

Umelá inteligencia

1. zadanie

Dávid Gavenda

Andrej Jackulík

Zadanie – úvod

Našou úlohou bolo pod zistiť ako by mala firma investovať 10 miliónov Euro do bežných akcií, do preferovaných akcií, do podnikových dlhopisov, do štátnych dlhopisov a do úspor v banke.

Odhadované ročné výnosy v jednotlivých prípadoch sa nachádzajú v tabuľke.

Ďalej existovali podmienky, ktoré keď sa porušili boli pokutované podľa viacerých štýlov

- Mŕtva pokuta (dostatočne veľká pokuta, ktorá by mala reprezentovať teoreticky čo najväčšie číslo)
- Stupňová pokuta (pokuta = $o \cdot p$, kde o reprezentuje počet nesplnených obmedzení a p dostatočne veľkú konštantu)
- Pokuta úmerná porušeniu (pokuta sa rovnala vzdialenosti od prípustnej oblasti ² za každé porušenie)

Zadanie sme riešili v Matlabe a snažili sa získať maximálny možný zisk.

Druh investície	Odhad výnosu	Veľkosť investície
Bežné akcie	4 %	x_1
Preferované akcie	7 %	x_2
Podnikové dlhopisy	11 %	x_3
Štátne dlhopisy	6 %	x_4
Úspory v banke	5 %	x_5

Príprava na riešenie

Na riešenie sme použili genetický algoritmus. Základom bolo vytvoriť si správnu populáciu, ktorá v našom prípade obsahovala 50 jednotlivcov a každý z nich reprezentoval „investovanie“ našej firmy, teda každý jednotlivec obsahoval 5 hodnôt, ktoré značili do akých investícií investoval koľko Eur.

oldPop					
50x5 double					
	1	2	3	4	5
1	6.5855e+06	8.1893e+06	5.2599e+06	4.1317e+06	2.8275e+06
2	7.1701e+06	8.0295e+06	6.0907e+06	6.4352e+06	7.6716e+06
3	1.0109e+06	7.4598e+06	1.5078e+06	1.4279e+06	9.7545e+06
4	8.8265e+06	2.2013e+06	3.3390e+05	4.9726e+06	7.9754e+06
5	5.1801e+06	3.3737e+06	2.3787e+06	8.0522e+06	3.6851e+06
6	8.9301e+06	3.5996e+06	2.5518e+06	3.0829e+06	8.1195e+06
7	9.8381e+06	2.8233e+06	4.4762e+06	6.0162e+06	2.7868e+06
8	5.8310e+05	1.5079e+06	4.1319e+06	1.2881e+06	5.7345e+05
9	8.3672e+06	7.4195e+06	6.9506e+06	2.0188e+06	4.8607e+06
10	2.3595e+06	7.4058e+06	2.8523e+06	5.7699e+06	6.8072e+06
11	4.1174e+06	1.8717e+06	3.3642e+05	3.8112e+06	9.4206e+06
12	3.7405e+06	6.3746e+06	3.3778e+06	8.6805e+06	4.4018e+06
13	8.2357e+06	2.9551e+06	5.4488e+06	9.1360e+06	9.0387e+06

Ako vyzerá príklad časti originálnej populácie

Populácia bola náhodne generovaná aby hodnoty x_1, x_2, x_3, x_4, x_5 boli v rozmedzí 0 až 10 miliónov.

Následne sme si vytvorili maticu ziskov, pôvodne sa jednalo len o čistý zisk bez ohľadu na pokuty, každý jednotlivec sa vynásobil maticou, ktorá obsahovala hodnoty ziskov podľa investovaných Eur a následne sčítal celý zisk

zisk	
50x1 double	
	1
1	-1.3504e+06
2	-1.7136e+06
3	-1.5196e+06
4	-1.0095e+06
5	-7.8848e+05
6	-1.5769e+06
7	-1.3155e+06
8	-2.0750e+06
9	-1.6191e+06
10	-2.3209e+06
11	-1.2835e+06
12	-2.1335e+06
13	-1.6185e+06

Príklad ziskov časti populácie

Môžeme si všimnúť že naše zisky sú záporne, jednalo sa čisto o zjednodušenie riešenia, keďže základné funkcie pracujú aby sme zistili čo najmenšiu hodnotu a v našom prípade sa snažíme získať čo najväčší zisk. V tomto prípade je matica ziskov naša fitness funkcia, podľa ktorej sa budeme riadiť.

Genetický algoritmus

Naším ďalším krokom bolo efektívne a správne pracovať s aktuálnou generáciou a vytvoriť novú generáciu, ktorá by bola lepšia ako predošlá a čo najviac sa približovala ideálnemu výsledku.

