

Continous optimalization part I.

Úlohu som riešil na 5 funkciách.

- Sphere
- Elipsoid
- Rastrigin
- BentCigar
- Schewel

V originálnom kóde existuje operátor GaussianMutation, čo je ovplyvnená mutácia, ktorá k aktuálnej pozícii pričítajú náhodné číslo z normálneho rozdelenia $[0,1]$. Pomimo tejto som vytvoril 4 druhy operátorov.

- UnBiased – Aktuálnu pozíciu nastaví na hodnotu z normálneho rozdelenia z rozsahu $[-5,5]$
- Biased – Ovplyvnená mutácia podobne ako originálny operátor. Pričítavam k aktuálnej pozícii hodnotu z normálneho rozdelenia no s priemerom -1 a odchýlkou 1
- AdaptiveUnBiased – Adaptívny operátor, ktorý podľa toho ako spomaľujeme pri hľadaní optimálneho riešenia, tak tým zväčšuje pravdepodobnosť mutácie. Pri mutácii vykoná nestrannú mutáciu a nastaví hodnotu na pozícii z normálneho rozdelenia z rozsahu $[-5,5]$
- AdaptiveBiased – Adaptívny operátor, ktorý podľa toho ako spomaľujeme pri hľadaní optimálneho riešenia, tak tým zväčšuje pravdepodobnosť mutácie. Pri mutácii vykoná ovplyvnenú mutáciu.

Funkcie som vyskúšal na originálnej GaussianMutation, na PolynomialMutation a na mojich vytvorených. Výsledky sú vidieť v grafoch.

V dosť prípadoch je vidieť, že adaptívne mutácie dávajú lepšie hodnoty ako tie statické (napr. Pri Rastrigin funkcii, dokonca pri Sphere funkcii AdaptiveBiased).

Operátor PolynomialMutation pri Sphere function a BentCigar poráža ostatné operátory o niekoľko rádov a pri iných funkciách si nevedie až tak zle.

V niektorých prípadoch sú statické operátory lepšie ako adaptívne (napr. Pri Schewel funkcii vychádza najlepšie nami vytvorená ovplyvnená mutácia)