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](mailto:xsmrecek@stuba.sk)

Predmet: UI

Deň a čas cvičenia: Utorok 16:00

Semester: ZS 20/21

Ročník: 2.

Obsah

[1. Zadanie 2](#_Toc56974084)

[2. Reprezentácia údajov 2](#_Toc56974085)

[2.1 Gén 2](#_Toc56974086)

[2.2 Chromozóm 2](#_Toc56974087)

[2.3 Populácia 3](#_Toc56974088)

[2.4 Fitnes 3](#_Toc56974089)

[3. Opis algoritmu 3](#_Toc56974090)

[3.1 Tvorba populácie 4](#_Toc56974091)

[3.2 Výber rodičov 4](#_Toc56974092)

[3.2.1 Ruleta 4](#_Toc56974093)

[3.2.2 Turnaj 5](#_Toc56974094)

[3.3 Kríženie rodičov 5](#_Toc56974095)

[3.4 Návratové hodnoty 6](#_Toc56974096)

[4. Testovanie 6](#_Toc56974097)

[4.1.1 Testovací scenár zo zadania 6](#_Toc56974098)

[4.1.2 Testovací scenár A 8](#_Toc56974099)

[4.1.3 Testovací scenár B 9](#_Toc56974100)

[4.1.4 Testovací scenár C 11](#_Toc56974101)

[5. Používateľská príručka 13](#_Toc56974102)

[5.1 Nastavenia vzorových vstupov 15](#_Toc56974103)

[5.2 Knižnice použité v riešení a prevzatý kód 18](#_Toc56974104)

[6. Zhodnotenie 18](#_Toc56974105)

# 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.

# 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/(dĺžka cesty)

## 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 miest, čo je vlastne dvojrozmerný zoznam, kde sa na každej z pozícií 0 až (počet miest -1) nachádzajú súradnice mesta.

## 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 populácií.

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.

## Populácia

Populácia je v programe reprezentovaná v zozname jedincov, ktorý je vytvorený a modifikovaný vo funkcií geneticky\_algoritmus.

## 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/(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.

# 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.

## Tvorba populácie

Z pôvodne zadaného zoznamu súradníc sa vytvorí inštancia triedy Graf. Vytváraním 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 fitnes.

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 fitnes všetkých jedincov v populácií. Index do poľa pravdepodobností, na ktorom je uložený fitnes 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 dvojice 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, permutáciami pôvodného grafu.

V prípade zvolenia možnosti „Ponechanie najlepšieho jedinca“ bude najhoršie dieťa v novej populácií 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.

## Výber rodičov

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

### 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.

### 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 fitnes. 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.

## 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.

## Návratové hodnoty

Funkcia geneticky\_algoritmus počas evolučného cyklu počíta a zapisuje do poľa priemerný fitnes každej generácie a fitnes najlepšieho jedinca v každej generácií. Po dokončení evolučného cyklu funkcia vypíše najlepšieho jedinca, jeho cestu, dĺžku cesty a jeho fitnes. 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 fitnes generácie, pole maximálnych hodnôt fitnes generácie a samotného najlepšieho jedinca. Z týchto hodnôt sa dá pomocou funkcie generuj\_excel vytvoriť tabuľka.

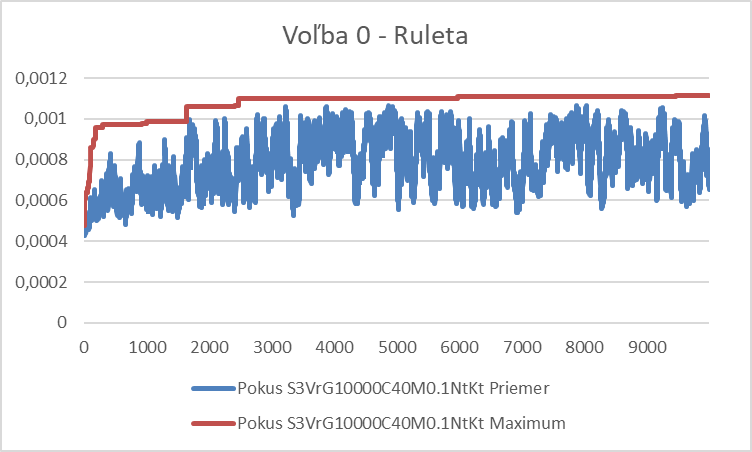
# Testovanie

Pre potreby testovania som si zvoli 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 populácií, 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 zachytila výstup z konzoly do textového súboru. Všetky súbory sú označené kódom a obsahujú príslušné hodnoty a grafy.

