

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

### 3.Zadanie – Zen evolučný algoritmus

Študijný program: Informatika  
Ročník: 2  
Cvičenie: Ut 14:00 -1.30b  
Cvičiaci: Ing. Matúš Pikuliak  
Predmet: Umelá inteligencia  
Akademický rok: 2019/2020

## Opis Algoritmu

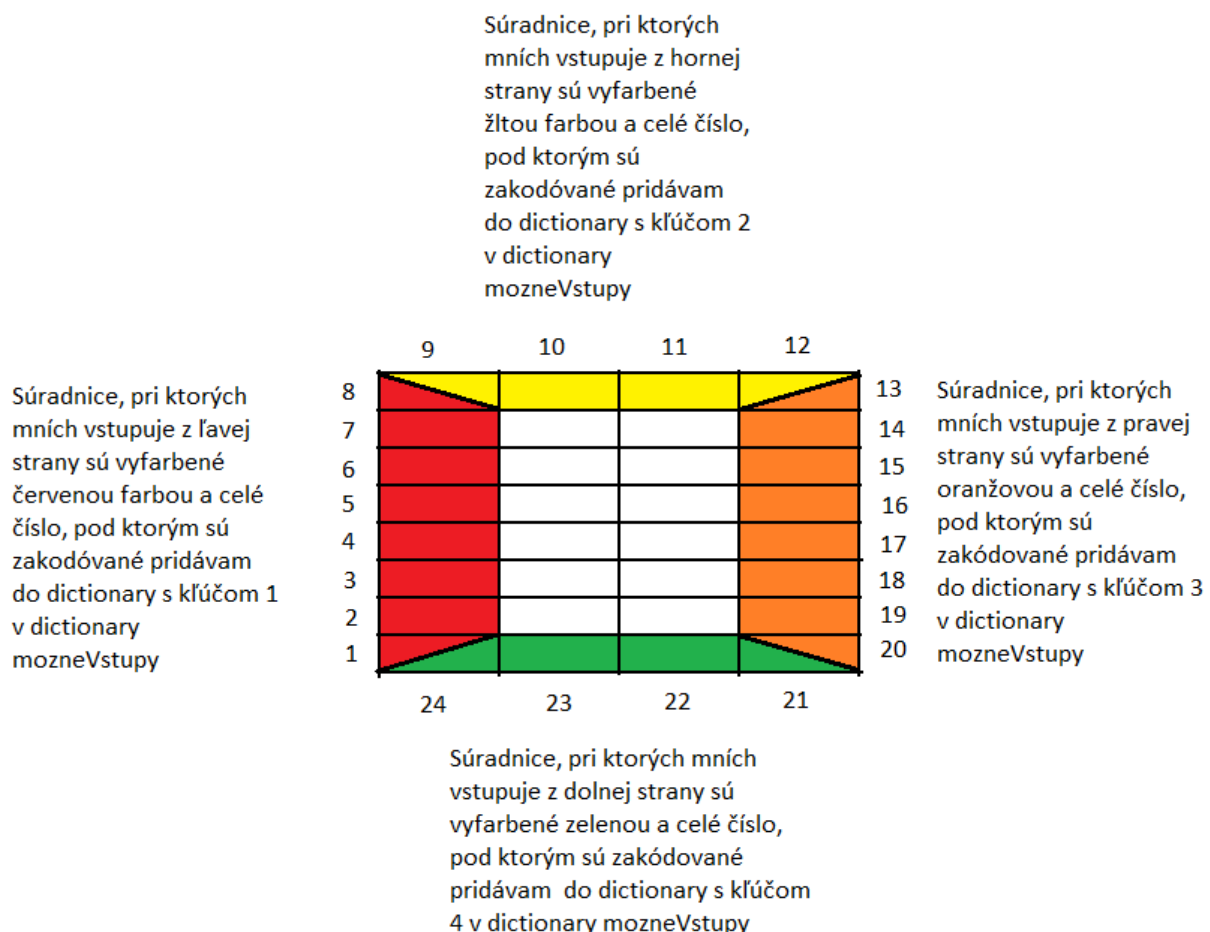
### Reprezentácia jedinca

Každého jedinca danej generácie si reprezentujem classou `ReprezentaciaJedinca`. V tejto classe si ukladám chromozóm (postupnosť génov orania) v jedno rozmernom liste s názvom **chromozom** a počet poraných políčok (fitness ohodnotenie) v atribúte **pocetZoranych**.

### Reprezentácia génu

Gén v mojom algoritme je reprezentovaný celým číslom od 1 po posledný možný vstup do mapy. Očíslovanie génov začínam od ľavého dolného rohu, pričom sa postupne posúvam po okraji mapy a každú odlišnú súradnicu pridávam zakódovanú pod celým číslom do **dictionary mozneVstupy**. Počas prechodu okraja sa inkrementuje toto celé číslo o 1, vždy keď zmením súradnicu. Súradnice rohu pridám do dictionary dva krát, **ale vždy je zakódovaná pod iným číslom a ide do rôzneho z vnorených dictionaries**.

Týmto myslím to, že ešte pred pridávaním vstupov do **dictionary mozneVstupy** v ňom deklarujem **štyri ďalšie dictionaries**, ktorých **kľúče** sú **1, 2, 3** alebo **4**. Každý z týchto dictionaries reprezentuje **rozdielnú stranu**, z ktorej mních vstupuje do záhradky.



### Príklad pridania vstupu do **dictionary mozneVstupy**:

Zoberme si, že aktuálna súradnica, ktorú je potrebné pridať je **(0, 3)** (teda pravý horný roh). Ako som vyššie avizoval, tak túto súradnicu pridám do **dvoch rozdielných vnorených dictionaries**. Najprv ju dám do **dictionary s kľúčom 2** a bude zakódovaná pod číslom **12**.

Meno: Jakub Hlavačka  
AIS ID: 96876

Následne ju pridám do **dictionary s kľúčom 3** ale už bude zakódovaná pod číslom **13** aby program dokázal rozpoznať, z ktorej strany záhradky ide mních orať.

V prípade, že sa na danej súradnici nachádza prekážka, cez ktorú mních nemôže vstúpiť, tak sa daná súradnica nepridá ani do jedného vnoreného dictionary.

Proces získavania všetkých možných vstupov(génov), skončí , keď sa mních vráti do ľavého dolného rohu.

### Pohyb a rozhodovanie mnícha

Pohyb a rozhodovanie mnícha riadi premenná **otocka**. Vždy, keď idem hodnotiť nové oranie podľa chromozómu konkrétneho jedinca, tak je hodnota premennej **otocka** inicializovaná na **True**.

V prípade, keď mních narazí na kameň, a hodnota premennej **otocka** je **True**, tak sa mních bude snažiť odbočiť doľava. Ak by mal mních opustiť mapu, pretože odbočil doľava, tak odbočí radšej doprava a hodnota premennej **otocka** ostane **True**. To samé platí v prípade, že sa doľava odbočiť nedá, pretože tam je prekážka.

Ak nenastane ani jeden z týchto prípadov, tak mních odbočí doľava a zmení hodnotu premennej **otocka** na **False**.

Keď je hodnota premennej **otocka False**, tak sa mních bude snažiť točiť doprava. Tak isto ako pri točení doľava, v prípade výstupu z mapy alebo nemožného otočenia doprava sa mních otočí doľava, pričom hodnota premennej **otocka** ostane **False**.

