

**UMINT Doplnkový projekt PGA Dokumentácia**

**Ladislav Rajcsányi**

## Obsah

Úloha .....	3
Genetický algoritmus (GA) .....	3
Funkcie ktoré som si vytvoril pre prácu s PGA .....	7
fitness_funkcia.....	7
GA_lower .....	7
GA_higher .....	7
GA_top .....	8
best_individual .....	8
GA .....	8
selworst.....	8
migration.....	8
migration_between_islands.....	8
Paralelný genetický algoritmus (PGA) .....	9

## Úloha

Mali sme riešiť úlohu z 2. prednášky( strana 55-57) pomocou GA a PGA . Pri riešení sme mali použiť všetky tri metódy pokutovania.

$$f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

$$127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \geq 0$$

$$282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \geq 0$$

$$196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \geq 0$$

$$-4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7 \geq 0$$

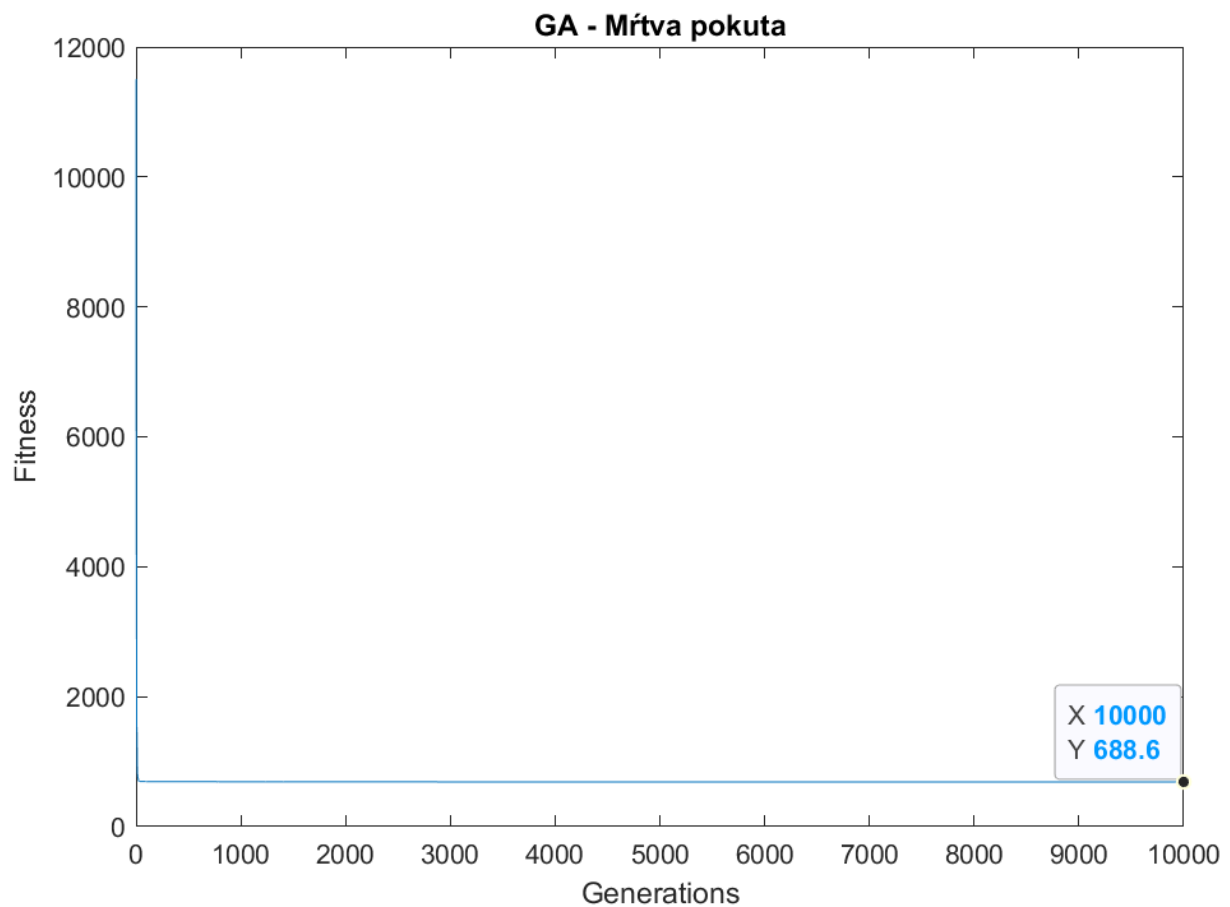
$$-10 \leq x_i \leq 10, i = 1, 2, \dots, 7$$

Riešenia sú priložené vo forme spustiteľných programov v Matlabe, v priečinku Figures sa nachádzajú grafy, ktoré som vytvoril pomocou programu. V priečinku Values sa nachádzajú výsledky jednotlivých spustení.

## Genetický algoritmus (GA)

Najprv podľa zadania som si vytvoril genetický algoritmus. V účelovej funkcii ( fitness funkcia) som si implementoval vyššie uvedený vzorec a jednotlivé metódy pokutovania. Vybral som si najlepších jedincov pomocou **selbest**, ktorých som odložil do matice, aby som ich náhodou nemodifikoval. Prvého najlepšieho som zobral 20-krát, druhého 15-krát a tretieho 10-krát. Potreboval som aj najlepšieho jedinca aby som mohol vykresliť graf priebehu evolúcie („fittrend“). Aby som mal náhodnosť pri výbere jedincov, som si zvolil výber pomocou váhovaného ruletového kolesa (**selsus**). Tento výber funguje ako otáčanie váhovaným ruletovým kolesom. Koleso je rozdelené na kruhové výseky ktoré budú reprezentovať jedince, a budú mať veľkosť na základe hodnoty účelovej funkcie. Čím je jedinec úspešnejší, tým bude mať väčšiu šancu byť vybraný. Po vybraní 55 jedincov som ich krížil pomocou **crossov**, ktorého som nastavil aby mal len 1 bod kríženia, a dvojice, ktorých budeme krížiť nám vyberal náhodne. Po krížení nastala 10%-ná globálna mutácia (**mutx**) a po nej 10%-ná lokálna mutácia (**muta**), kde vektor ohraničení prípustných aditívnych hodnôt mutácií (**Amp**) boli samé jednotky ([1 1 1 1 1 1 1]). Tento GA mal 10000 generácií po ktorom som dostal tieto výsledky:

Prvá metóda pokutovania, ktorú som si implementoval bola **metóda mŕtvej pokuty**:



Figures : GA\_Mrtva\_pokuta

Mŕtva pokuta

2.2601    2.1632    -0.5802    3.7244    -0.7623    1.0730    1.4985

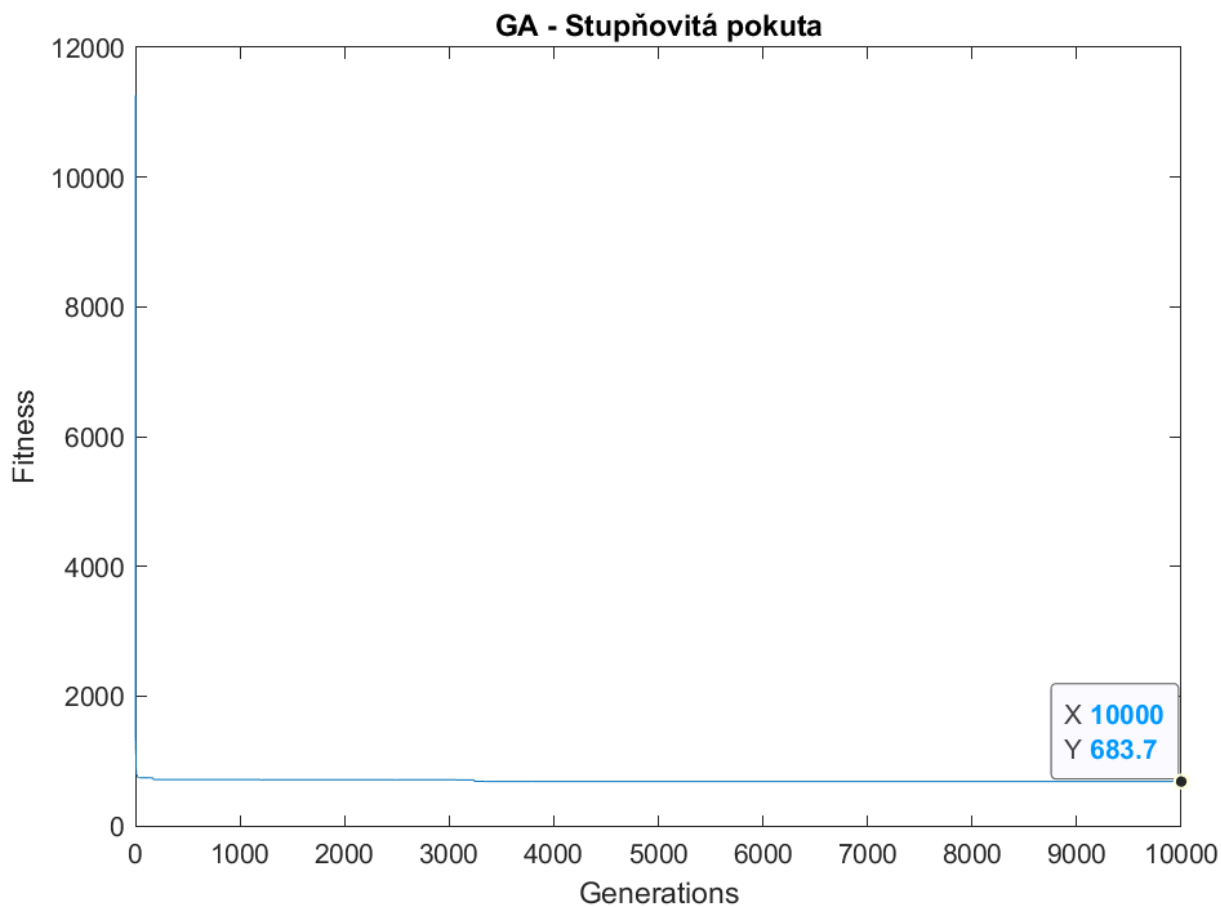
Funkčná hodnota najlepšieho jedinca

688.5510

Values : GA\_Mrtva\_pokuta

V prvom riadku som si zobrazil gény najlepšieho jedinca pri používaní mŕtvej pokuty. V druhom riadku som si zobrazil funkčnú hodnotu toho jedinca. Potom pri používaní PGA uvidíme, že táto hodnota, je príliš ďaleko od globálneho minima.

Druhá metóda bola **stupňovitá pokuta**:



Figures : GA\_Stupnovita\_pokuta

Stupňovitá pokuta

2.2344    2.0541    -0.0007    4.0505    -0.4028    1.1323    1.4631

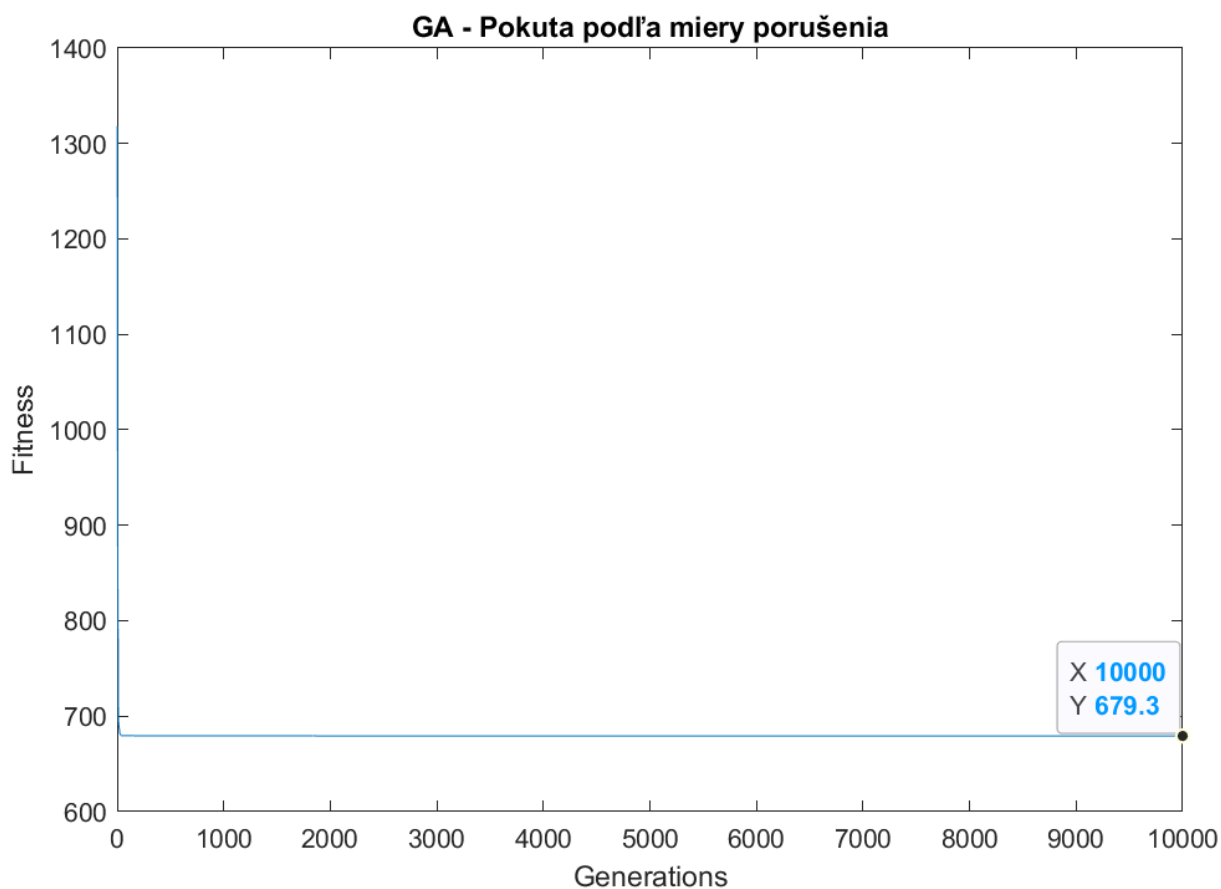
Funkčná hodnota najlepšieho jedinca

683.7462

Values : GA\_Stupnovita\_pokuta

Táto funkčná hodnota je síce lepšia ako v prípade mŕtvej pokuty, ale stále sme nenašli najoptimálnejšie riešenie teda globálny minimum.

