

Umelá inteligencia

1. zadanie - bonusová úloha

Dávid Gavenda

Andrej Jackulík

Zadanie – úvod

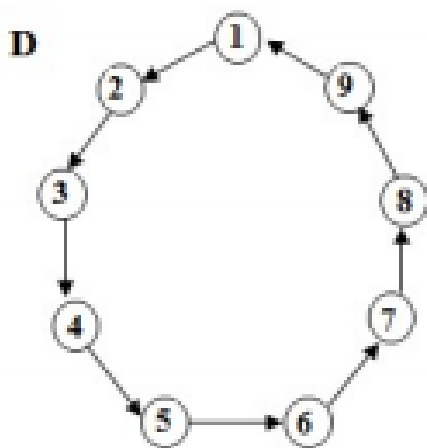
Našou úlohou bolo stručne porovnať hľadanie minima EggHolder funkcie pomocou genetického algoritmu a paralelného genetického algoritmu. Hľadali sme minimum funkcie Eggholder s 5 premennými (rozsah od -500 do 500).

Genetický algoritmus

V tomto prípade sme si vytvorili 5 subpopulácií. Pri každej z nich sme použili fitness funkciu (EggHolder), následne zmutovali a krížili

```
fit4 = eggholder(oldPop4);  
bestSel4 = selbest(oldPop4,fit4,[4 3 2]);  
theBest4 = selbest(oldPop4,fit4,1);  
newPop4 = seltourn(oldPop4,fit4,30);  
newPop4 = crossover(newPop4, 2, 1);  
mixPop4 = genrpop(10,space);  
newPop4 = mutx(newPop4, 0.2, space);  
newPop4 = muta(newPop4,0.1,amp,space);  
oldPop4 = [theBest4;bestSel4;newPop4;mixPop4];
```

Po každých 50 opakovaniach sme uskutočnili migráciu podľa nasledovného postupu.