Ak opäť nenastane ani jeden z týchto prípadov, tak sa mních otočí doprava a zmení hodnotu premennej **otocka** na **True**.

Takýto spôsob točenia mnícha som implementoval, pretože chcem aby mních pooral čo najväčšie množstvo záhradky pre daný vstup a nevytvoril v prípade viacerých točení do rovnakej strany slimáka v ktorom ostane zaseknutý.

### Spôsob orania mnícha

Mních na začiatku orania vstúpi do záhradky podľa súradníc prvého génu a podľa jeho hodnoty takisto bude vedieť, ktorým smerom sa začína oranie. Mních postupuje v oraní z políčka na políčko, pričom sa zvyšuje hodnota fitness pre daného jedinca vždy o 1. Ak mních narazí na prekážku, tak sa otáča podľa opísaného princípu v Pohyb a rozhodovanie mnícha.

Ak mních vyjde z mapy, tak prejde na ďalší gén chromozómu. Keď nastane situácia, že mních daný vstup už využil, tak sa pozrie na ďalší gén chromozómu. Ak patrí do skupiny génov, ktoré už boli využité, tak ho opäť preskočí. Takisto ak narazí na vstup, cez ktorý sa nemôže dostať do záhradky, pretože tam je poorané, tak ho preskočí. V prípade, že cez súradnice daného génu môže vstúpiť do záhradky, tak tak vykoná a pokračuje v oraní.

Mních ukončí oranie keď sa zasekne, nájde správne riešenie alebo prejde celú dĺžku chromozómu.

### Spôsoby selekcie

Pre tento genetický algoritmus som implementoval dva spôsoby selekcie. Prvý spôsob je **turnaj**, pri ktorom z generácie náhodne vyberiem štyroch jedincov, ktorým porovnáam ich fitness ohodnotenie a ten, ktorý je najlepší vyhráva turnaj a pôjde do kríženia alebo sa dostane do ďalšej generácie.

Druhým spôsobom je **ruleta**, pri ktorej sa vygeneruje náhodne číslo od 0 po celkový počet fitness ohodnotenia pre celú generáciu, z ktorej vyberám jedinca. Následne prechádzam jedincov generácie

a sčítam ich fitness ohodnotenie v pomocnej premennej, ktorá keď po pričítaní ďalšieho fitness ohodnotenia bude vyššia ako náhodne vygenerované číslo, tak z rulety vyberiem jedinca, ktorého fitness ohodnotenie sa malo pripočítať do pomocnej premennej.

### Tvorba novej generácie

Vytváranie novej generácie sa vykonáva v cykle, pokiaľ nová generácia nemá rovnaký počet jedincov ako tá predošlá, pričom sa počet jedincov počas vývoja nemení. Cyklus môže taktiež skončiť, keď sa nájde jedinec, ktorého mních dokáže poorať celú záhradku.

Pred spustením cyklu sa môže vybrať najlepší jedinec z predošlej generácie a pridať do novej, ak sa používateľ rozhodol pre elitizmus.

V každom opakovaní cyklu sa vykoná podmienka:

```
if (random.uniform(0, 1) <= crossover):
```

Ak hodnota výsledku **random.uniform(0, 1)** bude menšia ako hodnota premennej **crossover**, ktorú pred spustením algoritmu zadáva používateľ, tak sa vykoná kríženie. V opačnom prípade sa prenesie jeden prvok, ktorý bol vybraný ruletou alebo turnajom, zo starej generácie do novej alebo sa vytvorí nový jedinec, ktorého chromozóm vznikol náhodne tzv. „nová krv“.

Pri vykonávaní kríženia sa vyberú dvaja jedinci, jedným z dvoch spôsobov selekcie, podľa výberu používateľa pred začatím algoritmu. Následne vybraní jedinci vstupujú do procesu kríženia, kde nový jedinec vzniká dedením génov od rodičov, pričom každý rodič má rovnakú šancu predať daný gén z chromozómu svojmu dieťaťu.

Príklady implementovaného kríženia na 8 bitovom slove:

#### 1. Výsledok

1	1	0	1	0	0	0	1
0	1	1	0	0	1	1	1
1	1	1	0	0	1	0	1

#### 2. Výsledok

1	1	0	1	0	0	0	1
0	1	1	0	0	1	1	1
0	1	0	0	0	0	1	1

Počet rôznych výsledkov, ktoré môžu vzniknúť týmto krížením je 256.

Po skončení kríženia prechádza vytvorený jedinec procesom možnej mutácie, pričom jej pravdepodobnosť zadáva používateľ pred spustením algoritmu. Mutovať môžu všetky gény chromozómu. Gén, ktorý mutuje je nahradený ľubovoľným prvok z množiny možných vstupov uložených v **dictionary mozneVstupy**.

Po skončení možnej mutácie sa vytvorený jedinec ohodnotí a ak jeho mních dokáže poorať celú záhradku, tak program skončí v opačnom prípade sa pridá do novej generácie a proces sa opakuje až kým sa nevytvorí nová generácia alebo sa nenájde jedinec, ktorého mních dokáže poorať celú záhradku.

### Možnosti nastavenia parametrov

Používateľ pred spustením algoritmu zadáva počet jedincov v generácii a hodnoty parametrov pre kríženie a mutáciu. Následne vyberá algoritmus selekcie, prídavok novej krvi a vykonanie elitizmu.

## Celkový chod programu

- Načítanie rozloženia záhradky zo súboru vstup.txt.
- Zistenie počtu políčok na oranie, maximálnej dĺžky chromozómu a všetkých možných vstupov, ktoré figurujú ako gény.
- Používateľ vyberie, či chce aplikovanie elitarizmu, pridávanie novej krvi a konkrétny algoritmus selekcie. Potom zadá počet jedincov generácie a hodnoty parametrov pre kríženie a mutáciu.
- Následne sa turbo náhodne vygenerujú jedinci prvej generácie, ktorí sú hneď ohodnotení. V prípade, že sa ešte nenašlo riešenie, tak program pokračuje inak skončí.
- Vytvorená generácia vstupuje do procesu vytvorenia novej, kde sa novo vytvorený jedinec ohodnotí a ak jeho mních dokáže poorať celú záhradku, tak sa program skončí. Inak sa tento proces vytváranie generácie opakuje, pokiaľ nie je vytvorená posledná generácia.
- Po skončení algoritmu sa vypíše oranie jedinca, ktorého mních dokázal poorať celú záhradku. Ak sa takýto jedinec nevytvoril, tak sa vypíše oranie jedinca, ktorého mních dokázal poorať najviac políčok záhradky zo všetkých jedincov vo všetkých generáciách.

## Opis vylepšovania

Všetky testy boli vykonávané na **vzorovej záhradke** zo stránky zadania, kde bol počet políčok, ktoré bolo potrebné poorať 114. V každom teste začínali obidva spôsoby s rovnakou generáciou a skončili, keď každý spôsob vygeneroval 1000 generácií. **Našlo** v tabuľkách znamená, že algoritmus vytvoril jedinca, ktorého mních bol schopný poorať celú záhradku. Číslo v tabuľkách symbolizuje, že algoritmus nebol schopný vytvoriť jedinca, ktorého mních poora celú záhradku a najlepším mníchom z generácie je ten, ktorého maximálny počet pooraných políčok je zaznamenaný v tabuľkách.

