

3. prednáška

Genetický algoritmus (GA)

Použitie metafory darvinovskej
evolúcie v informatike pre konštrukciu
optimalizačných algoritmov

Formalizácie základných pojmov evolúcie

Populácia P je množina týchto chromozómov

$$P = \{\alpha_1, \alpha_2, \dots, \alpha_p\} \subseteq \{a, b, \dots\}^k$$

Každý chromozóm $\alpha \in P$ je ohodnotený **silou**
(fitness)

$$F: P \rightarrow R_+$$

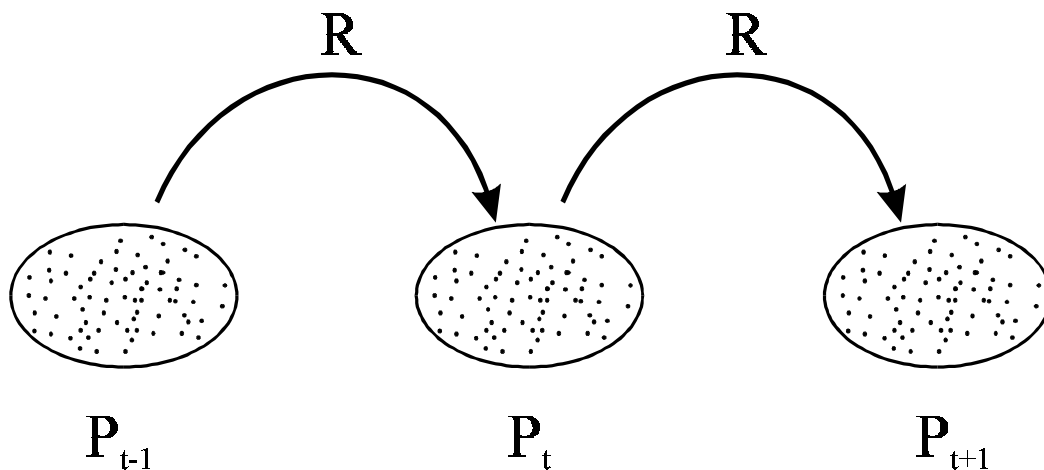
Reprodukcia je proces v ktorom z populácie sú kvázináhodne vybrané dva rodičovské chromozómy α_1 a α_2 v závislosti na ich sile (chromozómy s väčšou silou majú väčšiu pravdepodobnosť byť vybrané), pôvodné chromozómy sa reprodukujú na dva nové chromozómy – potomkov α'_1 a α'_2

$$(\alpha'_1, \alpha'_2) = O_{repro}(\alpha_1, \alpha_2)$$

Reprodukčný operátor obsahuje dve časti, a to *kríženie* a *mutáciu*. Navyac, tento operátor má stochastický charakter, operácie kríženia a mutácie sa vykonávajú len s určitou pravdepodobnosťou.

Evolúcia je rekurentný proces, v ktorom populácia chromozómov P_{t+1} v čase $t+1$ je určená len pomocou predchádzajúcej populácie P_t v čase t

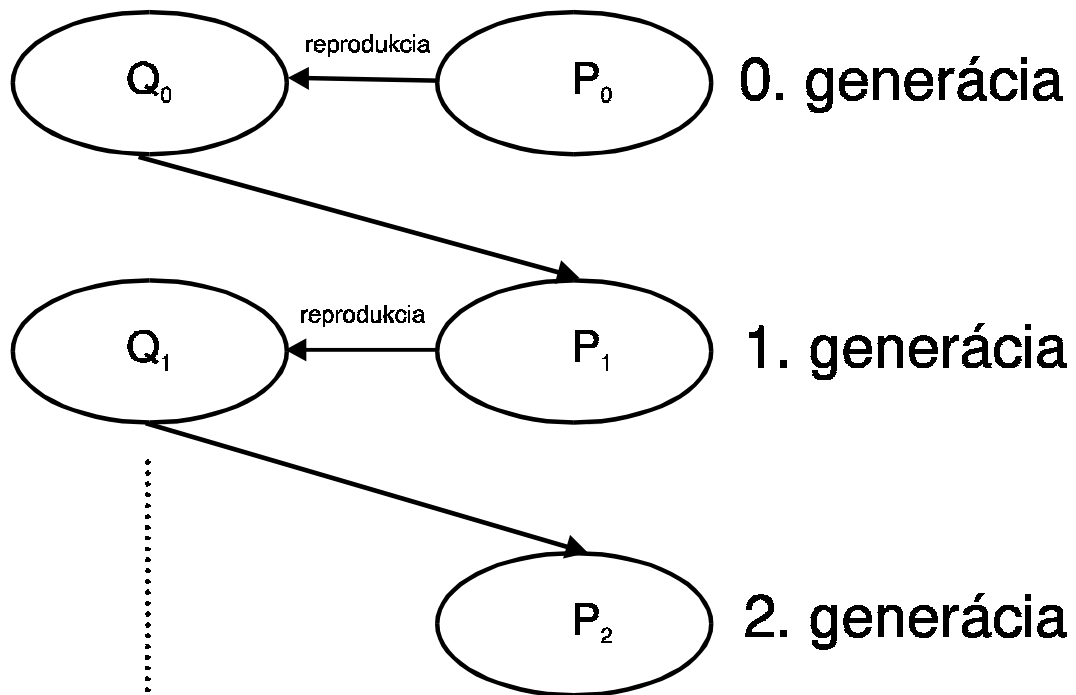
$$P_{t+1} = R(P_t)$$



Pseudopascalovská verzia algoritmu evolúcie v živej prírode, založenej na Darwinovskom prirodzenom výbere.

```
procedure Evolution;  
begin t:=0;  
      P:=randomly generated population of  
        chromosomes;  
      while t<tmax do  
      begin t:=t+1; Q:=∅;  
            while |Q|<|P| do  
            begin α1:=Oselect(P);  
                  α2:=Oselect(P);  
                  if random<Prepro then  
                  (α'1,α'2)=Orepro(α1,α2)  
                  else  
                  (α'1,α'2)=(α1,α2);  
                  Q:=Q∪{α'1,α'2};  
            end;  
            P:=Q;  
      end;  
      Pfin:=P;  
end;
```

Rekurentnosť evolúcie



Aký je vzťah medzi horolezeckým algoritmom a evolučným algoritmom?

