedinec z poslednej populácie:

[200, 160] [180, 100] [140, 180] [80, 160] [100, 80] [100, 40] [180, 60] [200, 40] [160, 20] [60, 20] [20, 20] [20, 40] [60, 80] [40, 120] [20, 160] [60, 200] [100, 180] [120, 120] [140, 140] [180, 200]

Jeho fitness je 0.0009012849177163344

4.1.2 Testovací scénár A

Testovací scénár A obsahuje náhodnú sadu súradníc.



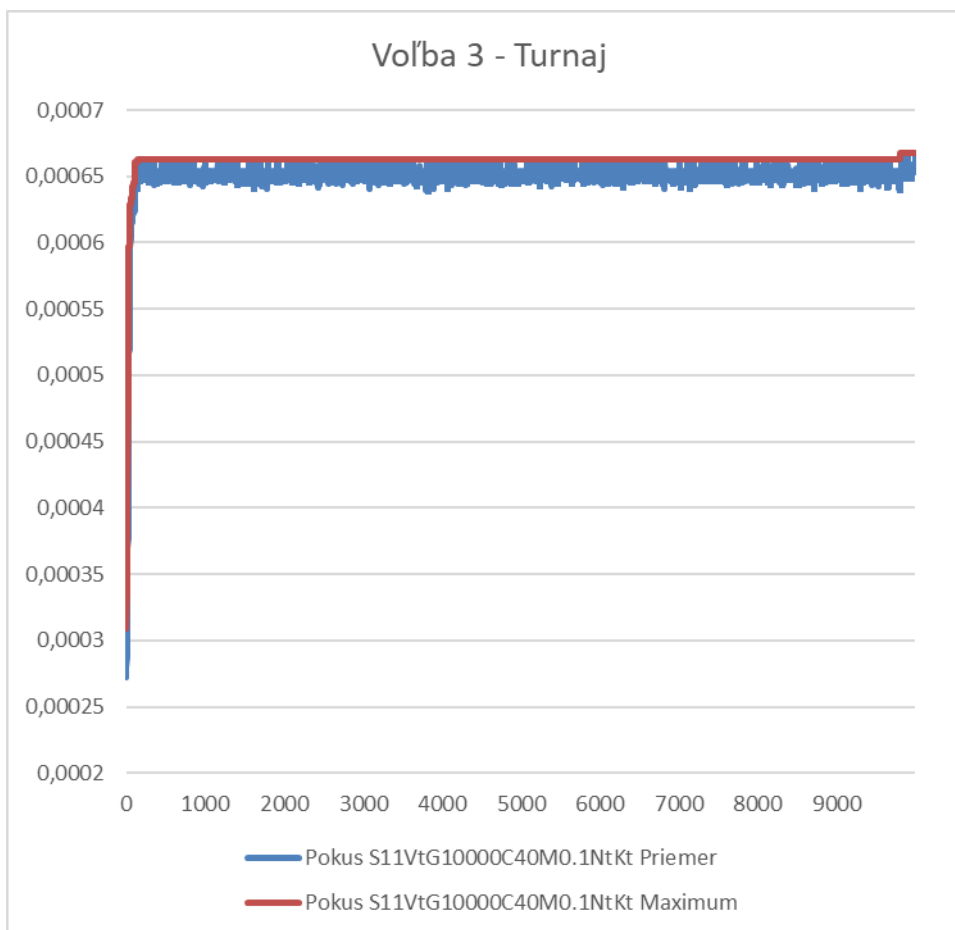
Použitie rulety prinieslo postupné zlepšovanie riešenia s narastajúcim počtom generácií. Program sa dokázal efektívne dostať z lokálnych optím. Z krivky priemerného fitness generácie je vidno, že v generáciách sa nachádzajú lepší, aj horší jedinci.

Výstup programu je:

S dĺžkou cesty 1046.6791010240556 je najlepší nájdený jedinec z poslednej populácie:

[114, 192] [138, 185] [166, 187] [178, 187] [192, 168] [125, 132] [91, 152] [92, 176] [70, 171] [19, 153] [1, 131] [24, 109] [72, 114] [89, 107] [71, 82] [91, 81] [117, 90] [125, 79] [129, 74] [136, 68] [145, 67] [155, 70] [160, 78] [183, 79] [181, 71] [160, 45] [183, 13] [142, 7] [96, 23] [27, 44] [17, 43] [18, 48] [37, 81] [27, 133] [34, 140] [76, 161]

Jeho fitness je 0.0009554026625941174



Riešenie turnajom na dlhú dobu uviazlo v lokálnom optime a v porovnaní s ruletou našlo podstatne horšie riešenie. Drobný posun k lepšiemu nastal až tesne pred ukončením testu a pravdepodobne bol spôsobený mutáciou. Z krivky priemerného fitness generácie je jasné, že generácia obsahovala len jedincov s veľmi podobnými hodnotami fitness.