Tvorba novej generácie sa skladala zo 6 častí.

- Vybrali sme prvok s najvyšším ziskom
- Vybrali sme 5x najlepšie prvok, 4x druhý najlepší, 3x tretí a 2x štvrtý
- Použili sme turnajový výber a vybrali 30 jedincov
- Použili sme obyčajnú mutáciu s mierou početnosti mutovania 0,35
- Použili sme aditívnu mutáciu s mierou početnosti mutovania 0,1
- Vygenerovali sme 5 čisto nových jednotlivcov

Následne sme spojili všetky prvky do novej generácie

```
theBest = selbest(oldPop, zisk, 1);  
  
bestSel = selbest(oldPop, zisk, [ 5 4 3 2 ]);  
  
newPop = seltourn(oldPop, zisk, 30);  
  
newPop = mutx(newPop, 0.35, space);  
  
newPop = muta(newPop, 0.1, amp, space);  
  
mixPop = genrpop(5, space);  
|  
oldPop = [theBest; bestSel; newPop; mixPop];
```

Dané hodnoty boli zvolené na základe testov, nie je to najviac efektívne ako by mohlo ale postačuje to. Kľúčovou časťou bola obyčajná mutácia, ktorá robila najväčší rozdiel v celkovom výsledku a samozrejme počet generácií.

Príkladom možnej „zlej“ zmeny je zvýšiť početnosť mutovania obyčajnej mutácie na 1 a tým pádom vždy len vytvárať 30 nových prvkov. Vyzeralo by to nasledujúco.

```
newPop = mutx(newPop, 1, space);
```

Táto zmena sa dosť podpísala na konečnom výsledku, ktorý sa stal menej presný a menej pravidelný.



50x5 double

	1	2	3	4	5
1	4.4335e+06	5.0532e+06	7.8983e+06	9.7898e+06	4.9023e+06
2	5.7368e+06	1.9642e+06	6.8903e+06	7.7641e+06	2.2711e+06
3	4.7461e+06	4.5809e+06	3.5118e+06	9.1939e+06	1.4287e+06
4	4.5676e+06	6.7834e+06	6.7160e+06	1.2749e+06	8.7712e+06
5	6.4024e+06	3.4745e+06	3.3606e+06	1.2703e+06	6.1797e+06
6	9.3094e+06	3.5277e+06	8.3944e+06	6.9690e+06	7.1668e+06
7	5.9947e+05	4.6311e+06	2.4740e+06	4.8554e+06	5.8875e+06
8	1.0966e+06	2.4002e+06	7.9447e+06	1.0548e+06	8.0149e+06
9	9.6799e+06	9.2642e+06	9.8084e+06	2.6388e+06	2.0135e+05
10	7.5813e+06	5.2041e+06	2.7497e+06	8.7705e+06	6.8551e+06
11	2.9344e+06	6.5470e+06	1.1665e+06	9.4511e+06	5.8531e+06
12	3.3281e+06	2.1009e+06	5.5571e+06	8.0281e+06	9.6354e+06
13	6.5852e+06	4.0528e+06	5.4180e+06	3.9318e+06	6.2282e+06
14	4.2860e+06	1.4785e+06	8.7992e+05	9.7123e+06	1.9889e+06
15	6.4380e+06	7.6746e+06	5.1004e+06	6.6402e+06	5.9947e+05
16	7.4854e+06	4.5677e+06	8.6258e+06	9.2714e+06	2.5062e+06
17	3.7101e+06	1.0564e+06	3.9797e+06	6.9565e+06	7.8463e+06
18	9.5372e+06	2.0784e+06	9.3294e+06	2.4829e+06	5.3455e+06
19	4.8791e+06	3.9653e+06	1.0536e+06	3.4396e+06	6.7232e+05
20	9.4490e+06	3.8249e+06	5.0217e+06	8.4941e+05	2.5253e+06
21	2.4055e+06	8.3810e+06	9.1727e+06	9.6658e+06	9.2077e+06
22	5.1972e+06	7.5501e+06	9.8175e+06	5.2282e+06	4.6915e+06
23	4.3147e+06	1.0233e+06	9.1851e+06	4.1019e+06	9.7375e+06
24	2.9362e+06	1.4780e+06	9.0913e+05	4.3729e+06	6.5531e+06
25	5.7135e+06	8.1822e+06	3.5399e+06	7.3295e+06	9.9995e+06
26	7.1750e+06	6.2846e+06	7.7177e+06	5.5609e+06	6.1707e+06
27	8.7671e+06	5.8143e+06	6.3540e+06	3.1875e+06	6.0704e+06
28	7.9940e+05	2.2523e+06	6.3326e+06	3.8546e+06	1.3149e+06
29	8.5649e+06	1.7475e+06	9.6939e+04	5.5096e+06	1.3406e+06
30	5.5507e+06	5.9749e+06	5.3910e+06	2.5900e+06	4.7705e+06
31	4.0462e+06	5.8452e+06	6.9762e+06	2.0376e+06	2.7991e+06
32	2.2983e+06	8.0127e+06	8.7666e+06	9.2267e+06	2.5560e+06
33	9.1428e+06	1.3548e+06	3.3362e+06	8.7425e+06	1.3465e+05
34	1.3111e+06	8.0693e+06	9.6037e+06	9.5658e+06	4.1199e+06
35	5.5631e+06	7.3317e+05	5.3936e+06	8.3200e+06	6.2527e+06
36	1.3431e+06	9.2594e+06	5.6506e+06	3.9260e+06	8.9119e+06
37	4.2290e+06	6.7365e+05	7.7529e+06	2.9269e+06	4.3102e+06
38	5.2263e+06	3.2936e+06	1.0807e+05	5.8723e+06	3.1117e+06
39	3.5357e+06	1.8150e+06	6.6764e+06	5.4565e+06	9.5167e+06
40	5.9004e+06	8.9469e+05	3.9215e+06	6.8813e+06	1.5266e+04

