

Genetické algoritmy

- Evolúcia k optimálnym riešeniam prebieha v mnohých generáciach populácií riešení pomocou prirodzeného výberu
- neúspešné riešenia vymierajú
- úspešné riešenia prežívajú a množia sa
- hybnou silou evolúcie je kríženie (výmena genetickej informácie medzi riešeniami) a mutovanie

1

Evolúcia

- veľmi zjednodušená inšpirácia tým, ako evolúcia prebieha v prírode (a študuje sa v biológii):
- živé organizmy (zvieratá alebo rastliny) vytvárajú potomkov, ktorí sú skoro (ale nie celkom) ako oni.
 - variáciu môže spôsobiť mutácia (náhodné zmeny)
 - variácia môže byť výsledkom pohlavného rozmnožovania (potomkovia majú nejaké vlastnosti od každého rodiča)
- niektorí z týchto potomkov prežijú a vytvoria vlastných potomkov, niektorí neprežijú
 - lepšie prispôbení potomkovia prežijú s väčšou pravdepodobnosťou
 - ako evolúcia pokračuje v čase, neskoršie generácie sa stávajú lepšie a lepšie prispôbenými
- genetické algoritmy používajú ten istý proces, aby vyvinuli lepšie programy

2

gén a chromozóm

- Gén je časťou molekuly DNA, je jednotkou dedičnosti, v ktorej je zakódovaná informácia pre vznik určitého znaku. Znak je určitá jednotlivá vlastnosť organizmu, v ktorej sa odlišuje jeden organizmus od druhého.
- Gény sú základné inštrukcie „budovania“ živého organizmu.
- Chromozóm obsahuje viacero génov. V každom chromozóme sú gény, zoradené v presnom poradí za sebou.
- Chromozóm je postupnosť génov.

3

genotyp a fenotyp

- Genotyp je súbor dedičných vloh, ktoré má organizmus.
- Na tento súbor vloh však pôsobí prostredie, ktoré vyvíjajúceho sa jedince obklopuje. Jeho genotyp a vplyvy prostredia vyvolávajú určité znaky a vlastnosti.
- Súbor týchto znakov a vlastností, ktoré sa prejavujú už navonok, sa nazýva fenotyp. Fenotyp je výsledkom vzájomného pôsobenia medzi genotypom a prostredím.
- Genotyp je viac alebo menej nemenný, zatiaľ čo fenotyp sa vplyvom prostredia môže meniť. Pravda, aj genotyp sa môže meniť, a to jednak krížením – kombináciou vloh obidvoch rodičov a jednak mutáciami. Mutácie sú náhle vzniknuté a trvalé dedičné zmeny.
- genotyp organizmu (gény a chromozómy) – jeho fenotyp (aký je v skutočnosti)

4

základný genetický algoritmus

- začni s veľkou „populáciou“ náhodne vygenerovaných „pokusných riešení“ daného problému
- opakuj:
 - ohodnot' každé z „pokusných riešení“
 - podrž si podmožinu z nich (tie najlepšie ohodnotené)
 - použi tieto riešenia na generovanie novej populácie
- skonči keď máš uspokojivé riešenie alebo nemáš viac času

5

naozaj jednoduchý príklad

- predpokladajme, že naše organizmy sú 32-bitové počítačové slová
- riešením je slovo so samými jednotkami
- ako to dosiahnuť:
 - vytvor 100 náhodne vygenerovaných počítačových slov
 - opakuj:
 - spočítaj jednotky v každom slove
 - skonči ak má nejaké slovo 32 jednotiek
 - podrž si 10 slov, ktoré majú najviac jednotiek (odhod' zvyšné)
 - z každého slova vygeneruj 9 nových slov takto:
 - zvol náhodne jeden bit v slove a zmeň ho
- všimnime si, že táto procedúra nezaručuje, že v ďalšej generácii budú slová s väčším počtom jednotiek, hoci je
- 6 to pravdepodobné

ozajstnejší príklad I

- predpokladajme, že máme veľký počet dvojíc čísel (bodov v dvojrozmernom priestore) (x, y)
 - napr $(1.0, 4.1), (3.1, 9.5), (-5.2, 8.6), \dots$
- treba preložiť týmito bodmi polynóm (najviac stupňa 5)
 - tj hľadáme výraz $y = ax^5 + bx^4 + cx^3 + dx^2 + ex + f$ taký, ktorý čo najlepšie vyhovuje daným údajom
 - bežný postup ako určiť „vyhovenie“:
 - vypočítať súčet (dané y - vypočítané y)² pre všetky body
 - najnižší súčet predstavuje najlepšie vyhovenie
- sú známe bežné metódy aproximácie, povedzme, že ich nepoznáme
- dá sa použiť genetický algoritmus na nájdenie „celkom dobrého“ riešenia