Výstup programu je:

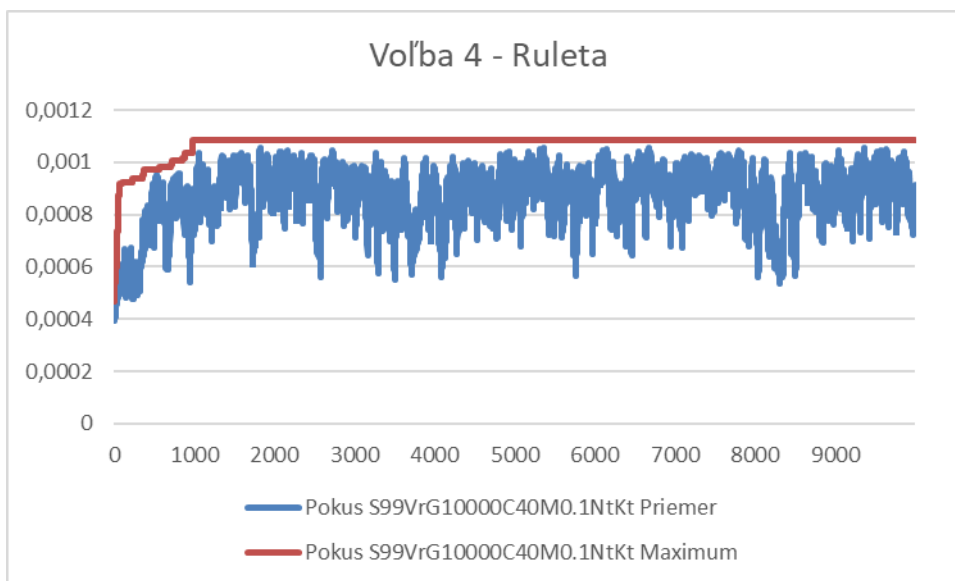
S dĺžkou cesty 1497.440883470884 je najlepší najdený jedinec z poslednej populácie:

```
[125, 132] [192, 168] [178, 187] [166, 187] [183, 79] [181, 71]
[160, 78] [155, 70] [160, 45] [136, 68] [89, 107] [96, 23] [27, 44]
[17, 43] [18, 48] [37, 81] [24, 109] [71, 82] [138, 185] [114, 192]
[91, 152] [91, 81] [125, 79] [142, 7] [183, 13] [145, 67] [129, 74]
[117, 90] [72, 114] [34, 140] [27, 133] [1, 131] [19, 153] [70, 171]
[76, 161] [92, 176]
```

Jeho fitness je 0.0006678059955743447

4.1.3 Testovací scénár B

Testovací scénár B obsahuje náhodnú sadu súradníc.



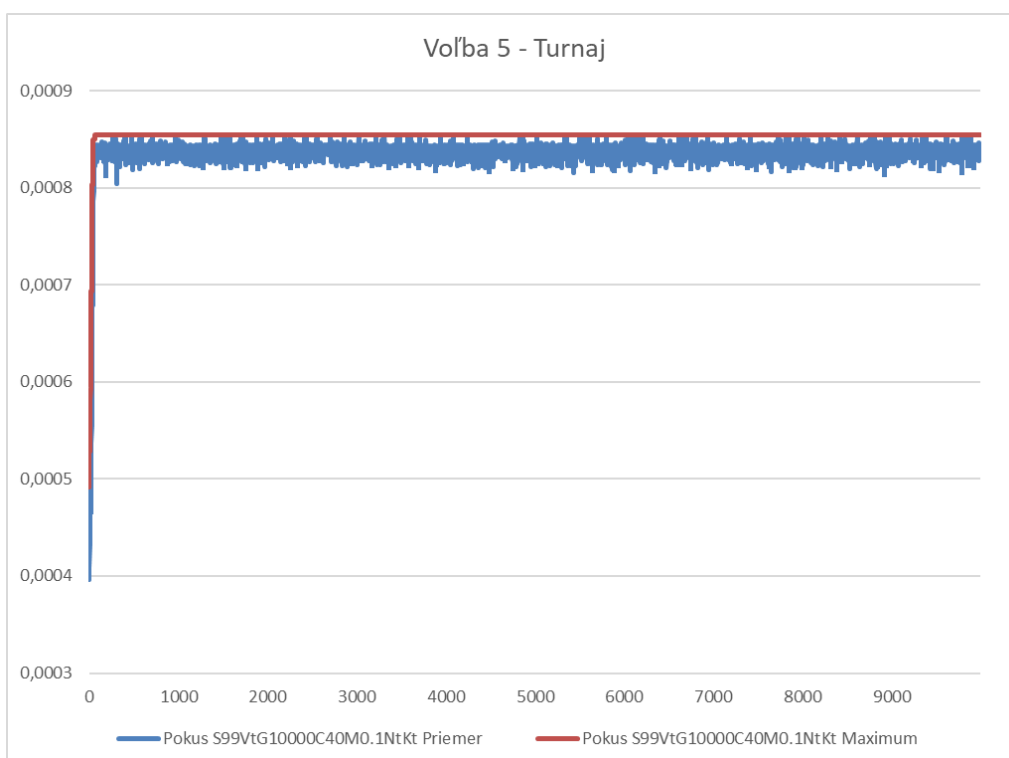
Použitie rulety prinieslo pomerne rýchle nájdenie optima. Z krivky priemerného fitness generácie je vidno, že v generácií sa nachádzajú lepší, aj horší jedinci, no ďalšie zlepšenie maximálneho fitness to neprinieslo.

