

Knapsack pomocí evoluce

Vojtěch Šára

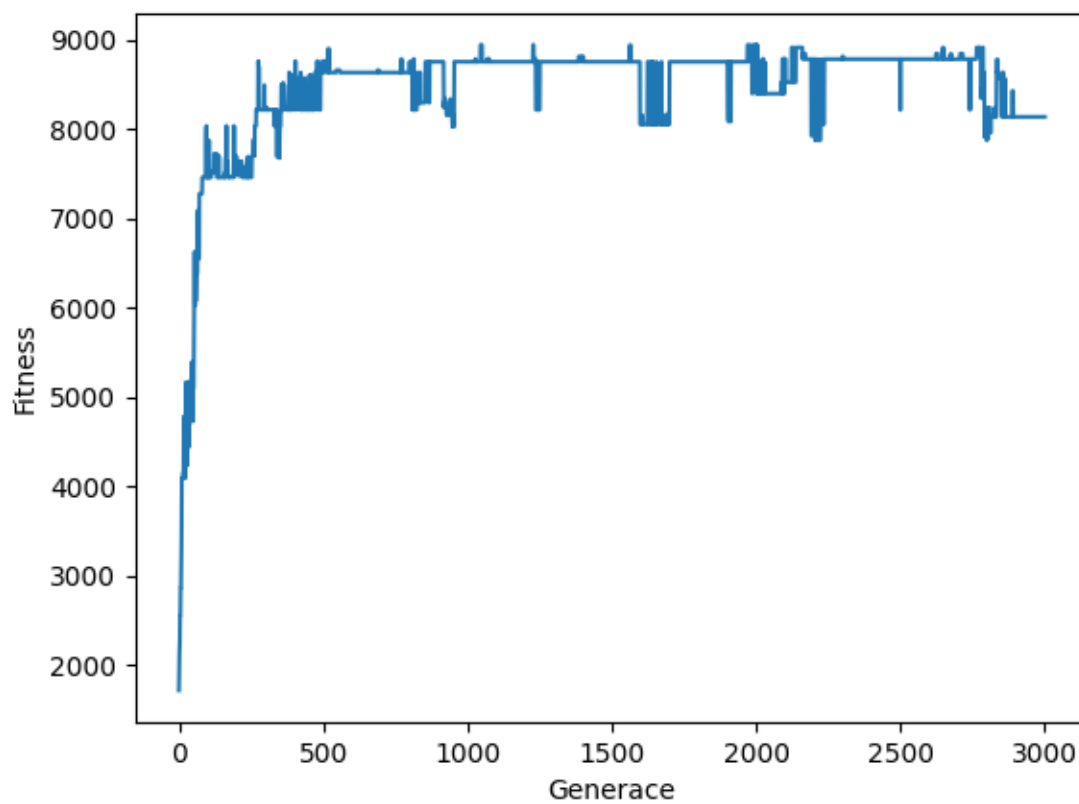
April 8, 2021

Jedince jsem kódoval pomocí indikátorových vektorů - tedy jedinec 1, 0, 1 odpovídá tomu, že první a třetí předmět se do batohu vyberou a prostřední ne. Díky tomu je spočítání fitness jedno maticové násobení. Selekcí jsem vybral ruletovou, protože mi experimentálně vyšla trochu lepší.

Důležitá se ukázala inicializace, pro kterou jsem si nejdříve spočítal počet jedniček, které chci do vektoru napsat, abych nevytvářel zbytečně tučné jedince. Křížení je jednobodové a fitness jsem určil jako váhu. Mutaci jsem změnil z původního bitflipu na průchod celým vektorem, s náhodnou změnou podle pravděpodobnosti `valuemutationprob`.

Mutaci se vyplatilo nastavovat vyšší, protože mi experimentálně vyšla takto trochu agresivnější mutace lépe, jinak se simulace příliš zasekává v latentním prostoru na nějakém lokálním minimu, agresivnější mutace pomáhá z těchto lokálních minim vyštoučovat. Optimální hodnoty pro mutaci záležely na velikosti testu, pro test 100 vyšly nejlepší parametry pro mutaci 0.55, 0.008, pro test 1000 to bylo 0.79, 0.001, kde první hodnota je pravděpodobnost, že bude jedinec mutován a druhá hodnota je agresivita mutace (viz kód).
Výsledky:

Test 100

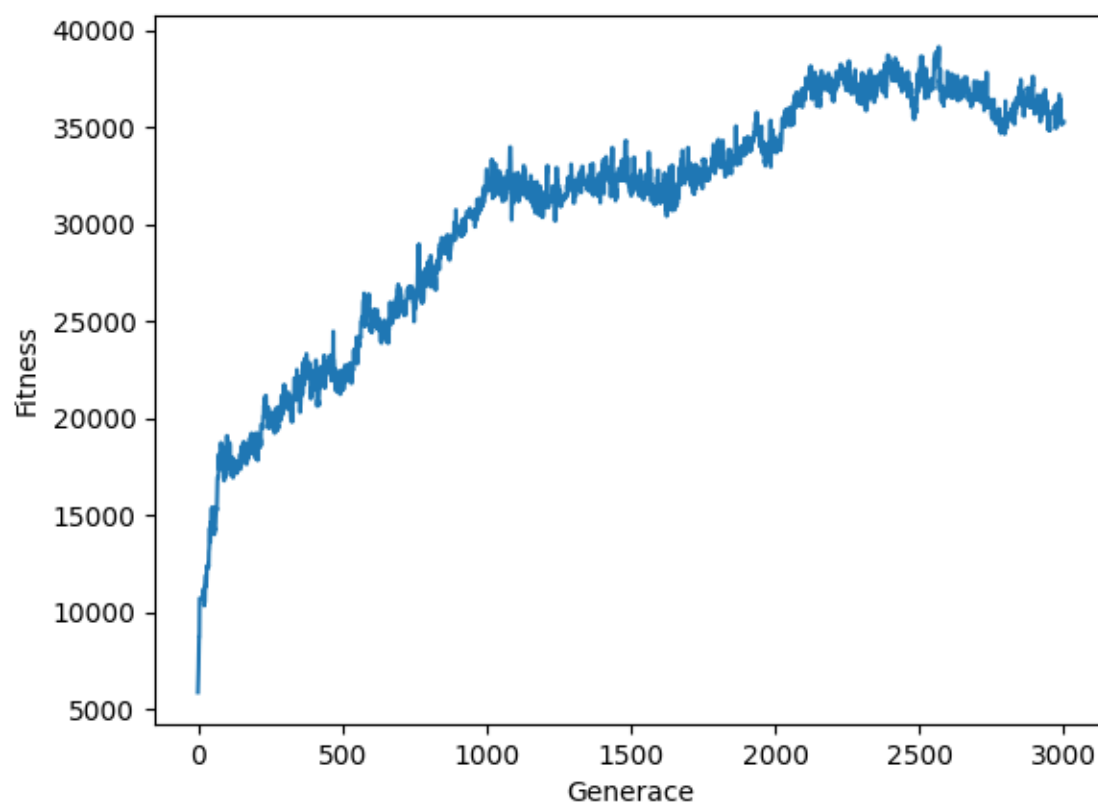


```

fitness: 8929.0 best fitness from last gen: 8900.0 best individual from last gen: [0 0 0 0 0 0 1 0 0 0 1 0 0 1 0
0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 1 0 0 1 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]

```

Test 1000

[illegible]