### 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.



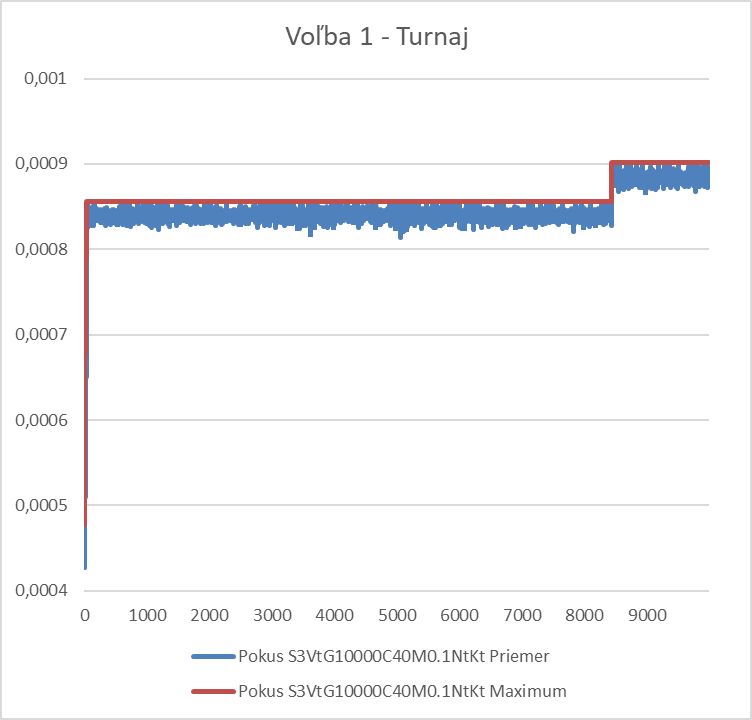
Použitie rulety prinieslo postupné zlepšovanie riešenia s narastajúcim počtom generácií. Program sa dokázal efektívne dostať z lokálnych optím. Z krivky priemerného fitnes generácie je vidno, že v generácií sa nachádzajú lepší, aj horší jedinci.

Výstup programu je:

S dlzkou cesty 895.706325022881 je najlepsi najdeny jedinec 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 fitnes 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 fitnes generácie je jasné, že generácia obsahovala len jedincov s veľmi podobnými hodnotami fitnes.

Výstup programu je:

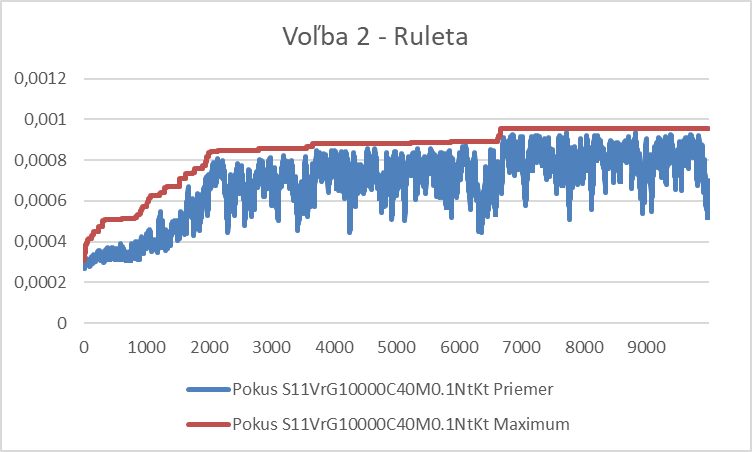
S dlzkou cesty 1109.527054478831 je najlepsi najdeny jedinec z poslednej populacie:

[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 fitnes je 0.0009012849177163344

### Testovací scenár A

Testovací scenár A obsahuje náhodnú sadu súradníc.