7

ozajstnejší príklad II

- náš výraz je $y = ax^5 + bx^4 + cx^3 + dx^2 + ex + f$
- naše „gény“ sú $a, b, c, d, e, a f$
- náš „chromozóm“ je pole $[a, b, c, d, e, f]$
- naša vyhodnocovacia funkcia pre jedno pole je:
 - pre každý bod (x, y) :
 - vypočítaj $\hat{y} = ax^5 + bx^4 + cx^3 + dx^2 + ex + f$
 - vypočítaj súčet $(y - \hat{y})^2$ cez všetky x
 - súčet je mierou „nedobrosti“ (suma štvorcov rozdielov má byť čo najmenšia, čím je väčšia, tým je horšie)
 - napr: pre pole $[0, 0, 0, 2, 3, 5]$ a body $(1, 12)$ and $(2, 22)$:
 - $\hat{y} = 0x^5 + 0x^4 + 0x^3 + 2x^2 + 3x + 5$ je $2 + 3 + 5 = 10$ pre x je 1
 - $\hat{y} = 0x^5 + 0x^4 + 0x^3 + 2x^2 + 3x + 5$ je $8 + 6 + 5 = 19$ pre x je 2
 - $(12 - 10)^2 + (22 - 19)^2 = 2^2 + 3^2 = 13$
 - ak by boli len tieto dva body, „nedobrosť“ poľa $[0, 0, 0, 2, 3, 5]$ je 13

8

ozajstnejší príklad III

- samotný algoritmus by bol:
 - vytvor 100 6-prvkových polí náhodných čísel
 - opakuj 500 razy (alebo hocikaký iný počet opakovaní):
 - pre každé z týchto 100 polí vypočítaj jeho nedobrosť (používaj všetky dátové body)
 - ponechaj 10 najlepších polí (zahod' ostatných 90)
 - z každého poľa, ktoré ponechávaš, vygeneruj 9 nových polí takto:
 - zvol ľubovoľný prvok poľa (spomedzi jeho 6 prvkov)
 - zvol náhodné reálne číslo medzi 0.0 a 2.0
 - vynásob náhodne zvolený prvok poľa s náhodne zvoleným číslom
 - po 500 pokusoch vráť pole s najlepším ohodnotením ako riešenie

9

bezpohlavné vs. pohlavné rozmnožovanie

- v doterajších príkladoch:
 - každý jedinec - „organizmus“ (alebo „riešenie“) mal len jedného rodiča
 - rozmnožovanie bolo bezpohlavné
 - jediný spôsob, ako vniesť variáciu, bol mutáciou (náhodnými zmenami)
- pri pohlavnom rozmnožovaní:
 - každý „organizmus“ (alebo „riešenie“) má dvoch rodičov (okrem členov počiatočnej populácie)
 - za predpokladu, že každý organizmus má iba jeden chromozóm, noví potomkovia vznikajú vytvaraním nového chromozómu z častí chromozómov každého rodiča

10

naozaj jednoduchý príklad

- predpokladajme, že naše organizmy sú 32-bitové počítačové slová
- riešením je slovo so samými jednotkami
- ako to dosiahnuť:
 - vytvor 100 náhodne vygenerovaných počítačových slov
 - opakuj:
 - spočítaj jednotky v každom slove
 - skonči ak má nejaké slovo 32 jednotiek
 - podrž si 10 slov, ktoré majú najviac jednotiek (odhod' zvyšné)
 - z každého slova vygeneruj 9 nových slov takto:
 - zvol jedno z ďalších slov
 - vezmi prvú polovicu z tohto slova a skombinuj ju s druhou polovicou toho ďalšieho slova

11

naozaj jednoduchý príklad - kríženie

- polovica z jedného, polovica z druhého:
 $0110\ 1001\ 0100\ 1110\ 1010\ 1101\ 1011\ 0101$
 $1101\ 0100\ 0101\ 1010\ 1011\ 0100\ 1010\ 0101$
 $0110\ 1001\ 0100\ 1110\ 1011\ 0100\ 1010\ 0101$
- alebo zvolíme „gény“ (bity) náhodne:
 $0110\ 1001\ 0100\ 1110\ 1010\ 1101\ 1011\ 0101$
 $1101\ 0100\ 0101\ 1010\ 1011\ 0100\ 1010\ 0101$
 $0100\ 0101\ 0100\ 1010\ 1010\ 1100\ 1011\ 0101$
- alebo považujeme za „gén“ väčšie zoskupenia bitov:
 $0110\ 1001\ 0100\ 1110\ 1010\ 1101\ 1011\ 0101$
 $1101\ 0100\ 0101\ 1010\ 1011\ 0100\ 1010\ 0101$
 $1101\ 1001\ 0101\ 1010\ 1010\ 1101\ 1010\ 0101$

