

# 4. kapitola

## Genetický algoritmus (GA)

### 4.1 Evolúcia v živej prírode

Genetický algoritmus patrí medzi základné stochastické optimalizačné algoritmy s výraznými evolučnými črtami. V súčasnosti je najčastejšie používaným evolučným optimalizačným algoritmom, so širokou paletou aplikácií od optimalizácie vysoko multimodálnych funkcií, cez kombinatoriálne a grafovo-teoretické problémy, až po aplikácie nazývané "umelý život".

V predchádzajúcej prednáške bol popísaný tzv. slepý algoritmus stochastickej optimalizácie funkcií. Aj keď je tento algoritmus extrémne jednoduchý, jeho hlavné ohraničenie spočíva v "slepom" náhodnom generovaní nového riešenia bez vzťahu k optimalizovanej funkcii. Je prakticky nepoužiteľný, čas potrebný pre nájdenie minima bude obrovský už pre jednoduché optimalizačné problémy. Metódy numerickej matematiky obchádzajú tento problém tak, že buď využívajú poznatky o tvare funkcie v danom bode (napr. gradientové metódy), alebo pomocou predchádzajúcich riešení sú schopné indikovať s veľkou efektívnosťou nádejný smer minimalizácie funkcie (napr. simplexová metóda). Ukazuje sa, že na základe analógie s evolučnými procesmi prebiehajúcimi v biologických systémoch existuje alternatívna možnosť, ako usmerniť náhodné generovanie bodov k hodnotám blízkym optimálnym. Práve táto analógia sa stala základom genetického algoritmu, ktorý vylepšuje čisto stochastický slepý algoritmus tak, že poskytuje v reálnom čase optimálne riešenie.

Darwinova teória evolúcie [1] sa zakladá na téze prirodzeného výberu, podľa ktorej prežívajú len najlepšie prispôbení jedinci populácie. Reprodukciou dvoch silných jedincov dostávame potomkov, ktorí s vysokou pravdepodobnosťou budú dobre prispôbení k úspešnému prežitiu. Pri podrobnej analýze (hlavne matematickej) sa ukazuje, že samotné pôsobenie reprodukcie nie je dostatočne efektívne na vznik dobre prispôbených jedincov s novými vlastnosťami, ktoré významne uľahčujú prežitie. Do evolúcie živej hmoty je nutné zapojiť aj tzv. mutácie. Tieto náhodným spôsobom ovplyvňujú (kladne alebo záporne) genetický materiál populácie jedincov.

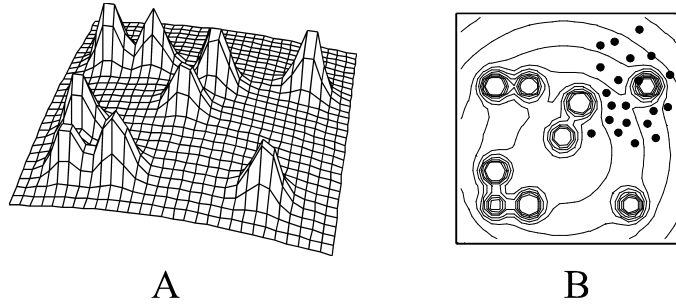
*Biologická evolúcia* je progresívna zmena obsahu genetickej informácie (genotypu) populácie v priebehu mnohých generácií. Prv ako pristúpime k jej špecifikácii, zavedieme pojem sily (fitness), ktorý hrá kľúčovú úlohu v našich úvahách o genetickom algoritme. V biológii je *sila* definovaná ako relatívna schopnosť prežitia a reprodukcie genotypu v danom prostredí. Podobne sa chápe aj v umelom živote, je to kladné číslo priradené genetickej informácii reprezentujúcej organizmus (obvykle vyjadrenej pomocou bitového reťazca), ktoré reprezentuje jeho relatívnu úspešnosť plniť si v danom prostredí svoje úlohy (napr. zber potravy) a vstupovať do reprodukcie, t.j. tvoriť nové organizmy.

Biologická evolúcia obsahuje dve základné zložky:

(1) *Prirodzený výber* (Darwin) - organizmy s väčšou silou produkujú viac potomkov, ktorí sa dožijú dospelosti a ďalej sa reprodukovujú, ako organizmy s malou silou.

(2) *Genetický drift* - náhodné udalosti (napr. mutácie alebo náhodná smrť organizmu s veľkou silou prv ako stihol vyprodukovať potomkov) môžu podstatne ovplyvniť evolučné procesy v malých populáciách.

Pojem "*povrch sily*" (fitness' surface/landscape) [2] podstatne uľahčuje heuristickú interpretáciu selekcie v populácii v priebehu evolúcie. Základná idea povrchu sily spočíva v grafickej reprezentácii sily vzhľadom k zloženiu genotypu organizmov v danom prostredí (pozri obr. 4.1A). Populácia potom môže byť znázornená ako "oblak" bodov na povrchu (pozri obr. 4.1B), každý bod je priradený organizmu z populácie. V priebehu mnohých generácií prirodzený výber spôsobuje pohyb populácie na povrchu do oblastí s väčšou hodnotou sily. Ak je dominantná selekcia, potom sa tento pohyb uskutočňuje v smere gradientu, v ktorom sila najrýchlejšie rastie. V opačnom prípade, ak je dominantná náhodnosť v evolúcii (napr. mutácie), pohyb bodov na povrchu sily má stochastický charakter.



**Obrázok 4.1.** Znáoznenie 3-rozmerného povrchu sily (A). Výška jednotlivých jeho vrcholov odpovedá hodnote sily genetickej informácie (genotypu) organizmu z danej populácie. Kontúrový diagram (B) je priradený povrchu sily, body reprezentujú organizmy v populácii.

7

Vo vyššom stupni abstrakcie [3] môže byť pojem biologický jedinec nahradený pojmom chromozóm, ktorý reprezentuje lineárne usporiadaným spôsobom informačný obsah jedinca (genotyp). Potom hovoríme o populácii chromozómov, ktoré sa reprodukujú s pravdepodobnosťou úmernou ich sile, pričom integrálnou súčasťou tejto reprodukcie sú mutácie. Mutácie zanášajú do chromozómov novú informáciu, ktorá môže zvýšiť silu chromozómov vznikajúcich reprodukciou pôvodných chromozómov. Nové chromozómy vytesňujú z populácie chromozómy s malou silou. Tento základný reprodukčný cyklus sa v danej populácii chromozómov neustále opakuje. Po určitom čase sa teda v populácii s vysokou pravdepodobnosťou objavia chromozómy s novými vlastnosťami, ktoré podstatne zvyšujú ich silu a vytesňujú pôvodné chromozómy bez týchto vlastností.