Výstup programu je:

S dĺžkou cesty 922.3814307818993 je najlepší nájdený jedinec z poslednej populácie:

[129, 151] [140, 168] [193, 176] [197, 180] [178, 187] [130, 192] [96, 181] [73, 160] [35, 185] [21, 169] [22, 154] [5, 140] [59, 107] [168, 68] [200, 20] [99, 19] [79, 47] [62, 17] [59, 39] [35, 55] [27, 62]

Jeho fitness je 0.0010841501862763041



Riešenie turnajom takmer okamžite uviazlo v lokálnom optime a nezmenili to ani mutácie. Rozdiel, medzi riešeniami ale nie je taký výrazný. Z krivky priemerného fitness generácie je jasné, že generácia obsahovala len jedincov s veľmi podobnými hodnotami fitness.

Výstup programu je:

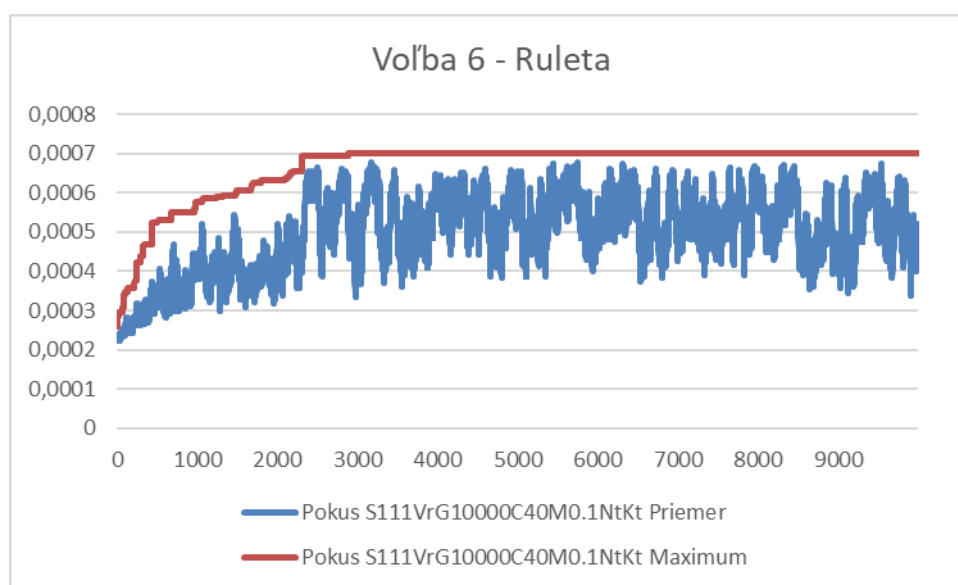
S dĺžkou cesty 1170.7134080200497 je najlepší nájdený jedinec z poslednej populácie:

[21, 169] [35, 185] [27, 62] [59, 107] [129, 151] [140, 168] [168, 68]
[200, 20] [99, 19] [79, 47] [62, 17] [59, 39] [35, 55] [73, 160] [96, 181]
[178, 187] [197, 180] [193, 176] [130, 192] [22, 154] [5, 140]

Jeho fitness je 0.0008541800180551737

4.1.4 Testovací scénár C

Testovací scénár C obsahuje náhodnú sadu súradníc.



Použitie rulety prinieslo pomerne rýchle nájdenie optima. Z krivky priemerného fitness generácie je vidno, že v generáciách sa nachádzajú lepší, aj horší jedinci, no ďalšie zlepšenie maximálneho fitness to neprinieslo.