12

porovnanie jednoduchých príkladov

- v naozaj jednoduchom prípade (dostať všetky 1):
 - pohlavný (dvaja rodičia, bez mutácie) prístup, ak uspeje, tak omnoho rýchlejšie
 - lebo sa v každom kroku mení až polovica bitov, nie iba jeden
 - pravda, bez mutácie nemusí vôbec uspieť
 - čírou nešťastnou náhodou sa môže napr. stať, že žiadne z pôvodne generovaných slov nemá napr. 27. bit nastavený na 1
 - vtedy nejestuje žiadny spôsob, aby sa môže 1 dostať na 27. miesto
 - ďalší problém je nedostatok genetickej rôznorodosti
 - možno aj malo niekoľko pôvodne generovaných slov na 27. mieste 1, ale ani jedno z nich sa nevybralo do druhej generácie
- najlepší prístup vo všeobecnosti sa javí byť pohlavné rozmnožovanie s malou pravdepodobnosťou mutácií

13

aproximácia pomocou pohlavného rozmnožovania

- náš výraz je $y = ax^5 + bx^4 + cx^3 + dx^2 + ex + f$
- naše „gény“ sú $a, b, c, d, e, a f$
- náš „chromozóm“ je pole $[a, b, c, d, e, f]$
- ako najlepšie kombinovať dva chromozómy do jedného?
 - môžeme vziať prvú polovicu z jedného a druhú polovicu z druhého: $[a, b, c, d, e, f]$
 - môžeme voliť gény náhodne: $[a, b, c, d, e, f]$
 - môžeme počítať „priemerné gény“: $[(a+a)/2, (b+b)/2, (c+c)/2, (d+d)/2, (e+e)/2, (f+f)/2]$
 - domnienka: posledný nápad by mohol byť najlepší

14

usmerňovaná evolúcia

- v doterajších príkladoch sa potomkovia vytvárali náhodne
 - nepokúšali sme sa vyberať „najlepšie“ gény z každého rodiča
 - takto funguje prirodzená (biologická) evolúcia
 - evolúcia v prírode nie je / sa nejaví byť usmerňovaná — nemá „cieľ“
- genetické algoritmy sa inšpirujú prírodou, ale neberú ju ako množinu pravidiel, ktoré treba otrocky dodržiavať
 - pre úlohu dostať slovo so samými 1 jestvuje evidentná miera toho, čo je „dobrý“ gén
 - lenže to vlastne vôbec nie je dobrý príklad problému
 - pre problém aproximácie je omnoho ťažšie rozpoznať „dobrý“ gén
 - ťažké pre praktický každý „skutočný“ problém riešený genetickým algoritmom

15

stochastický výber

- v doterajších príkladoch sa vyberalo N „najlepších“ organizmov za rodičov v ďalšej generácii
- bežnejší prístup je vyberať rodičov náhodne, s prihliadnutím na ich mieru „dobroti“, odhadovanej úspešnosti
 - čiže organizmus, ktorý je hodnotený ako dvakrát lepší než iný bude mať pravdepodobne dvakrát toľko potomkov
- výhody:
 - do ďalšej generácie najviac prispievajú najlepšie organizmy
 - každý organizmus má stále aspoň nejakú nádej, že bude vybraný ako rodič, preto sa nestráca genetická rôznorodosť

16

GA trochu podrobnejšie

- každé riešenie úlohy (každý stav) označuje pojmom **chromozóm**.
- populácia P obsahujúcou M chromozómov.
- chromozóm - *binárny reťazec* x dĺžky N
- Napríklad pre problém 8 dám by chromozóm zodpovedal usporiadanej množine políčok šachovnice, pričom jednotka by znamenala, že dané políčko je obsadené dámou a nula, že je prázdne.

17

GA trochu podrobnejšie

- Každému chromozómu $x \in P$ sa priradí hodnota funkcie úspešnosti **fitness** $f(x)$.
- vyjadruje úspešnosť daného riešenia.
- hľadáme globálne maximum fitness.

18

GA trochu podrobnejšie

```

function Genetický-Algoritmus(populácia, Vyhodnocovacia-Funkcia)
  returns riešenie
  inputs: populácia, množina riešení
           Vyhodnocovacia-Funkcia, funkcia, ktorá vyjadruje
           úspešnosť daného riešenia
  repeat
    rodičia ← Vyber(populácia, Vyhodnocovacia-Funkcia)
    populácia ← Reprodukcia(rodičia)
  until nejaké riešenie je dostatočne úspešné
  return najlepšie riešenie v populácii podľa Vyhodnocovacej-
  Funkcie
end
    
```

