

MI-PAA 2015 4.ukol

Tomas Nesrovnal
nesrotom@fit.cvut.cz

December 28, 2015

1 Zvolený algoritmus

Simulovaná evoluce

1.1 Jedinec

Jedinec v populaci je jednorozmerne bitove pole, kdy jednotlivy bit znaci pritomnost predmetu v batohu.

1.2 Fitness

Zde jsem zvolil jednoduche a primocare reseni a to, ze fitness jedince je celkova cena. Pokud je prekrocena kapacita, je fitness 0.

Mozne vylepseni by bylo priradit kladnou fitness i nevalidnim resenim, ale s dobrym potencialem.

1.3 Inicializace

Kazdy jedinec je nahodne generovan (kazdy bit ma 50 % sancí byt 1) do te doby, dokud nebude mit *fitness* > 0.

1.4 Crossover

Jednobodove a dvoubodove krizeni.

1.5 Mutace

Pri mutaci je s jistou (velmi malou) pravdepodobnosti zmutovan kazdy jedinec.

1.6 Selekcce

Turnajova selekcce.

1.7 Vliv nahody

Pred kazdym merenim byla vyresetovana nahoda. Protoze je implementace v C, vola se `srand(0)`.

2 Mereni

2.1 Metoda

Mereni jsem provedl na setu vygenerovanem pomoci generatoru:

```
.knapgen -I 42 -n 300 -N 1 -m 0.5 -k 0.5 -W 1000 -C 1000 -d 0
```

2.2 Spravnost vysledku

Spravnost vysledku byla overena porovnanim vysledku s vysledkem metody dynamickeho programovani. Je jasne, ze pro velke sety nejsou vysledky ty uplne nejlepsi, pro mensi sety s dobrym nastavenim byly nalezeny optimalni reseni.

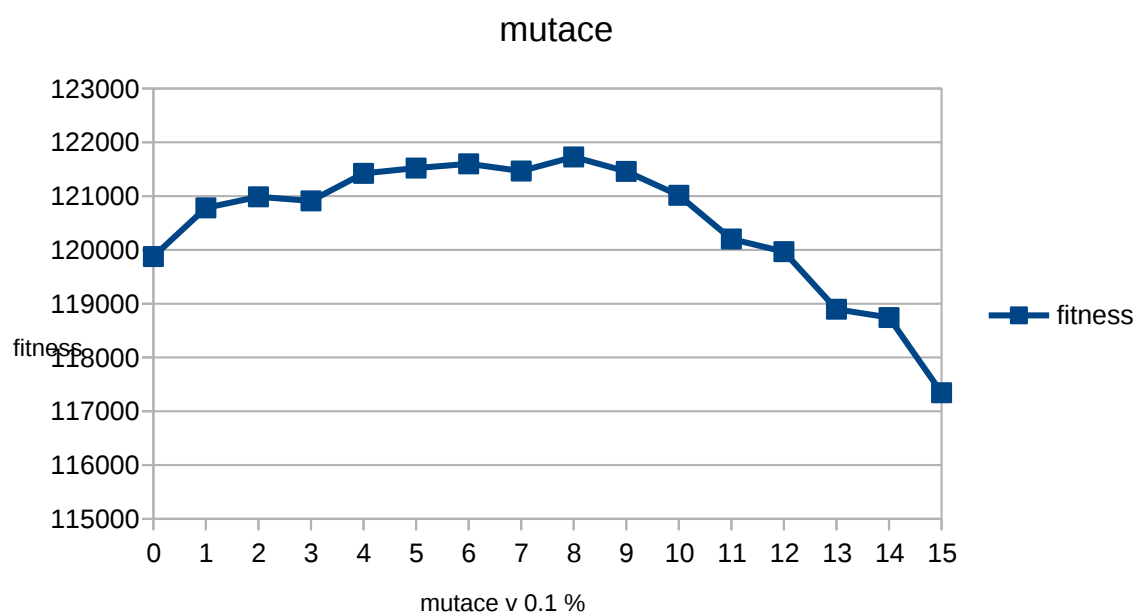
2.3 Na cem bylo mereno

Intel(R) Core(TM) i3-2328M Processor (3M Cache, 2.20 GHz), gcc 4.9.2 (-Ofast), OS GNU/Linux Ubuntu 14.04 64bit