50x5 double

	1	2	3	4	5
1	9.7613e+05	1.9796e+06	5.7246e+06	2.4814e+06	2.0810e+06
2	9.7613e+05	1.9796e+06	5.7246e+06	2.4814e+06	2.0810e+06
3	9.7613e+05	1.9796e+06	5.7246e+06	2.4814e+06	2.0810e+06
4	9.7613e+05	1.9796e+06	5.7246e+06	2.4814e+06	2.0810e+06
5	9.7613e+05	1.9796e+06	5.7246e+06	2.4814e+06	2.0810e+06
6	9.7613e+05	1.9796e+06	5.7246e+06	2.4814e+06	2.0810e+06
7	7.9940e+05	2.2523e+06	6.3326e+06	3.8546e+06	1.3149e+06
8	7.9940e+05	2.2523e+06	6.3326e+06	3.8546e+06	1.3149e+06
9	7.9940e+05	2.2523e+06	6.3326e+06	3.8546e+06	1.3149e+06
10	7.9940e+05	2.2523e+06	6.3326e+06	3.8546e+06	1.3149e+06
11	2.9362e+06	1.4780e+06	9.0913e+05	4.3729e+06	6.5531e+06
12	2.9362e+06	1.4780e+06	9.0913e+05	4.3729e+06	6.5531e+06
13	2.9362e+06	1.4780e+06	9.0913e+05	4.3729e+06	6.5531e+06
14	4.8791e+06	3.9653e+06	1.0536e+06	3.4396e+06	6.7232e+05
15	4.8791e+06	3.9653e+06	1.0536e+06	3.4396e+06	6.7232e+05
16	5.2263e+06	3.2936e+06	1.0807e+05	1.8498e+06	3.1117e+06
17	4.5676e+06	7.1112e+06	6.7160e+06	1.2749e+06	7.1408e+06
18	6.4380e+06	7.6746e+06	5.1004e+06	6.6402e+06	5.9947e+05
19	9.4490e+06	3.1441e+06	5.0217e+06	8.4941e+05	7.0491e+05
20	2.1915e+06	1.4780e+06	2.7731e+06	6.3199e+06	6.5531e+06
21	1.8933e+06	4.5809e+06	3.5118e+06	9.1939e+06	1.4287e+06
22	1.3431e+06	9.2594e+06	3.6854e+06	3.9260e+06	8.4621e+06
23	9.1428e+06	1.3548e+06	3.3533e+06	8.7425e+06	1.3465e+05
24	4.9309e+06	1.9796e+06	5.7246e+06	2.4814e+06	2.0810e+06
25	9.7613e+05	5.1248e+06	5.7246e+06	2.4803e+06	2.0810e+06
26	3.5357e+06	8.3167e+06	6.6764e+06	5.4565e+06	9.5167e+06
27	1.0966e+06	2.4002e+06	7.9559e+06	1.0660e+06	6.6453e+06
28	1.4898e+06	1.0564e+06	3.9797e+06	6.9565e+06	7.8463e+06
29	6.2796e+06	1.6385e+06	6.7917e+05	3.0767e+06	1.3573e+06
30	7.9940e+05	2.2523e+06	1.0102e+06	3.8387e+06	1.3149e+06
31	3.9485e+06	1.4780e+06	3.1888e+06	2.8459e+06	6.5531e+06
32	5.5631e+06	7.3317e+05	5.3936e+06	8.3200e+06	8.7277e+06
33	5.5507e+06	5.3278e+06	5.3910e+06	2.5926e+06	4.7705e+06
34	4.8791e+06	3.9653e+06	1.0536e+06	3.4396e+06	6.7232e+05
35	3.7101e+06	1.0564e+06	3.9797e+06	6.9565e+06	8.2547e+06
36	3.2742e+06	3.6583e+06	9.7617e+06	7.1588e+06	1.6711e+06
37	1.0966e+06	2.4002e+06	7.9447e+06	6.7818e+05	8.0149e+06
38	9.4490e+06	3.8249e+06	5.0217e+06	8.4941e+05	7.6328e+06
39	4.2860e+06	1.4785e+06	7.1059e+06	4.9969e+06	1.9889e+06
40	2.9344e+06	6.5470e+06	1.1665e+06	9.4511e+06	5.8531e+06