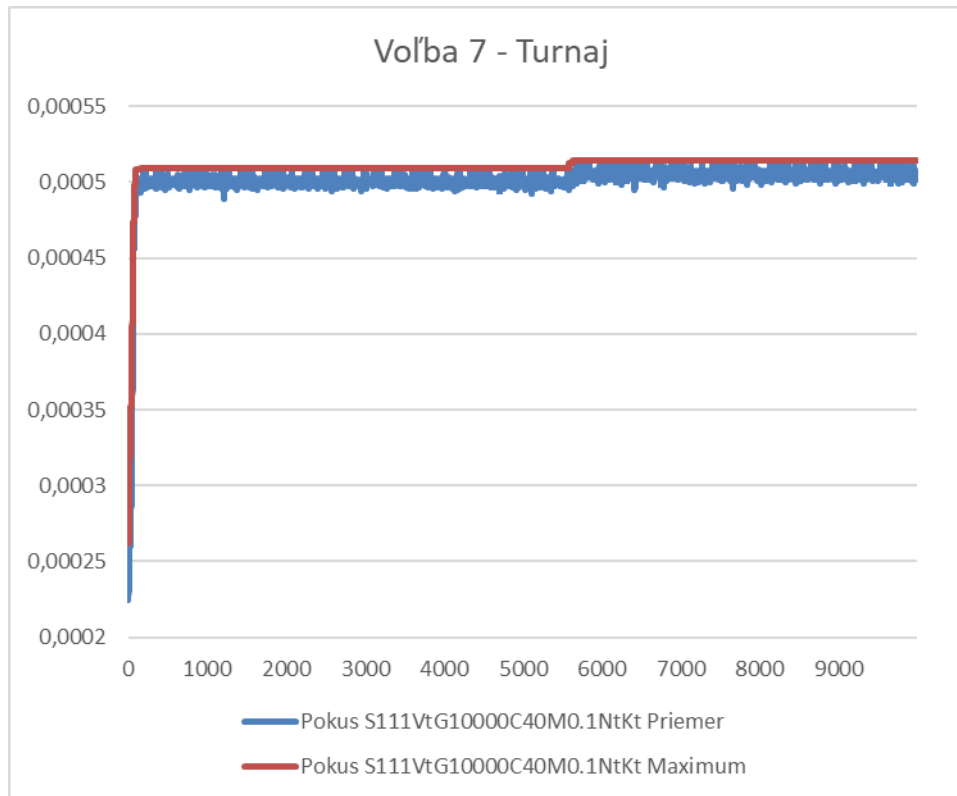
Výstup programu je:

S dĺžkou cesty 1428.9356680989954 je najlepší nájdený jedinec z poslednej populácie:

[135, 54] [162, 21] [185, 7] [198, 13] [173, 37] [170, 50] [171, 78]
[88, 123] [74, 139] [1, 137] [12, 173] [40, 199] [54, 200] [97, 166]
[157, 137] [170, 147] [193, 112] [197, 78] [200, 64] [197, 69] [187, 94]
[177, 103] [140, 131] [134, 173] [105, 199] [37, 186] [19, 169]

[33, 160] [66, 118] [108, 86] [123, 78] [91, 42] [75, 40] [28, 14]
 [12, 10] [4, 17] [50, 62] [50, 68] [31, 88] [59, 88]

Jeho fitness je 0.0006998215681258514



Riešenie turnajom na pomerne dlhú dobu uviazlo v lokálnom optime a v porovnaní s ruletou našlo podstatne horšie riešenie. Drobný posun k lepšiemu nastal až tesne pred ukončením testu a pravdepodobne bol spôsobený mutáciou. Z krivky priemerného fitness generácie je jasné, že generácia obsahovala len jedincov s veľmi podobnými hodnotami fitness.

Výstup programu je:

S dĺžkou cesty 1945.1003834768828 je najlepší nájdený jedinec z poslednej populácie:

[50, 68] [170, 147] [177, 103] [197, 69] [200, 64] [197, 78] [193, 112] [187, 94] [198, 13] [185, 7] [173, 37] [123, 78] [1, 137] [33, 160] [19, 169] [12, 173] [97, 166] [134, 173] [54, 200] [40, 199] [74, 139] [66, 118] [59, 88] [31, 88] [37, 186] [105, 199] [157, 137] [140, 131] [135, 54] [162, 21] [170, 50] [171, 78] [108, 86] [50, 62] [4, 17] [12, 10] [28, 14] [75, 40] [91, 42] [88, 123]

Jeho fitness je 0.0005141122836100067

5. Používateľská príručka

Program obsahuje 2 riadiace funkcie. Spúšťajú sa z funkcie main. Funkcia automaticky_testovac bola použitá na generovanie výstupov pre všetky testovacie scenáre. V main funkcii je zakomentované jej volanie. Jeho odkomentovaním sa opätovne dajú vygenerovať rovnaké výstupy, aké popisujem v časti Testovanie.

Funkcia riadic je funkcia obsahujúca konzolové používateľské rozhranie a používateľ ňou môže ovládať program.