19

GA trochu podrobnejšie

- 0 ŠTART : vytvor náhodnú populáciu n chromozómov
- 1 FITNES : vyhodnoť fitness $f(x)$ každého chromozómu v populácii
- 2 REPRODUKČIA
 - 0 VÝBER : založený na $f(x)$
 - 1 REKOMBINÁCIA : križenie chromozómov
 - 2 MUTÁCIA : zmutuj chromozómy
 - 3 PRIJATIE : odstráň alebo prijmi nový chromozóm
- 3 NÁHRADA : nahraď starú populáciu novou
- 4 TEST : otestuj, či je v populácii riešenie problému
- 5 CYKLUS : opakuj kroky 1 – 4 dovtedy, až nejaký chromozóm prejde úspešne testom

20

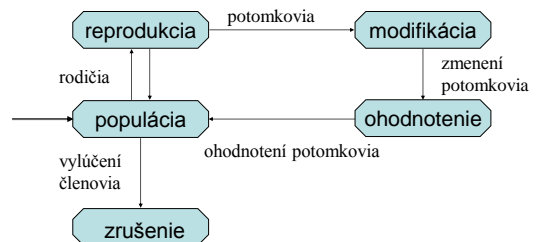
zložky genetického algoritmu

problém, ktorý treba vyriešiť a ...

- metóda kódovania (gén, chromozóm)
- procedúra inicializácie (vytvorenie)
- vyhodnocovacia funkcia (prostredie)
- výber rodičov (reprodukcia)
- genetické operátory (mutácia, križenie)
- nastavenia parametrov (skúsenosť a šikovnosť)

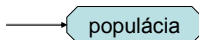
21

cyklus genetického algoritmu evolúcie



22

populácia

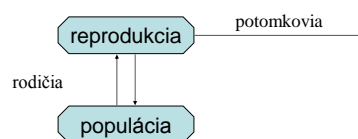


chromozómy môžu byť:

- bitové reťazce (0101 ... 1100)
- reálne čísla (43.2 -33.1 ... 0.0 89.2)
- permutácie prvků (E11 E3 E7 ... E1 E15)
- zoznam pravidiel (R1 R2 R3 ... R22 R23)
- prvky programu (genetické programovanie)
- ... ľubovoľná údajová štruktúra ...

23

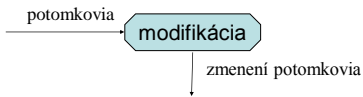
reprodukcia



Rodičia sa vyberajú náhodne, s pravdepodobnosťou úmernou ohodnoteniu chromozómu.

24

modifikácia



- modifikácie sa spúšťajú stochasticky
- typy operátorov:
 - mutácia
 - kríženie (rekombinácia)

25

Mutovanie

- mení náhodne potomkov
- prečo?
 - aby sa predišlo prechodu populácie do lokálneho optima
- ako?
 - niekoľko náhodne zvolených bitov sa invertuje
- príklad

pôvodný potomok 1	1001010000001111
pôvodný potomok 2	0101101100111010
zmutovaný potomok 1	1011010000001111
zmutovaný potomok 2	0101101110111010

26

mutovanie – lokálna modifikácia

pred: (1 0 1 1 0 1 1 0)
 po: (1 0 1 0 0 1 1 0)

pred: (1.38 -69.4 326.44 0.1)
 po: (1.38 -67.5 326.44 0.1)

- spôsobuje pohyb v priestore hľadania (lokálne alebo globálne)
- obnovuje v populácii stratenú informáciu

27

Pravdepodobnosť mutovania

- ako často sa budú mutovať časti chromozómu
 - 0 – potomok prechádza do novej generácie skrížený alebo nezmenený
 - (0, 1) – časť chromozómu sa zmení
 - 1 – celý chromozóm sa zmení
- prečo mutovať?
 - aby GA neupadol do lokálneho extrému
- ak je mutovanie príliš časté, hľadanie riešenia sa blíži náhodnému

28

kríženie - rekombinácia

$$\begin{array}{lcl}
 P1 & (0\ 1\ \overset{*}{1}\ 0\ 1\ 0\ 0\ 0) & \longrightarrow (0\ 1\ \overset{*}{0}\ 0\ 1\ 0\ 0\ 0) \ C1 \\
 P2 & (1\ 1\ 0\ 1\ 1\ 0\ 1\ 0) & \longrightarrow (1\ 1\ \overset{*}{1}\ 1\ 1\ 0\ 1\ 0) \ C2
 \end{array}$$

kríženie je kritická črta genetických algoritmov:

- silne urýchľuje hľadanie v začiatkoch evolúcie populácie
- vedie k účinnému kombinovaniu čiastočných riešení na rôznych chromozómoch