Pristúpme k formalizácii základných pojmov evolúcie tak, ako formulované v predchádzajúcej časti tejto kapitoly. Jedinci populácie sú reprezentovaní chromozómami, čo sú lineárne reťazce symbolov. Potom populácia  $P$  je množina týchto chromozómov

$$P = \{\alpha_1, \alpha_2, \dots, \alpha_p\} \subseteq \{a, b, \dots\}^k \quad (4.1)$$

Populácia  $P$  obsahuje  $p$  chromozómov, ktoré sú realizované ako reťazce dĺžky  $k$  zložené zo znakov  $a, b, \dots$ . Obvykle, tieto znaky sa rovnajú 0 a 1, potom chromozómy sú binárne reťazce,  $P \subseteq \{0,1\}^k$  (táto binárna konvencia bude použitá v celej tejto práci). Každý chromozóm  $\alpha \in P$  je ohodnotený silou, ktorá je interpretovaná ako zobrazenie chromozómov na kladné reálne čísla

$$F: P \rightarrow R_+ \quad (4.2)$$

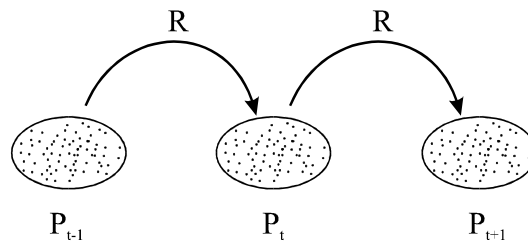
Proces reprodukcie chromozómov začína tým, že z populácie sú vybrané dva chromozómy kvázi náhodne v závislosti na ich sile (chromozómy s väčšou silou majú väčšiu pravdepodobnosť byť vybrané). Pod reprodukciou budeme rozumieť taký proces, v ktorom sa dva pôvodné chromozómy  $\alpha_1$  a  $\alpha_2$  reprodukujú (tvoria potomkov) na dva nové chromozómy  $\alpha'_1$  a  $\alpha'_2$

$$(\alpha'_1, \alpha'_2) = O_{repro}(\alpha_1, \alpha_2) \quad (4.3)$$

Konkrétny tvar tohto operátora reprodukcie nebudeme zatiaľ bližšie diskutovať. Ako bude ukázané neskôr, obsahuje dve časti, a to *kríženie* a *mutáciu*. Navyše, tento operátor má stochastický charakter, operácie kríženia a mutácie sa vykonávajú len s určitou pravdepodobnosťou. Mutácia už bola špecifikovaná v kapitole 2.4, operáciu kríženia, ktorá znamená výmenu častí medzi dvoma chromozómami, budeme špecifikovať podrobnejšie neskôr.

Vo všeobecnej rovine môže byť evolúcia populácie chápaná ako zmena populácie realizujúca sa v čase po jednotlivých krokoch (generáciách). Nech  $P_t$  je populácia chromozómov v čase  $t$ , potom v nasledujúcom čase  $t+1$  populácia  $P_{t+1}$  je určená rekurentne pomocou "reprodukčného"  $R$ -operátora (pozri obr. 4.2)

$$P_{t+1} = R(P_t) \quad (4.4)$$



**Obrázok 4.2.** Evolúcia môže byť charakterizovaná ako postupná rekurentná zmena populácie, formálne  $P_{t+1}=R(P_t)$ . R-operátor popisuje rekurentnú zmenu populácie z jednej generácie na druhú. Obsahuje niekoľko komponent, akými sú výber chromozómov - rodičov, kríženie, mutáciu a návrat chromozómov - potomkov do populácie.

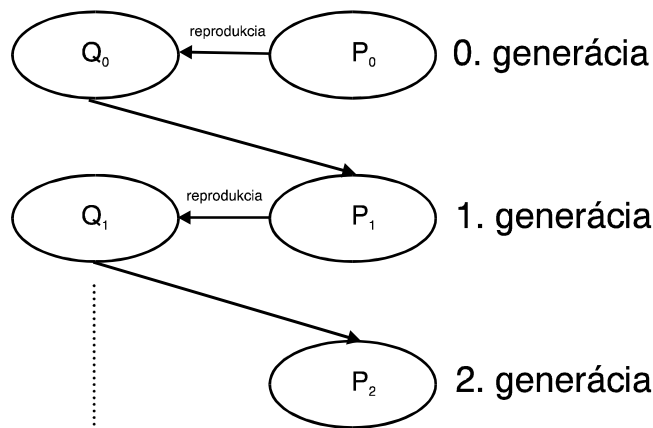
Prístup založený na všeobecnej formule (4.4) poskytuje veľmi všeobecný pohľad na evolúciu, avšak za cenu toho, že R-operátor nie je bližšie špecifikovaný. Našu pozornosť upriamime na jednoduchú realizáciu R-operátora tak, že obsahuje problém výberu chromozómov z populácie, reprodukciu (4.3) a návrat nových chromozómov do populácie. Dostaneme jednoduchý evolučný algoritmus, ktorý simuluje evolúciu chromozómov v populácii, pričom hnacou silou tejto evolúcie je Darwinovský prirodzený výber (pozri algoritmus 4.1 a obrázok 4.3).

```

procedure Evolution(input  $t_{max}$ ; output  $P_{fin}$ );
begin  $t:=0$ ;
       $P:=\{\text{randomly generated population of chromosomes}\}$ ;
      while  $t < t_{max}$  do
        begin  $t:=t+1$ ;  $Q:=\emptyset$ ;
          while  $|Q| < |P|$  do
            begin select quasirandomly two parental
              chromosomes  $\alpha_1, \alpha_2 \in P$  with greatest
              fitness;
              if  $\text{random} < P_{repro}$  then
                 $(\alpha'_1, \alpha'_2) = O_{repro}(\alpha_1, \alpha_2)$  else  $(\alpha'_1, \alpha'_2) = (\alpha_1, \alpha_2)$ ;
                 $Q := Q \cup \{\alpha'_1, \alpha'_2\}$ ;
              end;  $P := Q$ ;
            end;  $P_{fin} := P$ ;
        end;
end;

```

**Algoritmus 4.1.** Pseudopascalovská verzia algoritmu evolúcie v živej prírode, založenej na Darwinovskom prirodzenom výbere. Algoritmus je inicializovaný náhodným generovaním populácie  $P$ . Vonkajší while-cyklus sa opakuje  $t_{max}$  generácií (iterácií). Vo vnútornom while-cykle sa zostrojuje nová populácia tak, že sa kvázi náhodne vyberú dva chromozómy z populácie  $P$  (pravdepodobnosť ich výberu je úmerná ich sile) a aplikuje sa na ne reprodukčný operátor (4.3). Tento proces reprodukcie sa realizuje s pravdepodobnosťou  $P_{repro}$ , v opačnom prípade sa vybrané rodičovské chromozómy jednoducho okopírujú do nových chromozómov - potomkov. Takto vzniklé dva nové chromozómy sa uložia do novej populácie  $Q$ . Vnútorný while-cyklus sa opakuje tak dlho, až kardinalita novej populácie  $Q$  je rovná kardinalite starej populácie  $P$ . Populácia  $P$  sa potom nahradí novou populáciou  $Q$ .



**Obrázok 4.3.** Diagramatická vizualizácia evolučného algoritmu 4.1. Z počiatočnej populácie  $P_0$  aplikovaním reprodukčného procesu je generovaná nová populácia potomkov  $Q_0$ , ktorá do ďalšej generácie vstupuje ako obnovená populácia  $P_1$ . Tento proces sa iteračne opakuje  $t_{max}$ -krát.