Tu môžeme vidieť rozdiel medzi starou generáciou a novou .

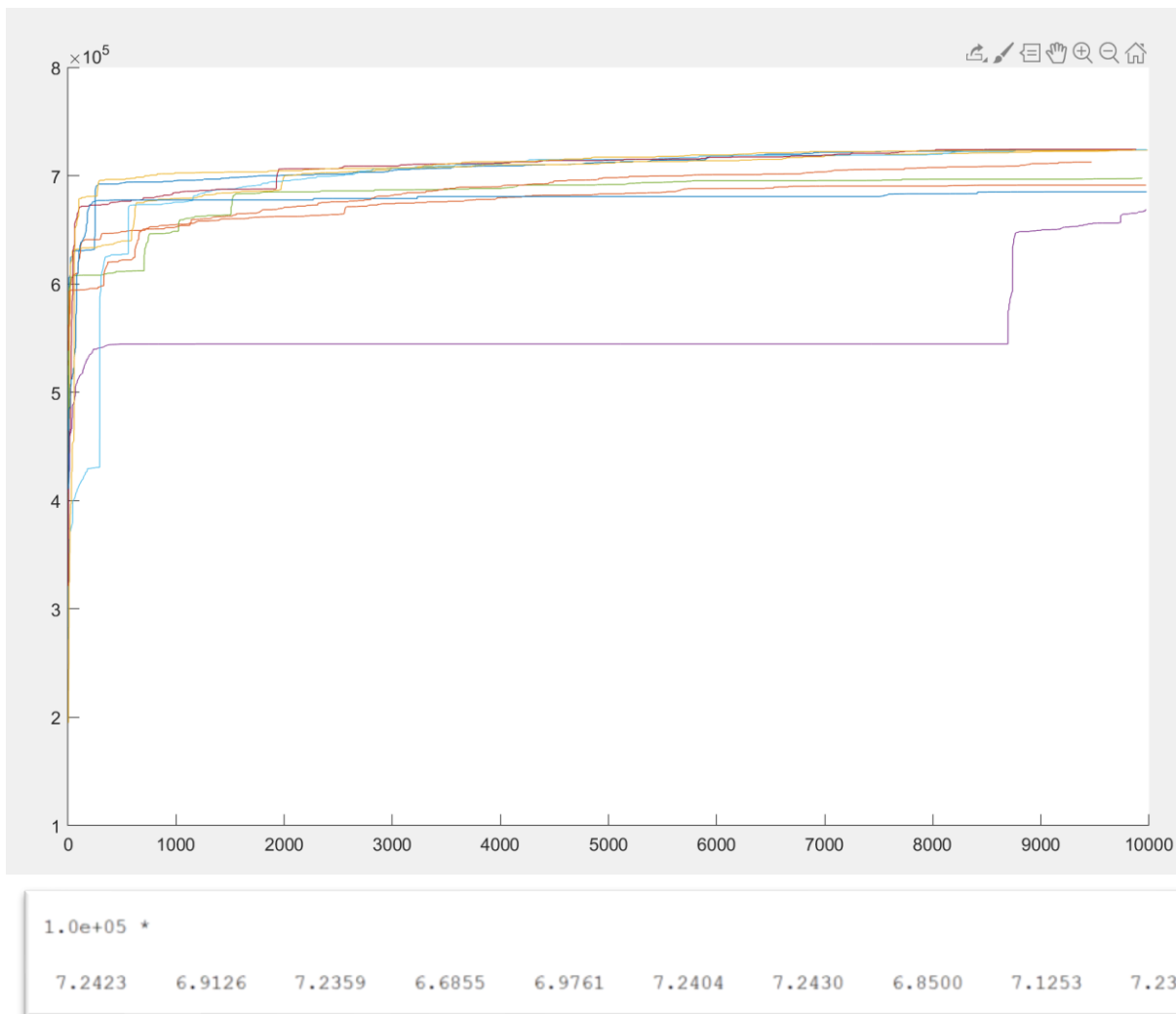
(je to len ich časť)

Dosiahnuté výsledky

Na všetky behy aj typy pokút bola použitá rovnaká tvorba nových generácií, spomenutá vyššie.

