Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 19 Bratislava 4

Umelá inteligencia

Strojové učenie sa – evolučný algoritmus a evolučné programovanie

Zenová záhrada

Vypracoval: Jozef Varga

Cvičiaci: Mgr. Irina Malkin Ondik, PhD.

Ak. rok: 2017/2018

Zadanie

Zenová záhradka je plocha vysypaná hrubším pieskom (drobnými kamienkami). Obsahuje však aj nepohyblivé väčšie objekty, ako napríklad kamene, sochy, konštrukcie, samorasty. Mních má upraviť piesok v záhradke pomocou hrablí tak, že vzniknú pásy ako na nasledujúcom obrázku.

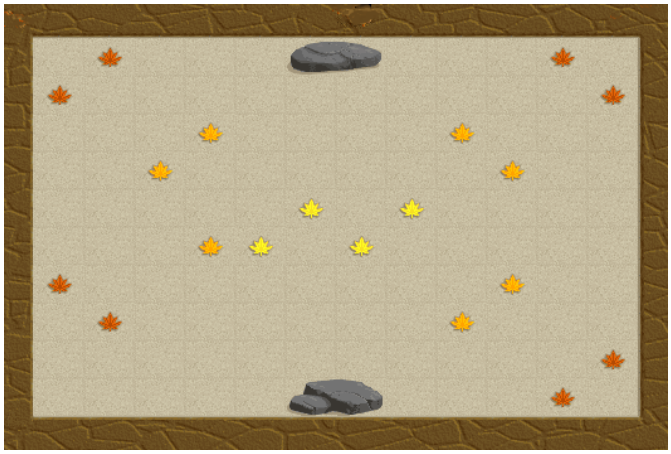
Obrázok 1Ukážka zenovej záhrady

0	0	1	0	0	0	0	0	10	10	8	9
0	0	1	0	0	K	0	0	10	10	8	9
0	K	1	0	0	0	0	0	10	10	8	9
0	0	1	1	K	0	0	0	10	10	8	9
0	0	K	1	0	0	0	0	10	10	8	9
2	2	2	1	0	0	0	0	10	10	8	9
3	3	2	1	0	0	0	0	K	K	8	8
4	3	2	1	0	0	0	0	5	5	5	5
4	3	2	1	0	0	0	11	5	6	6	6
4	3	2	1	0	0	0	11	5	6	7	7

Obrázok 2Priebežný stav zenovej záhrady

Pásy môžu ísť len vodorovne alebo zvislo, nikdy nie šikmo. Začína vždy na okraji záhradky a ťahá rovný pás až po druhý okraj alebo po prekážku. Na okraji - mimo záhradky môže chodiť ako chce. Ak však príde k prekážke - kameňu alebo už pohrabanému piesku - musí sa otočiť, ak má kam. Ak má voľné smery vľavo aj vpravo, je jeho vec, kam sa otočí. Ak má voľný len jeden smer, otočí sa tam. Ak sa nemá kam otočiť, je koniec hry. Úspešná hra je taká, v ktorej mních dokáže za daných pravidiel pohrabať celú záhradu, prípade maximálny možný počet políčok. Výstupom je pokrytie danej záhrady prechodmi mnícha. Pokrytie zodpovedajúce presne prvému obrázku (priebežný stav) je napríklad vidno na obr. 2.

Navyše úlohu je možné rozšíriť tak, že mních navyše zbiera popadané lístie. Listy musí zbierať v poradí: najprv žlté, potom pomarančové a nakoniec červené. Príklad vidno na obrazku 3



Obrázok 3 Zenová zahrada zbieranie popadanych listov

Opis riešenia a použitý algoritmus

Na vypracovanie tohoto zadania som využil klasický genetický algoritmus. Ako prvé som si vytvoril generáciu náhodne vygenerovaných jedincov. Každý jedinec obsahoval gény. Počet génov jedinca sa vypočítalo ako $počet\ génov = \frac{\text{obvod}}{2} + kamene$. Počet jedincov v generácií je 100. Každý jedinec obsahuje hodnotu fitnes funkcie (počet pohrabaných políčok) ktorá je na začiatku inicializovaná na 0. V programe je stanovené že program skončí ak nájde riešenie alebo ak vygeneruje 1500 generácií.
Po vytvorení prvej generácie postupne púšťam jedincov na mapu. Každý jedinec využije svoje gény v poradí akom ich má. Každý gén je vlastne číslo, ktoré definuje kadiaľ má mních vstúpiť.
Napríklad mapa veľkosti 2 x 2 by vyzerala takto:

	0	1	
7			2
6			3
	5	4	

Obrázok 4 Mapa 2x2 ukazujúca vstupné hodnoty (gény)

Čísla okolo mapy ukazujú možné vstupy na mapu. Tieto čísla (gény) hovoria taktiež o počiatočnom smere mnícha. Smer z týchto čísel dostaneme nasledovne:

- Ak je to číslo menšie ako šírka mapy mních ide dole (0, 1)
- Ak je to číslo väčšie ako šírka avšak menšie ako šírka + výška mapy mních ide doľava (2, 3)
- Ak je to číslo väčšie ako šírka + výška avšak menšie ako 2 * šírka + výška mapy mních ide hore (5, 4)
- Inač mních ide doprava (6, 7)

Ak spustíme náš program na hore uvedenej mape vypíše nám jedinca ktorý našiel cestu J[**3** 4 **7** 2] viď obr.5

	0	1	
7	4	4	2
6	2	2	**3**
	5	4	

Obrázok 5 Vyplnená mapa 2x2

Každý gén prejde mapu a zapisuje do nej číslo svojho poradia + hodnota 2 pretože v mape je neohrabané pole reprezentované 1-kou a kamene sú reprezentované 0. V našom prípade gén 3 nám bude zapisovať čísla 2 a gén 7 nám zapisuje čísla 4.
Každý jedinec si počíta fitnes funkciu -> počet pohrabaných políčok mapy. Každý gén pri svojom pohybe (pohrabaní políčka) inkrementuje fitnes hodnotu jedinca.
Ak narazí na prekážku začne sa rozhodovať ktorým smerom má ďalej ísť.
Rozhodovanie funguje podľa smeru akým sa pohybuje:

- Ak idem z pravej strany na ľavú ako prvé kontrolujem či sa môžem posunúť dole
- Ak idem z ľavej strany na pravú ako prvé kontrolujem či sa môžem posunúť dole
- Ak idem zhora dole najskôr kontrolujem pravú stranu
- Ak idem zdola hore najskôr kontrolujem pravú stranu

Ak sa daným smerom, ktorý ako prvé kontrolujem nedá ísť (je tam prekážka – kameň, políčko už bolo pohrabané) skontrolujem druhú stranu (ak som kontroloval najskôr ľavú stranu, tak následne skontrolujem pravú a naopak)

Ak daný gén vstúpi na mapu a vyjde z nej na kraji, všetko je v poriadku, avšak ak vojde a dostane sa na miesto z ktorého sa nemá kam pohnúť (z každej strany som obkolesený prekážkou alebo sám sebou) pustím tento gén ešte raz avšak mu zmením hodnotu ktorú nastavuje (svoje poradie) na hodnotu neohrabaného poľa -> 1. Na konci vynulujem hodnotu fitnes ktorú by daný gén vytvoril (tým pričítam nulu ku celkovej fitnes jedinca) daného pri posune, tým vrátim mapu do stavu akoby sa daný gén ani nespustil.
Takto pustím všetky gény daného jedinca. Na konci skontrolujem či daný jedinec nie je riešením alebo či aspoň nie je doposiaľ najlepším riešením. Ak je najlepším riešením uložím jeho gény, ak je to celkové riešenie teda fitnes jedinca je rovná šírke mapy * výška mapy – počet kameňov skončím algoritmus a vypíšem finálnu mapu.
Ak skontrolujem celú generáciu a nenájdem riešenie zoradím si jedincov v generácií podľa fitnes funkcie zostupne a danú generáciu zmutujem a skrížim čím vytvorím novú generáciu. Vždy pri vytváraní generácie zmutujem 50% najlepších jedincov predošlej generácie (pri zoradení je to prvá polovica) a zvyšných 50 % jedincov (druhú polovicu) vytvorím skrížením. Dané generácie vytváram a opakujem algoritmus pokiaľ nevytvorím 1500 generácií alebo nenájdem riešenie.