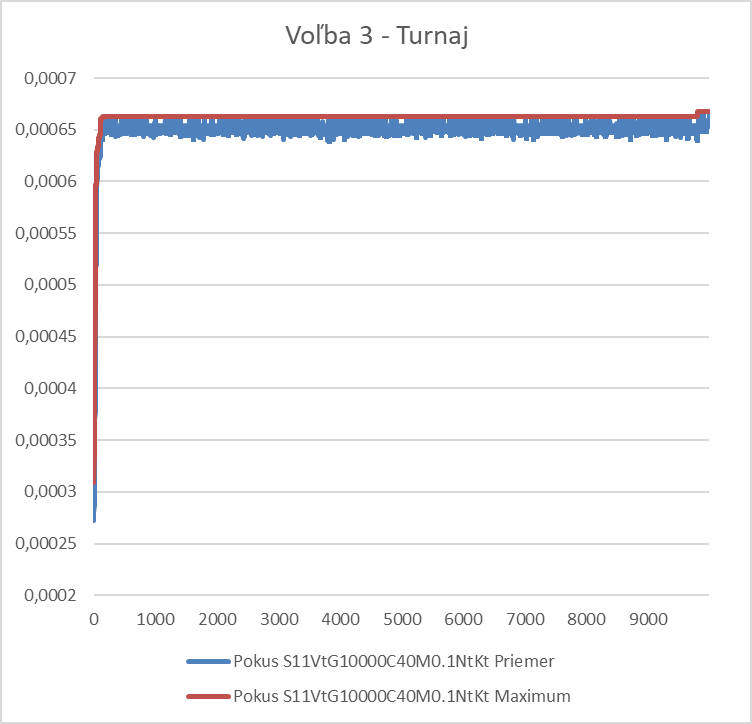
Použitie rulety prinieslo postupné zlepšovanie riešenia s narastajúcim počtom generácií. Program sa dokázal efektívne dostať z lokálnych optím. Z krivky priemerného fitnes generácie je vidno, že v generácií sa nachádzajú lepší, aj horší jedinci.

Výstup programu je:

S dlzkou cesty 1046.6791010240556 je najlepsi najdeny jedinec z poslednej populacie:

[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 fitnes 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 fitnes generácie je jasné, že generácia obsahovala len jedincov s veľmi podobnými hodnotami fitnes.

Výstup programu je:

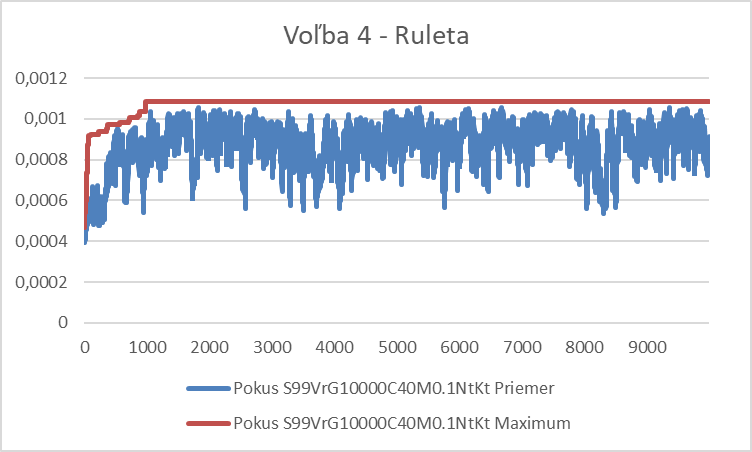
S dlzkou cesty 1497.440883470884 je najlepsi najdeny jedinec z poslednej populacie:

[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 fitnes je 0.0006678059955743447

### Testovací scenár B

Testovací scenár B obsahuje náhodnú sadu súradníc.