- Evolučný algoritmus sa zakladá na simultánnej optimalizácii celej populácie chromozómov, zatiaľ čo horolezecký algoritmus optimalizuje náhodne len jeden objekt - chromozóm.
- Ak v procese reprodukcie chromozómov uvažujeme len mutácie, tak evolučný algoritmus je veľmi podobný horolezeckému algoritmu.
- Možnosť úniku z lokálneho minima pomocou (spočiatku málo výhodnej) mutácie v evolučnom algoritme je spojená s potenciálnou nevýhodou, vygenerovaný chromozóm môže mať menšiu silu ako pôvodný chromozóm.
- V evolučnom algoritme je lokálne hľadanie optimálneho riešenia horolezeckého algoritmu nahradené efektívnejším spôsobom simultánnej optimalizácie celej populácie chromozómov.

Prečo v evolučnom algoritme do reprodukčného procesu vyberáme chromozómy kvázináhodne?

- Ak by sme do reprodukčného procesu vyberali len tie chromozómy, ktoré majú najväčšiu silu, potom by sme s určitou pravdepodobnosťou podstatne ohraničili oblasť D , na ktorej hľadáme optimálne riešenie.
- Evolučný algoritmus vyberá chromozómy do procesu reprodukcie *kvázináhodne*.
- Obrazne môžeme povedať, že v rámci evolučného algoritmu nevystupuje *deus ex machina*, ktorý vopred pozná výsledok evolúcie populácie chromozómov a môže zasahovať výberom chromozómov do procesu reprodukcie.

Základné pojmy genetického algoritmu - populácia a sila chromozómov

- Genetický algoritmus (GA) bol vynájdenný v polovici 70-tých rokov Hollandom. V súčasnosti patrí medzi najpopulárnejšie evolučné optimalizačné algoritmy.
- Základné princípy genetického algoritmu sú určitou špecifikáciou všeobecných princíпов evolúcie a jej algoritmizácie.

Chromozóm α je binárny vektor fixnej dĺžky k

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k) \in \{0,1\}^k$$

Populácia P je multimnožina obsahujúca chromozómy α

$$P = \{\alpha_1, \alpha_2, \dots, \alpha_p\}$$

Účelová funkcia f je definovaná nad množinou binárnych vektorov dĺžky k

$$f : \{0,1\}^k \rightarrow R$$

Evolučná úloha je nájsť globálne minimum tejto funkcie nad množinou $\{0,1\}^k$

$$\alpha_{opt} = \arg \min_{\alpha \in \{0,1\}^k} f(\alpha)$$

Funkcia f "reprezentuje prostredie" v ktorom existujú chromozómy populácie.

Biologická terminológia: chromozóm α reprezentuje *genotyp* organizmu, zatiaľ čo funkčná hodnota $f(\alpha)$ reprezentuje jeho *fenotyp*. Mierou *úspešnosti* chromozómu je jeho funkčná hodnota.

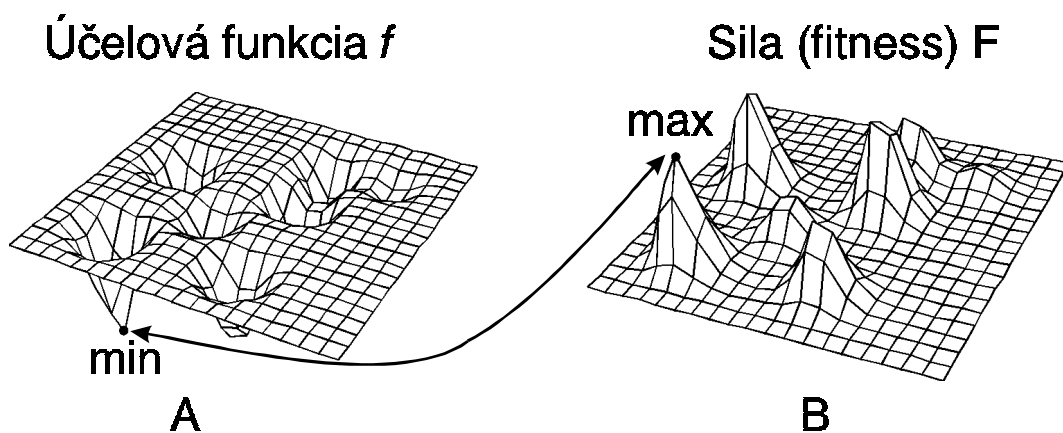
Chromozóm je tým **úspešnejší** (má **väčšiu silu**), čím je jeho funkčná hodnota (genotyp) menšia

$$\forall \alpha_1, \alpha_2 \in P: f(\alpha_1) \leq f(\alpha_2) \Rightarrow F(\alpha_1) \geq F(\alpha_2) \geq 0$$

Renormalizovaná sila

$$F'(\alpha) = \frac{F(\alpha)}{\sum_{\alpha' \in P} F(\alpha')}$$

$$\sum_{\alpha \in P} F'(\alpha) = 1$$



Akým konkrétnym spôsobom realizovať zobrazenie hodnôt účelovej funkcie na silu?