Mutácie
Pri mutovaný jedinca vymením jeho 2 ľubovoľne vygenerované gény. Každá mutácia je možná s pravdepodobnosťou 100% avšak je to možné meniť v programe no pri tomto nastavení to bolo najefektívnejšie.

Kríženie
Na začiatku pri spustení aplikácie si užívateľ vyberie typ výberu jedincov ktorý sa budú krížiť. Tieto možnosti sú dve a to proporcionálnou selekciou (ruleta) alebo selekciou ohodnotením.

Výber jedinca proporcionálnou selekciou (ruletou) je nasledovný:

- Sčítam do x všetky fitnes hodnoty jedincov danej generácie
- Náhodne vygenerujem číslo y v rozsahu 0-(x - 1)
- Postupne prechádzam všetkých jedincov v poradí akom sú a odčítavam od y ich fitnes funkciu pokiaľ nedostanem záporné číslo alebo nulu. Ak dostanem jedno z toho, uložím poradové číslo jedinca a spustím prehľadávanie ešte raz, nakoľko na kríženie potrebujeme dvoch

Výber jedinca selekciou ohodnotením:

- Vyberám jedincov postupne zo začiatku a z konca. Tým že máme generáciu zoradenú vyberáme tým najsilnejšieho z najslabším, tak sa posunieme a vyberieme 2-hého najsilnejšieho a 2-hého najslabšieho
- ak mám jedincov z fitnes v tomto poradí : 1 4 13 24 tak mi vráti jedinca 1 s 24, 4 s 13.

Po výbere 2 jedincov, vyberiem prvého z nich a skopírujem ho, následne prechádzam každý jeden jeho gén a s pravdepodobnosťou 50% zamením daný gén z génom z druhého jedinca. Čiže napr. Ak mám 2 jedincov a to : J1(1 **2 3** 4) a J2(**5** 6 7 **8**). Tak skrížený jedinec J3 môže vyzerať nasledovne J3(5 2 3 6), tu vidíme že 1. a 4. gén sa zmenil na gén z J2.
Týchto jedincov postupne ukladám za zmutovaných.

Zbieranie listov:
Zbieranie listov upravilo hore uvedený algoritmus tak, že na začiatku si načítam početnosti jednotlivých farieb listov. Následne nastavím žlté listy na -1, pomarančové listy na -2

a červené listy na -3. V algoritme stačilo zmeniť to aby všetky listy boli považované génmi jedinca za prekážku, avšak bola premenná „listy“ ktorá určovala výnimku. Ako prvé sa tá premenná nastavila na -1 čím sa zberali žlté listy. Toto nastavenie pretrvalo pokiaľ počet žltých listov na mape sa nerovnal 0. V tej chvíli sa premenná „listy“ nastavila na -2 a tým sme zberali pomarančové listy. Takto sme postupovali aj pri červených.

Používateľské rozhranie

Program si vypýta všetky potrebné informácie. Stačí ísť podľa nápovedi viď obr.

```
Zadajte typ vyberu jedinca
        0 - Proporcionálna selekcia (ruleta)
        1 - SELEKCIA OHODNOTENIM
0
Zadaj veklost mapy
x:6
y:6
Zadaj pocet kamenov:1
Kamen suradnice kamena v tvare 'x y' :
Zadaj 1. kamen:0 0
Zadaj pocet zltych listov na mape:1
Zadaj 0. zlty list v tvare 'x y' :1 1
Zadaj pocet pomarancovych listov na mape:1
Zadaj 0. pomarancove list v tvare 'x y' :2 1
Zadaj pocet cervenych listov na mape:1
Zadaj 0. cervene list v tvare 'x y' :3 1


X  1  1  1  1  1
1  -1 -2 -3 1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1

Generacia:  0   Max: 35   POHRABANE

Geny najlepsieho jedinca ktorý mal fitnes 35 najdeneho v 0 generacii:
8 22 18 17 19 16 5 11 4 14 9 2 15

X  10 10 10 10 8
3  3  3  3  3  3
2  2  2  2  2  2
12 12 12 12 12 12
6  6  6  6  6  6
4  4  4  4  4  4
```