Ako prvé som sa rozhodol medzi dvomi spôsobmi kríženia, prvým bolo kríženie, kde sa náhodne vygeneruje číslo od 1 po maximálnu dĺžku chromozómu - 1. Následne toto číslo slúžilo ako hranica génov, ktorú môže jednotlivý rodič predať dieťaťu.

Príklady prvého testovaného spôsobu kríženia na 8 bitovom slove:

Prvý príklad:

náhodne vygenerované číslo 3

1	1	0	1	0	0	0	1
0	1	1	0	0	1	1	1
0	1	1	1	0	0	0	1

Druhý príklad:

náhodne vygenerované číslo 6

1	1	0	1	0	0	0	1
0	1	1	0	0	1	1	1
0	1	1	1	0	0	0	1

Počet rôznych krížení, ktoré môžu vzniknúť týmto krížením je 7.

Druhý spôsob kríženie som finálne implementoval v programe a je opísaný vyššie.

Pri tomto testovaní bol použitý prvý spôsob orania, ktorý je opísaný v ďalšom testovaní.

### Výsledky testovania:

Počet génov generácie = 100, crossover = 0,8, mutácia = 0,1, pridávanie „novej krvi“, elitizmus

Prvý spôsob	Druhý spôsob
Našlo	Našlo
111	Našlo
112	Našlo
110	110
113	113
113	112
113	Našlo
113	Našlo
Našlo	Našlo
Našlo	113

Počet génov generácie = 50, crossover = 0,8, mutácia = 0,1, pridávanie „novej krvi“, elitizmus

Prvý spôsob	Druhý spôsob
113	110
113	Našlo
113	110
Našlo	Našlo
Našlo	Našlo
112	112
109	110
111	113
Našlo	109
113	108

Z výsledkov testovania som usúdil, že ďalej budem implementovať druhý spôsob kríženia.

Následne som skúšal, ktorý spôsob orania je lepší. Prvý spôsob bol, že ak mních vyšiel zo záhradky a ďalší gén chromozómu, cez ktorý má vstúpiť je buď použitý alebo je na jeho súradnici poraný, tak mních ukončí oranie.

Druhý spôsob fungoval na tom istom princípe, ale keď mních narazil na gén chromozómu, ktorý už využil, tak prešiel na ďalší.

Tieto spôsoby som testoval na rôzne hodnoty parametrov kríženia, mutácie, počtu jedincov v generácií a pridávanie „novej krvi“. Avšak používal som v nich len spôsob selekcie turnajom.

### Výsledky testovania:

Počet jedincov = 100, crossover = 0,8, mutácia = 0,1, nepridával som „novú krv“, elitizmus

Prvý spôsob	Druhý spôsob
112	112
107	113
Našlo	Našlo
110	Našlo

Meno: Jakub Hlavačka  
AIS ID: 96876

113	112
113	112
Našlo	111
112	Našlo
112	Našlo
113	Našlo

Počet jedincov 50, crossover = 0,8, mutácia = 0,1, nepridával som „novú krv“, elitizmus

Prvý spôsob	Druhý spôsob
Našlo	104
111	110
112	Našlo
112	113
110	Našlo
108	108
Našlo	113
Našlo	Našlo
Našlo	111
109	113

Počet jedincov = 100, crossover = 0,8, mutácia = 0,15, pridávanie „novej krvi“, elitizmus

Prvý spôsob	Druhý spôsob
Našlo	Našlo
Našlo	Našlo
Našlo	110
111	Našlo
113	Našlo
110	Našlo
Našlo	Našlo
112	Našlo
Našlo	112
110	Našlo

Počet jedincov = 50, crossover = 0,8, mutácia = 0,15, pridávanie „novej krvi“, elitizmus

Prvý spôsob	Druhý spôsob
112	109
Našlo	109
110	Našlo
113	111
Našlo	113
112	113
111	Našlo
111	Našlo
Našlo	112
Našlo	110

Meno: Jakub Hlavačka  
AIS ID: 96876

Počet jedincov = 100, crossover = 0,8, mutácia = 0,1, pridávanie „novej krvi“, elitizmus

Prvý spôsob	Druhý spôsob
Našlo	Našlo
108	113
Našlo	Našlo
Našlo	Našlo
113	Našlo
113	Našlo
Našlo	Našlo
Našlo	Našlo
Našlo	112
107	Našlo

Počet jedincov = 50, crossover = 0,8, mutácia = 0,1, pridávanie „novej krvi“, elitizmus

Prvý spôsob	Druhý spôsob
Našlo	113
113	108
109	112
Našlo	Našlo
111	113
Našlo	112
109	Našlo
Našlo	Našlo
Našlo	111
112	Našlo

Z výsledkov testovania som zistil, že druhý spôsob orania je lepší a môj algoritmus lepšie reaguje, keď sa do každej generácie pridáva „nová krv“ a gén má vyššiu šancu mutácie ako je bežné.

Následne som implementoval spôsob selekcie ruletou a snažil sa zistiť, pre ktorú selekciu algoritmus dostáva lepšie výsledky a ako na nich pôsobia zmeny parametrov kríženia, mutácie, počtu jedincov v generácií a pridávanie „novej krvi“. Výsledky boli 99% krát identické.

Nakoniec som náhodou vyskúšal ako algoritmus zareaguje, keď bude mních pri prechádzaní génov chromozómu preskakovať každý gén, ktorý použil alebo z jeho súradníc nie je možné vstúpiť do záhradky. Takéto oranie skončí iba v prípade zaseknutia mnícha pri oraní, nájdením správneho riešenia alebo prejdením celého chromozómu.

Posledný spôsob bol najlepší, algoritmus mi oproti predošlým spôsobom extrémne rýchlo našiel jedinca, ktorého mních poorá celú záhradku, pričom pri predošlých ju niekedy ani nepooral.