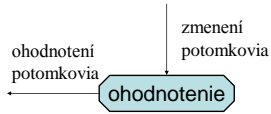
29

pravdepodobnosť kríženia

- ako často sa bude krížiť
 - 0 – celá nová generácia vzniká z presných kópií chromozómov starej generácie
 - (0, 1) – potomkovia vznikajú z častí rodičov
 - 1 – všetci potomkovia vzniknú krížením
- prečo krížiť?
 - nové chromozómy snáď zdedia dobré časti (vlastnosti) a budú lepšie
- odporúča sa nechať časť populácie prežiť do ďalšej generácie

30

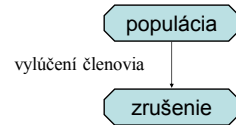
ohodnotenie



- ohodnocovač dekoduje chromozóm a priradí mu hodnotu fitness
- ohodnocovač je jediným spojením medzi GA a problémom, ktorý riešia

31

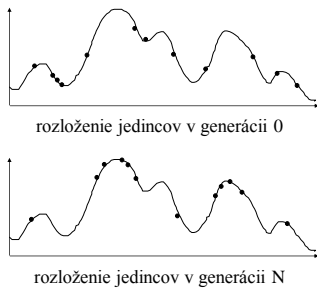
zrušenie



- celé populácie sa nahrádzajú v každom kroku iterácie
- len niekoľko členov sa nahrádza v každom kroku

32

abstraktný príklad



33

opäť naozaj jednoduchý príklad

- pre jednoduchosť, l -bitové slová, začiatočná populácia má n chromozómov.
- nech $l = 10$ a $n = 6$

34

naozaj jednoduchý príklad - inicializácia

hodíme mincu 60 razy, dostaneme začiatočnú populáciu:

$s_1 = 1111010101$	$f(s_1) = 7$
$s_2 = 0111000101$	$f(s_2) = 5$
$s_3 = 1110110101$	$f(s_3) = 7$
$s_4 = 0100010011$	$f(s_4) = 4$
$s_5 = 1110111101$	$f(s_5) = 8$
$s_6 = 0100110000$	$f(s_6) = 3$

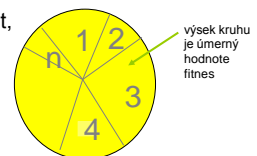
35

naozaj jednoduchý príklad - výber

výber v závislosti od ohodnotenia užitočnosti fitness metódou rulety:

jedinec i bude mať pravdepodobnosť výberu $\frac{f(i)}{\sum_{j=1}^n f(j)}$

výber sa opakuje toľkokrát, koľko potrebujeme jedincov do populácie rodičov – rovnaký počet ako je veľkosť populácie (6 v tomto prípade)



36

naozaj jednoduchý príklad - výber

predpokladajme, že po výberoch dostaneme takúto populáciu:

$$s_1' = 1111010101 \quad (s_1)$$

$$s_2' = 1110110101 \quad (s_3)$$

$$s_3' = 1110111101 \quad (s_5)$$

$$s_4' = 0111000101 \quad (s_2)$$

$$s_5' = 0100010011 \quad (s_4)$$

$$s_6' = 1110111101 \quad (s_6)$$

37

naozaj jednoduchý príklad - kríženie

nastupuje kríženie. pre každý rodičovský pár (dvojicu chromozómov) sa rozhodne na základe pravdepodobnosti kríženia (nech je určená napr 0.6), či sa vôbec skrížia.

nech sa rozhodlo, že skrížia sa iba páry (s_1', s_2') a (s_5', s_6') . Pre každý pár sa náhodne určí bod kríženia, napr. 2 pre prvý a 5 pre druhý

38

naozaj jednoduchý príklad - kríženie

pred skrížením:

$$s_1' = 1111010101$$

$$s_2' = 1110110101$$

$$s_5' = 0100010011$$

$$s_6' = 1110111101$$

po skrížení:

$$s_1'' = 1110110101$$

$$s_2'' = 1111010101$$

$$s_5'' = 0100011101$$

$$s_6'' = 1110110011$$

39

naozaj jednoduchý príklad - mutácia

posledný krok je zmutovať nové jedince: pre každý bit nového jedinca jestvuje nenulová pravdepodobnosť (napr 0.1), že sa zmutuje

pred aplikáciou mutácie:

$$s_1'' = 1110110101$$

$$s_2'' = 1111010101$$

$$s_3'' = 1110111101$$

$$s_4'' = 0111000101$$

$$s_5'' = 0100011101$$

$$s_6'' = 1110110011$$

40

naozaj jednoduchý príklad - mutácia

po aplikovaní mutácie:

$$s_1''' = 1110100101 \quad f(s_1''') = 6$$

$$s_2''' = 1111110100 \quad f(s_2''') = 7$$

$$s_3''' = 1110101111 \quad f(s_3''') = 8$$

$$s_4''' = 0111000101 \quad f(s_4''') = 5$$

$$s_5''' = 0100011101 \quad f(s_5''') = 5$$

$$s_6''' = 1110110001 \quad f(s_6''') = 6$$

41

naozaj jednoduchý príklad - záver

v jednej generácii sa celková užitočnosť (súčet fitness všetkých jedincov) zmenila z 34 na 37, zlepšenie o ~9%