Obrázok 6 zadanie vstupu

Testovania

Svoj program som testoval postupne, najskôr som testoval jednoduchá mapy scenár A, B. Neskôr som skúšal aj stavy ktoré nemajú riešenie scenáre C, D, E, F alebo stavy ktoré sú zložitejšie (z Obr.1) scenár G, H. Nakoniec som testoval aj bonusové riešenie ktorým bolo pozbierať listy v poradí a to najskôr žlté, potom pomarančové a nakoniec červené. Napríklad scenár I alebo obrázok 6. Testy boli dôležité hlavne kvôli nastaveniu parametrov ako sú : percento jedincov ktorý budú zaradený do mutovania / kríženia, aká je percentuálna šanca že nový jedinec bude mať pri krížení gén z prvého jedinca a aká je šanca že z druhého jedinca, počet jedincov, počet nutných generácií ...

Scenár A)
Vstup:

```
0
5 5
3
0 0
0 1
4 4
0
0
0
```

Výstup :

```
Zadaj veklost mapy
x:y:Zadaj pocet kamenov:Kamen suradnice kamena v tvare :x y
Zadaj 1. kamen:Zadaj 2. kamen:Zadaj 3. kamen:

X  1  1  1  1
X  1  1  1  1
1  1  1  1  1
1  1  1  1  1
1  1  1  1  X

Generacia:  0   Max: 22   POHRABANE

Geny najlepsieho jedinca ktorý mal fitnes 22 najdeneho v 0 generacii:
12 9 1 14 4 18 3 16 0 6 5 13 10

X  4  2  8  6
X  4  2  8  6
5  4  2  8  6
5  4  2  8  6
5  4  2  8  X
```

Scenár B)
Vstup

```
1
5 5
3
0 0
0 1
4 4
0
0
0
```

Výstup:

```
X  1  1  1  1
X  1  1  1  1
1  1  1  1  1
1  1  1  1  1
1  1  1  1  X

Generacia:  0   Max: 22   POHRABANE

Geny najlepsieho jedinca ktorý mal fitnes 22 najdeneho v 0 generacii:
15 0 19 1 12 6 18 16 4 10 2 5 17

X  5  12 12 10
X  5  7  7  7
9  5  7  7  7
9  5  5  5  5
2  2  2  2  X
```

Scenár C)
Vstup:

```
0
3 3
5
0 0
2 0
1 1
0 2
2 2
0
0
0
```

Výstup:

```
Generacia:  1493   Max: 0   Min:  0 AVG:  0
Generacia:  1494   Max: 0   Min:  0 AVG:  0
Generacia:  1495   Max: 0   Min:  0 AVG:  0
Generacia:  1496   Max: 0   Min:  0 AVG:  0
Generacia:  1497   Max: 0   Min:  0 AVG:  0
Generacia:  1498   Max: 0   Min:  0 AVG:  0
Generacia:  1499   Max: 0   Min:  0 AVG:  0


X  1  X
1  X  1
X  1  X
```

Scenár D)
Vstup:

```
1
3 3
5
0 0
2 0
1 1
0 2
2 2
0
0
0
```

Výstup:

```
Generacia:  1492   Max: 0   Min:  0 AVG:  0
Generacia:  1493   Max: 0   Min:  0 AVG:  0
Generacia:  1494   Max: 0   Min:  0 AVG:  0
Generacia:  1495   Max: 0   Min:  0 AVG:  0
Generacia:  1496   Max: 0   Min:  0 AVG:  0
Generacia:  1497   Max: 0   Min:  0 AVG:  0
Generacia:  1498   Max: 0   Min:  0 AVG:  0
Generacia:  1499   Max: 0   Min:  0 AVG:  0


X  1  X
1  X  1
X  1  X
```