Diskutujeme vzťah medzi horolezeckým algoritmom (algoritmus 4.3) a evolučným algoritmom (algoritmus 4.1). V čom sa tieto algoritmy odlišujú? Evolučný algoritmus sa zakladá na simultánnej optimalizácii celej populácie chromozómov, zatiaľ čo horolezecký algoritmus 3.3 optimalizuje náhodne len jeden objekt - chromozóm. Ak v procese reprodukcie chromozómov uvažujeme len mutácie, tak evolučný algoritmus je veľmi podobný horolezeckému algoritmu 3.3. 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. Pri väčšom množstve chromozómov v populácii sa najskôr takýto chromozóm nezúčastňuje reprodukčného procesu, no neskôršie môže nastať situácia, že tieto chromozómy sa ukážu vhodnými pre ďalšiu evolúciu (optimalizáciu) systému. V evolučnom algoritme je lokálne hľadanie optimálneho riešenia horolezeckého algoritmu nahradené efektívnejším spôsobom simultánnej optimalizácie celej populácie chromozómov, pričom jednotlivé chromozómy si v rámci reprodukcie pomocou kríženia vymieňajú informáciu.

Prečo pri prezentácii evolučných algoritmov neustále zdôrazňujeme, že do reprodukčného procesu *vyberáme chromozómy náhodne*? Ak by sme do reprodukčného procesu vybrali len tie chromozómy, ktoré majú najväčšiu silu, potom by sme s určitou pravdepodobnosťou podstatne ohraničili oblasť  $D$ , na ktorej hľadáme optimálne riešenie. Evolučný algoritmus by bol veľmi "oportunistický", za dočasne lepší výsledok by sme asi podstatne ohraničili celý proces evolúcie populácie chromozómov smerom k optimálnemu riešeniu a tým sa dostali k horšiemu konečnému riešeniu. Preto evolučný algoritmus vyberá chromozómy do procesu reprodukcie *kvázináhodne*. Obrazne môžeme povedať, s "licenciou básnika", že v rámci evolučného algoritmu nevystupuje *deus ex machina*, ktorý vopred pozná výsledok evolúcie populácie chromozómov a môže zasahovať výberom chromozómov do procesu reprodukcie. Keby sme boli schopní toto realizovať, potom nepotrebujeme evolučné algoritmy, pretože optimálne riešenie už poznáme. Všetky "častočne urýchľujúce" heuristiky sú nielen nedostatočné, no v konečnom dôsledku aj zavádzajúce, preto ich zabudovanie do evolučného algoritmu je vo väčšine prípadov veľmi problematické.

## 4.2 Základné pojmy genetického algoritmu - populácia a sila chromozómov

Genetický algoritmus (GA) bol vynájdený v polovici 70-tých rokov Hollandom [3]. V súčasnosti patrí medzi najpopulárnejšie evolučné optimalizačné algoritmy [4,5]. Základné princípy genetického algoritmu sú určitou špecifikáciou všeobecných princípov evolúcie a jej algoritmickej realizácie, ktoré boli prezentované v predchádzajúcej kapitole. Pod chromozómom  $\alpha$  budeme rozumieť binárny vektor fixnej dĺžky  $k$

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k) \in \{0,1\}^k \quad (4.5)$$

Populácia  $P$  obsahuje chromozómy - binárne vektory  $\alpha$

$$P = \{\alpha_1, \alpha_2, \dots, \alpha_p\} \quad (4.6)$$

kde  $p$  je kardinalita populácie  $P$ , vyjadruje počet chromozómov v  $P$ ,  $p=|P|$ . Nech  $f$  je účelová funkcia definovaná nad množinou binárnych vektorov dĺžky  $k$

$$f: \{0,1\}^k \rightarrow R \quad (4.7)$$

ktorá ohodnotí každý binárny vektor  $\alpha \in \{0,1\}^k$  reálnym číslom. Našou úlohou bude hľadať globálne minimum tejto funkcie nad množinou  $\{0,1\}^k$  (pozri (2.5))

$$\alpha_{opt} = \arg \min_{\alpha \in \{0,1\}^k} f(\alpha) \quad (4.8)$$

Funkcia  $f$  "reprezentuje prostredie" v ktorom existujú chromozómy populácie. Vyjadrené pomocou biologickej terminológie, chromozóm  $\alpha$  reprezentuje *genotyp* organizmu, zatiaľ čo funkčná hodnota  $f(\alpha)$  reprezentuje jeho *fenotyp*. Mierou úspešnosti chromozómu je jeho funkčná hodnota. Pretože hľadáme minimum účelovej funkcie, chromozóm je tým úspešnejší, čím je jeho funkčná hodnota (genotyp) menšia. Na základe týchto úvah môžeme zaviesť ohodnotenie chromozómov silou ako zobrazenie (4.2), ktoré vyhovuje nasledujúcej podmienke (pozri obr. 4.4)

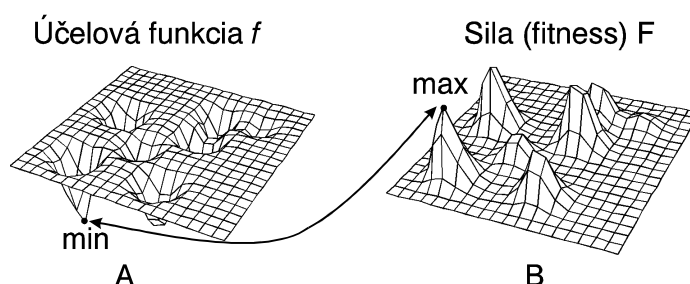
$$\forall \alpha_1, \alpha_2 \in P: f(\alpha_1) \leq f(\alpha_2) \Rightarrow F(\alpha_1) \geq F(\alpha_2) \geq 0 \quad (4.9)$$

Z praktických dôvodov je vhodné, aby numerické hodnoty sily boli z otvoreného intervalu  $(0,1)$ , preto sa zavádza tzv. *renormalizovaná sila*

$$F'(\alpha) = \frac{F(\alpha)}{\sum_{\alpha' \in P} F(\alpha')} \quad (4.10a)$$

$$\sum_{\alpha \in P} F'(\alpha) = 1 \quad (4.10b)$$

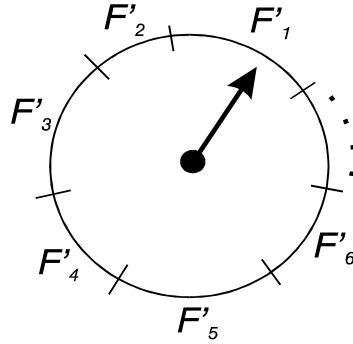
kde  $0 < F'(\alpha) < 1$ , pre každé  $\alpha \in P$ .



**Obrázok 4.4.** Diagram A znázorňuje účelovú funkciu v priestore všetkých možných binárnych vektorov, kde binárny vektor je formálne pretransformovaný na bod v dvojrozmernom priestore a tretí rozmer (výška) je daný hodnotou účelovej funkcie. Každému chromozómu z populácie  $P$  môžeme priradiť funkčnú hodnotu ležiacu na povrchu účelovej funkcie. Diagram B znázorňuje silu binárnych vektorov - chromozómov. V dôsledku podmienky (4.9), tam kde na povrchu A máme minimum, na povrchu B musí byť maximum (pozri šípku).