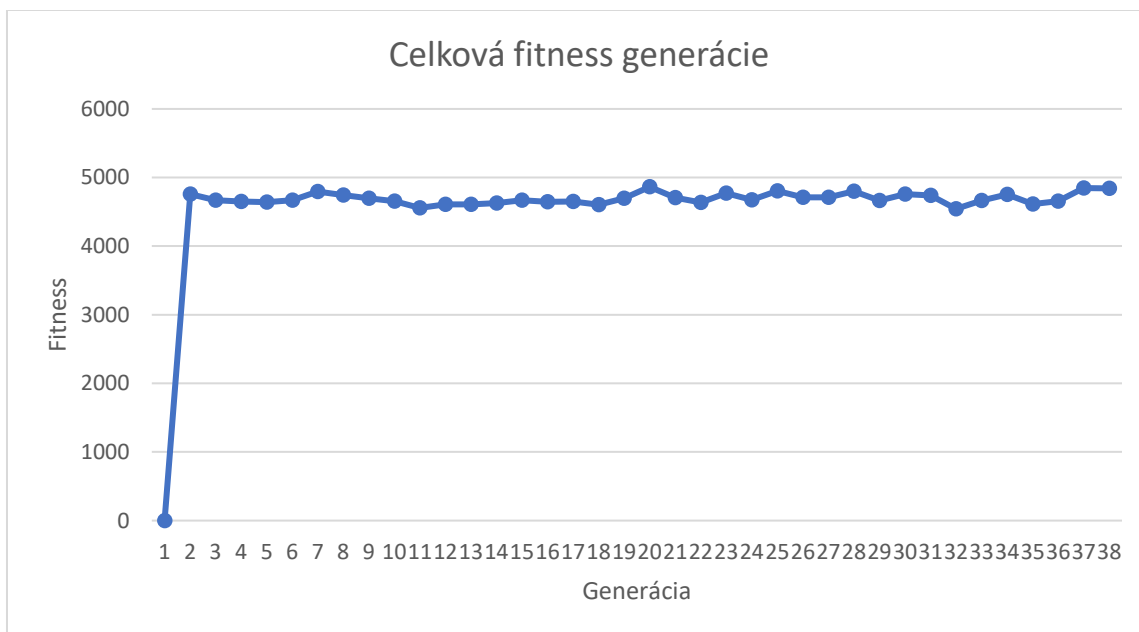
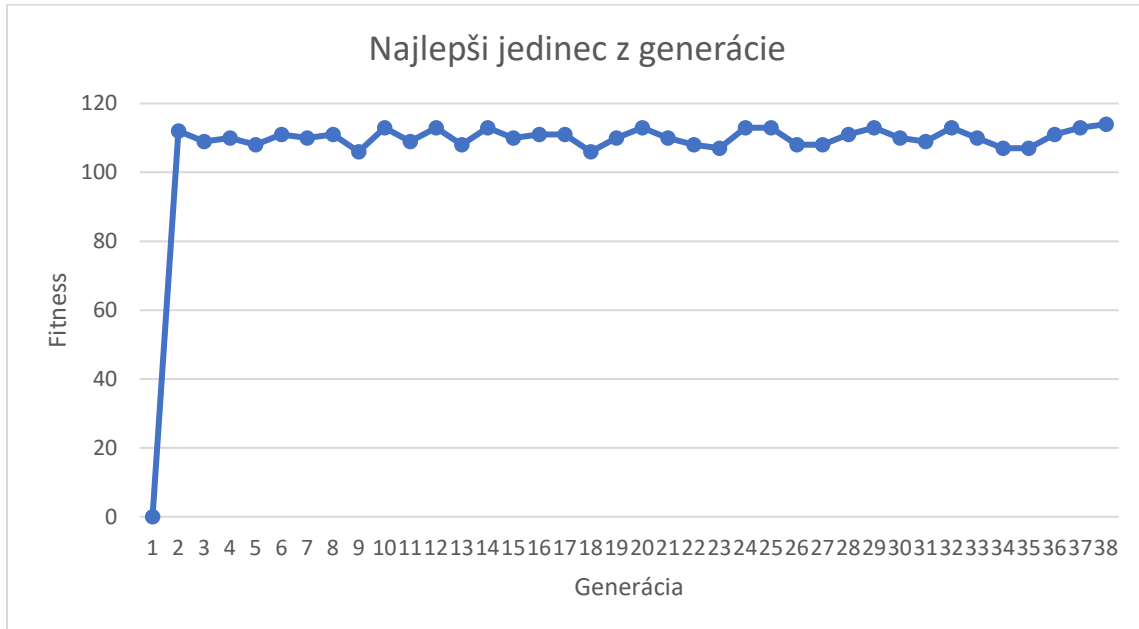


Meno: Jakub Hlavačka  
AIS ID: 96876

## Zhodnotenie vlastností algoritmu

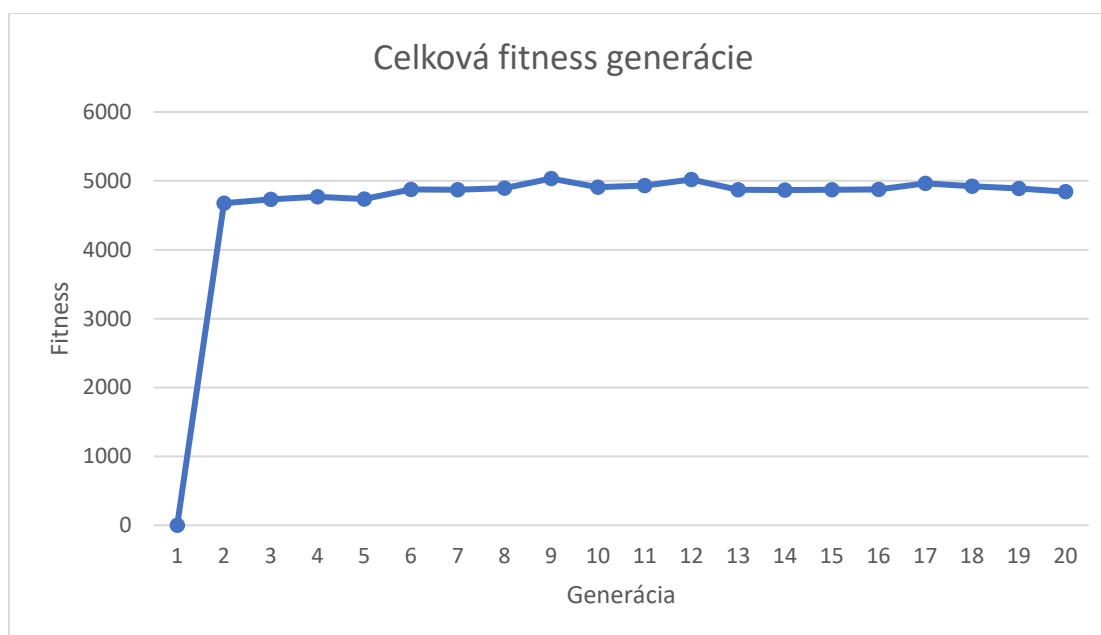
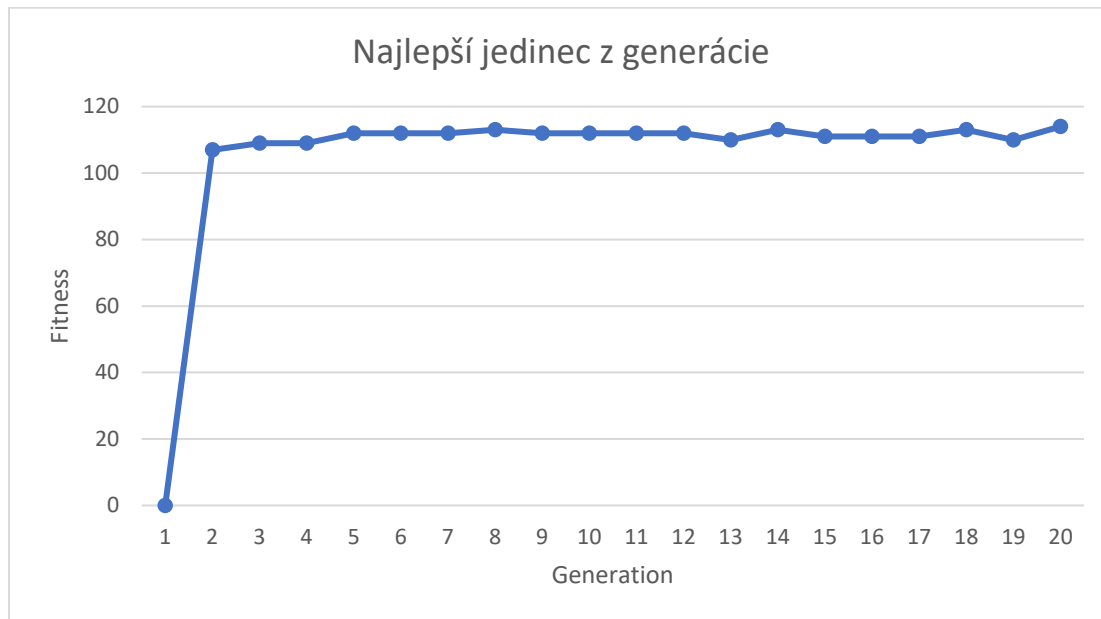
Zaznamenávania nasledujúcich grafov pre fitness funkciu algoritmu boli aplikované na vzorovej mape zo stránky pre zadanie.