Naším cieľom je dosiahnuť zisk 750 000 pri ideálnom rozdelení investícií.

0	2 500 000	2 500 000	2 500 000	2 500 000
---	-----------	-----------	-----------	-----------

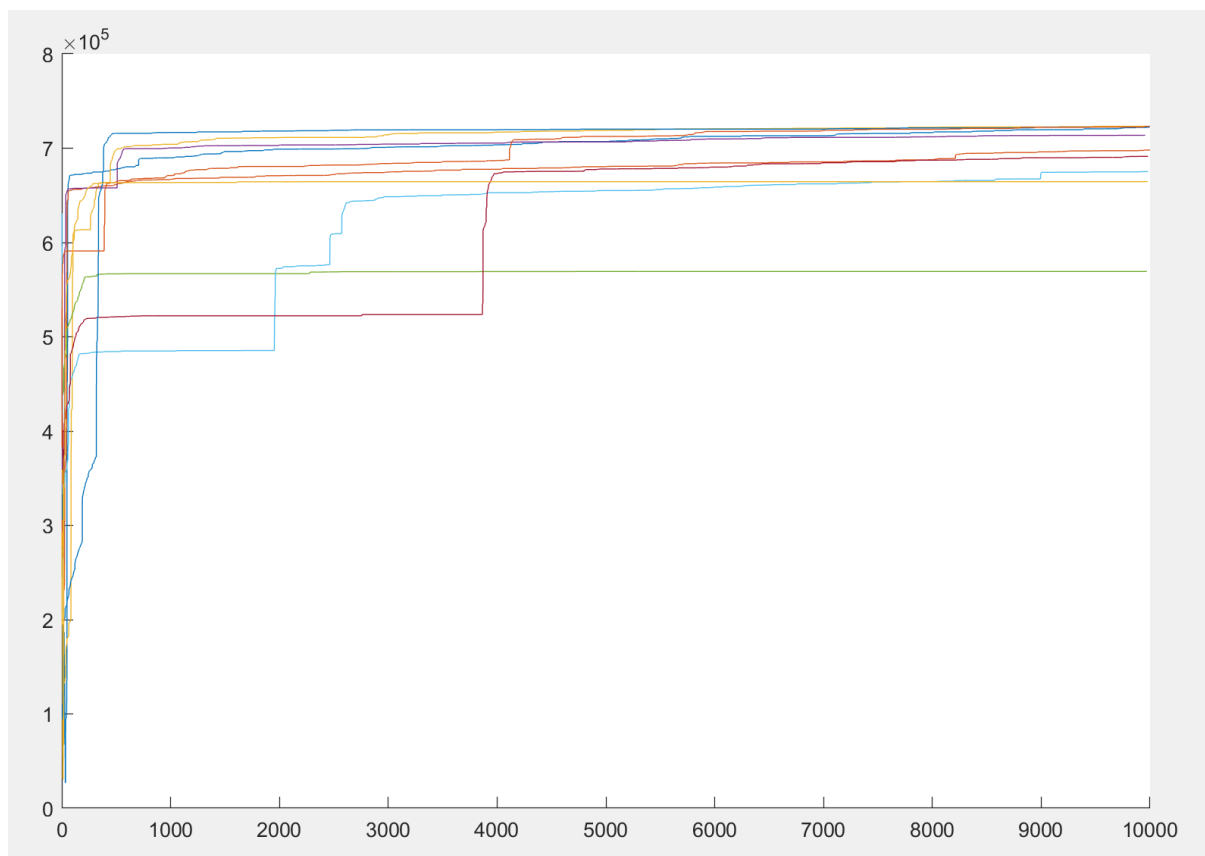


V tomto prípade používame „**death penalty**“, ktorá je nastavená na 100 000 000. Z daného grafu sú odstránené príliš veľké hodnoty, ktoré znehodnocovali graf. Môžeme vidieť že zisky sa väčšinou po pár sto pokusoch ustália a následne dochádza len k drobným zmenám. Priemerný výsledok z 10 behov bol **707 478** a **najlepší mal 724 230**. Bližší výsledok by sa dal dosiahnuť správnou zmenou tvorenia nových generácií.