Akým konkrétnym spôsobom realizovať zobrazenie hodnôt účelovej funkcie na silu tak, aby bola splnená podmienka (4.9)? Toto zobrazenie sa dá uskutočniť mnohými spôsobmi, v tejto kapitole ukážeme dva prístupy zostrojenia síl pomocou hodnôt účelovej funkcie. Prvý postup sa zakladá na jednoduchom lineárnom zobrazení funkčných hodnôt na silu tak, že maximálnej (minimálnej) funkčnej hodnote je priradená minimálna (maximálna) sila, ostatné funkčné hodnoty sú lineárne interpolované medzi týmito dvoma limitnými hodnotami

$$F(\alpha) = \frac{F_{max} - F_{min}}{f_{min} - f_{max}} f(\alpha) + \frac{f_{min} F_{min} - f_{max} F_{max}}{f_{min} - f_{max}} \quad (4.11a)$$



**Obrázok 4.5.** Schematické vyjadrenie rulety pomocou jednotkovej kružnice, na ktorej sú vynesené renormalizované sily. Obrazne povedané, čím väčšia je renormalizovaná sila, tým väčší je priradený oblúk na kružnici. Ručička (šípka) sa tlmene točí okolo svojej osi (bodu) a je chápaná ako generátor náhodnej polohy na kružnici. Po jej zastavení ukazuje kvázináhodne vybraný chromozóm.

kde  $f_{\max}$  ( $f_{\min}$ ) je maximálna (minimálna) hodnota účelovej funkcie pre chromozómy populácie  $P$

$$f_{\max} = \max_{\alpha \in P} f(\alpha), \quad f_{\min} = \min_{\alpha \in P} f(\alpha) \quad (4.11b)$$

a  $F_{\max}$  ( $F_{\min}$ ) je maximálna (minimálna) hodnota sily, obvykle sa položí  $F_{\max}=1$  a  $F_{\min}=\varepsilon$ , kde  $\varepsilon$  je malé kladné číslo (napr.  $\varepsilon=0.01$ ). Dosadením týchto hraničných hodnôt sily do (4.11b) dostaneme konečný výraz pre výpočet sily chromozómov na základe ich hodnôt účelovej funkcie

$$F(\alpha) = \frac{1}{f_{\min} - f_{\max}} [(1 - \varepsilon)f(\alpha) + f_{\min}\varepsilon - f_{\max}] \quad (4.11c)$$

Táto formula je často používaná v rôznych implementáciách genetického algoritmu v prípade, že hodnoty účelovej funkcie sa moc nemenia (napr. sú rovnakého poriadku). V opačnom prípade, ak tieto funkcie sa menia vo veľkom intervale idúcom cez mnoho poriadkov funkčných hodnôt, formula (4.11c) priraduje skoro nulovú silu tým chromozómom, ktoré odpovedajú funkčným hodnotám blízky maximálnej hodnote a týmto sa neopodstatnene zosilňuje preferencia chromozómov s malou funkčnou hodnotou. Tento nedostatok je odstránený nasledujúcim alternatívnym prístupom k určaniu síl chromozómov.

Nech  $Q=(q_1, q_2, \dots, q_{|P|})$  je taká permutácia  $|P|$  chromozómov z populácie  $P$ , ktorá usporiada hodnoty účelovej funkcie do neklesajúcej postupnosti

$$f(\alpha_{q_1}) \leq f(\alpha_{q_2}) \leq \dots \leq f(\alpha_{q_{|P|}}) \quad (4.12a)$$

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

$$F(\alpha_{q_i}) = \frac{1}{1 - \varepsilon} [(1 - \varepsilon)i + \varepsilon - |P|] \quad (\text{pre } i = 1, 2, \dots, |P|) \quad (4.12b)$$

V tomto prípade je sila chromozómov z populácie  $P$  rovnomerne rozložená v celom intervale  $[\varepsilon, 1]$ , kde  $\varepsilon$  je malé kladné číslo. Určitým nedostatkom tejto formuly je, že ohodnocuje chromozómy s rovnakou funkčnou hodnotou rôznymi silami, čo môže byť na závalu hlavne pre záver behu genetického algoritmu, keď populácia je pomerne homogénna. Z týchto dôvodov ohodnotenie formulou (4.12b) je upravené takto

$$f(\alpha_{q_{i-1}}) = f(\alpha_{q_i}) \Rightarrow F(\alpha_{q_i}) = F(\alpha_{q_{i-1}}) \quad (\text{pre } i = 2, 3, \dots, |P|) \quad (4.12c)$$

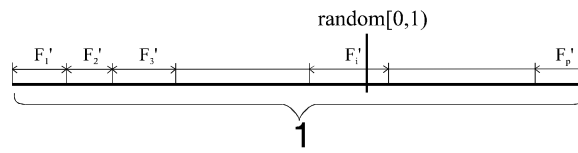
Potom chromozómy s rovnakou hodnotou účelovej funkcie sú ohodnotené rovnakou silou.

**Príklad 4.1.** V kapitole 4.2 sú popísané dve rôzne metódy transformácie hodnôt účelovej funkcie na sily tak, aby bola splnená podmienka (4.9). Navrhnite iné alternatívne spôsoby tejto transformácie a vykonajte ich interpretáciu v zmysle, ako sa líšia od pôvodných dvoch prístupov.

Pomocou renormalizovaných síl môžeme jednoducho realizovať kvázináhodnosť výberu chromozómov (čo patrí medzi základné imperatívy evolučných algoritmov) pomocou pojmu ruleta [4], pozri obr. 4.5. Chromozóm s indexom  $i$  je vybraný (kvázináhodne), ak pre náhodne generované číslo z intervalu  $[0, 1)$  s rovnomernou distribúciou pravdepodobnosti platí

$$\sum_{k=1}^{i-1} F'_k \leq \text{random} < \sum_{k=1}^i F'_k \quad (4.13)$$

Táto formula má jednoduchú grafickú interpretáciu znázornenú na obr. 4.6 (pozri tiež algoritmus 4.2).



**Obrázok 4.6.** Jednotková úsečka je rozdelená na úseky podľa veľkostí renormalizovaných síl chromozómov z populácie  $P$ . Náhodne generované číslo z intervalu  $[0,1]$  predstavuje polohu na úsečke (reprezentované hrubou vertikálnou čiarou) a podľa danej polohy určuje odpovedajúci chromozóm. Z konštrukcie tejto realizácie "rulety" vyplýva, že čím väčšia je sila chromozómu, tým väčšia je pravdepodobnosť jeho výberu.

```
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;
```

**Algoritmus 4.2.** Jednoduchá implementácia rulety pre náhodný výber chromozómov podľa ich sily. Vo while-cykle sa postupne (rekurentne) realizujú súčty renormalizovaných síl (pozri rov. (4.11) a obr. 4.5). Sofistikovanejší prístup k implementácii rulety je založený na tom, že pred aktiváciami rulety sa spočítajú dolné a horné hranice podintervalov, na ktoré je rozdelený interval  $[0,1]$ . Potom hľadáme, napr. metódou binárneho hľadania [xx], do ktorého intervalu patrí premenná  $aux$ .

**Príklad 4.2.** Algoritmus 4.2 implementuje ruletu podľa vzťahu (4.13) tak, že sa postupne hľadá interval  $[bound\_lower, bound\_upper]$ , ktorý obsahuje náhodne vygenerované číslo  $0 < aux \leq 1$ . Podstatne efektívnejšia metóda implementácia rulety je pomocou binárneho hľadania [6] (porovnaj s deťskou hrou určenia prirodzeného čísla menšieho alebo rovného ako 100, dobrý hráč sa pýta oponenta či dané číslo je väčšie (menšie) tak, že neustále delí nádejné intervaly na polovice). Navrhните algoritmus rulety pomocou binárneho hľadania, vykonajte jeho implementáciu.