Počet jedincov generácie 50, kríženie 0.9, mutácia 0.15, typ selekcie ruleta, elitarizmus nebol aplikovaný a v prípade, že sa nevykonalo kríženie, tak sa pridala „nová krv“ do generácie.



Meno: Jakub Hlavačka  
AIS ID: 96876

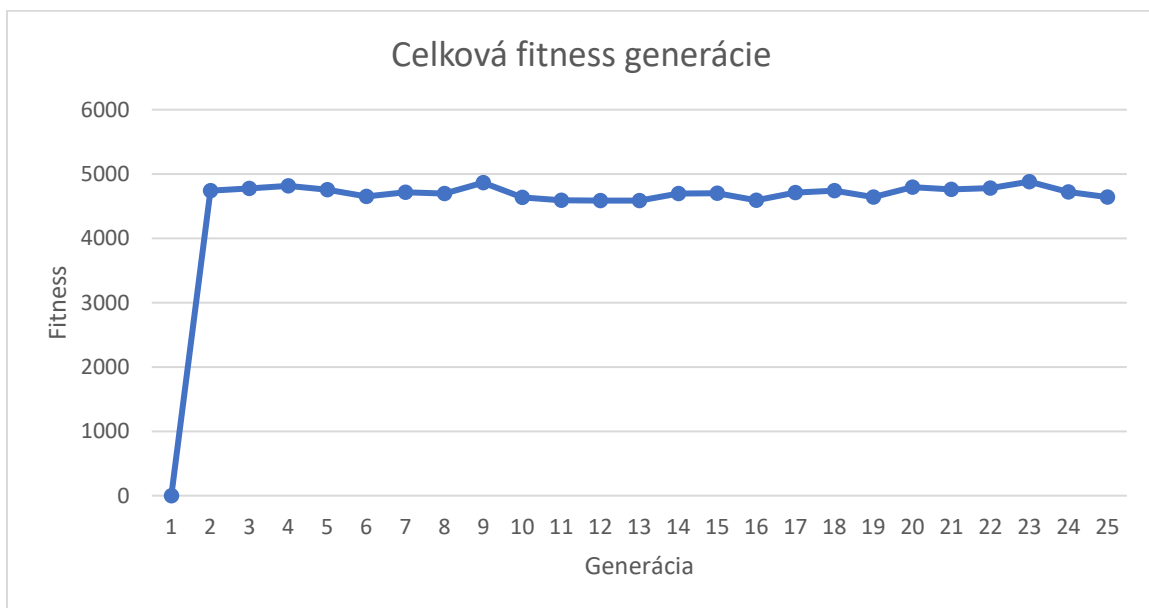
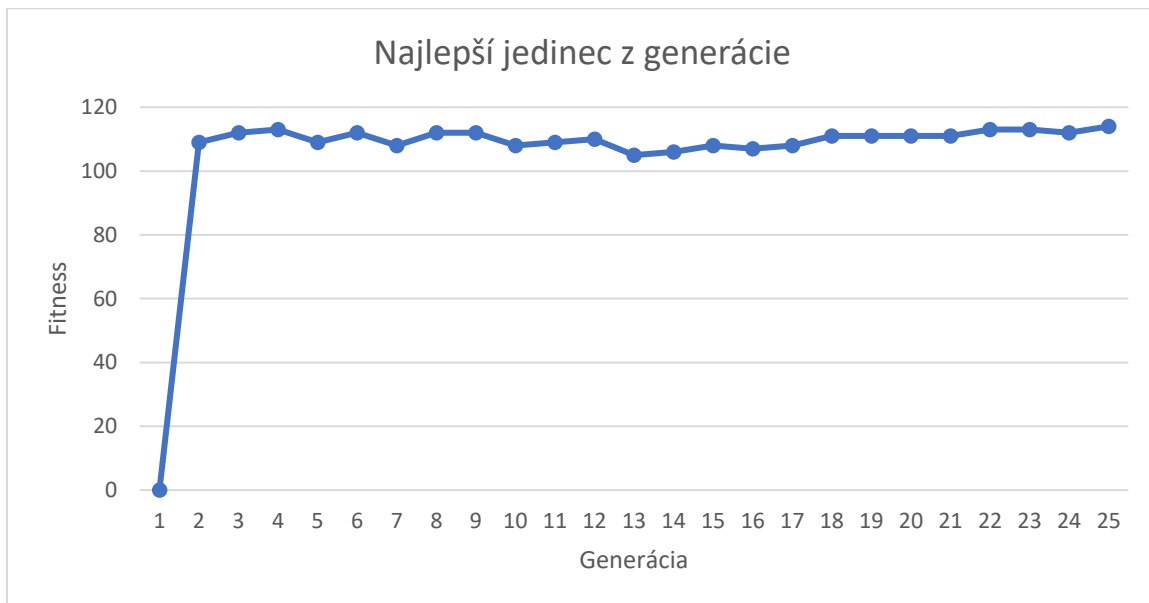
Počet jedincov generácie 50, kríženie 0.9, mutácia 0.15, typ selekcie turnaj, elitarizmus nebol aplikovaný a v prípade, že sa nevykonalo kríženie, tak sa pridala „nová krv“ do generácie.