Naposledy som si implementoval **pokutovanie podľa miery porušenia obmedzení**:



Figures : GA\_Pokuta\_podla\_miery\_porusenien

Pokuta podľa miery porušenia

2.3723    2.0348    -0.4331    4.7142    -0.6084    0.9415    1.5685

Funkčná hodnota najlepšieho jedinca

679.2755

Values : GA\_Pokuta\_podla\_miery\_porusenien

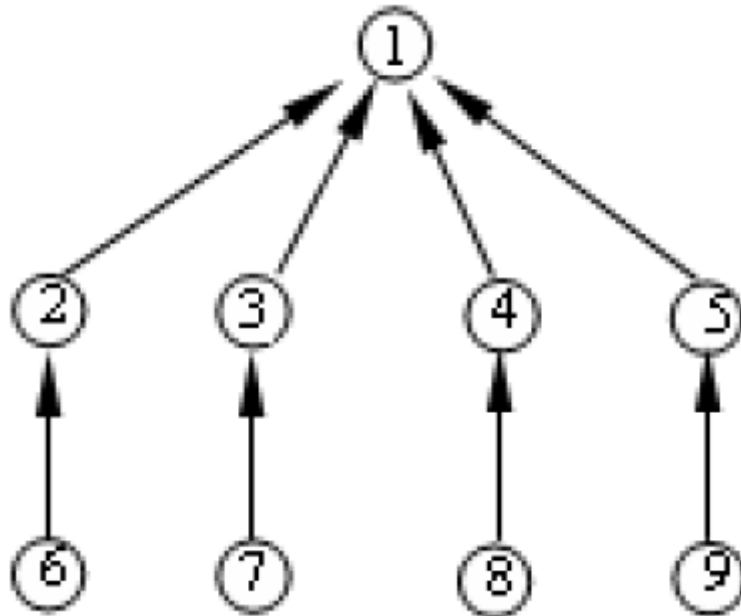
Pomocou GA táto funkčná hodnota bola najmenšia v porovnaní s ostatnými metódami ( táto metóda pokutovania generovala najlepšie výsledky).

## Funkcie ktoré som si vytvoril pre prácu s PGA

**[Fit]=fitness\_funkcia(Pop)**

Túto funkciu sme už používali pri GA, obsahuje vzorec, ktorú sme používali na výpočet funkčnej hodnoty a jednotlivé metódy pokutovania.

Pri riešení PGA som si vybral 3 vrstvovú topológiu, pre každú som vytvoril zvláštny genetický algoritmus:



Pre vrstvu 6, 7, 8, 9:

**GA\_lower(Island,Space,graph,j,i)**

Kde **Island** bude reprezentovať jednotlivé ostrovy z tejto vrstvy, **selbest** som nastavil na nízke hodnoty pre nižší selektívny tlak a preto aby som mal dosť vysokú diverzitu, ktorá nám bude produkovať nové smery hľadania, som si použil náhodný výber (**selrand**).

**graph** bude reprezentovať maticu, do ktorého vložíme funkčnú hodnotu najlepšieho jedinca v danej generácii.

**j** bude reprezentovať číslo grafu, **i** bude číslo aktuálnej generácie.

Pre vrstvu 2, 3, 4, 5:

**GA\_higher(Island,Space,graph,j,i)**

Pre zvýšenie selektívneho tlaku, som tu zadal väčšie hodnoty v **selbest**. Aby sme už začali vyhľadávať lepšie riešenia, ale napriek tomu sme mali trochu diverzity som si použil výber pomocou váhovaného ruletového kola (**selsus**).

Pre vrstvu 1 :

**GA\_top(Island,Space,graph,j,i)**

Tu som už hľadal najlepších jedincov, teda tu som mal najväčší selektívny tlak, a aby som mal čo najnižšiu diverzitu som si zvolil turnajový výber (**seltourn**).

Tieto funkcie nám vrátia v každej generácii modifikovaný ostrov a najlepšieho jedinca z daného ostrova.

**best\_individual(best\_one1,best\_one2,best\_one3,best\_one4,best\_one5,best\_one6,best\_one7,best\_one8,best\_one9,graph,j,i)**

Táto funkcia berie tých najlepších jedincov a zistí, ktorý je z nich úplne najlepší.

**GA(Island1,Island2,Island3,Island4,Island5,Island6,Island7,Island8,Island9,Space,i,graph)**

Táto funkcia v každej generácii spustí genetické algoritmy (**GA\_lower**, **GA\_higher**, **GA\_top**) pre jednotlivé vrstvy a funkciu na vyhľadávanie najlepšieho jedinca (**best\_individual**)

**[index\_of\_worst\_individual] = selworst(Fvpop)**

Táto funkcia nám na základe účelovej funkcie (fitness funkcie) vyberie index najhoršieho jedinca, teda ktorý má najväčšiu funkčnú hodnotu.

**[destination] = migration(destination,replacement)**

Táto funkcia nám umožní migráciu. Pomocou **selworst** zistíme, na cieľovom ostrove (**destination**), ktorý je najslabší jedinec, a keď už máme jeho index nahradíme ho s jedincom **replacement**.

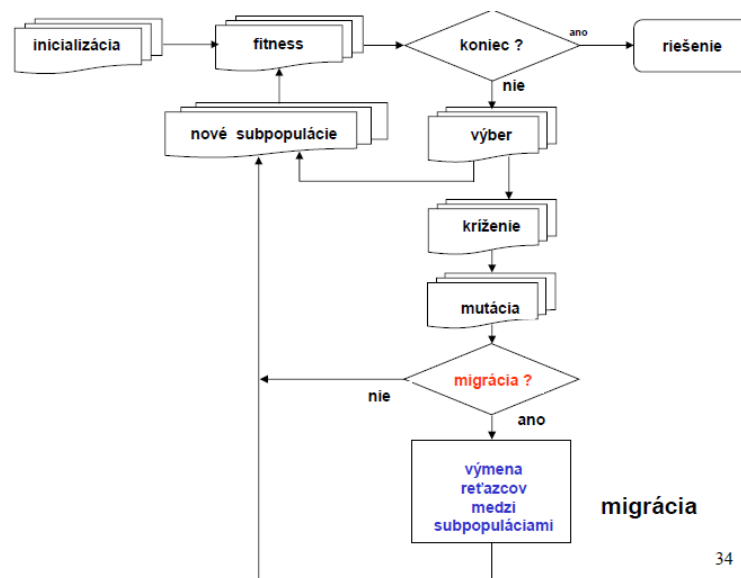
**migration\_between\_islands(Island1,Island2,Island3,Island4,Island5,Island6,Island7,Island8,Island9)**

Pomocou tejto funkcie som si realizoval na základe topológie migrácie medzi jednotlivými ostrovmi.