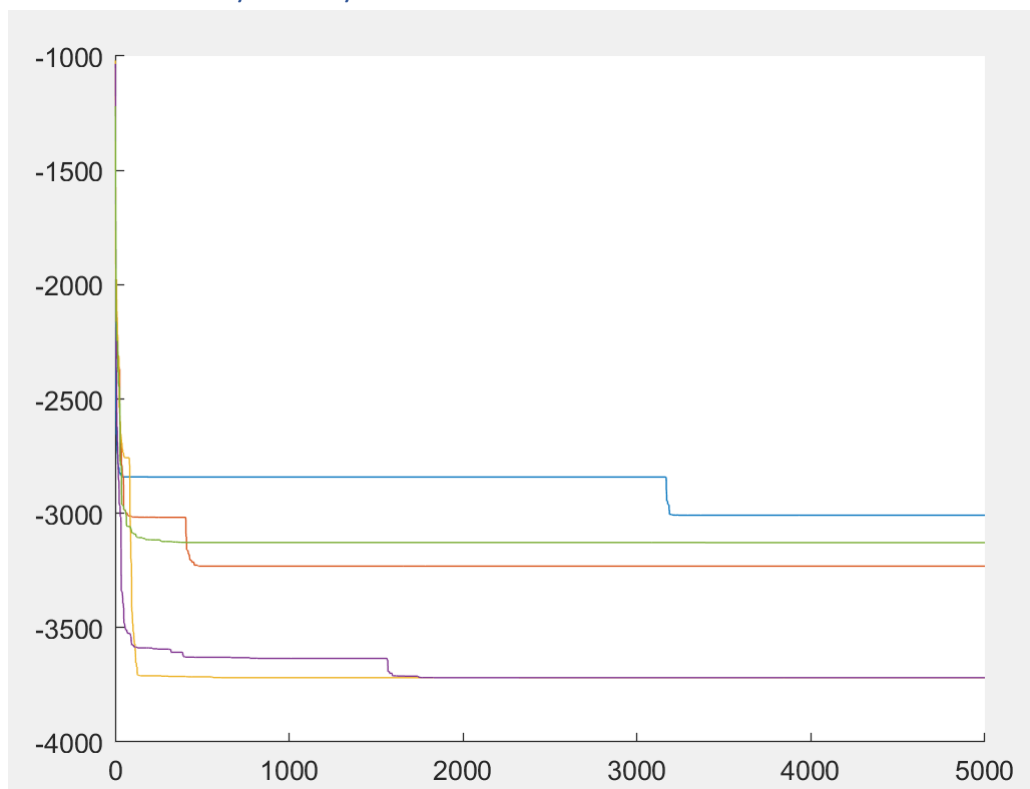


Migrovanie prebiehalo skopírovaním najlepšieho jedinca subpopulácie a jeho skopírovanie na náhodne miesto v nasledujúcej.

```
if mod(i,50)==0  
    oldPop1((randi(48)+2),:)=theBest5;  
    oldPop2((randi(48)+2),:)=theBest1;  
    oldPop3((randi(48)+2),:)=theBest2;  
    oldPop4((randi(48)+2),:)=theBest3;  
    oldPop5((randi(48)+2),:)=theBest4;  
end
```

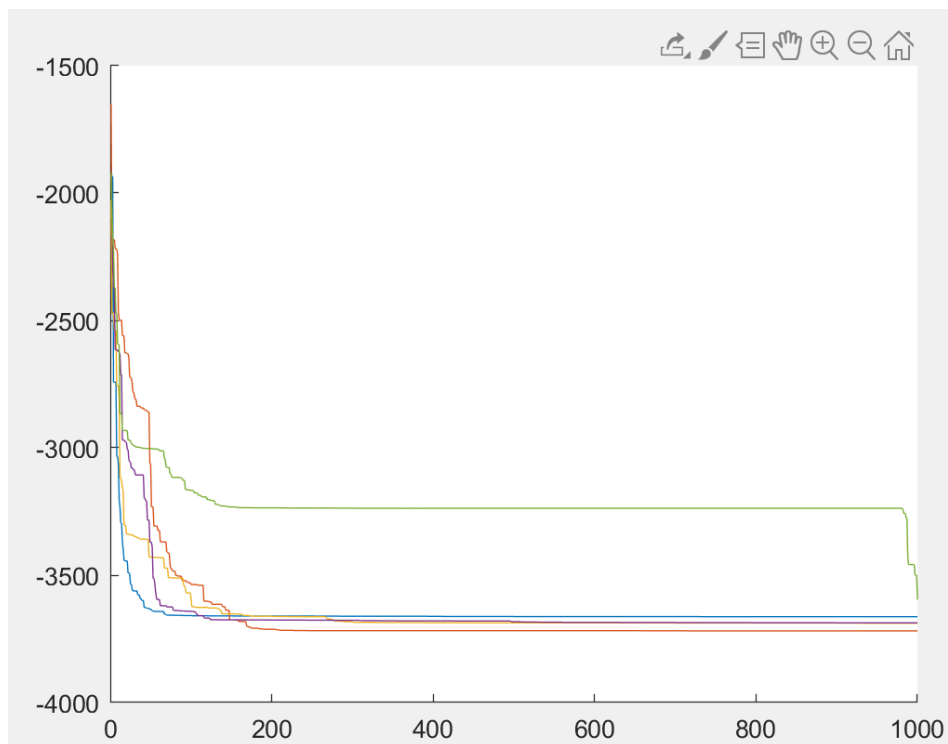
Na konci sa vybral najlepší prvok zo všetkých subpopulácií a ten sa vypísal.

Dosiahnuté výsledky



5000 generácií genetického algoritmu

Priemer z piatich behov bol -336 120



1000 generácií paralelného genetického algoritmu

Priemer z piatich behov bol 367 080

Môžeme vidieť ze paralelný genetický algoritmus bol o dost presnejší a menej stagnoval. Počet generácii sa líšil 5-násobne pretože PGA používal 5 subpopulácii.

Záver

Naším cieľom bolo porovnať GA a PGA. Lepšie na tom vyšiel PGA, ktorý bol presnejší aj efektívnejší.