

Genetické algoritmy

Lokálne prehľadávanie a optimalizačné problémy

Matúš Goljer

Fakulta informatiky
Masarykova univerzita, Brno

7. December, 2010

Obsah

- 1 Optimalizačné problémy
- 2 Genetické algoritmy
- 3 Aplikácie

Optimalizačné problémy

- problém nájsť najlepšie riešenie z množiny všetkých riešení

Optimalizačné problémy

- problém nájsť najlepšie riešenie z množiny všetkých riešení
- alebo aspoň “dostatočne vhodné” riešenie

Optimalizačné problémy

- problém nájsť najlepšie riešenie z množiny všetkých riešení
- alebo aspoň “dostatočne vhodné” riešenie
- každý problém má definovanú *účelovú funkciu* (cenovú funkciu)

Optimalizačné problémy

- problém nájsť najlepšie riešenie z množiny všetkých riešení
- alebo aspoň “dostatočne vhodné” riešenie
- každý problém má definovanú *účelovú funkciu* (cenovú funkciu)
- problém obchodného cestujúceho - najkratšia cesta
- problém batohu - maximalizovať cenu v zadanom obsahu

Prehľadavacie algoritmy

- riešenie hľadáme prehľadávaním stavového priestoru

Prehľadavacie algoritmy

- riešenie hľadáme prehľadávaním stavového priestoru
- úplné prehľadávanie
- lokálne prehľadávanie
- genetické algoritmy

Úplné prehľadávanie

- generuje všetky možnosti (DFS, BFS...)
- uchováva najlepšie riešenie, ktoré na konci vracia

Úplné prehľadávanie

- generuje všetky možnosti (DFS, BFS...)
- uchováva najlepšie riešenie, ktoré na konci vracia
- kombinatorická explózia
- TSP - $n!$ možných ciest
- pre väčšinu reálnych problémov nepoužiteľné ...

Úplné prehľadávanie

- generuje všetky možnosti (DFS, BFS...)
- uchováva najlepšie riešenie, ktoré na konci vracia
- kombinatorická explózia
- TSP - $n!$ možných ciest
- pre väčšinu reálnych problémov nepoužiteľné ...
- ... ale je úplné :)

Lokálne prehľadávanie

- definuje reláciu *byť susedom*

Lokálne prehľadávanie

- definuje reláciu *byť susedom*
- 1 začína s nejakým známim riešením
- 2 postupne prehľadáva susedné stavy
- 3 vyberie “lepšie” (vzhľadom na výberovú funkciu) z aktuálneho a susedného riešenia
- 4 ak není vybrané riešenie koncové, nastaví aktuálne riešenie na vybrané riešenie, a pokračuje z bodu 2

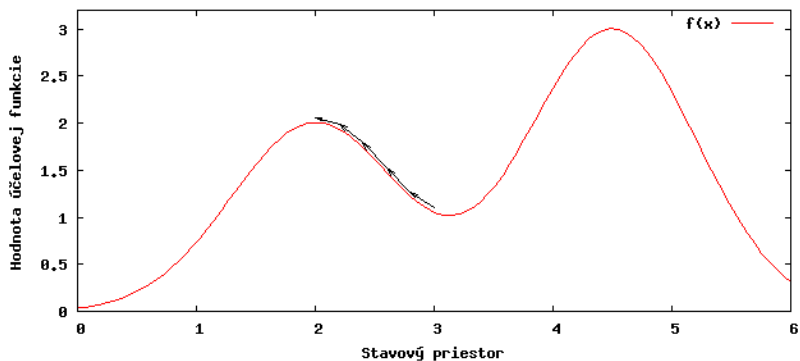
Lokálne prehľadávanie

- definuje reláciu *byť susedom*
 - 1 začína s nejakým známim riešením
 - 2 postupne prehľadáva susedné stavy
 - 3 vyberie “lepšie” (vzhľadom na výberovú funkciu) z aktuálneho a susedného riešenia
 - 4 ak není vybrané riešenie koncové, nastaví aktuálne riešenie na vybrané riešenie, a pokračuje z bodu 2
-
- môže uviaznuť v lokálnom optime