### 4.3 Kríženie chromozómov

Pre chromozómy z populácie  $P$  sú definované dve základné operácie: operácia mutácie a operácia kríženia. Prvá operácia (mutácia) už bola podrobne diskutovaná v rámci horolezeckého algoritmu (pozri kapitolu 1.4), preto sa teraz budeme podrobne venovať len operácii 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

$$(\alpha', \beta') = O_{cross}(\alpha, \beta) \quad (4.14)$$

kde rôzne spôsoby realizácie operátora  $O_{cross}$  sú znázornené na obr. 4.7 (pozri algoritmus 4.3). Tento operátor tiež má stochastický charakter, podobne ako operátor mutácie. Táto stochastičnosť je založená na tom, že pri aplikovaní operátora kríženia na dvojicu chromozómov z populácie sa generuje náhodne tzv. bod (alebo body) kríženia, potom už je aplikácia kríženia plne deterministická.

V prípade, že chromozóm kóduje  $n$  reálnych premenných pomocou binárnych reťazcov dĺžky  $k$  (celková dĺžka chromozómu je  $kn$ ), operácia kríženia sa môže aplikovať separátne pre každý binárny podreťazec reprezentujúci jednotlivú premennú účelovej funkcie, pozri obr. 4.8.

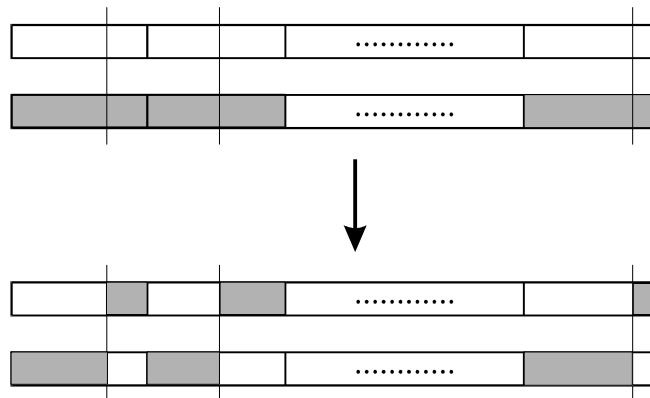
```

1-bodové kríženie
procedure Crossover(input:  $\alpha, \beta$ ; output:  $\alpha', \beta'$ );
begin cross_point :=  $O_{rand}(k)$ ;
  for  $i := 1$  to cross_point do
     $\alpha_i := \beta_i$ ;  $\beta_i := \alpha_i$ ;
  for  $i :=$  cross_point + 1 to  $n$  do
     $\alpha_i := \alpha_i$ ;  $\beta_i := \beta_i$ ;
end;
2-bodové kríženie
begin  $\alpha'_1 := \beta_1$ ;  $\alpha'_{i+1} := \beta_{i+1}$ ;  $\alpha'_j := \beta_j$ ;
   $\beta_1 := \alpha_1$ ;  $\beta_{i+1} := \alpha_{i+1}$ ;  $\beta_j := \alpha_j$ ;
end;

```

**Algoritmus 4.3.** Jednoduchá pseudopascalovská implementácia kríženia. Premenná  $cross\_point$  je náhodne generovaný bod kríženia. Do tohto bodu sa chromozómy  $\alpha$  a  $\beta$  jednoducho kopírujú do nových chromozómov  $\alpha'$  resp.  $\beta'$ . Za týmto bodom až do konca chromozómu sa do  $\alpha'$  kopíruje zvyšok  $\beta$  a do  $\beta'$  zvyšok  $\alpha$ . Každý z dvoch nových chromozómov je zložený z dvoch častí. V 1-bodovom krížení je náhodne generovaný bod kríženia a chromozómy si vymenia svoje časti za týmto bodom. V 2-bodovom krížení sú náhodne generované dva rôzne body kríženia, chromozómy si vymenia časti medzi nimi.

**Príklad 4.3.** Vykonajte rozšírenie algoritmu 4.3 (realizujúceho 1-bodové kríženie binárneho reťazca dĺžky  $k$ ) tak, aby sa realizovalo kríženie znázornené na obr. 4.8, kde chromozóm sa skladá z  $n$  binárnych podreťazcov dĺžky  $k$ .



**Obrázok 4.8.** Schematické znázornenie kríženia dvoch chromozómov dĺžky  $kn$ , kde  $k$  je dĺžka binárnej reprezentácie reálnej premennej a  $n$  je počet premenných účelovej funkcie. Operácia kríženia prebieha pre každý podreťazec zvlášť pre náhodne generované body kríženia (vertikálne čiary v hornej časti diagramu).

Pomocou operátorov mutácie a kríženia môžeme pristúpiť ku konštrukcii operátora reprodukcie (4.3). Nech  $\alpha, \beta \in P$  sú dva chromozómy z populácie  $P$ , pomocou operátora reprodukcie zostrojíme z týchto dvoch "rodičovských" chromozómov dva nové chromozómy "potomkov", formálne (pozri (4.3))  $(\alpha', \beta') = O_{repro}(\alpha, \beta)$ . V prvom kroku zostrojíme pomocou kríženia dva nové chromozómy

$$(\hat{\alpha}, \hat{\beta}) = O_{cross}(\alpha, \beta) \tag{4.15}$$

V nasledujúcom kroku sú tieto dva nové chromozómy modifikované mutáciou na výsledné chromozómy - potomkov

$$\alpha' = O_{mut}(\hat{\alpha}) \quad \text{a} \quad \beta' = O_{mut}(\hat{\beta}) \tag{4.16}$$

Implementácia reprodukčného operátora je vyjadrená pomocou algoritmu 4.4.



```

procedure Reproduction(input: $\alpha, \beta$ ; output: $\alpha', \beta'$ );
begin Crossover( $\alpha, \beta, \hat{\alpha}, \hat{\beta}$ );
      Mutation_Bin( $\hat{\alpha}, \alpha'$ );
      Mutation_Bin( $\hat{\beta}, \beta'$ );
end;

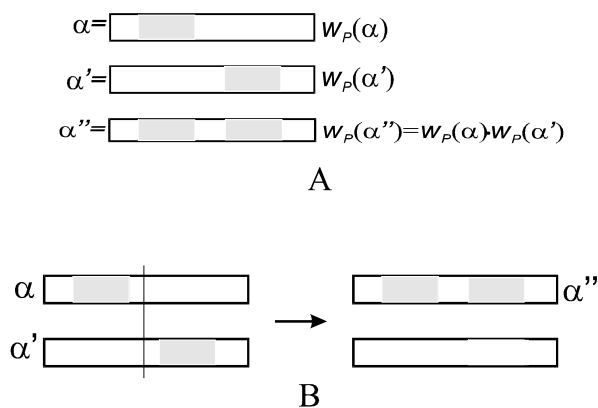
```

**Algoritmus 4.4.** V prvom kroku sa aktivuje procedúra kríženia (pozri algoritmus 4.3), výstupné parametre tejto procedúry  $\hat{\alpha}$  a  $\hat{\beta}$  sú modifikované pomocou mutácie (pozri algoritmus 1.2).