Meno: Jakub Hlavačka

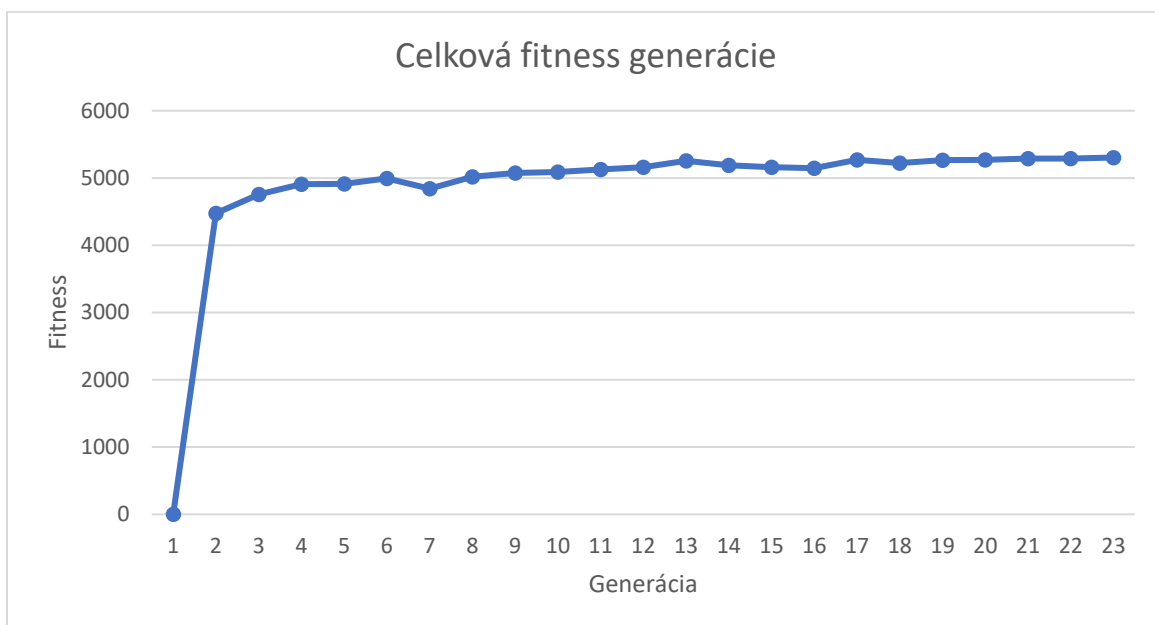
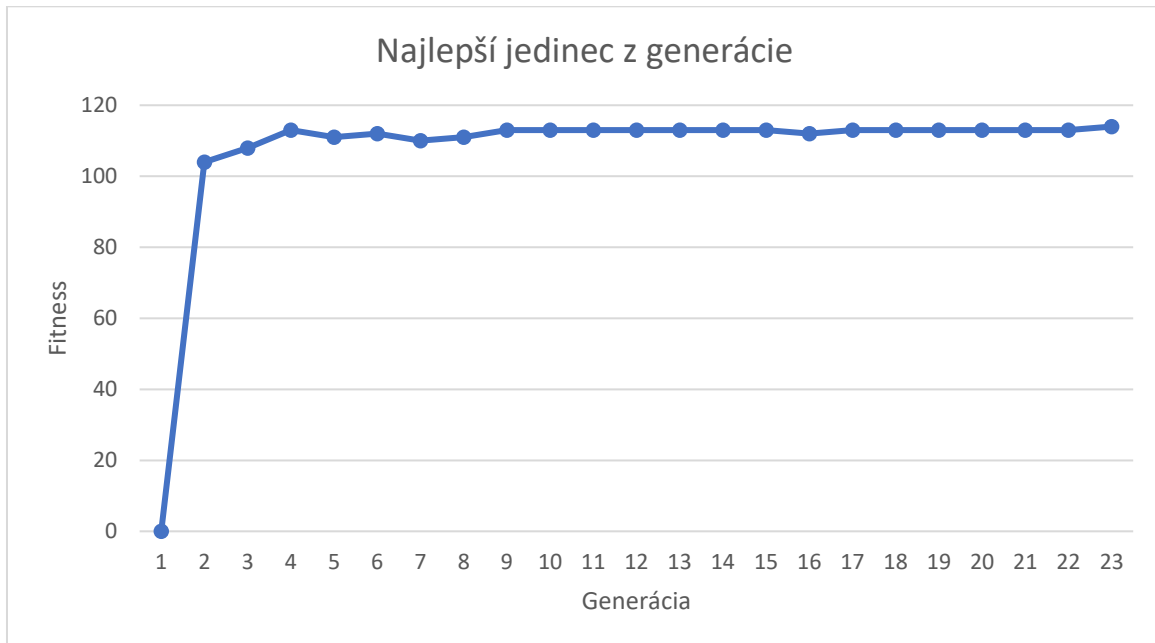
AIS ID: 96876

Počet jedincov generácie 50, kríženie 0.8, mutácia 0.1, typ selekcie ruleta, elitarizmus nebol aplikovaný a v prípade, že sa nevykonalo kríženie, tak sa pomocou rulety vybral jedinec zo starej generácie, ktorý prešiel do novej.



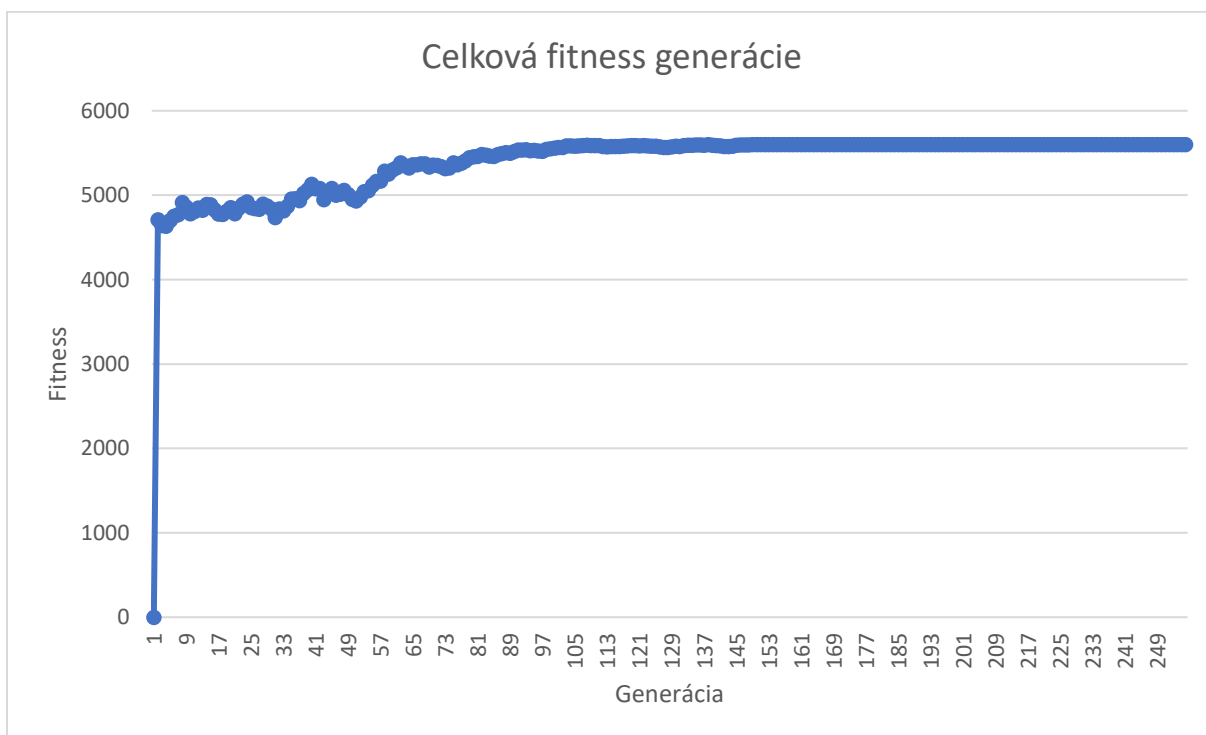
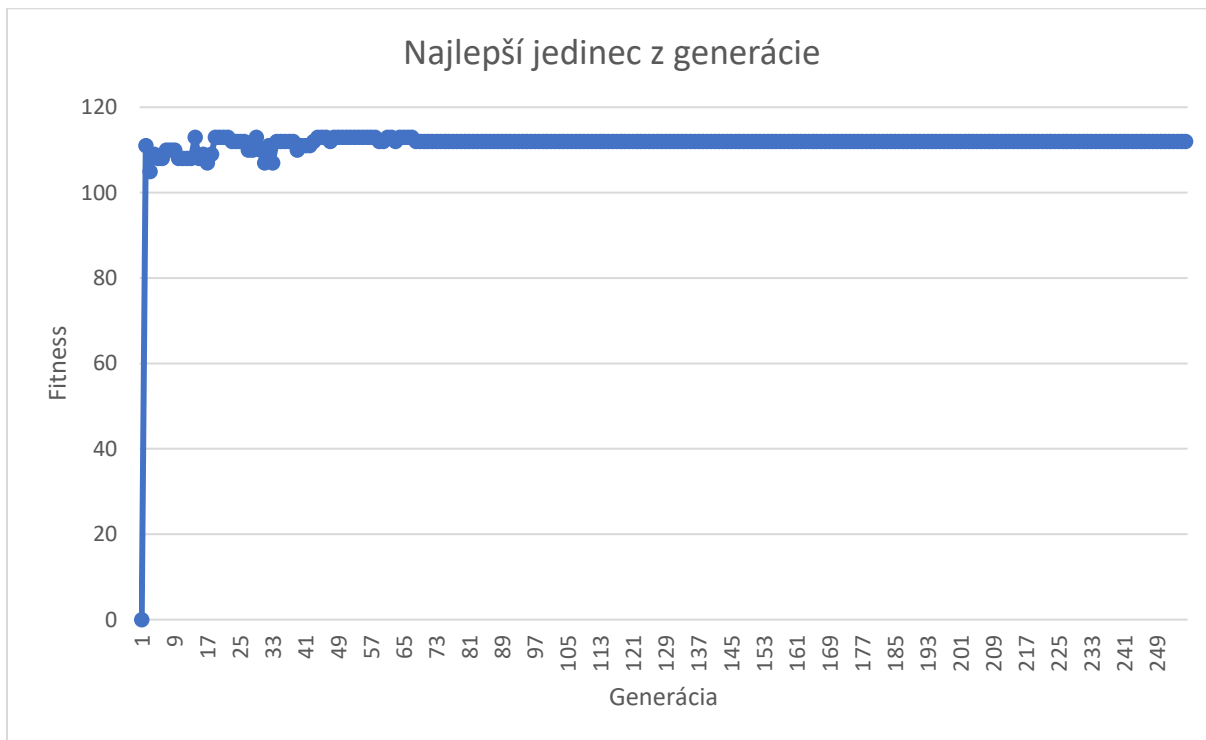
Meno: Jakub Hlavačka  
AIS ID: 96876