Lokálne prehľadávanie

- konkrétna stratégia záleží na akceptačnej funkcii
 - hill climbing - vyberie vždy lepšie z dvoch riešení
 - simulované žíhanie (simulated annealing) - občas povoľuje výber horšieho riešenia, zabraňuje uviaznutiu v lok. optime
 - tabu search - nepovoľuje opakovaný výber rovnakej zmeny

Lokálne prehľadávanie



Genetické algoritmy

- aplikujú princípy biologickej evolúcie
- prirodzený výber - lepšie riešenie prežije
- dedičnosť
- rekombinácia (crossover)
- mutácia

Genetické algoritmy - pojmy

- gény - pravidlá podľa ktorých je organizmus zostavený

Genetické algoritmy - pojmy

- gény - pravidlá podľa ktorých je organizmus zostavený
- chromozómy - reťazce génov

Genetické algoritmy - pojmy

- gény - pravidlá podľa ktorých je organizmus zostavený
- chromozómy - reťazce génov
- genotyp - súbor génov týkajúcich sa jedného znaku

Genetické algoritmy - pojmy

- gény - pravidlá podľa ktorých je organizmus zostavený
- chromozómy - reťazce génov
- genotyp - súbor génov týkajúcich sa jedného znaku
- fenotyp - pozorovateľné vlastnosti organizmu

Kódovanie chromozómu

- bitové reťazce
- vektory reálnych čísel
- permutácie
- stromy
- grafy

Algoritmus

- 1 inicializuj populáciu
- 2 vyber časť populácie na základe vhodnosti (fitness)
- 3 aplikuj rekombináciu
- 4 aplikuj mutácie
- 5 ak sme presiahli max. počet generácií, vráť najlepšie riešenie z aktuálnej generácie, inak pokračuj v bode 2 s novou generáciou

Inicializácia

- iníciaľna populácia náhodne vygenerovaná
- desiatky až stovky jedincov - doménovo závislé
- možnosť “seedovať” predpripravených jedincov - možné urýchlenie

Selekcia

- vybraný jedinci majú možnosť kríženia
- ďalej sú náhodne mutovaný
- tvoria novú generáciu
- Darwin - najlepší prežije...

Selekcia

- vybraný jedinci majú možnosť kríženia
- ďalej sú náhodne mutovaný
- tvoria novú generáciu
- Darwin - najlepší prežije...
- ...ale nie vždy - slabší jedinci pomáhajú udržiavať genovú diverzitu

Výber úmerný vhodnosti

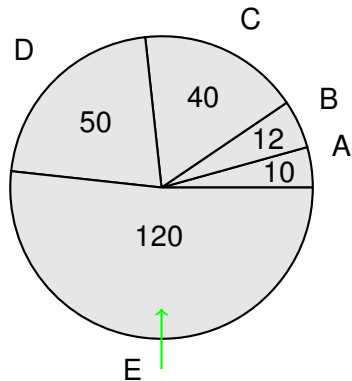
- “roulette-wheel selection”
- pravdepodobnosť výberu úmerná pomeru vhodnosti jedinca voči celej populácii:

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}$$

Výber úmerný vhodnosti

- “roulette-wheel selection”
- pravdepodobnosť výberu úmerná pomeru vhodnosti jedinca voči celej populácii: $p_i = \frac{f_i}{\sum_{j=1}^N f_j}$
- pridelí časť kolesa každému jedincovi
- roztočí koleso, vyberie jedinca na značke

Výber úmerný vhodnosti



Obr.: Výber úmerný vhodnosti

Turnajový výber

- spúšťa n turnajov
- pre každý turnaj vyberie k náhodných jedincov
- šanca výberu i -teho jedinca $p(1 - p)^i$
- $p = 1 \Rightarrow$ deterministický výber - vždy najlepší jedinec
- $k = 1 \Rightarrow$ náhodný výber

Elitizmus (Elitism)

- niekoľko najlepších riešení automaticky pridá do novej generácie

- najlepšie riešenie sa nestratí

Rekombinácia

- simuluje biologické “párenie”

Rekombinácia

- simuluje biologické “párenie”
- nemusí sa obmedzovať na dvoch rodičov a jedného potomka
- od každého rodiča prevezme časť chromozómu a skombinuje do nového jedinca

Jednobodová rekombinácia

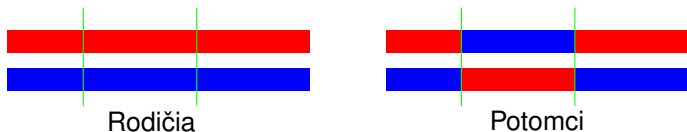
- rozdelí rodičovské chromozómy v jednom bode a prehodí ich, vytvorí dvoch potomkov



Obr.: Jednobodová rekombinácia

Dvojbodová rekombinácia

- rozdelí rodičovské chromozómy v dvoch bodoch
- vymení obsah chromozómu medzi hraničnými bodmi.