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. Toto presvedčenie je čiastočne založené na osobných skúsenostiach s aplikáciami genetického algoritmu a na analógiách s evolúciou biologických objektov, ktoré vyústili vo formulácii slávnej Hollandovej vety [3] tvoriacej teoretický základ genetického algoritmu.

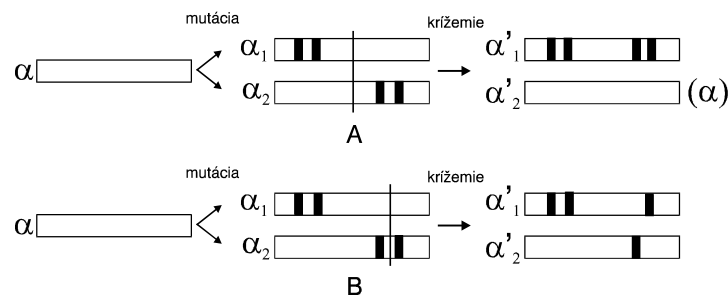
Pohlavná reprodukcia je v prírode všeobecne využívaná hlavne medzi zložitejšími druhmi našej planéty. V porovnaní s reprodukčnými technikami, ktoré vyžadujú len jedného rodiča, pohlavná reprodukcia je neobyčajne zložitý proces produkcie potomkov. Druh využívajúci pohlavnú reprodukciu musí byť podstatne zložitejší ako druh, ktorý ju nevyužíva. Tieto jedinci sa musia odlišovať v určitých podstatných znakoch a musia tiež vynaložiť určitý čas a energiu, aby sa vzájomne stretli k reprodukčnému procesu. 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 pomocou mutácií.



**Obrázok 4.9.** Schematické znázornenie významu kríženia v genetickom algoritme. Diagram A znázorňuje chromozómy  $\alpha$  a  $\alpha'$ , ktoré obsahujú dva rôzne neprekrývajúce sa podreťazce, tieto podreťazce zvyšujú silu chromozómov. Pravdepodobnosť výskytu týchto chromozómov je označená  $w_p(\alpha)$  a  $w_p(\alpha')$ . Chromozóm  $\alpha''$  obsahuje obidva podreťazce, pravdepodobnosť jeho výskytu je zhruba rovná súčtu pravdepodobností výskytu chromozómov  $\alpha$  a  $\alpha'$ ,  $w_p(\alpha'') = w_p(\alpha) \times w_p(\alpha')$ . Diagram B Ukazuje, že použitím kríženia môžeme zostrojiť chromozóm  $\alpha''$ , pravdepodobnosť tejto konštrukcie je podstatne vyššia ako pravdepodobnosť súčasného výskytu dvoch podreťazcov  $w_p(\alpha'')$ .

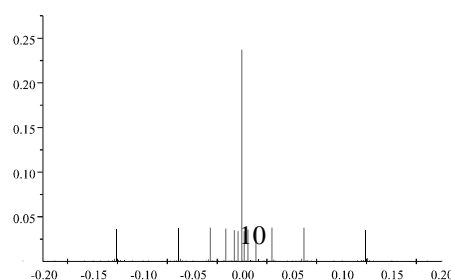
Prevedieme abstrakciu tohto problému (pozri obr. 4.9). Nech  $\alpha$  a  $\alpha'$  sú dva chromozómy, a každý z nich obsahuje nejaký podreťazec, ktorý podstatne zvyšuje silu chromozómu. Nech sú tieto reťazce na takých pozíciách, že keby boli na jednom chromozóme, tak sa ani čiastočne neprekrývajú. Nech pravdepodobnosť výskytu chromozómu s jedným z týchto dvoch výhodných podreťazcov v populácii je označená  $w_P(\alpha)$  resp.  $w_P(\alpha')$ . Potom pravdepodobnosť  $w_P(\alpha'')$  výskytu chromozómu  $\alpha''$ , ktorý súčasne obsahuje podreťazce z chromozómov  $\alpha$  a  $\alpha'$ , sa približne rovná súčinu týchto pravdepodobností,  $w_P(\alpha'')=w_P(\alpha) \times w_P(\alpha')$ . Položme, *ad-hoc*,  $w_P(\alpha)=10^{-6}$  a  $w_P(\alpha')=10^{-6}$ , potom  $w_P(\alpha'')=10^{-12}$ . Môžeme teda konštatovať, že súčasný výskyt podreťazcov z chromozómov  $\alpha$  a  $\alpha'$  v jednom chromozóme  $\alpha''$  v dôsledku len mutácií je vysoko nepravdepodobný jav. Situácia sa avšak dramaticky zmení, ak pripustíme kríženie medzi chromozómami. Výskyt chromozómu  $\alpha''$  v populácii sa stáva vysoko pravdepodobným javom za predpokladu, že chromozómy  $\alpha$  a  $\alpha'$  sa v populácii už vyskytujú. Pohlavné kríženie (alebo v abstraktnej rovine, kríženie medzi chromozómami) je efektívny spôsob ako prekonať pravdepodobnostnú bariéru tvorby chromozómov s vysokým informačným obsahom a teda aj s vysokou silou. Ak nezapojíme do prirodzeného výberu pohlavné kríženie, tak sa



**Obrázok 4.10.** Dva rôzne prípady pôsobenia kríženia na dva chromozómy, ktoré vznikli mutáciou (v tomto prípade práve dva bity zmenili svoje hodnoty) templátu  $\alpha$ . V prípade A bod kríženia bol náhodne vybraný tak, že krížením chromozómov sme dostali jedného potomka, ktorý je totožný s templátom  $\alpha$ . V prípade B, bod kríženia bol vybraný tak, že ani jeden z potomkov kríženia nie je totožný s templátom.

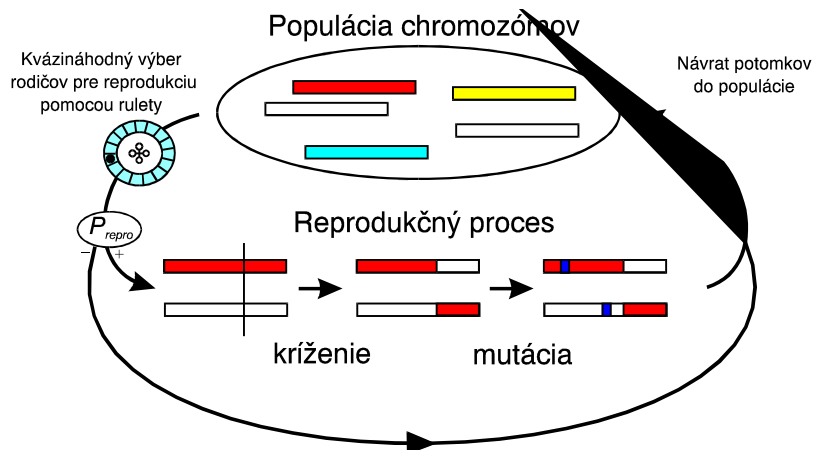
rýchlosť evolúcie podstatne zníži tak, že vznik zložitejších druhov v prírode je len málo pravdepodobný.