	1	2	3	4	5
1	5.1657e+03	2.4807e+06	2.3298e+06	2.6696e+06	2.5148e+06

Jedno z najlepších riešení pre predstavu jedná sa o hodnoty

5165	2 480 700	2 398 000	2 669 600	2 514 800
------	-----------	-----------	-----------	-----------



1.0e+05 *

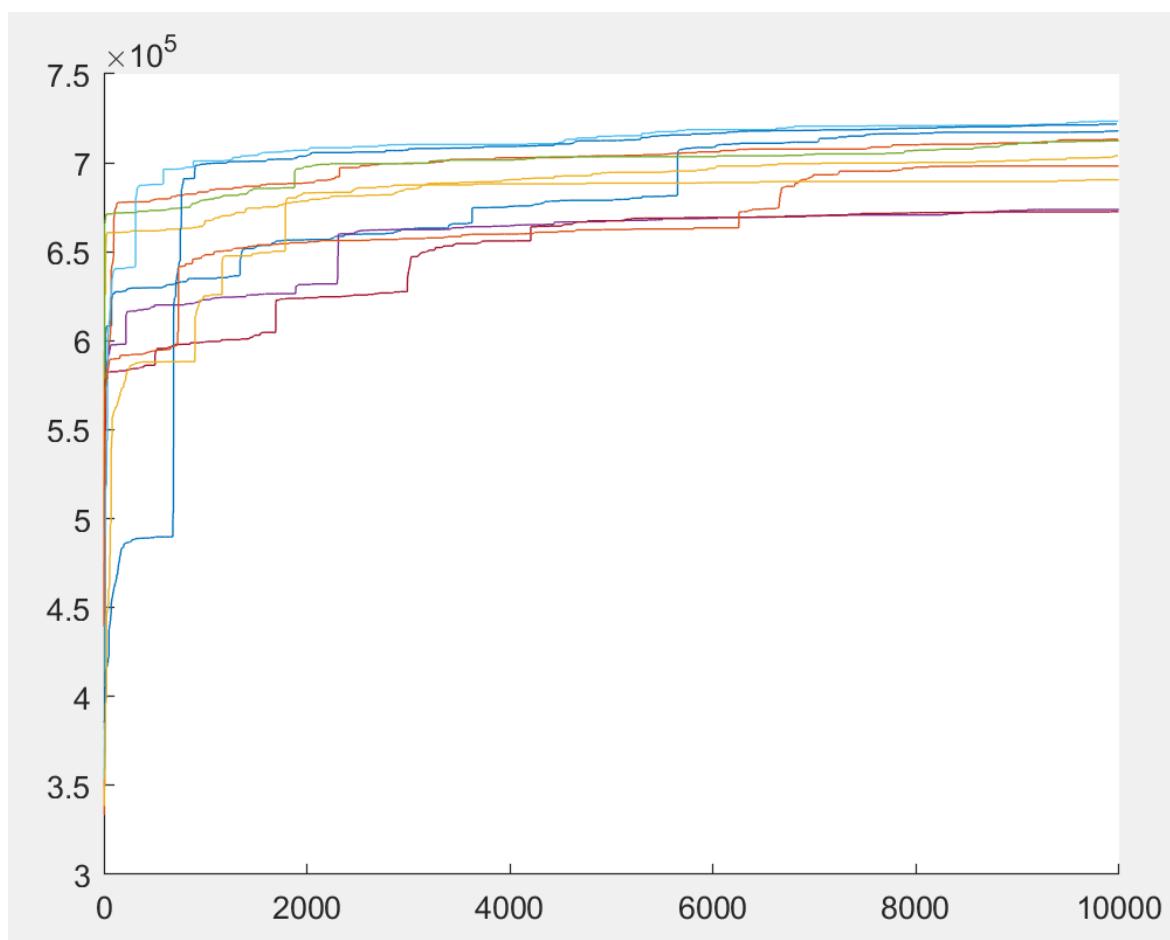
7.2227 6.9803 7.2322 7.1371 5.6922 6.7515 6.9133 7.2234 7.2328 6.6452

V tomto prípade používame „**stupňovu pokutu**“, ktorá je nastavená na $1\,000\,000^p$, kde p je počet porušení. Z daného grafu sú odstránené príliš veľké hodnoty, ktoré znehodnocovali graf. Môžeme vidieť že zisky sa väčšinou po pár sto pokusoch ustália a následne dochádza len k drobným zmenám. Priemerný výsledok z 10 behov bol **690 310** a **najlepší mal 723 280**.