Prvá metóda je jednoduché lineárne zobrazenie funkčných hodnôt na silu tak, že maximálnej (minimálnej) funkčnej hodnote je priradená minimálna (maximálna) sila

$$F(\alpha) = \frac{F_{max} - F_{min}}{f_{min} - f_{max}} f(\alpha) + \frac{f_{min} F_{min} - f_{max} F_{max}}{f_{min} - f_{max}}$$

$$f_{min} \leq f(\alpha) \leq f_{max}$$

$$F_{min} = \varepsilon \leq F(\alpha) \leq F_{max} = 1$$

Finálna formula pre lineárnu transformáciu funkčných hodnôt na silu

$$F(\alpha) = \frac{1}{f_{min} - f_{max}} \left[(1 - \varepsilon) f(\alpha) + f_{min} \varepsilon - f_{max} \right]$$

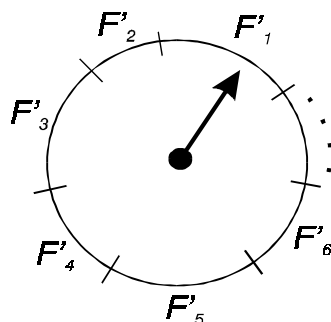
Druhá metóda je založená na usporiadaní funkčných hodnôt do rastúcej postupnosti. Nech $Q=(q_1,q_2,\dots,q_{|P|})$ je taká permutácia chromozómov z populácie P , ktorá usporiada hodnoty účelovej funkcie do neklesajúcej postupnosti

$$f(\alpha_{q_1}) \leq f(\alpha_{q_2}) \leq \dots \leq f(\alpha_{q_{|P|}})$$

Potom chromozómu α_{q_1} ($\alpha_{q_{|P|}}$) priradíme maximálnu (minimálnu) silu $F_{\max}=1$ ($F_{\min}=\varepsilon$)

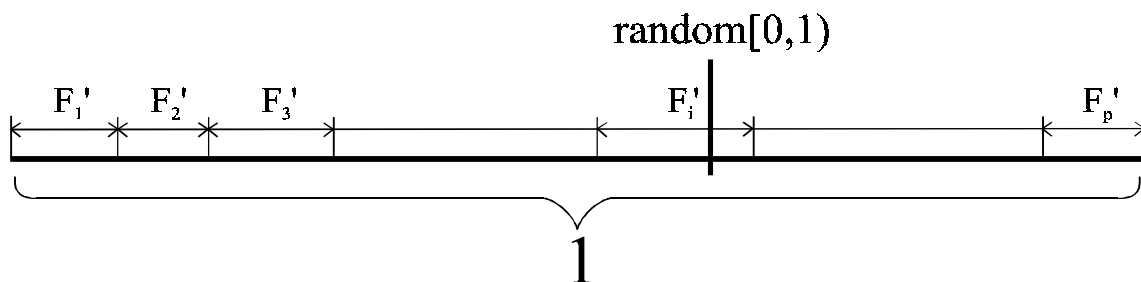
$$F(\alpha_{q_i}) = \frac{1}{1-|P|} [(1-\varepsilon)i + \varepsilon - |P|] \quad (\text{pre } i = 1, 2, \dots, |P|)$$

Pomocou renormalizovaných síl môžeme jednoducho realizovať kvázináhodnosť výberu chromozómov (čo patrí medzi základné imperatívy evolučných algoritmov) pomocou pojmu **ruleta**



Chromozóm s indexom i je vybraný (kvázináhodne), ak pre náhodne generované číslo z intervalu $[0,1)$ s rovnomernou distribúciou pravdepodobnosti platí

$$\sum_{k=1}^{i-1} F'_k \leq z < \sum_{k=1}^i F'_k$$



Jednoduchá implementácia rulety

```
function Roulette_Wheel:integer;  
begin aux:=random;  
    bound_lower:=0; bound_upper:=F'1;  
    i:=0; criterion:=false;  
    while (i<|P|) and (not criterion) do  
    begin i:=i+1;  
        criterion:=(bound_lower≤aux)  
        and (aux<bound_upper);  
        bound_lower:=bound_upper;  
        bound_upper:=bound_upper+F'i+1;  
    end;  
    Roulette_wheel:=i;  
end;
```

Poznámka: Korektná implementácia algoritmu je založená na binárnom hľadaní.

Kríženie chromozómov

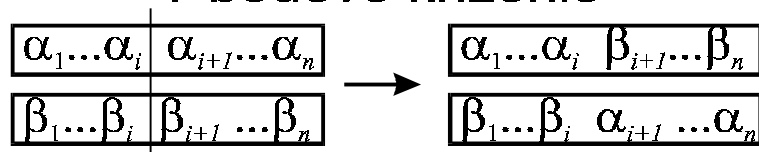
Pre chromozómy z populácie P sú definované dve základné operácie:

1. operácia mutácie a
2. operácia kríženia.

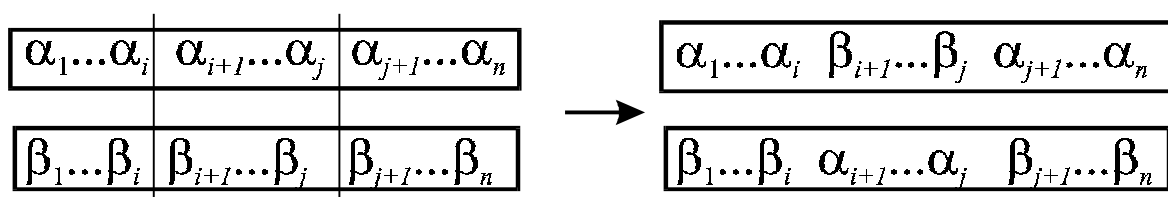
Majme dva chromozómy $\alpha, \beta \in \{0,1\}^k$, operácia kríženia bude formálne interpretovaná ako operátor - zobrazenie, ktorý tejto dvojici chromozómov priradí dva nové chromozómy o rovnakej dĺžke ako pôvodné

$$(\alpha', \beta') = O_{cross}(\alpha, \beta)$$

1-bodové kríženie



2-bodové kríženie



Pseudopascalovská implementácia kríženia

```
procedure Crossover;  
begin cross_point:=1+random(k-1);  
      for i:=1 to cross_point do  
      begin  $\alpha'_i := \alpha_i$ ;  $\beta'_i := \beta_i$  end;  
      for i:=cross_point+1 to k do  
      begin  $\alpha'_i := \beta_i$ ;  $\beta'_i := \alpha_i$  end;  
end;
```