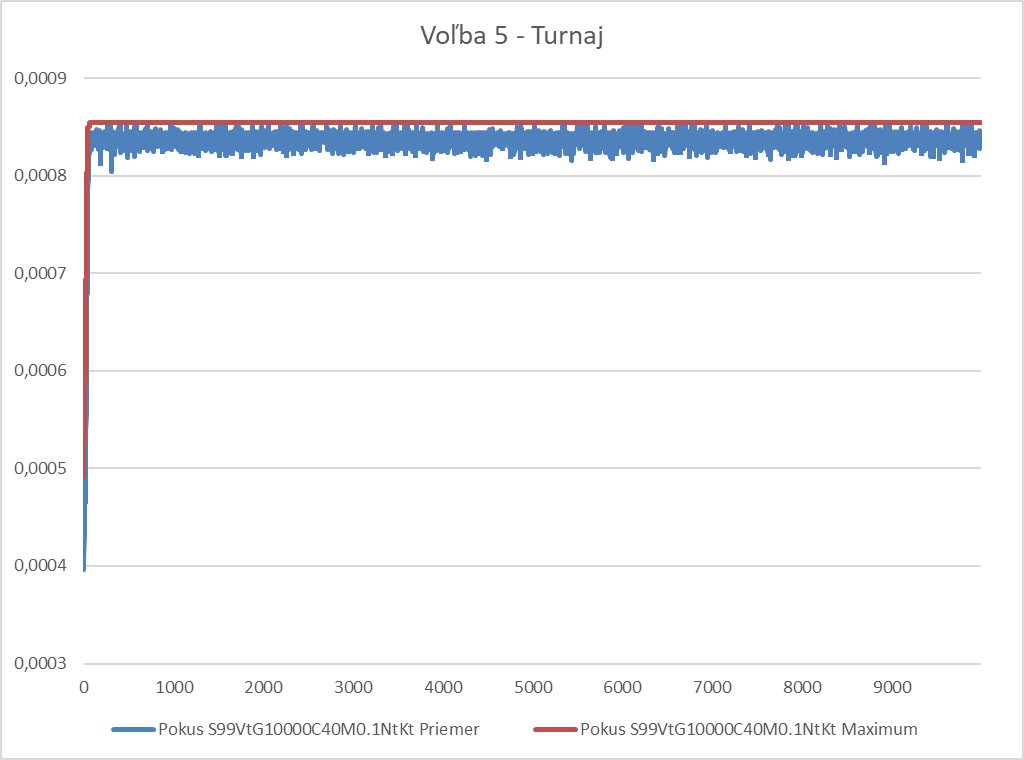
Použitie rulety prinieslo pomerne rýchle nájdenie optima. Z krivky priemerného fitnes generácie je vidno, že v generácií sa nachádzajú lepší, aj horší jedinci, no ďalšie zlepšenie maximálneho fitnes to neprinieslo.

Výstup programu je:

S dlzkou cesty 922.3814307818993 je najlepsi najdeny jedinec z poslednej populacie:

[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 fitnes 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 fitnes generácie je jasné, že generácia obsahovala len jedincov s veľmi podobnými hodnotami fitnes.

Výstup programu je:

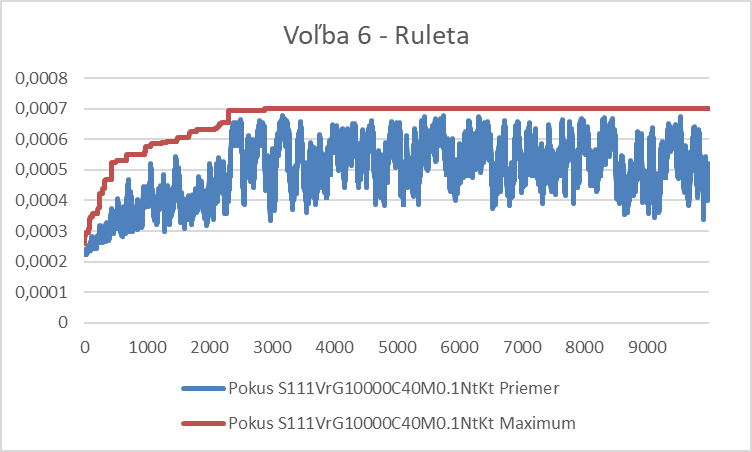
S dlzkou cesty 1170.7134080200497 je najlepsi najdeny jedinec z poslednej populacie:

[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 fitnes je 0.0008541800180551737

### Testovací scenár C

Testovací scenár C obsahuje náhodnú sadu súradníc.