Vyššie uvedené úvahy o význame kríženia v genetickom algoritme budú konkretizované pre chromozómy reprezentované binárnymi reťazcami. Vykonajme nasledujúci jednoduchý výpočtový experiment. Nech  $\alpha$  je binárny reťazec - templát, špeciálnou operáciou mutácie  $O_{mut}^{(i)}$  pretransformujeme tento chromozóm na dva nové chromozómy,  $\alpha_1 = O_{mut}^{(i)}(\alpha)$  a  $\alpha_2 = O_{mut}^{(i)}(\alpha)$ . Operátor mutácie  $O_{mut}^{(i)}$  mutuje - preklopí práve  $i$  bitov v reťazci  $\alpha$ . Stochastičnosť tohto operátora spočíva v tom, že bity v ktorých dochádza k preklopeniu sú náhodne vybrané, ich počet je určený parametrom  $i$ . Aplikovaním operácie kríženia na tieto dva chromozómy dostaneme  $(\alpha'_1, \alpha'_2) = O_{cross}(\alpha_1, \alpha_2)$ . Definujme si nasledujúce dva výrazy:  $\Delta_1 = \text{real}(\alpha) - \text{real}(\alpha'_1)$  a  $\Delta_2 = \text{real}(\alpha) - \text{real}(\alpha'_2)$ , vyjadrujúci rozdiel medzi reálnou hodnotou templátu  $\alpha$  a chromozómu vzniklého mutáciou a krížením. V prípade, že  $\Delta_i=0$  (pre  $i=1,2$ ), potom operácia kríženia "opravila" mutáciou zmenený chromozóm  $\alpha_i$  na pôvodný templát. V opačnom prípade, ak  $\Delta_i \neq 0$ , kríženie produkuje nové chromozómy, ktoré sú rôzne od templátu (pozri obr. 4.10). Na obr. 4.11 je uvedený graf znázorňujúci distribúciu hodnôt diferencií  $\Delta$  pre mutáciu špecifikovanú tak, že zmení v reťazci práve jeden bit. Na tomto grafe vidíme významný "pík" odpovedajúci  $\Delta=0$ , t.j. kríženie opravuje mutované reťazce na pôvodný templát. Ako ukazujú naše modelové výpočty, distribúcia



**Obrázok 4.11.** Distribúcia pravdepodobnosti výskytu  $\Delta$  rozdielov medzi templátom (náhodne generovaným) dĺžky 10 a chromozómami z neho tvorených mutáciou (jeden bit sa zmení) a potom skrížených, pričom reálne čísla priradené chromozómom sú z intervalu (0,1). Distribúcia má významný "pík" pre  $\Delta=0$ , t.j. kríženie významne "opravuje" zmutované chromozómy do tvaru pôvodného templátu

pravdepodobnosti tohto "píku"  $\Delta=0$  rýchlo klesá s rastúcim počtom zmenených bitov, pre štyri zmenené bity je táto pravdepodobnosť takmer nulová. Ako interpretovať tieto pozorovania? Operácia kríženia je veľmi efektívnou vtedy, keď do procesu kríženia vstupujú chromozómy, ktoré majú veľké neprekrývajúce sa časti z "cieľového" chromozómu - templátu. V tomto prípade operácia kríženia



**Obrázok 4.12.** Schematické znázornenie genetického algoritmu (pozri algoritmus 4.5). Pomocou rulety sa vyberie dvojica chromozómov, ktorá vstupuje do procesu reprodukcie s pravdepodobnosťou  $P_{repro}$ , pričom reprodukčný proces sa skladá z kríženia a mutácie.

produkuje nové chromozómy, ktoré tieto rôzne časti integrujú do jedného chromozómu - potomka veľmi blízkeho (alebo totožného) s cieľovým templátom. V opačnom prípade, ak dvojica chromozómov vstupujúca do kríženia neobsahuje takéto časti, operácia kríženia pôsobí podobne ako mutácia, produkuje dvojicu chromozómov, ktorých sila je v podstate podobná sile chromozómov vstupujúcich do kríženia.

**Príklad 4.4.** Napíšte program pomocou ktorého budete schopný zreprodukovať graf na obr. 4.11. Preved'te štúdiu pre rôzne dĺžky binárneho reťazca a počet zmenených bitov v binárnych reťazcoch  $\alpha_1$  a  $\alpha_2$ .

### 4.3 Implementácia genetického algoritmu [4]

Konkrétnu implementáciu genetického algoritmu dostaneme zo všeobecného evolučného algoritmu (pozri algoritmus 4.1), ak je operátor reprodukcie  $O_{repro}$  špecifikovaný tak, že obsahuje dve časti: časť kríženia a časť mutácie (pozri algoritmus 4.5). Výber chromozómov do reprodukčného procesu sa realizuje pomocou rulety (pozri algoritmus 4.2). Pseudopascalovský kód genetického algoritmu je vyjadrený algoritmi 4.5 (pozri obr. 4.12).

Problém zastavenia genetických algoritmov v týchto implementáciách je zatiaľ riešený tak, že počet generácií je zhora ohraničený číslom  $t_{max}$ , čo je avšak veľmi zjednodušený pohľad na tento dôležitý aspekt genetických algoritmov. Budeme diskutovať dve metódy ukončenia genetického algoritmu.

Prvá metóda spočíva v tom, že v každom iteračnom kroku - generácii nájdeme v populácii chromozóm s najväčšou hodnotou sily, označíme ho  $\tilde{\alpha}_{opt}$ . Nech  $0 \leq w(\tilde{\alpha}_{opt}) \leq 1$  je frakcia chromozómov populácie, ktoré sa rovnajú  $\tilde{\alpha}_{opt}$ . Potom genetický algoritmus môže byť ukončený, ak frakcia  $w(\tilde{\alpha}_{opt}) > w_{max}$ , kde  $w_{max}$  je prahová hodnota frakcie (obvykle 0.8-0.95). To znamená, že

populácia obsahuje  $100 \times w(\tilde{\alpha}_{opt})$  % chromozómov, ktoré sú identické chromozómu  $\tilde{\alpha}_{opt}$ , čiže je už malá pravdepodobnosť, že v rámci ďalších generácií genetického algoritmu vzniknú chromozómy s

```

procedure Genetic_Algorithm(input  $t_{max}$ ; output  $\alpha_{opt}$ );
begin  $t:=0$ ; stop_criterion:=false;
       $P:=\{\text{randomly generated population of chromosomes}\}$ ;
      while ( $t < t_{max}$ ) and (not stop_criterion) do
        begin  $t:=t+1$ ;  $Q:=\emptyset$ ;
          each chromosome is evaluated by fitness;
          while  $|Q| < |P|$  do
            begin select by roulette wheel two
              chromosomes  $\alpha_1, \alpha_2 \in P$ ;
              if random  $< P_{repro}$  then
                Reproduction( $\alpha_1, \alpha_2, \alpha'_1, \alpha'_2$ ) else
                  begin  $\alpha'_1 := \alpha_1$ ;  $\alpha'_2 := \alpha_2$  end;
                 $Q := Q \cup \{\alpha'_1, \alpha'_2\}$ ;
            end;
           $P := Q$ ;
          if convergence criteria are fulfilled
            then stop_criterion:=true;
        end;
       $\alpha_{opt} := \text{best chromosome of } P$ ;
end;