Operátor reprodukcie obsahuje kríženie a mutáciu

$$(\alpha', \beta') = O_{repro}(\alpha, \beta)$$

$$(\tilde{\alpha}, \tilde{\beta}) = O_{cross}(\alpha, \beta)$$

$$\alpha' = O_{mut}(\tilde{\alpha}) \quad \text{a} \quad \beta' = O_{mut}(\tilde{\beta})$$

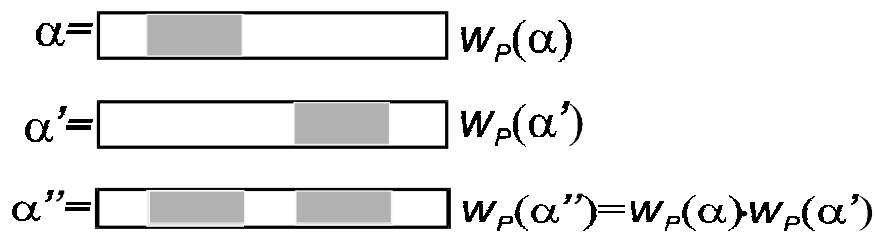
Význam kríženia

- Kríženie patrí medzi základnú črtu genetického algoritmu, ak ho odstránime, potom sa nám genetický algoritmus zredukuje na niečo blízke horolezeckému algoritmu.
- Používanie kríženia odlišuje genetický algoritmus od ostatných stochastických evolučných algoritmov, ktoré v rámci populácie objektov - riešení - chromozómov tiež používajú reprodukciu založenú na ich sile a aplikujú mutácie na jednotlivé objekty.
- Medzi pracovníkmi zaoberajúcimi sa aplikáciami genetického algoritmu panuje presvedčenie, že efektívnosť týchto algoritmov je vo všeobecnosti menšia ako efektívnosť genetického algoritmu hlavne z toho dôvodu, že nepoužívajú kríženie.

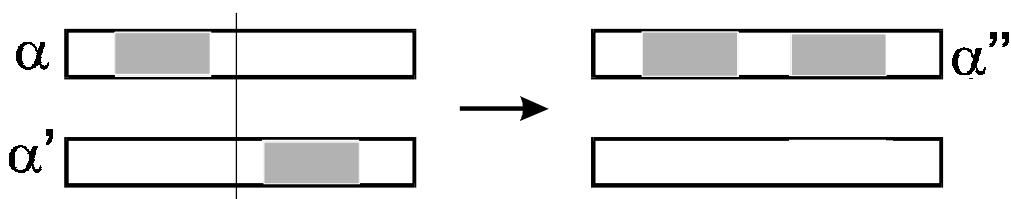
Význam kríženia (pohlavnej reprodukcie)

- Pohlavná reprodukcia je v prírode všeobecne využívaná hlavne medzi zložitejšími druhmi našej planéty. Pohlavná reprodukcia je neobyčajne zložitý proces produkcie potomkov.
- Druh využívajúci pohlavnú reprodukciu musí byť podstatne zložitejší ako druh, ktorý ju nevyužíva.
- Pretože pohlavná reprodukcia jednoznačne zvíťazila v aréne prirodzeného výberu, musia existovať vážne dôvody, prečo tento komplikovaný spôsob reprodukcie patrí medzi základné črty, ktoré odlišujú nižšie druhy od vyšších druhov.

Jeden z hlavných dôvodov (podporený dôkladnou matematickou analýzou) výhodnosti kríženia je fakt, že pohlavná reprodukcia umožňuje rýchlu výmenu výhodných vlastností, ktoré podstatne zvyšujú silu jedincov - potomkov, čo je len veľmi obtiažne dosiahnuteľné len pomocou mutácií.