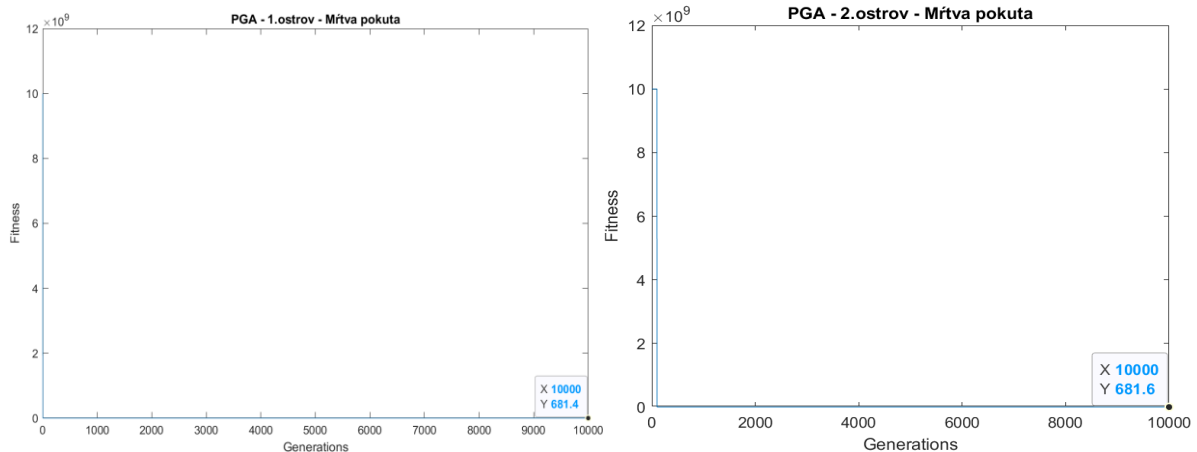
## Paralelný genetický algoritmus (PGA)

Po vygenerovaní 900 jedincov som ich rozdelil do 9 skupín , ktorých som nazval **Island** (Ostrov). Podľa obrázku z 2. prednášky -

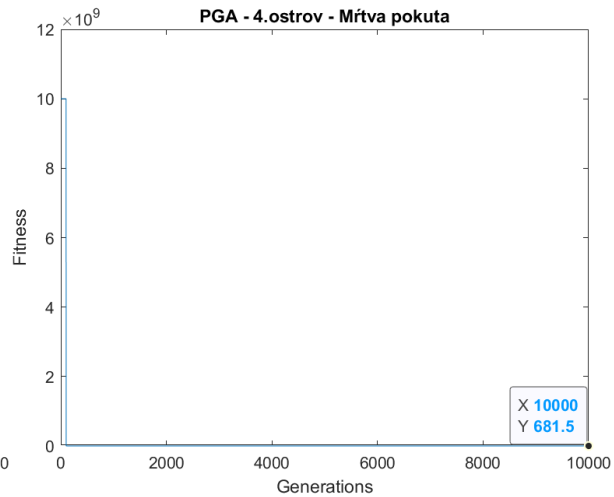
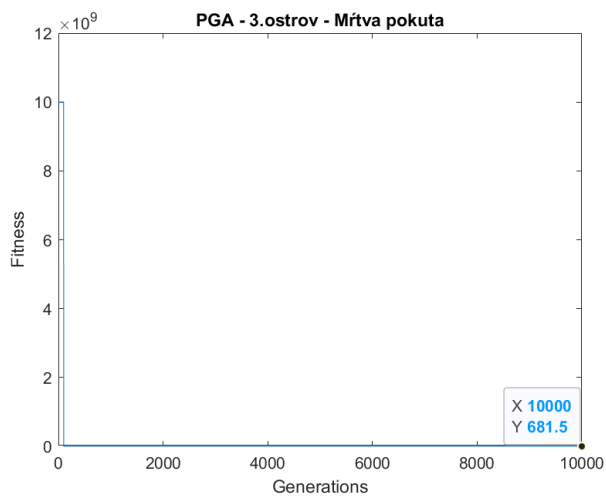


34

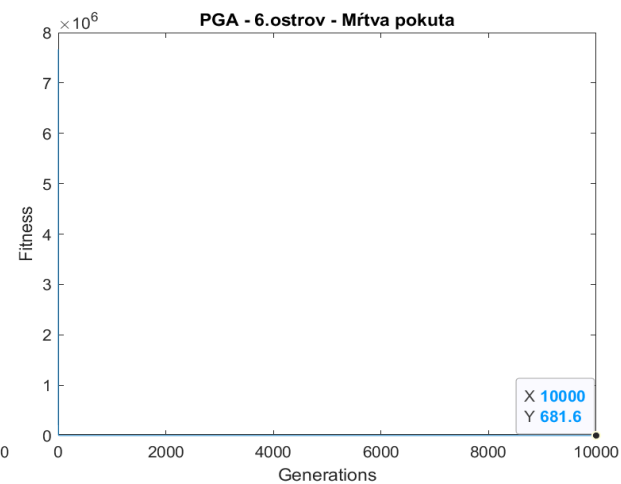
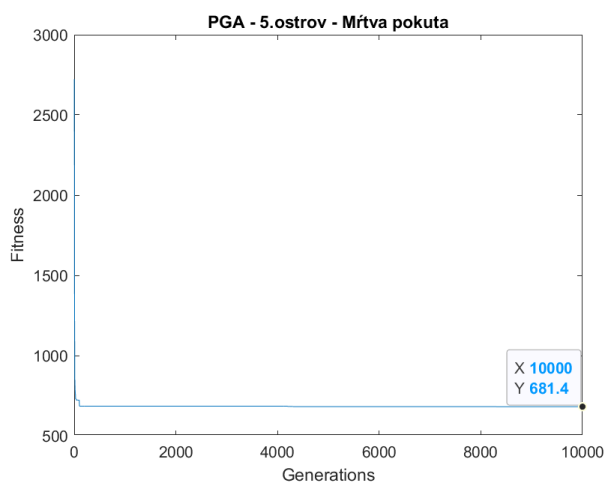
som si v každom cykle najprv spustil funkciu **GA** , ktorá nám umožní genetickú modifikáciu jednotlivých ostrovov, na základe vrstvy v topológii. V každej 100. generácii sa uskutočnila migrácia medzi ostrovmi pomocou funkcie **migration\_between\_islands**. Takisto a v prípade GA aj tu sme mali otestovať 3 metódy pokutovania.