Počet jedincov generácie 50, kríženie 0.8, mutácia 0.1, typ selekcie turnaj, elitarizmus nebol aplikovaný a v prípade, že sa nevykonalo kríženie, tak sa pomocou turnaju vybral jedinec zo starej generácie, ktorý prešiel do novej.



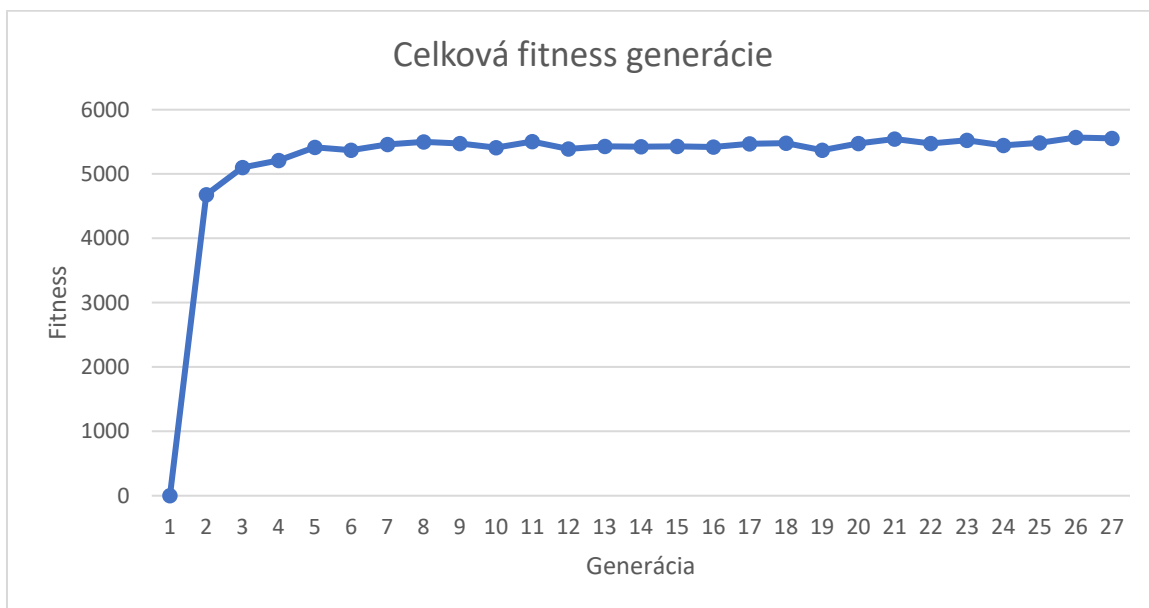
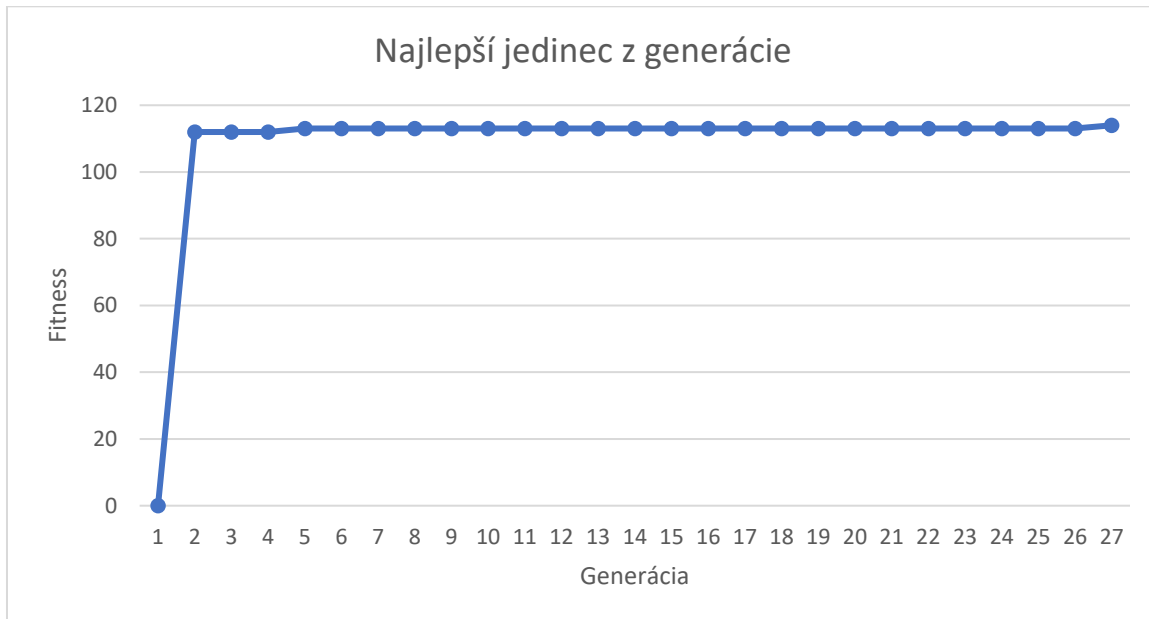
Meno: Jakub Hlavačka  
AIS ID: 96876

Počet jedincov generácie 50, kríženie 0.5, mutácia 0, typ selekcie ruleta, elitarizmus nebol aplikovaný a v prípade, že sa nevykonalo kríženie, tak sa pomocou rulety vybral jedinec zo starej generácie, ktorý prešiel do novej. Algoritmus ani len nenašiel riešenie, pretože nebola aplikovaná mutácia.