a tak ďalej sa pokračuje od začiatku, pokiaľ nie splnený test riešenia

42

d'alší príklad: problém obchodného cestujúceho

obchodný cestujúci musí navštíviť každé mesto v
jemu pridelenej oblasti práve raz a potom sa vrátiť do
miesta, kde začal. Dané sú ceny prepravy medzi
všetkými dvojicami miest. Aký má byť plán jeho
okružnej cesty, aby bola celková cena najmenšia?

TSP \in NP-úplné problémy

pozn: začneme jedným možným prístupom k
hľadaniu približného riešenia problému obchodného
cestujúceho pomocou GA

43

TSP (reprezentácia, vyhodnotenie, inicializácia a výber)

vektor $v = (i_1, i_2, \dots, i_n)$ reprezentuje cestu (v je
permutácia $\{1, 2, \dots, n\}$)

Fitnes f riešenia je prevrátená hodnota ceny
zodpovedajúcej cesty

Inicializácia: použiť nejakú heuristiku alebo
náhodný výber permutácií $\{1, 2, \dots, n\}$

Použijeme náhodný výber s pravdepodobnosťou
úmernou fitnes

44

TSP (kríženie1)

Potomok vznikne zvolením podpostupnosti cesty z jedného rodiča a
zachovaním relatívneho poradia miest z druhého rodiča

napr:

$p_1 = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9)$ a

$p_2 = (4\ 5\ 2\ 1\ 8\ 7\ 6\ 9\ 3)$

Najprv sa podpostupnosti medzi bodmi kríženia skopírajú do
potomkov

$o_1 = (x\ x\ x\ 4\ 5\ 6\ 7\ x\ x)$ a

$o_2 = (x\ x\ x\ 1\ 8\ 7\ 6\ x\ x)$

45

TSP (kríženie2)

Ďalej, počínajúc od druhého bodu kríženia jedného rodiča, mestá z
druhého rodiča sa kopírujú pri zachovaní poradia

postupnosť miest v druhom rodičovi je

$9 - 3 - 4 - 5 - 2 - 1 - 8 - 7 - 6$

po odstránení miest prevzatých z prvého potomka dostaneme

$9 - 3 - 2 - 1 - 8$

Táto postupnosť sa vloží do prvého potomka

$o_1 = (2\ 1\ 8\ 4\ 5\ 6\ 7\ 9\ 3)$, a podobne do druhého

$o_2 = (3\ 4\ 5\ 1\ 8\ 7\ 6\ 9\ 2)$

46

TSP (inverzia)

Podpostupnosť medzi dvoma náhodne zvolenými bodmi sa
obrátí

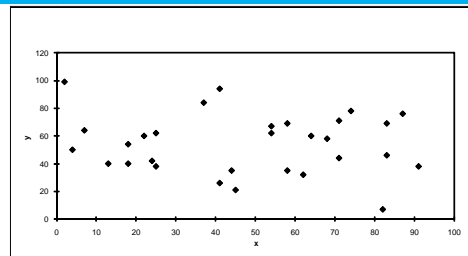
napr:

$(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9)$ sa zmení na $(1\ 2\ 7\ 6\ 5\ 4\ 3\ 8\ 9)$

Také jednoduché obrátenie poradia génov (miest) zaručuje, že
výsledný potomok je prípustná cesta

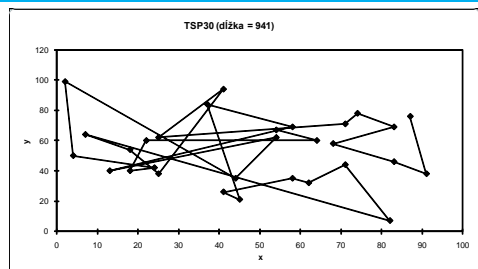
47

TSP príklad – 30 miest



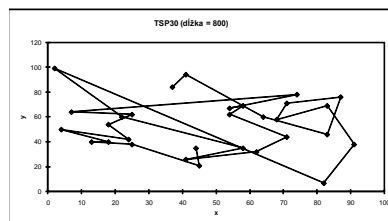
48

TSP príklad – 30 miest



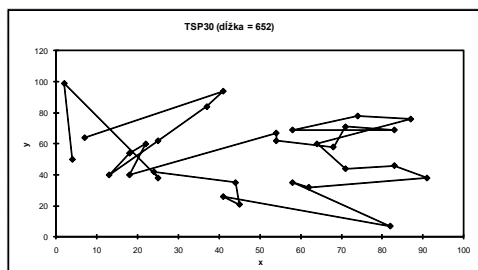
49

riešenie_j (dĺžka 800)