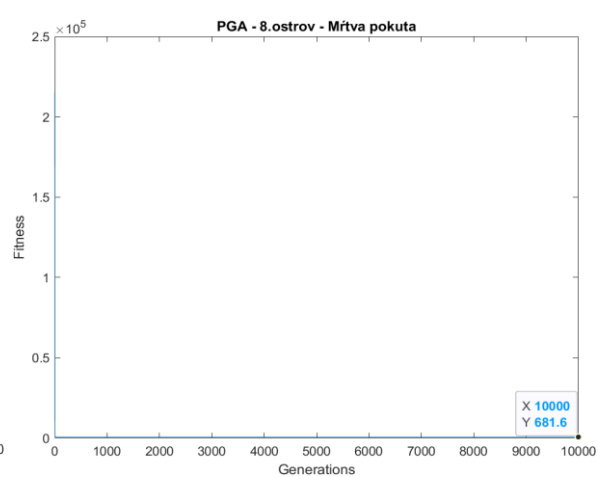
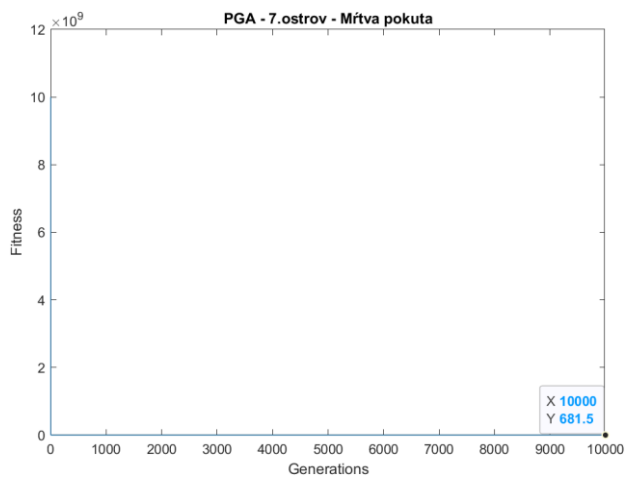
Figures : PGA\_1\_ostrov\_Mrtva\_pokuta a PGA\_2\_ostrov\_Mrtva\_pokuta



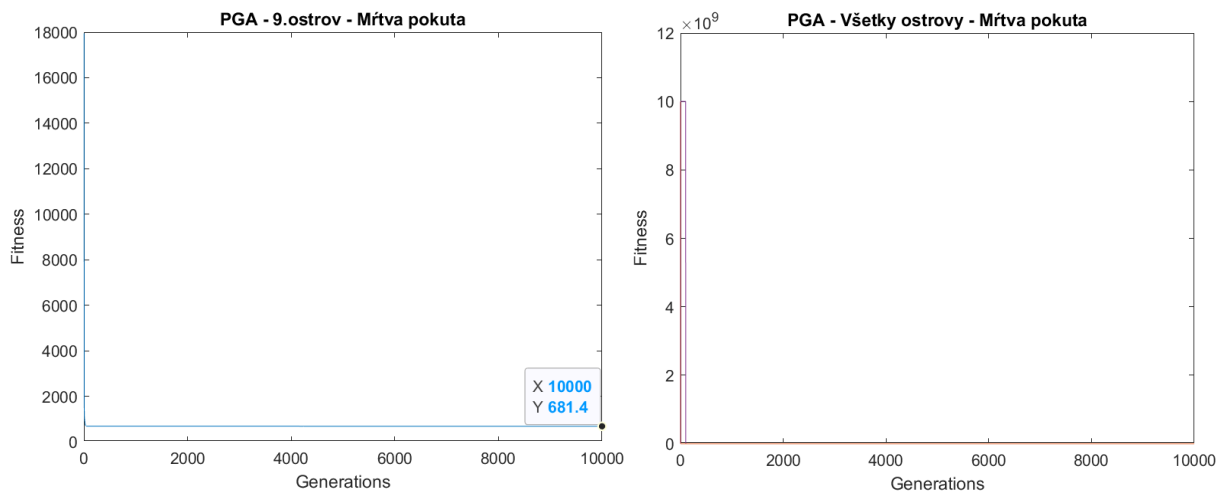
Figures : PGA\_3\_ostrov\_Mrtva\_pokuta a PGA\_4\_ostrov\_Mrtva\_pokuta



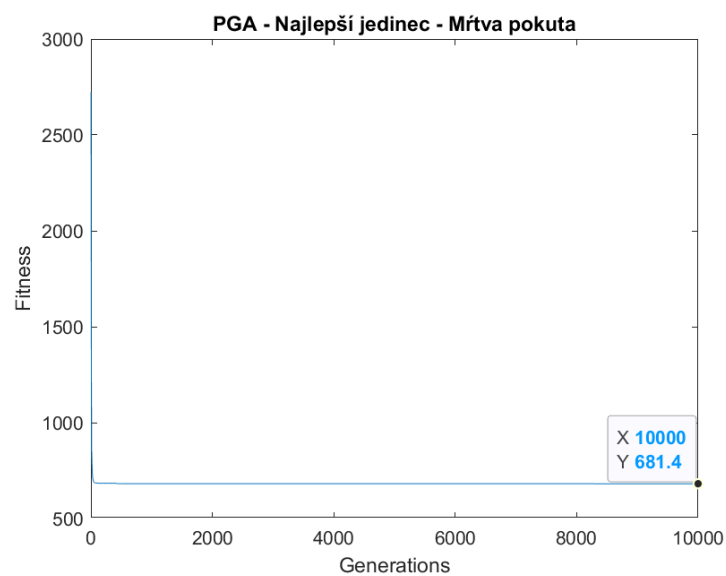
Figures : PGA\_5\_ostrov\_Mrtva\_pokuta a PGA\_6\_ostrov\_Mrtva\_pokuta



Figures : PGA\_7\_ostrov\_Mrtva\_pokuta a PGA\_8\_ostrov\_Mrtva\_pokuta



Figures : PGA\_9\_ostrov\_Mrtva\_pokuta a PGA\_Vsetky\_ostrovy\_Mrtva\_pokuta



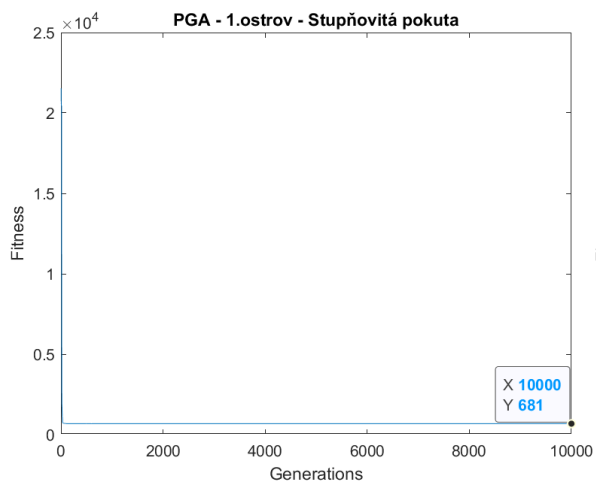
Figures : PGA\_Najlepsi\_jedinec\_Mrtva\_pokuta

PGA - Mŕtva pokuta - Gény najlepšieho jedinca  
 2.1901    1.9812    -0.3894    4.3092    -0.5403    1.1323    1.4632