Meno: Jakub Hlavačka  
AIS ID: 96876

Počet jedincov generácie 50, kríženie 0.3, mutácia 0.1, typ selekcie turnaj, elitizmus nebol aplikovaný a v prípade, že sa nevykonalo kríženie, tak sa pomocou turnaju vybral jedinec zo starej generácie, ktorý prešiel do novej.



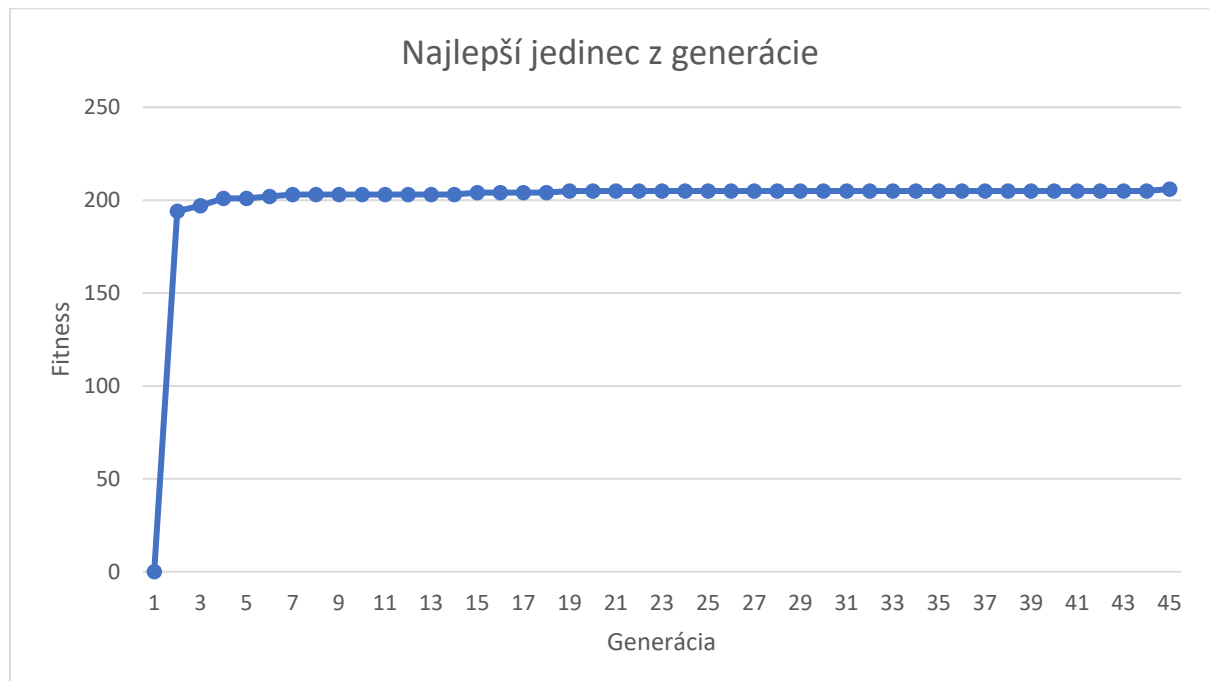
Meno: Jakub Hlavačka  
AIS ID: 96876

Nasledujúce zachytávania fitness funkcie bolo aplikované na tejto mape:

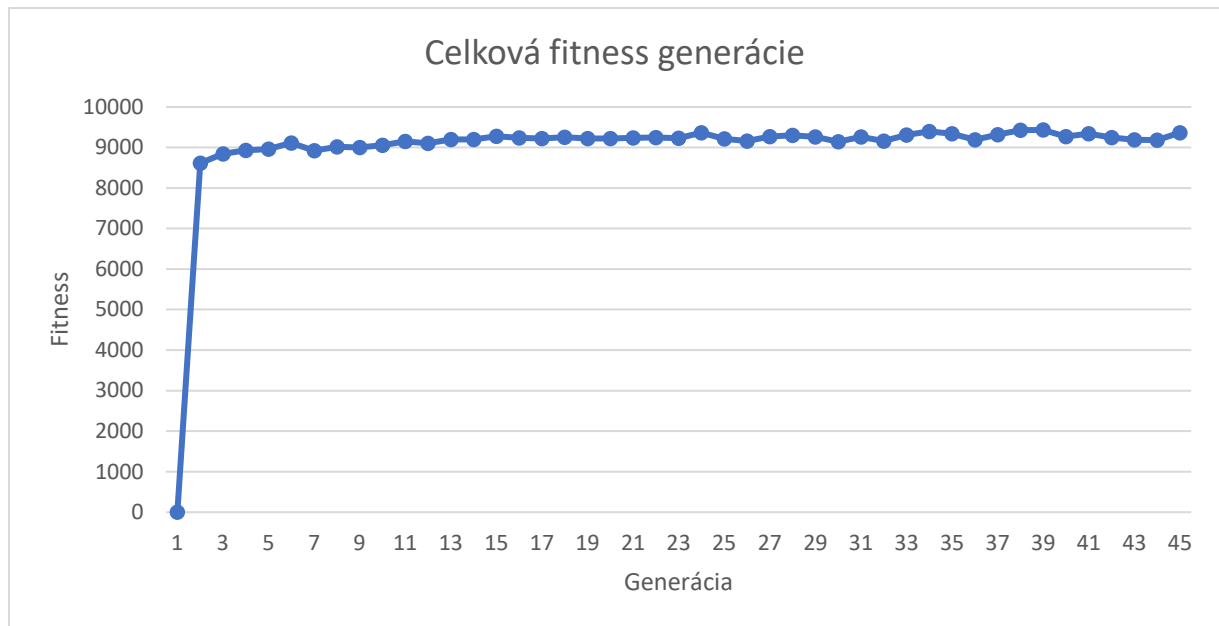
```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 K K K K K K K
0 0 0 0 0 K 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 K 0 0 0 0 0 0 0 0 0 0 0 0 0 K K 0 0 0 0
0 0 0 0 K 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 K 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 K K 0 0 0
0 0 0 0 0 0 0 0 0 K K 0 0 0 0 0 0 0 0 0
0 0 0 K K 0 K 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 K K K K 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 K K K K 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 K 0 0 0 K K 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 K K K 0 0 0
```

Počet jedincov generácie 50, kríženie 0.9, mutácia 0.15, typ selekcie turnaj, elitizmus bol aplikovaný a v prípade, že sa nevykonalo kríženie, tak sa pomocou turnaju vybral jedinec zo starej generácie, ktorý prešiel do novej.

Pre takúto mapu som pomocou rulety nedokázal porať celú záhradku.



Meno: Jakub Hlavačka  
AIS ID: 96876



## Záver

V súbore vstupy.txt sú najzaujímavejšie a najťažšie záhradky, ktoré som testoval. Záhradku, ktorú má algoritmus porať je potrebné zadať do súboru vstup.txt a ostatné parametre sa zadávajú po spustení programu.