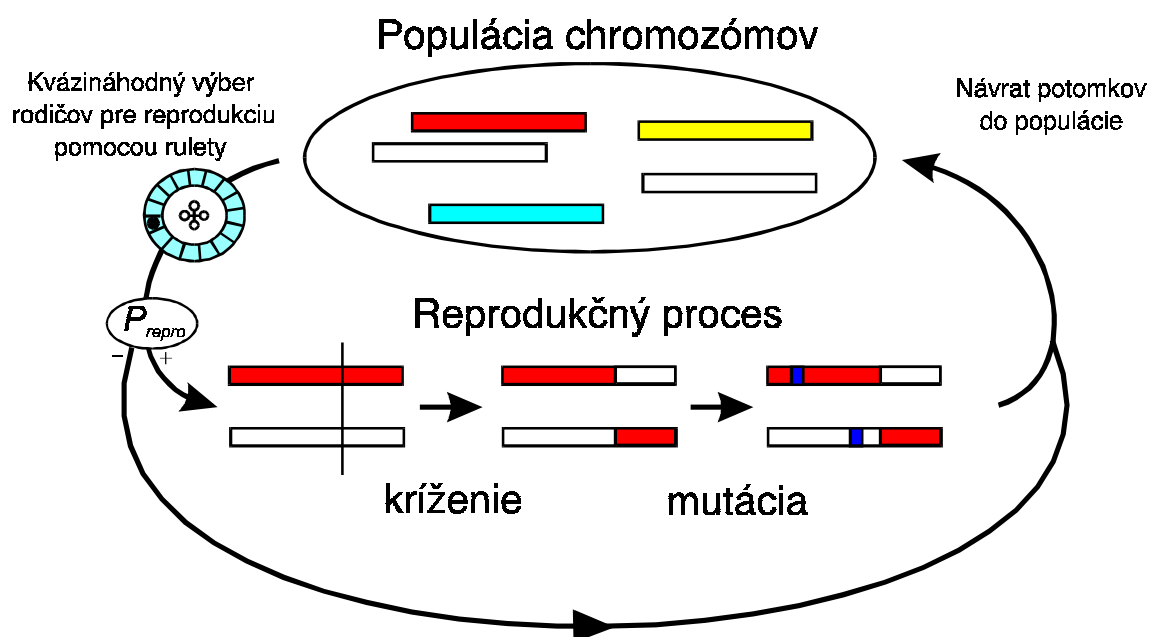


A



B

Schematické znázornenie genetického algoritmu



Implementácia genetického algoritmu

```
procedure Genetic_Algorithm;  
begin t:=0; stop_criterion:=false;  
      P:=randomly generated population;  
      while (t<tmax) and  
            (not stop_criterion) do  
      begin t:=t+1;  
            Q:=∅;  
            each chromosome is evaluated  
            by fitness;  
            while |Q|<|P| do  
            begin  $\alpha_1 := O_{\text{select}}(P)$  ;  
                    $\alpha_2 := O_{\text{select}}(P)$  ;  
                   if random<Prepro then  
                       R( $\alpha_1, \alpha_2, \alpha'_1, \alpha'_2$ ) else  
                   begin  $\alpha'_1 := \alpha_1$  ;  
                            $\alpha'_2 := \alpha_2$   
                   end ;  
                   Q:= Q ∪ { $\alpha'_1, \alpha'_2$ } ;  
            end ;  
            P:=Q;  
            if convergence criteria are  
            fulfilled then  
            stop_criterion:=true;  
      end ;  
       $\alpha_{\text{opt}} :=$ best chromosome of P;  
end ;
```

Problém zastavenia genetického algoritmu

1. Prepísaný počet evolučných epoch (premenná t).
2. Výpočet analógu pravdepodobnostného vektora $\mathbf{w}=(w_1, w_2, \dots, w_k)$ z horolezeckého algoritmu s učením. Nech populácia P v i -tom kroku (generácii) obsahuje chromozómy $P = (\alpha_1, \alpha_2, \dots, \alpha_p)$, kde $\alpha_i = (\alpha_1^{(i)}, \alpha_2^{(i)}, \dots, \alpha_k^{(i)})$. Komponenty pravdepodobnostného vektora sú určené takto

$$w_j = \frac{1}{p} \sum_{i=1}^p \alpha_j^{(i)} \quad (\text{pre } j = 1, 2, \dots, k)$$

Parameter usporiadania definovaný pre takto určený pravdepodobnostný vektor má tvar

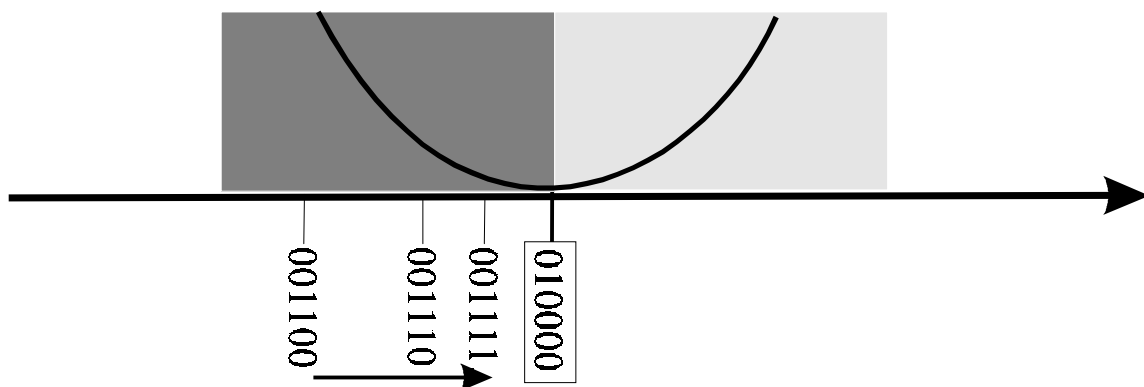
$$\chi(\mathbf{w}) = \frac{4}{k} \sum_{j=1}^k (w_j - 0.5)^2$$

Tento parameter s funkčnými hodnotami z intervalu $[0,1]$ má tú vlastnosť, že pre náhodne generovanú populáciu P (prvé iteračné kroky genetického algoritmu) sa blíži nule (je malé kladné číslo). Tak ako evolúcia populácie pokračuje, hodnota parametru rastie a asymptoticky sa blíži jednotke.

Hammingova bariéra

- Dva chromozómy $\alpha=(00100\dots0)$ a $\beta=(000111\dots1)$ reprezentujú dve susedné celé čísla
- Tieto dva chromozómy aj keď reprezentujú dve "susedné" čísla, ich Hammingovská vzdialenosť je veľká.
- Predpokladajme, že optimálne riešenie odpovedá binárnemu reťazcu $\alpha=(00100\dots0)$ a že populácia obsahuje chromozómy, ktoré sú blízke alebo totožné s binárnym reťazcom $\beta=(000111\dots1)$.
- Chromozómy populácie $(00011**\dots*)$ sa vyvíjajú smerom k optimálnemu riešeniu tak, že po určitom počte iteračných krokov populácie bude obsahovať v prevažnej miere chromozómy $\beta=(000111\dots1)$.
- Pravdepodobnosť zmeny $(00011**\dots*)$ mutáciou na $(00100**\dots*)$ je veľmi malá.

- Evolúcia populácie chromozómov, ktorá bola inicializovaná chromozómami typu $(00011^{**}...^{*})$, má len veľmi malú šancu byť smerovaná do výslednej populácie, ktorá by obsahovala optimálne riešenie $\alpha=(00100...0)$, a s najväčšou pravdepodobnosťou skončí u riešenia $\beta=(000111...1)$.
- Táto skutočnosť sa v genetickom algoritme nazýva Hammingova bariéra a predstavuje vážne ohrozenie aplikovateľnosti genetického algoritmu za predpokladu, že sa používa štandardné kódovanie binárnych reťazcov.



Ako sa vyhnúť Hammingovej bariére v genetickom algoritme?