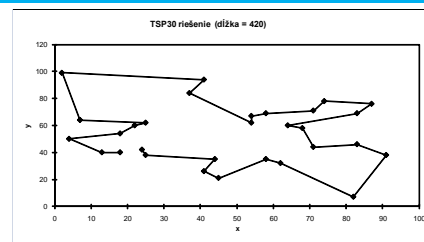
50

riešenie_k (dĺžka 652)



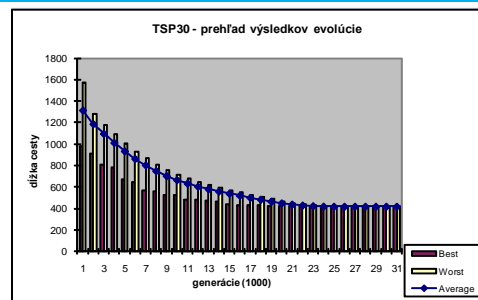
51

najlepšie riešenie (dĺžka 420)



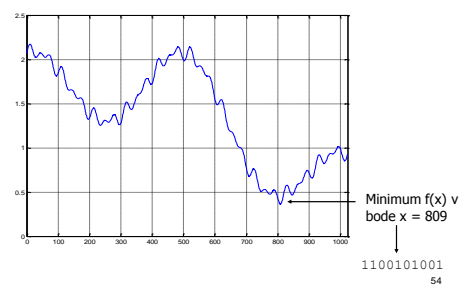
52

prehľad výsledkov evolúcie



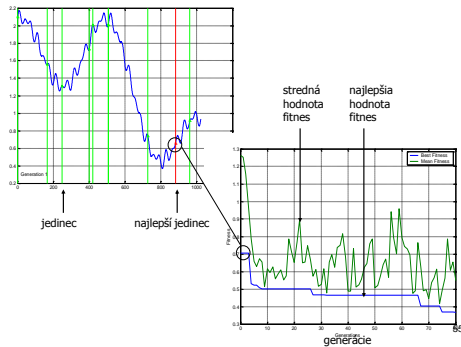
53

príklad: hľadanie minima funkcie



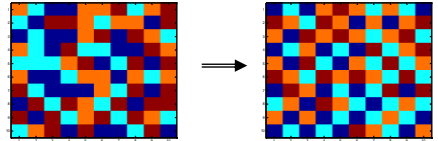
54

príklad: hľadanie minima funkcie



príklad: hracia doska

- daná je hracia doska $n \times n$, na ktorej každé políčko môže mať svoju farbu, jednu z množiny 4 farieb.
- cieľ je dosiahnuť konfiguráciu hracej dosky takú, že žiadne susediace políčka nemajú rovnakú farbu (políčka na uhlopriečke nesusedia)



56

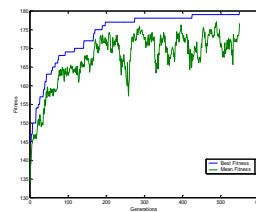
príklad: hracia doska

- chromozómy reprezentujú spôsob, ako je hracia doska ofarbená.
- chromozómy sa nerepresentujú reťazcami bitov, ale 4-bitovými maticami
- 4-bity v matici môžu mať jednu z hodnôt 0, 1, 2 alebo 3, v závislosti od farby
- kríženie zahŕňa manipuláciu matice namiesto operácií nad 4-bitmi. Kríženie môže kombinovať rodičovské matice vodorovne, zvisle, do trojuholníka alebo do štvorca
- mutácia zostáva menením bitov

57

príklad: hracia doska

priebeh funkcie fitness



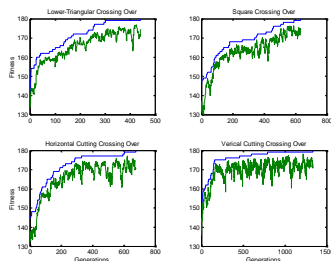
tento problém sa dá opísať grafom s n uzlami a $(n-1)$ hranami, takže funkcia fitness $f(x)$ sa dá ľahko definovať ako:

$$f(x) = 2 \cdot (n-1) \cdot n$$

58

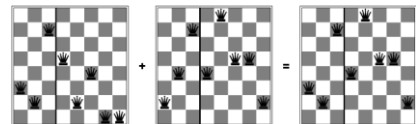
príklad: hracia doska

priebehy funkcie fitness pre rôzne pravidlá kríženia



59

príklad: 8 dám



32752411 24748552 32748552

60

príklad: 8 dám



- Funkcia fitness: počet nenapádajúcich sa dvojíc dám (min = 0, max = $8 \times 7/2 = 28$)
- fitness(chromozóm[24748552]) = $24/(24+23+20+11) = 31\%$
- fitness(chromozóm[32752411]) = $23/(24+23+20+11) = 29\%$
- atď.