Scenár E)
Vstup:

```
0
5 5
7
0 0
1 2
2 0
0 1
2 1
3 4
3 3
0
0
0
```

Výstup:

```
Generacia:  1493   Max: 16   Min:  10 AVG:  15
Generacia:  1494   Max: 16   Min:  10 AVG:  15
Generacia:  1495   Max: 16   Min:  10 AVG:  15
Generacia:  1496   Max: 16   Min:  10 AVG:  15
Generacia:  1497   Max: 16   Min:  10 AVG:  15
Generacia:  1498   Max: 16   Min:  10 AVG:  15
Generacia:  1499   Max: 16   Min:  10 AVG:  15

Geny najlepsieho jedinca ktorý mal fitnes 16 najdeneho v 0 generacii:
15 12 4 11 14 13 17 1 7 10 3 9 0 5 19 8 18

X  1  X  8  4
X  1  X  8  4
8  X  8  8  4
8  8  8  X  4
2  2  2  X  4
```

Scenár F)
Vstup:

```
1
5 5
7
0 0
1 2
2 0
0 1
2 1
3 4
3 3
0
0
0
```

Výstup:

```
Generacia:  1489   Max: 16   Min:  9 AVG:  13
Generacia:  1490   Max: 16   Min:  9 AVG:  13
Generacia:  1491   Max: 16   Min:  9 AVG:  13
Generacia:  1492   Max: 16   Min:  9 AVG:  13
Generacia:  1493   Max: 16   Min:  9 AVG:  13
Generacia:  1494   Max: 16   Min:  9 AVG:  13
Generacia:  1495   Max: 16   Min:  9 AVG:  13
Generacia:  1496   Max: 16   Min:  10 AVG:  13
Generacia:  1497   Max: 16   Min:  11 AVG:  13
Generacia:  1498   Max: 16   Min:  10 AVG:  13
Generacia:  1499   Max: 16   Min:  10 AVG:  13

Geny najlepsieho jedinca ktorý mal fitnes 16 najdeneho v 2 generacii:
0 4 7 5 9 15 14 2 12 18 13 17 3 10 19 6 11

X  1  X  13 3
X  1  X  13 3
13 X  13 13 3
13 13 13 X  3
7  7  7  X  3
```

Scenár G)

Vstup:

```
0
12 10
6
1 2
2 4
4 3
5 1
9 6
10 6
0
0
0
```

Výstup:

```
Generacia:  0   Max: 108   Min:  59 AVG:  88
Generacia:  1   Max: 108   Min:  59 AVG:  91
Generacia:  2   Max: 110   Min:  60 AVG:  94
Generacia:  3   Max: 108   Min:  61 AVG:  95
Generacia:  4   Max: 108   Min:  59 AVG:  94
Generacia:  5   Max: 108   Min:  70 AVG:  95
Generacia:  6   Max: 109   Min:  61 AVG:  97
Generacia:  7   Max: 113   Min:  61 AVG:  95
Generacia:  8   Max: 114   POHRABANE

Geny najlepsieho jedinca ktorý mal fitnes 114 najdeneho v 8 generacii:
6 0 22 33 31 1 19 14 8 16 29 35 36 11 13 26 20 30 12 21 25 5 27 41 24 34 4 9

3  7  23 23 23 23 2  17 10 29 29 4
3  7  7  7  7  X  2  17 10 29 29 4
3  X  7  7  7  7  2  17 10 29 29 4
3  7  7  7  X  7  2  17 10 29 29 4
3  7  X  7  7  7  2  17 10 29 29 4
3  7  6  6  6  6  2  17 10 29 29 4
3  7  6  12 12 6  2  17 10 X  X  4
3  7  6  12 12 6  2  17 10 26 26 4
3  7  6  12 12 6  2  17 10 26 26 4
3  7  6  12 12 6  2  17 10 26 26 4
```

Scenár H)