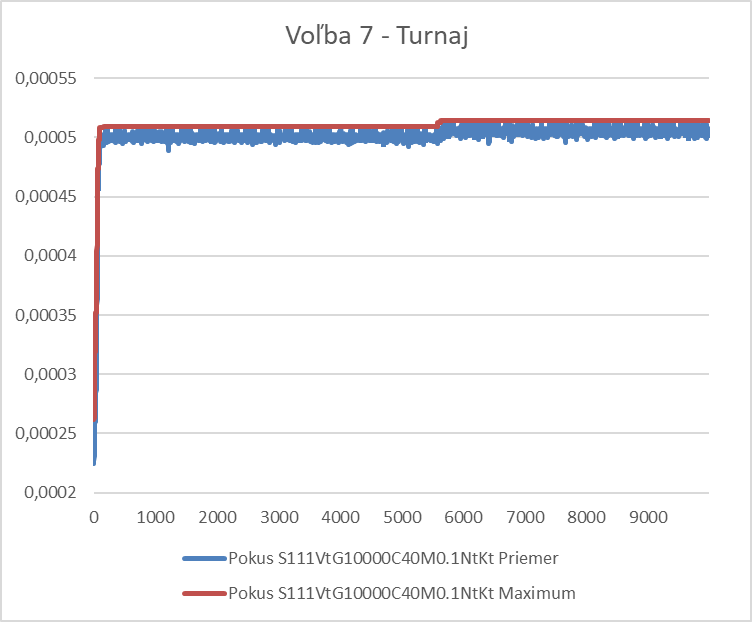
Použitie rulety prinieslo pomerne rýchle nájdenie optima. Z krivky priemerného fitnes generácie je vidno, že v generácií sa nachádzajú lepší, aj horší jedinci, no ďalšie zlepšenie maximálneho fitnes to neprinieslo.

Výstup programu je:

S dlzkou cesty 1428.9356680989954 je najlepsi najdeny jedinec z poslednej populacie:

[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 fitnes 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 fitnes generácie je jasné, že generácia obsahovala len jedincov s veľmi podobnými hodnotami fitnes.

Výstup programu je:

S dlzkou cesty 1945.1003834768828 je najlepsi najdeny jedinec z poslednej populacie:

[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]

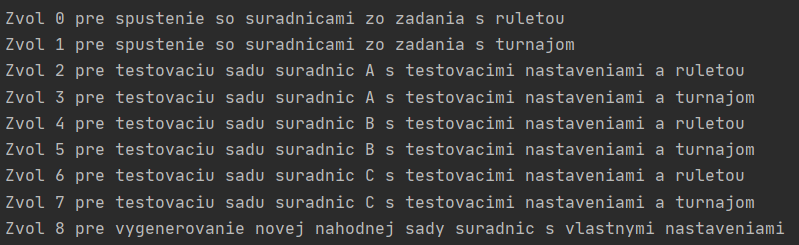
Jeho fitnes je 0.0005141122836100067

# 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 funkcií je zakomentované jej volanie. Jeho odkomentovaním sa opätovne dajú vygenerovať rovnaké výstupy, aké popisujem v časti Testovanie.

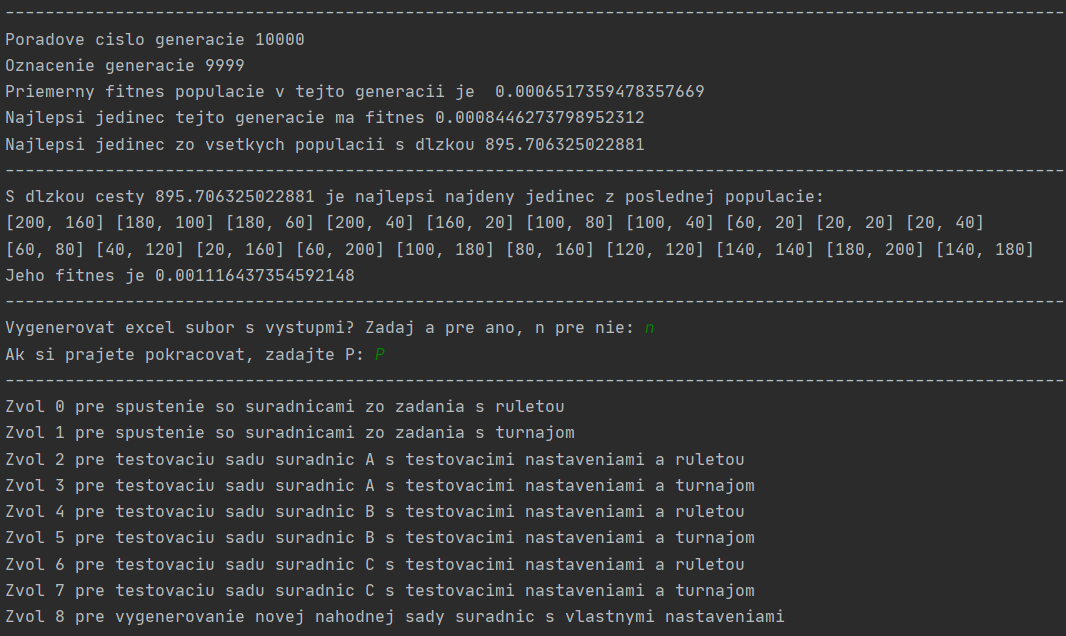
Funkcia riadic je funkcia obsahujúca konzolové používateľské rozhranie a používateľ ňou môže ovládať program.