61

implementačné aspekty

- parametre
 - reprezentácia kandidátov riešení
 - veľkosť populácie, pravdepodobnosť mutácie, ...
 - pravidlá výberu, vylučovania
 - operátory kríženia, mutácie
- kritérium ukončenia
- výkon, škálovateľnosť
- riešenie je len tak dobré, ako je dobrá vyhodnocovacia funkcia (navrhnuť ju je často najťažšie)

62

počet chromozómov

- koľko chromozómov je v populácii (v jednej generácii)
 - príliš málo
 - málo možností na kríženie -> malá časť stavového priestoru sa dá prehľadať
 - príliš veľa
 - GA sa spomalí
 - zvyšovanie nad nejakú hranicu (závislú najmä od spôsobu kódovania problému) neprináša užitočnosť (zrýchlenie)

63

výber rodičov

- ako vybrať chromozómy, ktoré sa stanú rodičmi?
- (Darwin:) prežiť (a mať potomkov) majú najlepší
- ako vybrať najlepšie chromozómy?
 - ruleta
 - poradie
 - ďalšie

64

ruleta

- výber sa robí na základe fitness
 - čím má chromozóm lepšiu fitness, tým dostane väčšiu šancu
- na kolese rulety má každý chromozóm vyhradený segment úmerný jeho fitness
 - náhodný hod, guľka skončí v niektorom segmente
 - chromozóm s lepšou fitness sa častejšie vyberie
- algoritmus výberu ruletou:
 - S := súčet všetkých ohodnotení fitness v celej populácii
 - r := generuj náhodné číslo v intervale (0, S)
 - s := preberaj jedného člena populácie za druhým a spočítavaj ich fitness začínúc od 0. keď s > r, zastav a vráť chromozóm, pri ktorom si.

65

výber podľa poradia

- výber ruletou môže príliš potlačiť šancu ostatných chromozómov, ak fitness jedného priveľmi dominuje
- výber sa robí na základe fitness
 - chromozóm s lepšou fitness dostane väčšiu šancu
- na kolese má každý chromozóm vyhradený segment úmerný jeho poradiu v usporiadaní podľa fitness
 - chromozóm s najmenšou fitness – 1
 - chromozóm s druhou najmenšou fitness – 2
 - ...
 - chromozóm s najlepšou fitness – N (počet chromozómov v populácii)
- všetky chromozómy dostanú lepšiu šancu byť vybrané
- najlepšie sa nie veľmi líšia od ostatných -> pomalšie konverguje k riešeniu

66

d'alšie možnosti

- stabilita
 - vybrať zopár chromozómov s vysokou fitness ako rodičov
 - vybrať niekoľko chromozómov so slabou fitness, vylúčiť ich z populácie a nahradiť potomkami rodičov
- elitárstvo
 - najlepší alebo niekoľko najlepších chromozómov sa skopírujú do novej populácie
 - ďalej štandardne

67

prínosy GA

- základný pojem je ľahko pochopiteľný
- môže byť samostatný modul, nie súčasť aplikácie
- podporuje aj multikriteriálnu optimalizáciu
- vhodný pre „zašumené“ prostredia
- vždy dáva nejakú odpoveď; odpoveď sa zlepšuje s časom
- z podstaty paralelný; ľahko distribuovaným spracovaním

68

prínosy GA

- veľa možností, ako sa pokúsiť o zrýchlenie, najmä ak pribúdajú znalosti o doméne problému
- ľahko sa dá využiť nejaké predchádzajúce alebo alternatívne riešenia
- pružná softvérová súčiastka vhodná na tvorbu hybridných aplikácií
- významný záznam doterajších prípadov použitia, široký rozsah

69

kedy použiť GA

- alternatívne riešenia sú príliš pomalé alebo príliš zložitá
- treba nástroj na prieskum nových prístupov
- problém sa podobá inému, ktorý sa už úspešne vyriešil pomocou GA

70

niektoré oblasti použitia

Doména	Druh aplikácie
riadenie	plynovod, udržiavanie rovnováhy, uhýbanie pred streľami
navrhovanie	polovodičová doska, lietadlo, klávesnica, komunikačné siete
rozvrhovanie	výroba, prideľovanie zdrojov
robotika	plánovanie trasy
strojové učenie	navrhovanie neurónových sietí, vylepšovanie klasifikačných algoritmov, klasifikácia
spracovanie signálov	navrhovanie filtrov
hranie hier	poker, dáma, vážhova dilema
optimalizácia	TSP, prepojovanie, balenie plezniakov, ofarbovanie grafov

71