Po spustení programu sa vypíše menu:

```
Zvol 0 pre spustenie so suradnicami zo zadania s ruletou
Zvol 1 pre spustenie so suradnicami zo zadania s turnajom
Zvol 2 pre testovaciu sadu suradnic A s testovacimi nastaveniami a ruletou
Zvol 3 pre testovaciu sadu suradnic A s testovacimi nastaveniami a turnajom
Zvol 4 pre testovaciu sadu suradnic B s testovacimi nastaveniami a ruletou
Zvol 5 pre testovaciu sadu suradnic B s testovacimi nastaveniami a turnajom
Zvol 6 pre testovaciu sadu suradnic C s testovacimi nastaveniami a ruletou
Zvol 7 pre testovaciu sadu suradnic C s testovacimi nastaveniami a turnajom
Zvol 8 pre vygenerovanie novej nahodnej sady suradnic s vlastnymi nastaveniami
```

Používateľ si môže zvoliť jeden z testovacích scenárov 0 – 7 alebo voľbou 8 nastaviť vlastný testovací scenár.

Ak používateľ v tomto menu zvolí číslo 0, spustí sa algoritmus s nastaveniami pre testovací scenár 0. Po úspešnom vypísaní výstupu sa program spýta, či si používateľ praje vygenerovať excel tabuľku s výsledkami. Po tejto voľbe sa program spýta, či chce používateľ pokračovať. Ak používateľ zadá „P“ pre pokračovanie, opäť sa vráti do menu.

```
-----
Poradove cislo generacie 10000
Oznacenie generacie 9999
Priemerny fitness populacie v tejto generacii je 0.0006517359478357669
Najlepsi jedinec tejto generacie ma fitness 0.0008446273798952312
Najlepsi jedinec zo vsetkych populacii s dlzkou 895.706325022881
-----
S dlzkou cesty 895.706325022881 je najlepsijedenec z poslednej populacie:
[200, 160] [180, 100] [180, 60] [200, 40] [160, 20] [100, 80] [100, 40] [60, 20] [20, 20] [20, 40]
[60, 80] [40, 120] [20, 160] [60, 200] [100, 180] [80, 160] [120, 120] [140, 140] [180, 200] [140, 180]
Jeho fitness je 0.001116437354592148
-----
Vygenerovat excel subor s vystupmi? Zadaaj a pre ano, n pre nie: n
Ak si prajete pokračovat, zadajte P: P
-----
Zvol 0 pre spustenie so suradnicami zo zadania s ruletou
Zvol 1 pre spustenie so suradnicami zo zadania s turnajom
Zvol 2 pre testovaciu sadu suradnic A s testovacimi nastaveniami a ruletou
Zvol 3 pre testovaciu sadu suradnic A s testovacimi nastaveniami a turnajom
Zvol 4 pre testovaciu sadu suradnic B s testovacimi nastaveniami a ruletou
Zvol 5 pre testovaciu sadu suradnic B s testovacimi nastaveniami a turnajom
Zvol 6 pre testovaciu sadu suradnic C s testovacimi nastaveniami a ruletou
Zvol 7 pre testovaciu sadu suradnic C s testovacimi nastaveniami a turnajom
Zvol 8 pre vygenerovanie novej nahodnej sady suradnic s vlastnymi nastaveniami
```

Ak používateľ v menu zvolí číslo 8, program sa ho spýta na nastavenie seedu. Seed sa v tejto možnosti vždy nastavuje manuálne, kvôli nutnosti opakovať niektoré testy. Potom sa program bude postupne pýtať na nastavenie vstupných argumentov:

```
Zvol 8 pre vygenerovanie novej nahodnej sady suradnic s vlastnymi nastaveniami
8
-----
Bola zvolena moznost 8
Zadaj seed pre random: 1453
Tato moznost zopoveda suradniciam vygenerovany randomom so seedom 1453
Zadaj parny pocet clenov populacie: 30
Zadaj pocet generacii: 2000
Zadaj sposob vyberu rodicov, pre ruletu zvol r, pre turnej zvol t: r
Zadaj pravdepodobnosti mutacii deti: 0.07
Ponechat najlepsiho? Zadaj a pre ano, n pre nie: n
Pouzit Novu krv? Zadaj a pre ano, n pre nie: n
Vypisat vypis kazdych 1000 generacii? Zadaj a pre ano, n pre nie: n
#####
Start genetickeho algoritmu
Povodne suradnice [[134, 164], [166, 120], [177, 83], [21, 92], [98, 184], [163, 58], [144, 45], [129, 70]]
Vyber rodicov ruleta
Pocet generacii 2000
Pocet clenov populacie 30
Pravdepodobnost mutacii deti 0.07
S dlzkou cesty 2007.3161877075718 je najlepsi najdeny jedinec z poslednej populacie:
[177, 83] [132, 16] [32, 14] [9, 33] [130, 111] [175, 95] [144, 45] [131, 84] [118, 61] [145, 75]
[129, 70] [195, 121] [166, 120] [95, 70] [73, 76] [70, 118] [179, 77] [161, 10] [94, 55] [26, 101]
[41, 171] [99, 198] [98, 184] [88, 110] [38, 74] [21, 92] [68, 73] [6, 82] [33, 72] [163, 58]
[134, 164] [134, 163] [174, 167]
Jeho fitnes je 0.0004981776195119695
Najlepsi jedinec z poslednej populacie je iny ako najlepsi jedinec zo vsetkych populacii.
Najlepsi jedinec zo vsetkych populacii s dlzkou 1857.5074798059673 je:
[177, 83] [132, 16] [32, 14] [9, 33] [130, 111] [175, 95] [144, 45] [118, 61] [129, 70] [131, 84]
[145, 75] [166, 120] [195, 121] [95, 70] [73, 76] [70, 118] [179, 77] [161, 10] [94, 55] [26, 101]
[41, 171] [99, 198] [98, 184] [88, 110] [21, 92] [6, 82] [38, 74] [33, 72] [68, 73] [163, 58]
[134, 164] [134, 163] [174, 167]
Jeho fitnes je 0.0005383558402168366
-----
Vygenerovat excel subor s vystupmi? Zadaj a pre ano, n pre nie: n
Bol vygenerovany subor s vystupmi "2020-11-22--21-07-40-S1453VrG2000C30M0.07NfKf.xlsx"
Ak si prajete pokracovat, zadajte P: n
Process finished with exit code 0
```