	1	2	3	4	5
1	6.3534e+05	1.8647e+06	2.4999e+06	2.1236e+06	2.1235e+06

634 340	1 864 700	2 499 900	2 123 600	2 123 500
---------	-----------	-----------	-----------	-----------

V tomto prípade sa nejednalo práve o najlepší výsledok ale ukazuje nám to že náš algoritmus nefunguje práve ideálne. Na rozdiel od death penalty, došlo k ustáleniu neskôr aj priemerný zisk bol nižší.



V tomto prípade používame **pokutu podľa miery porušenia**, ktorá je nastavená na „rozdiel“². Z daného grafu sú odstránené príliš veľké hodnoty, ktoré znehodnocovali graf. Môžeme vidieť že zisky sa ustálili celkom neskoro a sú tam veľké skoky Priemerný výsledok z 10 behov bol **702 790** a **najlepší mal 723 420**.

	1	2	3	4	5
1	9.9776e+05	1.4413e+06	2.4340e+06	2.5636e+06	2.5633e+06

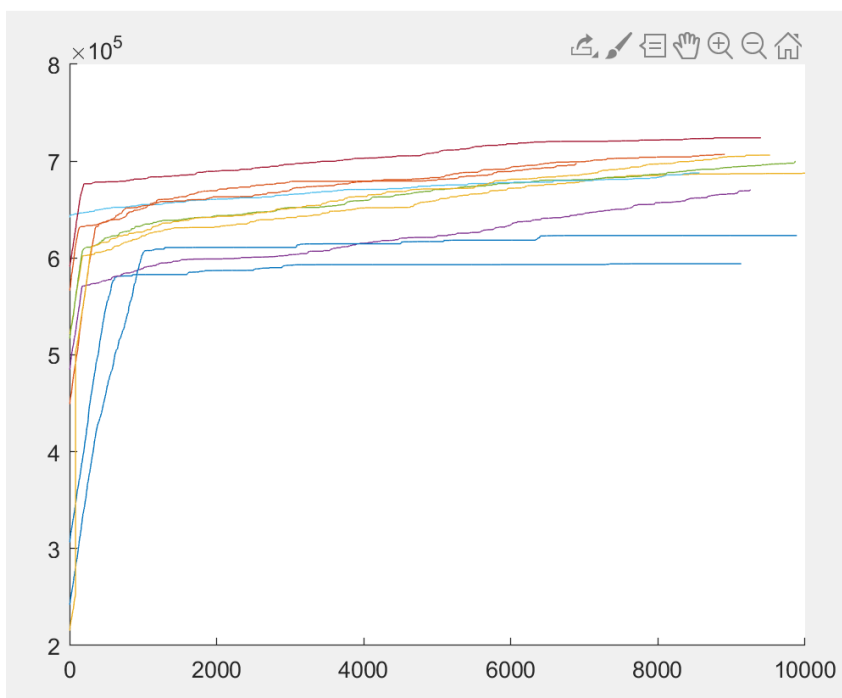
997 760	1 441 300	2 434 000	2 563 600	2 563 300
----------------	------------------	------------------	------------------	------------------

V tomto prípade je súčet prvých dvoch investícií približne 2 500 000 avšak náš program sa nedostal k ideálnej hodnote.

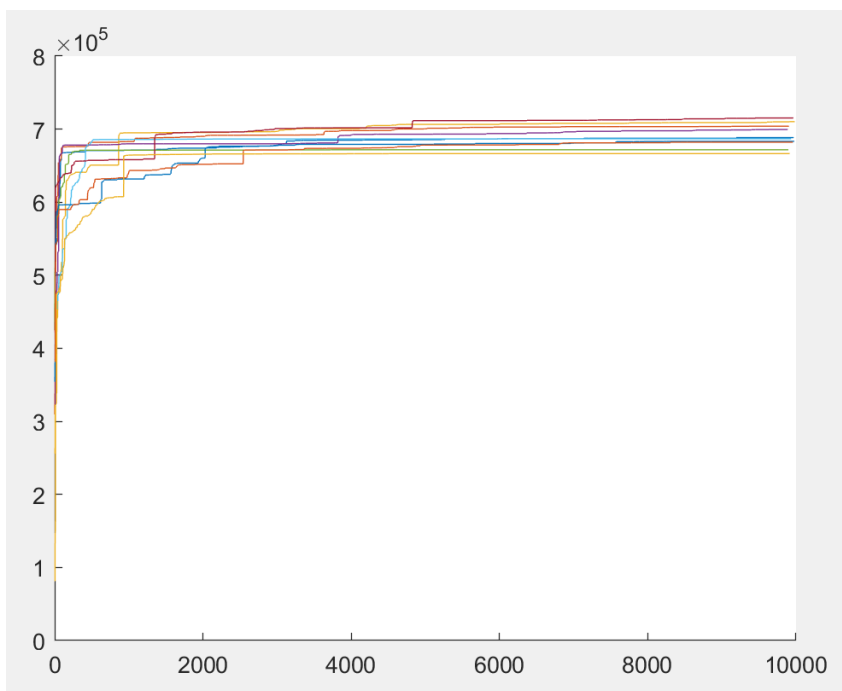
Na rozdiel od predošlých tipov penált, nie je kolísanie ziskov priemerné.