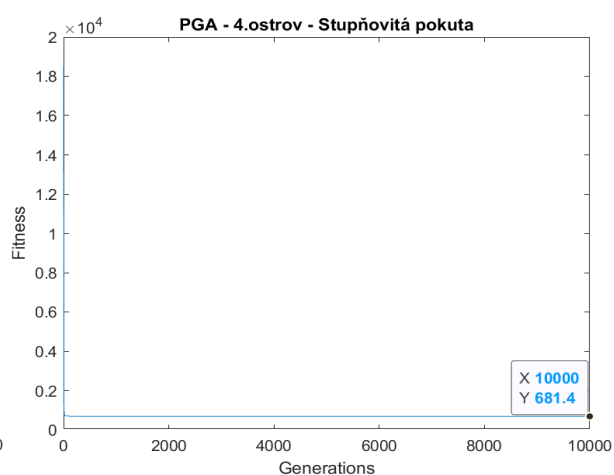
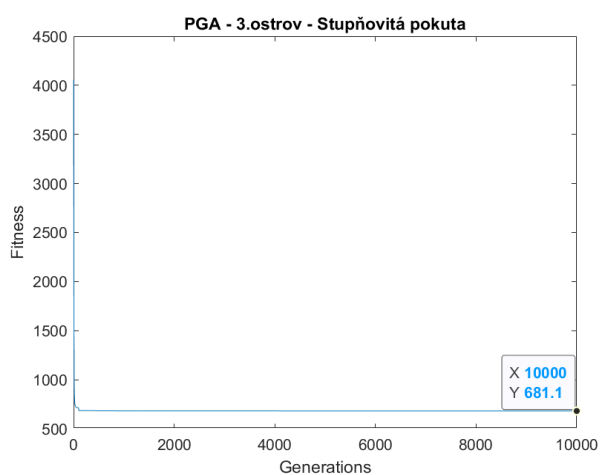
Funkčná hodnota najlepšieho jedinca  
 681.3526

Values : PGA\_Mrtva\_pokuta

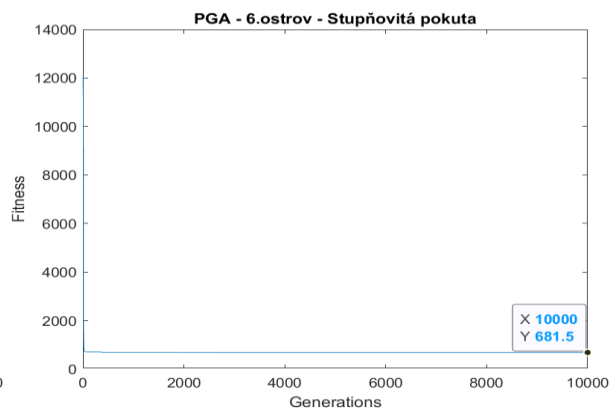
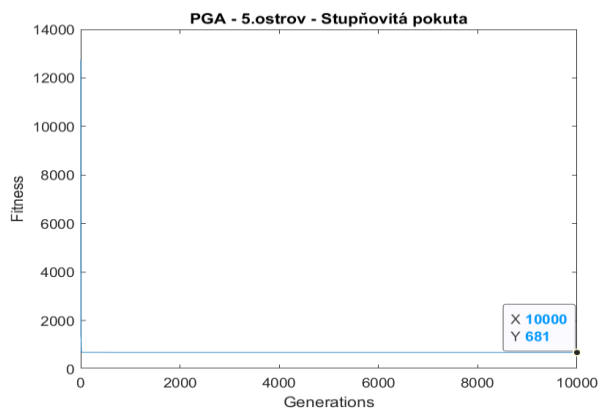
Pri používaní mŕtvej pokuty sme dosiahli tieto výsledky, z ktorých najlepšia bola 681.4.  
 Pomoco tejto metódy sme nedostali k optimálnemu riešeniu.



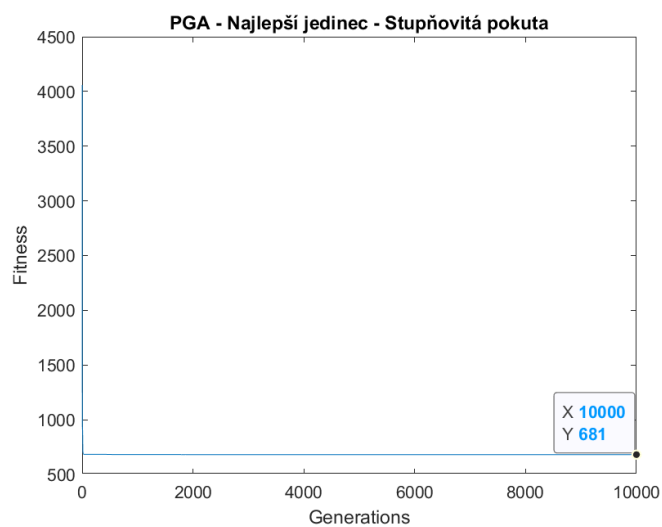
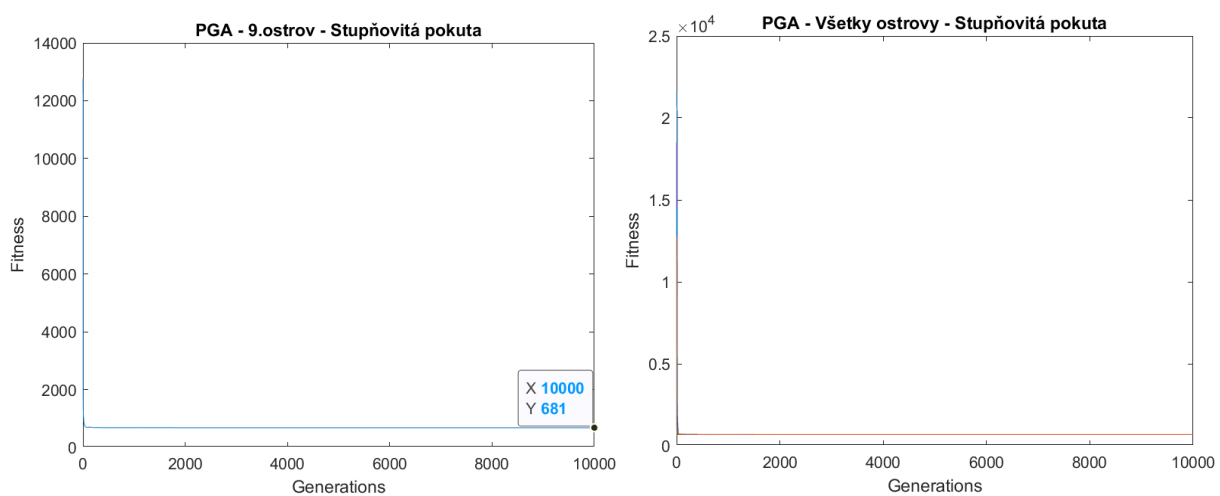
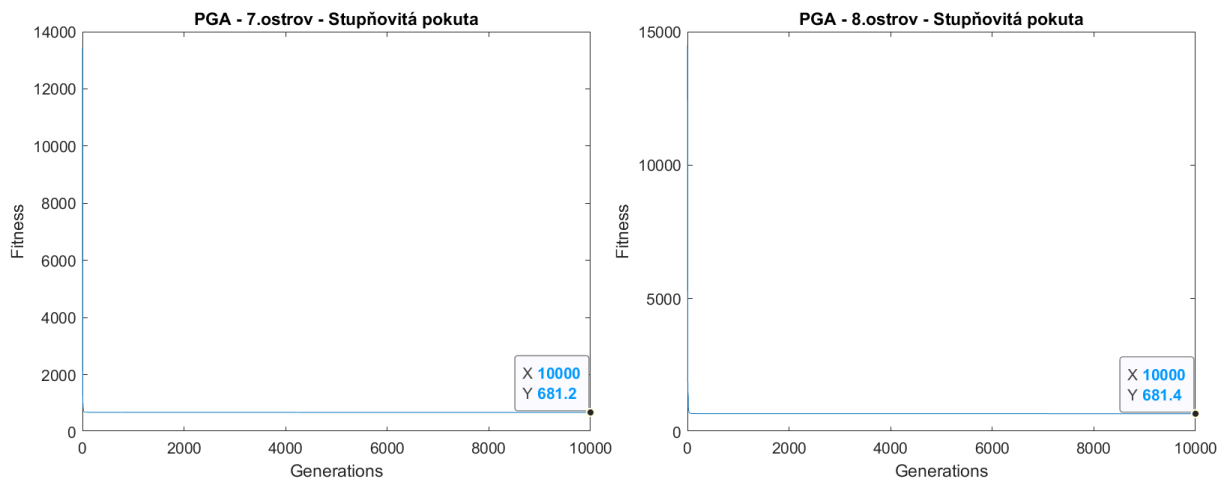
Figures : PGA\_1\_ostrov\_Stupnovita\_pokuta a PGA\_2\_ostrov\_Stupnovita\_pokuta