Program neošetruje vstupy. Preto ak sa do nejakého poľa zadá nesprávna hodnota, program spadne. Počet členov populácie musí byť párne číslo.

V prípade, že používateľ zvolí možnosť vygenerovať excel súbor s výstupmi, ten sa vygeneruje v mieste zdrojového kódu. Jeho názov bude vo formáte:

RRRR-MM-DD-HH-MM-SS-SsVvGgCcMmNnKk.xlsx

Malé písmená označené červenou symbolizujú:

s – číslo – zadaný seed

v – písmeno – výber rodiča – r pre ruletu, t pre turnaj

g – číslo – počet generácií

c – číslo – počet členov generácie, počet jedincov v populácii

m – číslo – pravdepodobnosť mutácií detí

n – písmeno – t pre aktívne zabezpečenie postupu najlepšieho jedinca z generácie do nasledujúcej generácie, f pre nasledujúcu generáciu zloženú iba z detí predoslej generácie

k – písmeno – t pre používanie Novej krvi, f pre nepoužívanie Novej Krví

5.1 Nastavenia vzorových vstupov

Pri zvolení konkrétnej možnosti program uvedie nastavenie, s ktorým pracuje. Tieto nastavenia ktoré boli použité pri testovaní uvádzam aj v tejto časti. Výstup z programu obsahuje aj pôvodné súradnice, ktoré vstupovali do evolučného algoritmu.

Bola zvolená možnosť 0

Tato možnosť zodpovedá súradniciam zo zadania, čiže sa negenerujú. Seed je nastavený na 3

#####

Start genetického algoritmu

Pôvodné súradnice [(60, 200), (180, 200), (100, 180), (140, 180), (20, 160), (80, 160), (200, 160), (140, 140), (40, 120), (120, 120), (180, 100), (60, 80), (100, 80), (180, 60), (20, 40), (100, 40), (200, 40), (20, 20), (60, 20), (160, 20)]

Vyber rodičov ruleta

Počet generácií 10000

Počet členov populácie 40

Pravdepodobnosť mutácií detí 0.1

Najhoršie z detí je nahradené najlepšími členmi z predoslej populácie

2 náhodne deti v každej generácii sú nahradené 2 novými permutáciami

Ciastkové vypisy počas evolúcie aktívne

Bola zvolená možnosť 1

Tato možnosť zodpovedá súradniciam zo zadania, čiže sa negenerujú. Seed je nastavený na 3

#####

Start genetického algoritmu

Pôvodné súradnice [(60, 200), (180, 200), (100, 180), (140, 180), (20, 160), (80, 160), (200, 160), (140, 140), (40, 120), (120, 120), (180, 100), (60, 80), (100, 80), (180, 60), (20, 40), (100, 40), (200, 40), (20, 20), (60, 20), (160, 20)]