Vstup:

```
1
12 10
6
1 2
2 4
4 3
5 1
9 6
10 6
0
0
0
```

Výstup:

```
Generacia:  0   Max: 108   Min:  46 AVG:  88
Generacia:  1   Max: 108   Min:  45 AVG:  89
Generacia:  2   Max: 108   Min:  45 AVG:  89
Generacia:  3   Max: 108   Min:  59 AVG:  90
Generacia:  4   Max: 110   Min:  63 AVG:  91
Generacia:  5   Max: 112   Min:  59 AVG:  92
Generacia:  6   Max: 110   Min:  59 AVG:  91
Generacia:  7   Max: 110   Min:  50 AVG:  91
Generacia:  8   Max: 114   POHRABANE

Geny najlepsieho jedinca ktorý mal fitnes 114 najdeneho v 8 generacii:
13 18 41 11 9 21 19 10 31 26 25 8 22 36 16 32 24 29 42 27 37 28 15 6 20 4 39 3

13 13 13 13 13 13 13 13 13 6  6  5
20 20 20 20 20 X  2  2  2  2  2  2
4  X  20 20 20 20 2  11 11 11 11 11
4  20 20 20 X  20 2  11 12 12 12 12
4  20 X  20 20 20 2  11 12 16 16 16
4  20 10 10 10 10 2  11 12 16 16 16
4  20 10 19 19 10 2  11 12 X  X  3
4  20 10 19 19 10 2  11 12 18 18 3
4  20 10 19 19 10 2  11 12 18 18 3
4  20 10 19 19 10 2  11 12 18 18 3
```

Scenár (LIST) I)

- Tento scenár je aj s obrázkom ktorý ukazuje kroky ako išiel daný jedinec.

Vstup:

0 12 10 4 5 0 6 0 5 9 6 9 4 4 5 5 4 6 5 7 4 7 2 3 3 2 3 5 8 2 9 3 8 7 9 6 8 1 0 0 1 0 6 1 7 10 0 11 1 10 9 11 8

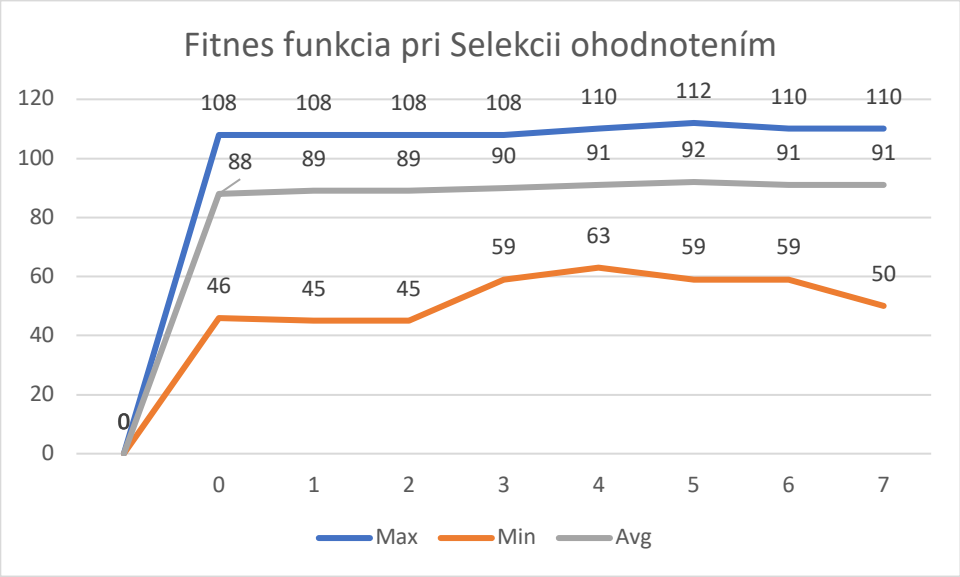
Výstup:

```
Generacia:  0   Max: 116   POHRABANE

Geny najlepsieho jedinca ktorý mal fitnes 116 najdeneho v 0 generacii:
5 35 1 21 27 13 0 7 4 2 31 34 42 16 19 43 12 8 28 29 39 24 32 33 20 38

8  4  4  4  4  X  X  9  9  9  9  9
8  3  3  3  3  3  3  3  3  3  3  7
8  3  3  3  3  3  3  3  3  3  3  7
8  3  3  3  3  3  3  3  3  3  3  7
8  3  3  3  3  3  3  3  3  3  3  7
8  3  3  3  3  3  3  3  3  3  3  7
8  3  3  3  3  3  3  3  3  3  3  7
3  3  3  3  3  3  3  3  3  3  3  7
3  3  3  3  3  3  3  3  3  3  3  7
13 13 12 12 12 X  X  5  5  5  5  5
```