Prvý prístup je založený na použití operátora inverzie, ktorý zmení binárny reťazec $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$ na iný binárny reťazec $\beta = (\beta_1, \beta_2, \dots, \beta_k)$

$$\beta = O_{inv}(\alpha)$$

Pre náhodne vygenerované celé číslo $1 < a < k$ (bod inverzie), komponenty binárneho vektora pre indexy $i > a$ sú zamenené za ich komplementy

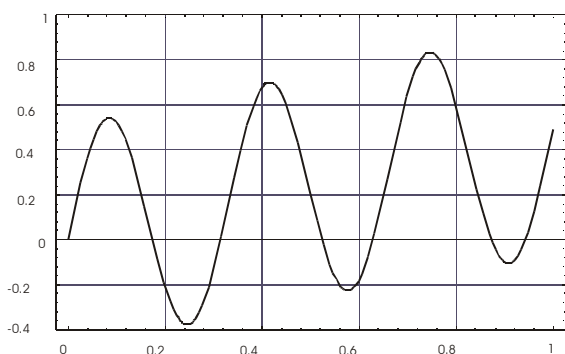
$$\beta_i = \begin{cases} \alpha_i & (\text{pre } i \in [1, a]) \\ 1 - \alpha_i & (\text{pre } i \in [a + 1, k]) \end{cases}$$

Stochastický operátor inverzie odstraňuje vznikajúce Hammingove bariéry.

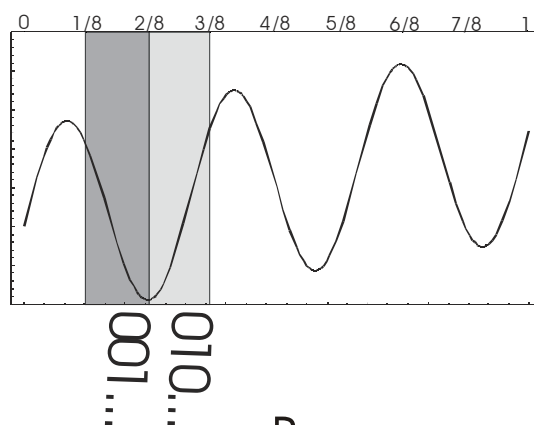
Ilustračný príklad

Základné pojmy z predchádzajúcej kapitoly budú ilustrované jednoduchou implementáciou genetického algoritmu s nasledujúcou účelovou funkciou

$$f(x) = e^{-x^2/100} \times \sin(10x) \times \cos(9x) \quad \forall x \in [0,1]$$



A



B

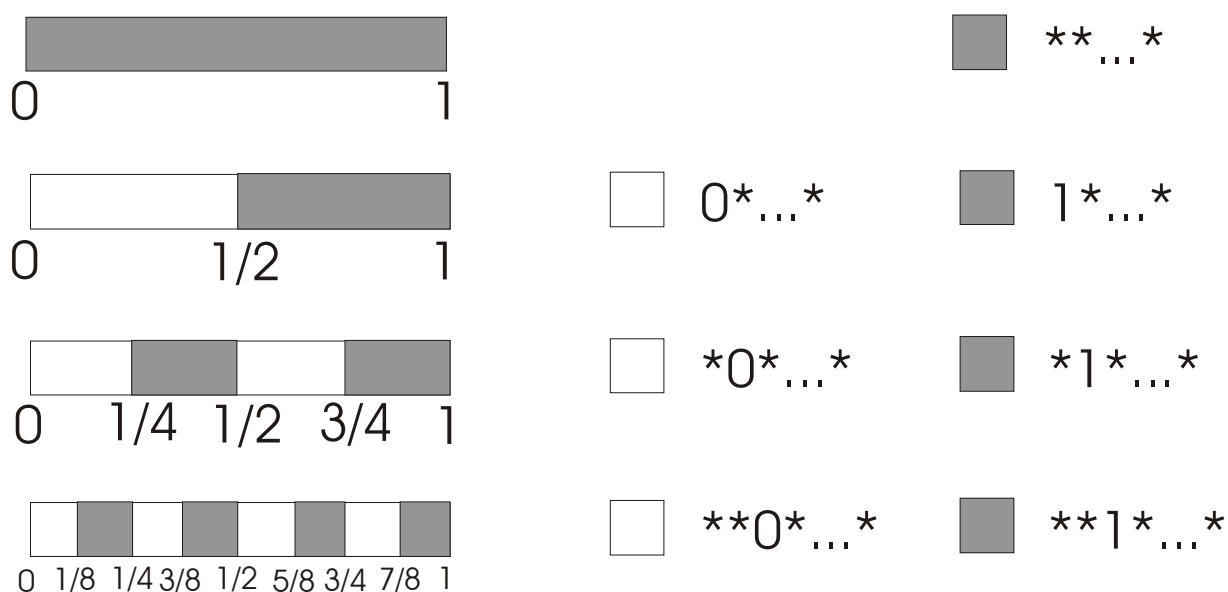
- (A) Funkcia má tri minimá: (1) $x_1=0.245322$, $f(x_1)=-0.377681$, globálne minimum, (2) $x_2=0.576377$, $f(x_2)=-0.226259$, (3) $x_3=0.907692$, $f(x_3)=-0.104824$.
- (B) Rozdelenie intervalu $[0,1]$ na osem podintervalov rovnakej dĺžky. Každému tomuto podintervalu môže byť priradený binárny kód.

No.	Schéma	Rozdelenie intervalu [0,1]							
		1/8	2/8	3/8	4/8	5/8	6/8	7/8	
1	***...	█	█	█	█	█	█	█	█
2	**0...	█	█	█	█	█	█	█	█
3	**1...	█	█	█	█	█	█	█	█
4	*0*	█	█	█	█	█	█	█	█
5	*00...	█	█	█	█	█	█	█	█
6	*01...	█	█	█	█	█	█	█	█
7	*1*	█	█	█	█	█	█	█	█
8	*10...	█	█	█	█	█	█	█	█
9	*11...	█	█	█	█	█	█	█	█
10	0**...	█	█	█	█	█	█	█	█
11	0*0...	█	█	█	█	█	█	█	█
12	0*1...	█	█	█	█	█	█	█	█
13	00*	█	█	█	█	█	█	█	█
14	000...	█	█	█	█	█	█	█	█
15	001...	█	█	█	█	█	█	█	█
16	01*	█	█	█	█	█	█	█	█
17	010...	█	█	█	█	█	█	█	█
18	011...	█	█	█	█	█	█	█	█
19	1**...	█	█	█	█	█	█	█	█
20	1*0...	█	█	█	█	█	█	█	█
21	1*1...	█	█	█	█	█	█	█	█
22	10*	█	█	█	█	█	█	█	█
23	100...	█	█	█	█	█	█	█	█
24	101...	█	█	█	█	█	█	█	█
25	11*	█	█	█	█	█	█	█	█
26	110...	█	█	█	█	█	█	█	█
27	111...	█	█	█	█	█	█	█	█