Vyber rodičov turnaj

Počet generácií 10000

Počet členov populácie 40

Pravdepodobnosť mutácií detí 0.1

Najhoršie z detí je nahradené najlepšími členmi z predoslej populácie

2 náhodne deti v každej generácii sú nahradené 2 novými permutáciami

Ciastkové vypisy počas evolúcie aktívne

Bola zvolená možnosť 2

Tato možnosť zodpovedá súradniciam vygenerovaným randomom so seedom 11


```
#####
#####
Start genetickeho algoritmu
Povodne suradnice [[91, 81], [183, 13], [76, 161], [71, 82], [24, 109], [92, 176], [160, 45], [125,
132], [34, 140], [129, 74], [37, 81], [91, 152], [145, 67], [70, 171], [114, 192], [125, 79], [96,
23], [155, 70], [89, 107], [138, 185], [27, 44], [142, 7], [183, 79], [192, 168], [27, 133], [181,
71], [178, 187], [1, 131], [136, 68], [72, 114], [160, 78], [19, 153], [117, 90], [166, 187], [17,
43], [18, 48]]
Vyber rodicov ruleta
Pocet generacii 10000
Pocet clenov populacie 40
Pravdepodobnost mutacii deti 0.1
Najhorsie z deti je nahradene najlepsim clenom z predoslej populacie
2 nahodne deti v kazdej generacii su nahradene 2 novymi permutaciami
Ciastkove vypisy pocas evolucie aktivne
```

```
Bola zvolena moznost 3
Tato moznost zopoveda suradniciam vygenerovanim randomom so seedom 11
#####
#####
Start genetickeho algoritmu
Povodne suradnice [[91, 81], [183, 13], [76, 161], [71, 82], [24, 109], [92, 176], [160, 45], [125,
132], [34, 140], [129, 74], [37, 81], [91, 152], [145, 67], [70, 171], [114, 192], [125, 79], [96,
23], [155, 70], [89, 107], [138, 185], [27, 44], [142, 7], [183, 79], [192, 168], [27, 133], [181,
71], [178, 187], [1, 131], [136, 68], [72, 114], [160, 78], [19, 153], [117, 90], [166, 187], [17,
43], [18, 48]]
Vyber rodicov turnaj
Pocet generacii 10000
Pocet clenov populacie 40
Pravdepodobnost mutacii deti 0.1
Najhorsie z deti je nahradene najlepsim clenom z predoslej populacie
2 nahodne deti v kazdej generacii su nahradene 2 novymi permutaciami
Ciastkove vypisy pocas evolucie aktivne
```

```
Bola zvolena moznost 4
Tato moznost zopoveda suradniciam vygenerovanim randomom so seedom 99
#####
#####
Start genetickeho algoritmu
Povodne suradnice [[35, 185], [168, 68], [197, 180], [129, 151], [35, 55], [193, 176], [59, 107],
[130, 192], [140, 168], [200, 20], [59, 39], [79, 47], [27, 62], [62, 17], [73, 160], [21, 169], [99,
19], [22, 154], [178, 187], [5, 140], [96, 181]]
Vyber rodicov ruleta
Pocet generacii 10000
Pocet clenov populacie 40
Pravdepodobnost mutacii deti 0.1
Najhorsie z deti je nahradene najlepsim clenom z predoslej populacie
2 nahodne deti v kazdej generacii su nahradene 2 novymi permutaciami
Ciastkove vypisy pocas evolucie aktivne
```

Bola zvolena moznost 5

Tato moznost zopoveda suradniciam vygenerovanim randomom so seedom 99

#####

Start genetickeho algoritmu

Povodne suradnice [[35, 185], [168, 68], [197, 180], [129, 151], [35, 55], [193, 176], [59, 107], [130, 192], [140, 168], [200, 20], [59, 39], [79, 47], [27, 62], [62, 17], [73, 160], [21, 169], [99, 19], [22, 154], [178, 187], [5, 140], [96, 181]]

Vyber rodicov turnaj

Pocet generacii 10000

Pocet clenov populacie 40

Pravdepodobnost mutacii deti 0.1

Najhorsie z deti je nahradene najlepsim clenom z predoslej populacie

2 nahodne deti v kazdej generacii su nahradene 2 novymi permutaciami

Ciastkove vypisy pocas evolucie aktivne

Bola zvolena moznost 6

Tato moznost zopoveda suradniciam vygenerovanim randomom so seedom 111

#####

Start genetickeho algoritmu

Povodne suradnice [[108, 86], [19, 169], [66, 118], [40, 199], [185, 7], [12, 10], [193, 112], [28, 14], [162, 21], [54, 200], [37, 186], [12, 173], [197, 78], [31, 88], [198, 13], [50, 68], [157, 137], [170, 147], [97, 166], [59, 88], [4, 17], [74, 139], [177, 103], [88, 123], [123, 78], [134, 173], [187, 94], [200, 64], [1, 137], [140, 131], [173, 37], [197, 69], [33, 160], [135, 54], [171, 78], [105, 199], [50, 62], [75, 40], [170, 50], [91, 42]]

Vyber rodicov ruleta

Pocet generacii 10000

Pocet clenov populacie 40

Pravdepodobnost mutacii deti 0.1

Najhorsie z deti je nahradene najlepsim clenom z predoslej populacie

2 nahodne deti v kazdej generacii su nahradene 2 novymi permutaciami

Ciastkove vypisy pocas evolucie aktivne

Bola zvolena moznost 7

Tato moznost zopoveda suradniciam vygenerovanim randomom so seedom 111

#####

Start genetickeho algoritmu

Povodne suradnice [[108, 86], [19, 169], [66, 118], [40, 199], [185, 7], [12, 10], [193, 112], [28, 14], [162, 21], [54, 200], [37, 186], [12, 173], [197, 78], [31, 88], [198, 13], [50, 68], [157, 137], [170, 147], [97, 166], [59, 88], [4, 17], [74, 139], [177, 103], [88, 123], [123, 78], [134, 173], [187, 94], [200, 64], [1, 137], [140, 131], [173, 37], [197, 69], [33, 160], [135, 54], [171, 78], [105, 199], [50, 62], [75, 40], [170, 50], [91, 42]]