2.4 Mutace

Vyvoj mutace v 0.1 %. Nastaveni: opakovani 1000, populace 1000, velikost turnaje 50

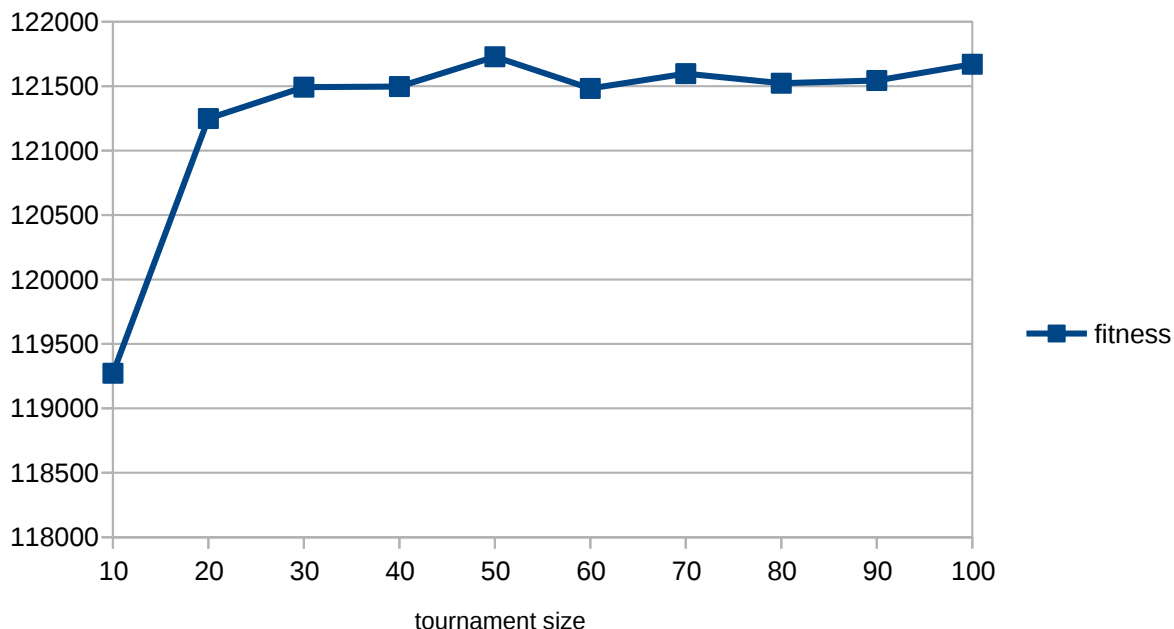
Figure 1: Vliv mutace



2.5 Velikost turnaje

Nastaveni: opakovani 1000, populace 1000, pravdepodobnost mutace jedince 0.8 %

Figure 2: Vliv velikosti tournamentu



2.6 Krizení

Pomocí jednobodového krizení bylo dosaženo výsledku 121365. Pomocí dvoubodového 121728.

2.7 Porovnání s dynamickým programováním: čas a výsledek

Dynamicke programování za 0.079516 sekundy spočítalo výsledek 121794. Simulovaná evoluce za 6.359731 sekund spočítala výsledek 121728.

Nutno podotknout, že složitost simulované evoluce závisí na parametrech, hlavně je to počet opakování, velikost populace a v malé míře záleží i na velikosti turnaje.

3 Závěr

Simulovaná evoluce má výhodu v tom, že po tom co zakódujeme jedice do genomu a vytvoříme fitness funkci, udělá zbytek práce za nás.