

Evolučné algoritmy - Batoh:

Našou úlohou je pomocou evolučných algoritmov nájsť čo najlepšie riešenie problému batohu. Máme počet predmetov n , maximálnu nosnosť batohu k a váhy a ceny jednotlivých predmetov c_i a v_i kde i reprezentuje i -ty predmet. Snažíme sa nájsť také predmety aby súčet ich váh nepresiahol nosnosť batohu, ale aby v súčte mali čo najvyššiu hodnotu.

Reprezentácia:

Binárni jedinci:

Na začiatku som zvolil cestu binárnych jedincov. Jedinec má veľkosť n , a má binárnu doménu:

- 1 = predmet je v batohu
- 0 = predmet nie je v batohu

Fitness funkcia:

vracia cenu predmetov v batohu, ale pokiaľ predmety presahujú váhu, tak vracia 1.

Algoritmus dobre fungoval na debug dátach a našiel optimum, avšak keď som sa pustil do testovacích dát, vôbec nefungoval. Premýšľal som ako zmeniť fitness funkciu, aby maximalizovala cenu, ale zároveň brala do úvahy váhu predmetov, a vtedy som sa rozhodol od binárnych jedincov upustiť a vyskúšať permutačných.

Permutačný jedinci:

Jedinec má opäť veľkosť n , ale teraz sa skladá z čísiel 1 až n , ktoré sa neopakujú, takže ide o permutáciu.

Fitness funkcia:

predmety si zoradíme do poľa. Potom do batohu pridávame postupne predmety v poradí permutácie. Keď predmet ktorý práve pridávame do batohu sa už nezmestí, tak ho aj s ostatnými predmetmi zahodíme a vraciame súčet cien predmetov, ktoré už sú v batohu. Takže prvých niekoľko predmetov permutácie sú tie, čo sa dostanú do batohu.

Evolučný algoritmus:

Selekcia:

Najskôr som sa rozhodol pre ruletovú selekciu, tá dobre fungovala na menších batohoch a našla optimá, ale na najväčšom batohu úplne zlyhávala a nedokázala sa dostať cez 20000.

Tak som vyskúšal turnajovú selekciu s počtom kôl = 3 a výsledky sa o mnoho zlepšili. Zamyslel som sa, prečo by to tak mohlo byť, a napadlo mi, že čím vyššie hodnoty fitness funkcie

generácia dosahuje, tým sa aj distribúcia rulety približuje k uniformnej, pretože jedinec s fitness 40_000 bude mať podobnú pravdepodobnosť ako jedinec s 45_000, aj keď druhý jedinec je o mnoho lepší. To práve rieši turnaj, kde jedinci s vyššími fitness majú stále vysokú šancu na to, že si ich vyberieme na kríženie.

Kríženie:

Na kríženie som zvolil PMX, metódu, ktorú sme preberali na prednáške a ide o často používanú metódu kríženia pre permutačných jedincov.

Mutácia:

Ako mutáciu som zvolil transpozíciu, takže výmenu 2 náhodných prvkov permutácie. Neskôr som mi zdalo, že mutácia volí až príliš malé okolie jedinca, a často sa mi stávalo že v neskorších generáciách celá populácia konvergovala k lokálnemu minimu. Preto som zvýšil počet transpozícií, na náhodne zvolené celé číslo z intervalu 1 až t.

Parametre evolúcie:

100 predmetov:

- Veľkosť populácie: 200
- Počet epoch: 200
- Pravdepodobnosti operátorov:
 - o Kríženie - 0.8
 - o mutácia - 0.2

1000 predmetov:

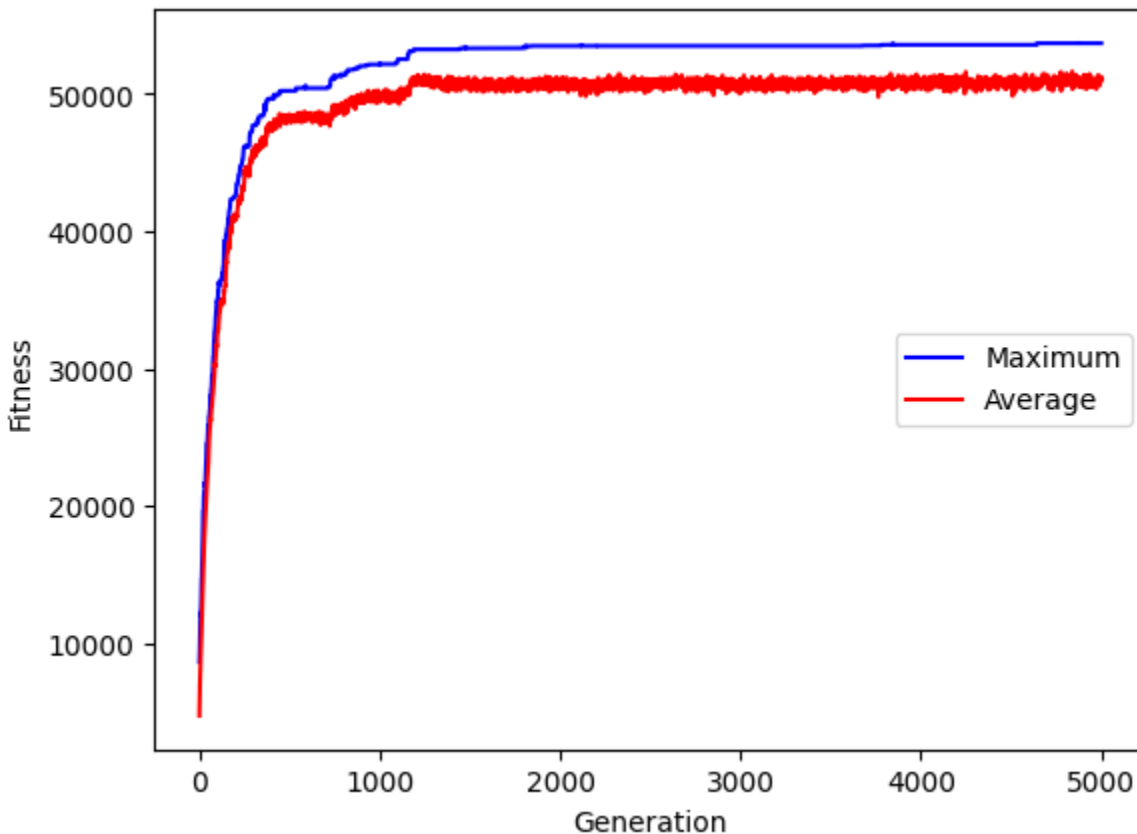
- Veľkosť populácie: 2000
- Počet epoch: 5000
- Pravdepodobnosti operátorov:
 - o Kríženie – 0.7
 - o Mutácia – 0.4

Pri 1000 predmetoch som začal s nižším počtom jedincov aj epóch, ale nedarilo sa mi dostať do konvergenzie, tak som zvýšil veľkosť populácie, aj počet generácií. 5 000 epóch som trochu prestrelil, pretože väčšina behov algoritmu už pri 2 000 generáciách dosahovala fitness 52 000 až 53 000. Potom už populácia stála kľudne aj 500 generácií, kým sa zvýšila max fitness len o niekoľko bodov. Na druhú stranu, vďaka 5000 epochám sa mi podarilo dostať cez 54 000.

Najlepšie výsledky:

100 predmetov - 9147

1000 predmetov - 54178



Záver:

Vo všetkých testoch sa mi podarilo dostať optimum, alebo sa k nemu aspoň priblížiť. Poznatky, na ktoré som prišiel, som okomentoval už v minulých odstavcoch, až na jeden, a to je elitizmus (skúšal som rôzne implementácie, napr. zobrať najlepších z rodičov + detí). Ten som sa pokúšal implementovať do svojho algoritmu, avšak skoro vždy sa mi stalo, že okolo fitness 50 000 celá populácia zdegenerovala a všetci jedinci boli rovnakí. Potom sa už max fitness pohybovala len vďaka náhodným mutáciám, ale to sa stalo raz za 1000 generácií. Takže som od elitizmu upustil, a hľadal iné cesty.