Obrázok 4.14. Delenie intervalu [0,1] podľa schém, ktorých prvé tri znaky sú špecifikované. Napríklad, schéma č. 20 tvaru 1*0...určuje reálnu premennú z prieniku intervalov $[1/2, 5/8] \cup [6/8, 7/8]$.

Globálne minimum funkcie je približne vyjadrené v binárnej reprezentácii chromozómami, ktoré obsahujú schémy $001^*...^*$ (tieto schémy odpovedajú reálnym číslam z intervalu $[1/8, 1/4]$) alebo $010^*...^*$ (tieto schémy odpovedajú reálnym číslam z intervalu $[1/4, 3/8]$).

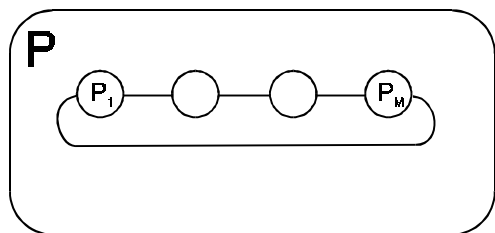
Pretože globálne minimum je v bode $x=0.245.. < 1/4$, toto minimum sa vyskytuje v blízkosti hornej hranici prvého intervalu, čiže binárny reťazec odpovedajúci riešeniu je približne vyjadrený ako $001111... .$



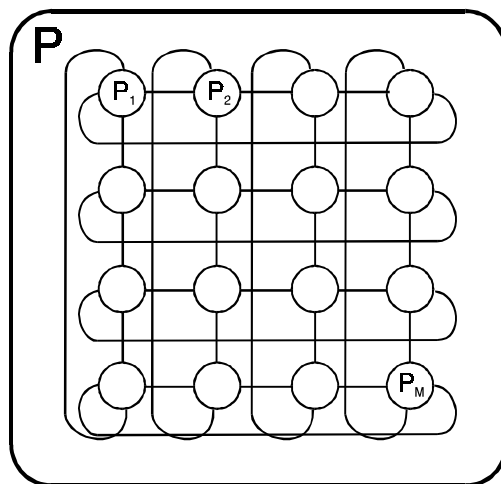
Druhý prístup je založený na použití použítie Grayovho kódu pre binárnu reprezentáciu reálnych premenných.

Paralelný genetický algoritmus

Genetický algoritmus je sekvenčný algoritmus. V literatúre existuje mnoho verzií tzv. **paralelného genetického algoritmu** (PGA) v ktorých je populácia rozdelená na podpopulácie, pričom evolúcia prebieha "nezávisle" nad týmito podpopuláciami, t.j. medzi nimi dochádza k občasnej stochastickej interakcii v rámci ktorej si vymenia chromozómy. Obvykle, tieto paralelné verzie sú prezentované (a tiež aj aplikované) ako spôsob diverzifikácie a intenzifikácie genetického algoritmu.



A

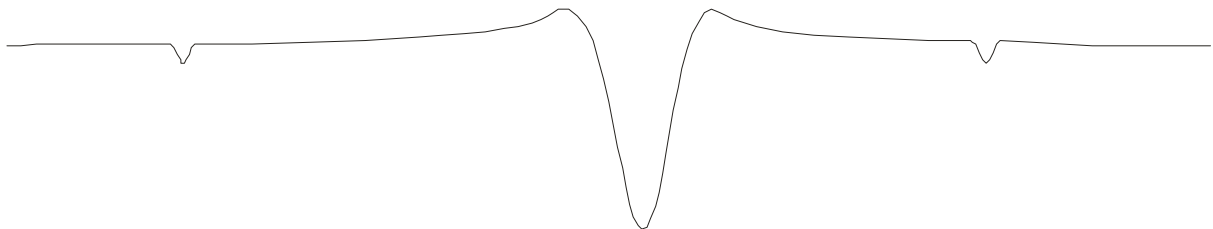


B

$$P = P_1 \cup P_2 \cup \dots \cup P_M \text{ a } P_i \cap P_j = \emptyset, \text{ pre } i \neq j.$$

"Messy" chromozómy (M-chromozómy)

- V r. 1989 Goldberg so spolupracovníkmi navrhli „messy“ genetický algoritmus (MGA) na prekonanie určitých potiaží evolučných algoritmov s deceptívnymi účelovými funkciami.
- *Deceptívna účelová funkcia* je multimodálna funkcia s mnohými minimami ale len s jedným globálnym minimom, ktoré je oddelené od ostatných miním bariérou.



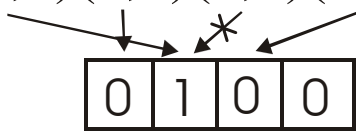
M-chromozóm (messy chromozóm) – premenlivá dĺžka, jeho komponenty sú špecifikované súčasne tak indexom ako aj hodnotou. Táto flexibilita m-chromozómov obvykle podstatne zvyšuje efektivitu a robustnosť MGA.