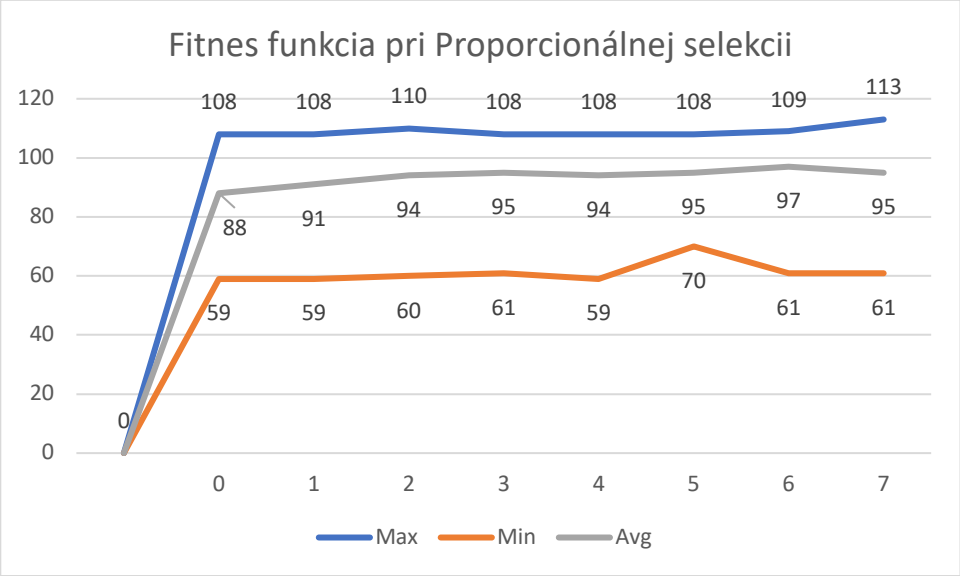
	0	1	2	3	4	5	6	7	8	9	10	11	
43	100	83	84	85	86	X	X	107	108	109	110	111	
42	101	27	26	25	24	23	22	21	20	19	18	92	13
41	102	28	61	62	63	64	65	66	67	68	17	93	14
40	103	29	60	47	46	45	44	43	42	69	16	94	15
39	104	30	59	48	49	50	51	52	41	70	15	95	16
38	105	31	58	57	56	55	54	53	40	71	14	96	17
37	106	32	33	34	35	36	37	38	39	72	13	97	18
36	82	81	80	79	78	77	76	75	74	73	12	98	19
35	1	2	3	4	5	6	7	8	9	10	11	99	20
34	115	116	112	113	114	X	X	91	90	89	88	87	21
	33	32	31	30	29	28	27	26	25	24	23	22	

Analýza

Scenár G)



Celkový priemer za všetky generácie je 90



Celkový priemer za všetky generácie je 94

Zhodnotenie

Tento projekt je podľa môjho názoru úspešný a jeho riešenie je dostatočne vyladené. Môj algoritmus bez akýchkoľvek problémov vyriešil aj zložitejšie zadania. Možné vylepšenie by bolo pridať ďalšie možnosti výberu jedincov na kríženie. Ďalej je možnosť zmeniť mutácie, napríklad aby mutácia gén ovplyvnila pri kontrole strán, takže napr. gén bude mať kladný a aj záporný, teda ak by bol gén kladný najskôr by skontroloval pravú stranu a naopak. Mutácia by ho následne s nejakou pravdepodobnosťou menila z kladného na záporné a naopak. Toto zadanie mi dalo nový pohľad na svet algoritmov a uviedlo ma do problematiky evolučných algoritmov a evolučného programovanie čo si myslím že v živote ešte využijem.