Vyber rodicov turnaj

Pocet generacii 10000

Pocet clenov populacie 40

Pravdepodobnost mutacii deti 0.1

Najhorsie z deti je nahradene najlepsim clenom z predoslej populacie

2 nahodne deti v kazdej generacii su nahradene 2 novymi permutaciami

Ciastkove vypisy pocas evolucie aktivne

5.2 Knížnice použité v riešení a prevzatý kód

V riešení používam knižničné funkcie z balíka `numpy`. Používam najmä `randint` na generovanie celých čísel z rozsahu a `choice` na náhodný vážený výber 2 rodičov z populácie.

Z balíka `math` používam funkciu `sqrt` pre odmocňovanie.

Z balíka `pandas` používam funkcie na tvorbu data frameov a generovanie excel tabuliek z číselných polí.

Z balíka `datetime` používam funkcie na zisťovanie aktuálneho dátumu a času a úpravu ich formátu pre potreby použitia týchto informácií v názvoch súborov.

Z balíka `sys` používam možnosť presmerovať výstup z konzoly do súboru pre potreby zaznamenania výstupov testovaní.

Pre generovanie excel súborov pomocou balíka `pandas` je potrebné mať nainštalovaný balík `openpyxl`.

6. Zhodnotenie

Problém obchodného cestujúceho je optimalizačná úloha, pri ktorej riešenie brutálnou silou eventuálne nájde globálne optimum, ale v nereálnom čase. Preto sa na riešenie tohto problému používajú optimalizačné algoritmy, ktoré hľadajú dostatočne dobré riešenie v reálnom čase. V tomto prípade som riešil Problém obchodného cestujúceho pomocou genetického algoritmu implementovaného v programovacom jazyku Python.

Môj genetický algoritmus vyberá rodičov ruletou alebo turnajom, križi ich výmenou úseku génov a prípadne mutuje ich deti. Môj implementovaný algoritmus dovoľuje používateľovi nastaviť vlastné hodnoty vstupných argumentov, ako sú počet členov populácie, počet generácií, výber rodiča, pravdepodobnosť mutácií detí, zabezpečenie prežitia najlepšieho jedinca z generácie a používanie Novej Krvi.

V testovaní som používal nastavenie: 40 jedincov v populácii, 10000 generácií, pravdepodobnosť mutácií 0,1 a ponechávanie najlepšieho jedinca spolu s aktívnou Novou Krvou. Tieto nastavenia dosahovali najlepšie výsledky. Mení sa seed pre random, aby sa generovali iné súradnice a mení sa spôsob výberu rodičov.

V testovaní sa jasne ukázalo, že výber rodičov ruletou zabezpečil postupné vylepšovanie nájdeného riešenia a dokázal sa dostať z lokálnych optím a výsledok sa blížil globálnemu optimu. Dá sa povedať, že s narastajúcim počtom generácií sa vylepšovalo riešenie. Z kriviek priemerného fitness generácie je vidno, že v generácií sa vždy nachádzajú lepší, aj horší jedinci. To zvyšuje pravdepodobnosť toho, že riešenie sa blíži ku globálnemu optimu.

Výber rodičov turnajom spôsobil uviaznutie v lokálnom optime, z ktorého sa nedalo dostať ani mutáciami. Do lokálneho optima sa riešenie dostalo veľmi rýchlo a priemerné hodnoty fitness generácií v grafe ukazujú, že v generáciách sa nachádzali len jedinci so zhruba rovnakými hodnotami fitness, ktorí riešenie neposúvali ku globálnemu optimu, ale stagnovali v lokálnom optime. Preto je vhodnejšie používať riešenie ruletou.

Vylepšenie by mohli priniesť ďalšie, komplexnejšie spôsoby výberu rodičov z populácie. Turnajový výber zlepšenie oproti rulete nepriniesol, pretože uviazol v lokálnom optime s jedincami, ktorých fitness je navzájom veľmi podobné. Pri inej implementácii turnajového výberu rodičov by sa výsledky mohli zlepšiť. Zlepšením, už implementovaným, je zabezpečenie prežitia najlepšieho jedinca v populácii a nahradzovanie 2 jedincov v každej populácii úplne novými jedincami – teda Nová Krv. Tieto zlepšenia výrazne kladným spôsobom ovplyvnili výsledky, preto som ich používal aj pri testovaní riešenia.