M-chromozóm premennej dĺžky je definovaný takto: Nech $Q = \{1, 2, \dots, k\} \times \{0, 1\}$ je množina obsahujúca usporiadané dvojice (μ, ν) , kde $\mu \in \{1, 2, \dots, k\}$ je index a $\nu \in \{0, 1\}$ je binárna hodnota. M-chromozóm dĺžky l je definovaný takto

$$c = ((\mu_1, \nu_1), (\mu_2, \nu_2), \dots, (\mu_l, \nu_l)) \in Q^l$$

$((2, 1)(1, 0)(2, 0)(3, 0))$

m-chromozóm



dekódovaný chromozóm

(1100)

templát

Dekódovanie m-chromozómu pomocou **templátu**. Poloha špecifikovaná indexom je ohodnotená binárnou hodnotou, ktorá doprevádza špecifikovaný index. Keď daná poloha už bola špecifikovaná v predchádzajúcom kroku dekódovania, potom sa aktuálna špecifikácia ignoruje. V prípade, že niektorá poloha nie je špecifikovaná m-chromozómom, potom sa pre špecifikáciu použije príslušná hodnota z templátu.

Mutácia m-chromozómu

M-chromozóm χ je transformovaný na iný m-chromozóm χ' operáciou mutácie

$$\chi' = O_{mut}(\chi)$$

$$\begin{aligned}\chi &= ((\mu_1, \nu_1), (\mu_2, \nu_2), \dots, (\mu_l, \nu_l)) \\ \chi' &= ((\mu'_1, \nu'_1), (\mu'_2, \nu'_2), \dots, (\mu'_l, \nu'_l)).\end{aligned}$$

Stochastický operátor mutácie O_{mut}

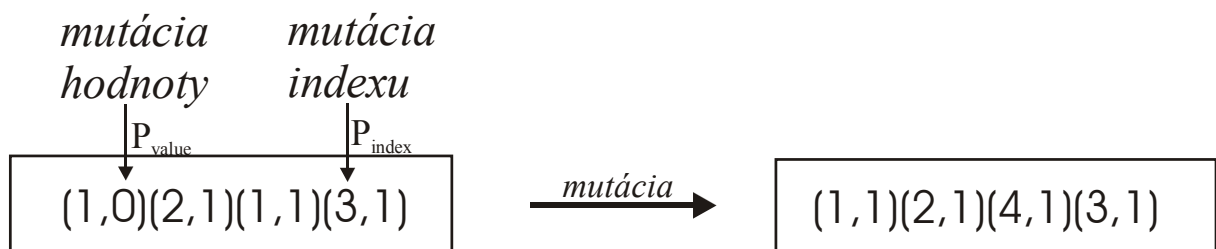
$$\mu'_i = \begin{cases} \hat{\mu}_i & (\text{ak } random < P_{index}) \\ \mu_i & (\text{ostatné prípady}) \end{cases}$$

Pravdepodobnosť P_{index} určuje stochastičnosť zmeny indexovej časti m-chromozómu.

Podobným spôsobom sa mení aj binárna časť m-chromozómu

$$v'_i = \begin{cases} 1 - v_i & (\text{ak } random < P_{value}) \\ v_i & (\text{ostatné prípady}) \end{cases}$$

Pravdepodobnosť P_{value} určuje stochastičnosť zmeny binárnej časti m-chromozómu.



Mutácia m-chromozómu má dve časti, mutáciu indexu a mutáciu binárnej hodnoty. Každá takáto mutácia sa vykonáva s pravdepodobnosťou P_{index} resp. P_{value} .

Kríženie m-chromozómov

Operácia kríženia dvoch rodičovských m-chromozómov

$$\chi = \left((\mu_1, \nu_1), (\mu_2, \nu_2), \dots, (\mu_{l_\chi}, \nu_{l_\chi}) \right)$$

$$\kappa = \left((\xi_1, \zeta_1), (\xi_2, \zeta_2), \dots, (\xi_{l_\kappa}, \zeta_{l_\kappa}) \right)$$

poskytuje dva nové m-chromozómy - potomkov

$$\chi' = \left((\mu'_1, \nu'_1), (\mu'_2, \nu'_2), \dots, (\mu'_{l'_\chi}, \nu'_{l'_\chi}) \right)$$

$$\kappa' = \left((\xi'_1, \zeta'_1), (\xi'_2, \zeta'_2), \dots, (\xi'_{l'_\kappa}, \zeta'_{l'_\kappa}) \right)$$

$$(\chi', \kappa') = O_{cross}(\chi, \kappa)$$



Operácia kríženia dvoch m-chromozómov, tieto si vymenia svoje časti nasledujúce za náhodne generovaným bodom kríženia. Operácia kríženia mení, vo všeobecnosti, dĺžku m-chromozómov

```

time:=0;
stop_criterion:=false;
P:={randomly generated population of
    m-chromosomes};
t:=randomly generated binary vector-template;
while (time<timemax) and (non stop_criterion) do
begin time:=time+1;
    Q:=∅;
    each m-chromosome is evaluated by
        fitness;
    while |Q|<|P| do
    begin select by roulette wheel two
        m-chromosomes  $\chi, \kappa \in P$ ;
        if random<Prepro then
        Reproduction( $\chi, \kappa, \chi', \kappa'$ ) else
        begin  $\chi' := \chi$ ;  $\kappa' := \kappa$  end;
        Q:=Q∪{ $\chi', \kappa'$ };
    end;
    P:=Q; t:=arg min $\chi \in P$  f(Gt( $\chi$ ));
    if convergence criteria are
        fulfilled then
        stop_criterion:=true;
    end;
 $\alpha_{opt} := t$ ;
end;

```

- Základný problém, ktorý musí byť riešený pri použití m -chromozómov v evolučných algoritmoch je konštrukcia templátu t potrebného pre dekódovanie m -chromozómov do tvaru binárnych vektorov.
- Pri inicializácii evolučného algoritmu sa templát t náhodne generuje ako binárny vektor, zatiaľ čo v neskorších fázach algoritmu je templát určený ako binárny vektor odpovedajúci najlepšiemu m -chromozómu z predchádzajúcej iterácie.
- Na záver algoritmu sa templát použije ako najlepšie výstupné riešenie. Po týchto jednoduchých úvahách môžeme pristúpiť k modifikácii genetického algoritmu 4.5 pre m -chromozómy (pozri algoritmus 4.9).