Po spustení programu sa vypíše menu:

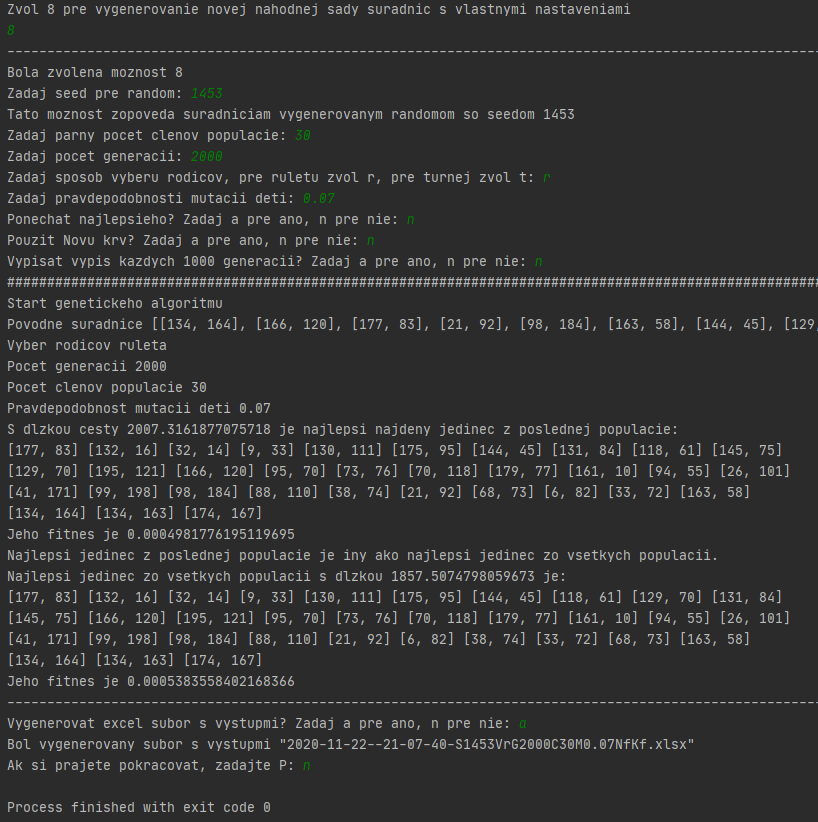


Používateľ si môže zvoliť jeden z testovacích scenárov 0 – 7 alebo voľbou 8 nastaviť vlastný testovací scenár.

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.



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



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 populácií

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 predošlej generácie

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

## 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 zvolena moznost 0

Tato moznost zodpoveda suradniciam zo zadania, cize sa negeneruju. Seed je nastaveny na 3

######################################################################################################################################################

Start genetickeho algoritmu

Povodne suradnice [(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 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 1

Tato moznost zodpoveda suradniciam zo zadania, cize sa negeneruju. Seed je nastaveny na 3

######################################################################################################################################################

Start genetickeho algoritmu

Povodne suradnice [(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 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 2

Tato moznost zopoveda suradniciam vygenerovanym 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 vygenerovanym 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 vygenerovanym 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 vygenerovanym 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 vygenerovanym 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 vygenerovanym 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

## Knižnice 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.

# 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. V tomto prípade som riešil Problém obchodného cestujúceho pomocou genetického algoritmu implementovaného v programovacom jazyku Python.

Môj genetický algoritmus vyberá rodičov ruletou alebo turnajom, kríži 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 populácií, 10000 generácií, pravdepodobnosť mutácií 0,1 a ponechávanie najlepšieho jedinca spolu s aktívnou Novou Krvou. 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 fitnes 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 fitnes generácií v grafe ukazujú, že v generáciách sa nachádzali len jedinci so zhruba rovnakými hodnotami fitnes, 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 fitnes je navzájom veľmi podobné. Zlepšením, už implementovaným, je zabezpečenie prežitia najlepšieho jedinca v populácií a nahradzovanie 2 jedincov v každej populácií úplne novými jedincami – teda Nová Krv. Tieto zlepšenia výrazne kladným spôsobom ovplyvnili výsledky, preto som ich používal aj pri testovaní riešenia.