Obr.: Dvojbodová rekombinácia

Rekombinácia permutácií

- podobne ako jednobodová rekombinácia
- druhá časť nie je vymenená (mohlo by dôjsť k duplikácii)
- dopočítaná ako $P_2 \setminus O_1$, so zachovaním poradia

Mutácia

- udržuje diverzitu
- zabraňuje rýchlej konvergencii k lokálnemu optimu

Mutácia

- udržuje diverzitu
- zabraňuje rýchlej konvergencii k lokálnemu optimu
- implementácia závislá na kódovaní
 - binárny reťazec: flip $0 \leftrightarrow 1$
 - vektor reálnych čísel: pričítanie Δx
 - permutácia: prehodí dva prvky
 - strom a graf: zmení hodnotu uzlu, pridá hranu/uzol

Scramble

- evolúcia smerom k zadanému reťazcu
- účelová funkcia - Hamming distance
- chromozóm je reprezentovaný ako permutácia znakov
- mutácie - výmena dvoch znakov

Problém obchodného cestujúceho

- NP-ťažký problém
- úplný graf na n vrchoch (vrchol = mesto)
- riešením je nájsť najkratšiu Hamiltonovskú kružnicu
- účelová funkcia - dĺžka cesty
- chromozóm je reprezentovaný ako permutácia vrcholov

Problém batohu

- NP-úplný problém
- množina predmetov, každý má hmotnosť a cenu
- batoh konečnej veľkosti (resp. človek odnesie len určitú hmotnosť)
- chceme maximalizovať hodnotu predmetov v batohu

Problém batohu

- NP-úplný problém
- množina predmetov, každý má hmotnosť a cenu
- batoh konečnej veľkosti (resp. človek odnesie len určitú hmotnosť)
- chceme maximalizovať hodnotu predmetov v batohu
- problém od nepamäťi riešili vykrádači pyramíd (a Indiana Jones)

Problém batohu

- NP-úplný problém
- množina predmetov, každý má hmotnosť a cenu
- batoh konečnej veľkosti (resp. človek odnesie len určitú hmotnosť)
- chceme maximalizovať hodnotu predmetov v batohu
- problém od nepamäťi riešili vykrádači pyramíd (a Indiana Jones)
- v reálnom prostredí sa jedná o pridelovanie zdrojov
 - batoh reprezentuje počet zamestnancov
 - váha predmetu reprezentuje veľkosť teamu
 - cena predmetu reprezentuje zisk z predaja produktu

Problém batohu

- chromozóm - bitový reťazec. 1 = predmet sme vybrali, 0 = predmet sme nevybrali
- jednobodová rekombinácia
- mutácia - flip bitu

Evolúcia motorky

- má dve kolesá a dve závažia prepojené “každé s každým” pružinami
- rekombinácia - náhodne vážený priemer rodičovských parametrov
- mutácia - zmena polomeru kolesa/závažia, zmena váhy pružiny
- <http://www.qubit.devisland.net/ga/>

Minesweeper

- úlohou robota je pozbierať čo najviac mín
- pohyb je ovládaný neurónovou sieťou
- 4 vstupy
 - x a y zložka vektoru, kde robot smeruje
 - x a y zložka vektoru k najbližšej míne
- 2 výstupy: rotácia ľavého a pravého tracku
- pohybuje sa ako pásový “tank”
- účelová funkcia - počet pozbieraných mín

Minesweeper

- neurónová sieť reprezentovaná ako vektor váh (reálne čísla)
- pevná topológia, existuje teda jednoznačné mapovanie

Minesweeper

- neurónová sieť reprezentovaná ako vektor váh (reálne čísla)
- pevná topológia, existuje teda jednoznačné mapovanie
- jednobodová rekombinácia
- mutácia upraví vybranú váhu o $\pm(0, 0.3)$

Neurónová sieť