Figures : PGA\_3\_ostrov\_Stupnovita\_pokuta a PGA\_4\_ostrov\_Stupnovita\_pokuta



Figures : PGA\_5\_ostrov\_Stupnovita\_pokuta a PGA\_6\_ostrov\_Stupnovita\_pokuta



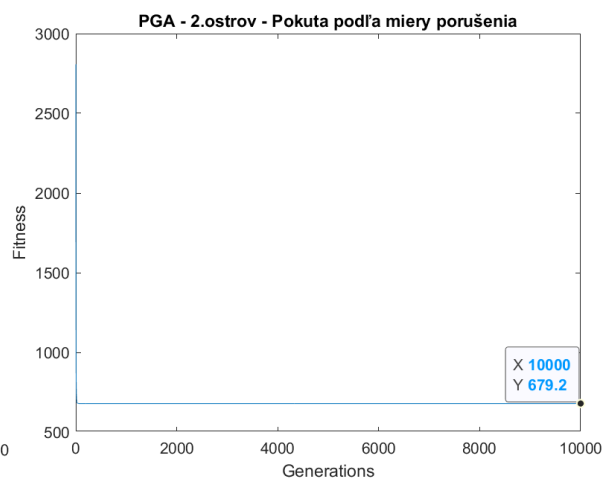
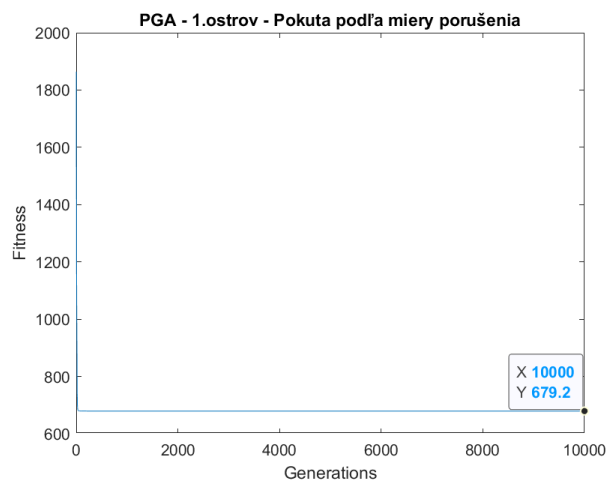
Figures : PGA\_Najlepsi\_jedinec\_Stupnovita\_pokuta

PGA - Stupňovitá pokuta - Gény nejlepšího jedince  
 2.2280 1.9652 -0.3518 4.3508 -0.6077 1.1365 1.5012

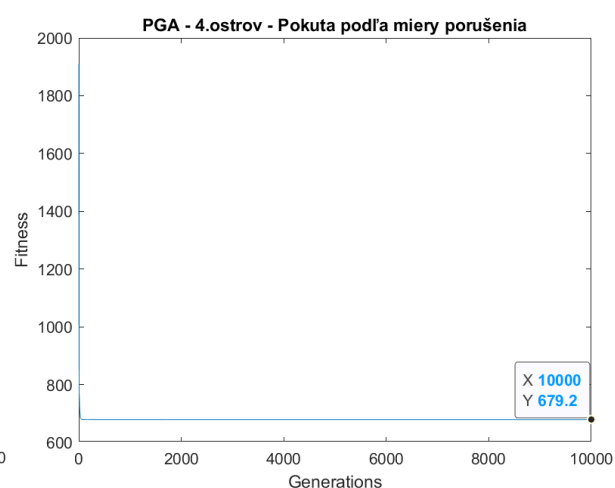
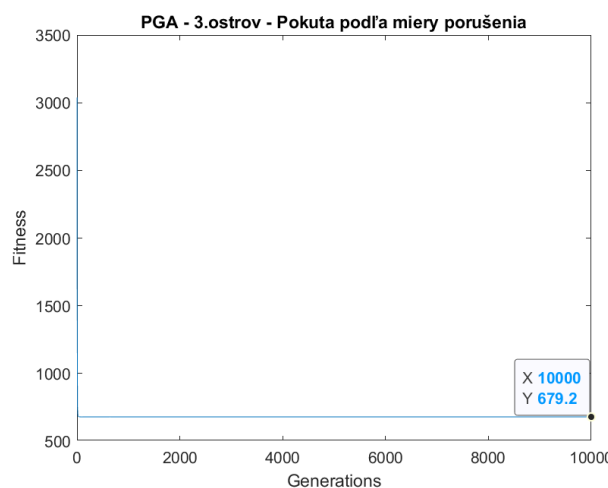
Funkčná hodnota nejlepšího jedince  
 680.9668

Values : PGA\_Stupnovita\_pokuta

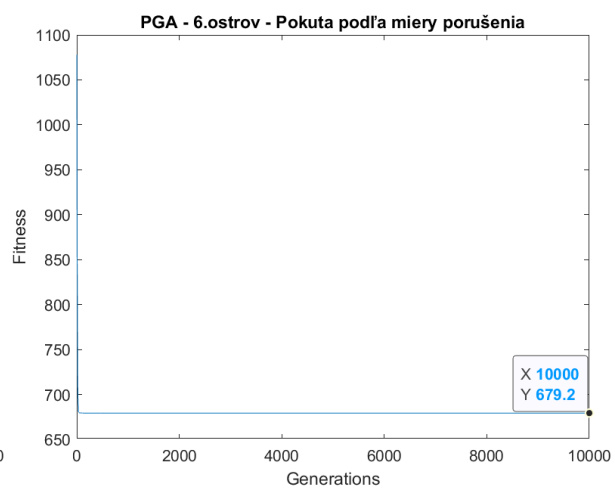
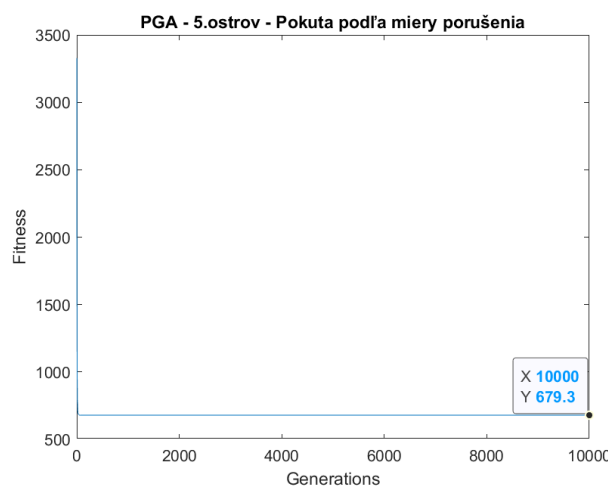
Síce pri používaní stupňovitej pokuty sme dosiahly lepšiu hodnotu (680,9) ako pri používaní mŕtvej pokuty (681,4), ale napriek tomu nedostali sme moc bližšie k optimálnemu riešeniu.