Modifikácia GA

Na jednoduchom príklade ukážeme následky zmeny v našom algoritme. Pre tento konkrétny príklad budeme meniť mieru početnosti mutovania v obyčajnej mutácii.



Miera početnosti 0



Miera početnosti 1

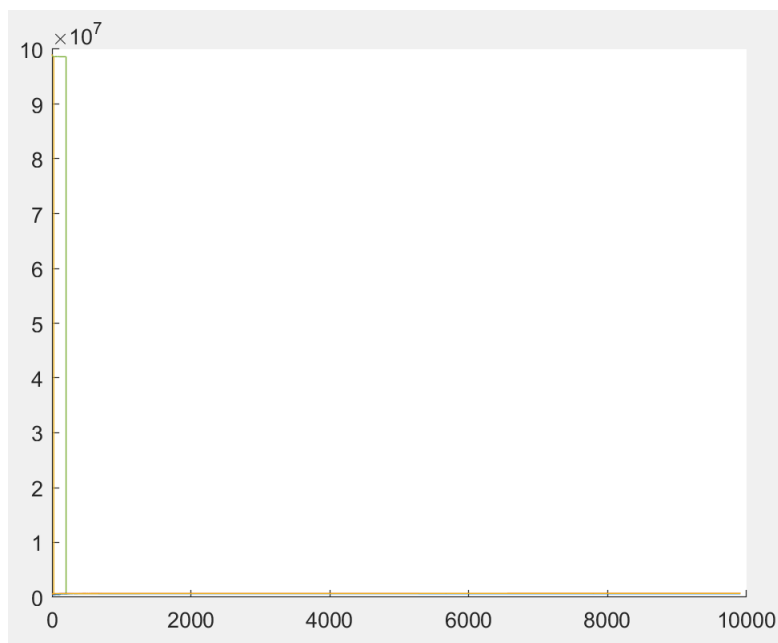
Môžeme vidieť že pri väčšej miere početnosti je omnoho viac „schodíkov“, ktoré sú prácou náhody, ktorá v grafe hore je 0. Zisky ktoré oba zistili nie sú ideálne ale lepšie je mať veľa náhody ako žiadnu, aspoň v tomto prípade. Ideálne výsledky boli niekde medzi 0.1 – 0.3.

Miera početnosti	0	0.1	0.3	0.7	1
1	5.9412	7.1113	6.9990	6.8745	6.9032
2	7.0702	7.1824	6.8831	7.1822	6.7752
3	6.8757	7.0222	7.2445	6.9356	6.8010
4	6.7064	7.0121	7.0971	7.0375	6.8452
5	6.9965	6.7521	6.9243	6.9113	7.1565
6	6.8783	6.9870	6.5755	7.0355	7.2131
7	7.2412	7.2277	7.0544	6.8828	7.0471
8	6.2325	7.2407	7.0341	6.6896	6.9385
9	6.9759	6.9037	7.0488	7.0158	7.1231
10	7.0619	7.2258	7.2296	6.9315	6.9076

Jedná sa o stovky tisíc

Záver

Naším cieľom bolo porovnať tri rôzne typy pokutovania. Najlepší a najstabilnejší sa v tomto prípade ukázal prípad, kde sa používa „death penalty“. Väčšinou sa tam bohužiaľ objavila aj možnosť, kde aspoň jeden získal pokutu aj ten najlepší a v tom prípade bol celý graf bezcenný, na záver jeden priložím. Pre jednoduchosť a hlavne dobrú viditeľnosť postupného zlepšovania boli veľkosti hodnôt v grafe obmedzené. Presnosť nášho algoritmu by sa dala zlepšiť, mohol by byť viac efektívny. Genetický algoritmus je v podstate riadený výber najlepších a ako platí v prírode tak platí aj tu, vždy sa dá byť lepší.



Graf z ktorého toho moc nevidno, len jeden veľký skok