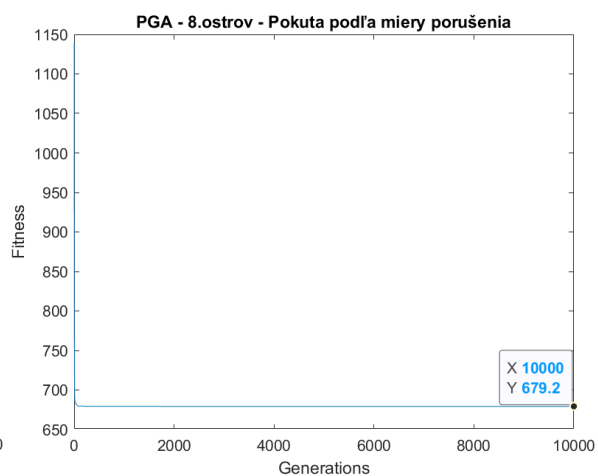
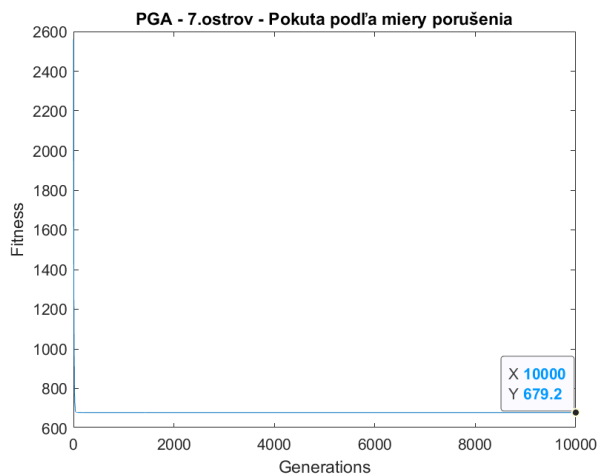
Figures : PGA\_1\_ostrov\_Pokuta\_podla\_miery\_porusenien a PGA\_2\_ostrov\_Pokuta\_podla\_miery\_porusenien



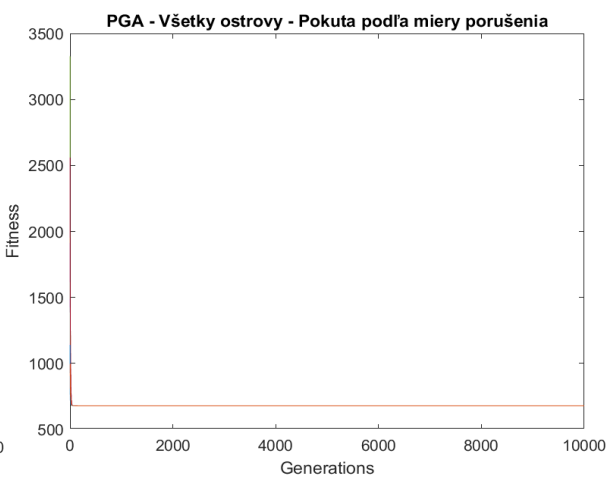
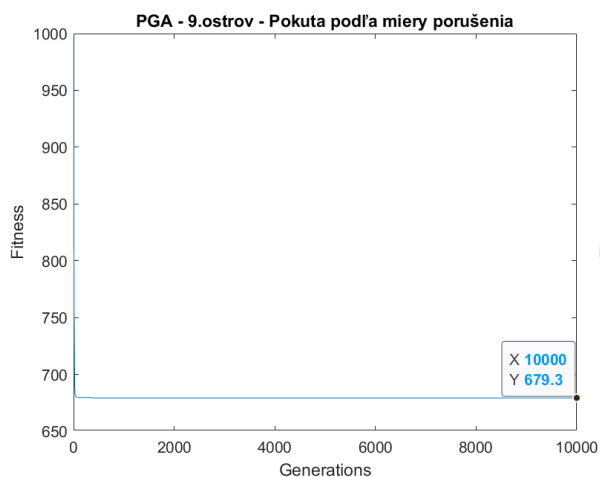
Figures : PGA\_3\_ostrov\_Pokuta\_podla\_miery\_porusenien a PGA\_4\_ostrov\_Pokuta\_podla\_miery\_porusenien



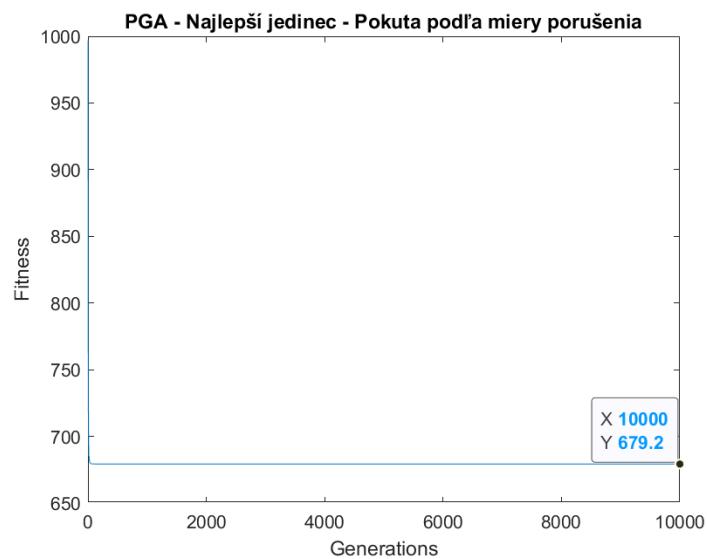
Figures : PGA\_5\_ostrov\_Pokuta\_podla\_miery\_porusenien a PGA\_6\_ostrov\_Pokuta\_podla\_miery\_porusenien



Figures : PGA\_7\_ostrov\_Pokuta\_podla\_miery\_porusenia a PGA\_8\_ostrov\_Pokuta\_podla\_miery\_porusenia



Figures : PGA\_9\_ostrov\_Pokuta\_podla\_miery\_porusenia a PGA\_Vsetky\_ostrovy\_Pokuta\_podla\_miery\_porusenia



Figures : PGA\_Najlepsi\_jedinec\_Pokuta\_podla\_miery\_porusenia

```
PGA - Pokuta podľa miery porušenia - Gény najlepšieho jedinca
2.3741 2.0358 -0.4052 4.7143 -0.6083 0.9787 1.5830

Funkčná hodnota najlepšieho jedinca
679.2351
```

Values : PGA\_Pokuta\_podla\_miery\_porusenja

Metóda pokutovania podľa miery porušenia obmedzení nám umožnila dostať sa čo najbližšie (v našom prípade 679,23) k optimálnemu riešeniu, teda ku globálnemu minimu.