```

**Algoritmus 4.5.** Pseudopascalovská implementácia genetického algoritmu, ktorá vznikla z evolučného algoritmu 4.1 tak, že reprodukčný operátor  $O_{repro}$  je špecifikovaný algoritmom 4.5, t.j. obsahuje kríženie a mutáciu. Vonkajší while-cyklus riadi výmenu generácií - epoch algoritmu, ktorých maximálny počet je určený vstupným parametrom  $t_{max}$ . Vnútorý while-cyklus generuje novú populáciu  $Q$  tak, že sú opakovane pomocou rulety vyhladané dva chromozómy a tieto s pravdepodobnosťou  $P_{repro}$  vstupujú do reprodukčného procesu (v opačnom prípade sa vybrané chromozómy jednoducho skopírujú do novej populácie  $Q$ ). Ako výstup z genetického algoritmu je  $\alpha_{opt}$ , určený ako chromozóm s najväčšou silou z poslednej populácie. Z konštrukcie novej populácie  $Q$  vyplýva, že populácia  $P$  musí mať párný počet chromozómov. Genetický algoritmus obsahuje dve pravdepodobnosti, a to pravdepodobnosť reprodukcie  $P_{repro}$  a pravdepodobnosť 1-bitovej mutácie  $P_{mut}$ , tieto možno pokladať spolu s kardinalitou populácie  $P$  za základné parametre algoritmu.

ešte väčšou silou ako má chromozóm  $\tilde{\alpha}_{opt}$ .

Druhý prístup k zastaveniu genetického algoritmu je založený na výpočte analógu pravdepodobnostného vektora  $w=(w_1, w_2, \dots, w_k)$  známeho z horolezeckého algoritmu s učením (pozri kapitolu 1.7). Nech populácia  $P$  v  $i$ -tom kroku (generácii) obsahuje chromozómy  $P=(\alpha_1, \alpha_2, \dots, \alpha_p)$ , kde  $\alpha_i=(\alpha_1^{(i)}, \alpha_2^{(i)}, \dots, \alpha_k^{(i)})$ . Potom komponenty pravdepodobnostného vektora sú určené takto

$$w_j = \frac{1}{p} \sum_{i=1}^p \alpha_j^{(i)} \quad (\text{pre } j=1, 2, \dots, k) \quad (4.17)$$

ktoré vyjadrujú "pravdepodobnosť" výskytu "1" v  $j$ -tej polohe vo všetkých chromozómoch populácie  $P$ . V prípade, že  $w_j=0$  (alebo  $w_j=1$ ), potom  $100 \times w_j$  % chromozómov z populácie  $P$  má v  $j$ -tej polohe 0 (1). Parameter usporiadania (pozri rov. (1.45)) definovaný pre takto určený pravdepodobnostný vektor má tvar

$$\chi(w) = \frac{4}{k} \sum_{j=1}^k (w_j - 0.5)^2 \quad (4.18)$$

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. Preto môžeme zaviesť prahovú hodnotu parametra usporiadania,  $\chi_{crit}$ , potom ak platí  $\chi(w) > \chi_{crit}$ , genetický algoritmus je ukončený. V tomto prípade populácia obsahuje chromozómy, ktoré v každej polohe majú s veľkou frekvenciou buď "0" alebo "1", takže môžeme povedať, že populácia obsahuje s veľkou

prevahou rovnaké chromozómy a preto je malá pravdepodobnosť, že v rámci pokračovania genetického algoritmu dostaneme nové lepšie riešenie.

**Príklad 4.5.** Napíšte program pre genetický algoritmus podľa algoritmu 4.5, pričom účelová funkcia obsahuje  $n$  premenných z intervalu  $[a,b]$ , pričom tieto reálne premenné sú v binárnej reprezentácii aproximované reťazcami dĺžky  $k$ . V binárnej reprezentácii nech sa používa tak štandardné, ako aj Grayovo kódovanie. Vykonaajte tieto štúdie:

(1) Účelová funkcia z príkladu 2.7 pre  $n=1,2,3,4$  (pozri príklad 2.7).

(2) Účelová funkcia z príkladu 2.8 pre  $n=1,2,3,4$  (pozri príklad 2.8).

Pre obe tieto funkcie porovnajme efektívnosť algoritmus tak pre Grayovo kódovanie, ako aj pre štandardné kódovanie binárnej reprezentácie reálnych čísel. Sila chromozómov z populácie nech je počítaná formulami (4.11) resp. (4.12). Pokúste sa optimalizovať pravdepodobnosti  $P_{mut}$  a  $P_{repro}$  pre rôzne veľkosti populácie, spôsoby kódovania a dĺžku  $k$  binárnej reprezentácie.

**Príklad 4.6.** Implementujte do genetického algoritmu z príkladu 4.5 kritéria pre zastavenie genetického algoritmu z kapitoly 4.3. Vykreslite priebeh pravdepodobností  $w_i$  definovaných vzťahom (4.17) vzhľadom k počtu epoch (iteračných krokov) genetického algoritmu. Výsledky porovnajme s obr. 1.16.

## 4.4 Teória genetického algoritmu

Väčšina teoretických prác venovaná genetickému algoritmu [3,7] je zameraná na význam tzv. schém, ktoré určujú celé triedy podobných binárnych reťazcov dĺžky  $k$  (chromozómov) z množiny  $\{0,1\}^k$ . Schéma je definovaná ako reťazec dĺžky  $k$  zostrojený nad množinou symbolov  $\{0,1,*\}$ , formálne schéma  $\sigma \in \{0,1,*\}^k$ , kde "\*" je tzv. divoký symbol (wildcard). Hovoríme, že reťazec  $\alpha \in \{0,1\}^k$  je príkladom schémy  $\sigma \in \{0,1,*\}^k$ , formálne  $\sigma \in \alpha$ , vtedy, ak v každej polohe schémy  $\sigma$  s iným ako divokým symbolom je hodnota znaku reťazca  $\alpha$  totožná so znakom v  $\sigma$ . Formálne

$$\sigma \in \alpha \Leftrightarrow (\forall i \in \{1,2,\dots,k\}: \sigma_i \in \{0,1\} \Rightarrow \sigma_i = \alpha_i) \quad (4.19)$$

Zavedený pojem môžeme jednoducho ilustrovať schémou  $\sigma=(11**0**)$ , binárny reťazec dĺžky  $k=7$   $\alpha=(1101011)$  je príkladom tejto schémy, prvé dva divoké symboly sú nahradené podreťazcom 01, zatiaľ čo posledné dva divoké symboly podreťazcom 11.

Množina  $I(\sigma)$  obsahuje všetky možné príklady  $\alpha$  schémy  $\sigma$ ,

$$I(\sigma) = \{\alpha; \alpha \in \sigma\} \quad (4.20)$$

Pre schéma  $\sigma=(11**0**)$  množina  $I(\sigma)$  obsahuje 16 binárnych reťazcov, ktoré môžu byť zostrojené tak, že divoké symboly "\*" systematicky nahradíme "0" alebo "1",

$$I(\sigma) = \{(1100000), (1100001), (1100010), \dots, (1111011)\} \quad (4.21)$$

Rád schémy  $\sigma$ , označený  $o(\sigma)$ , popisuje počet symbolov "0" alebo "1" v  $\sigma$

$$o(\sigma) = \left| \left\{ i; \sigma_i \in \{0,1\} \right\} \right| \quad (4.22)$$

táto formula určujúca rád schémy  $\sigma$  má jednoduchú interpretáciu, rád je definovaný ako kardinalita množiny indexov komponent schémy  $\sigma$ , ktoré nie sú rovné divokému symbolu. Pre  $\sigma=(11**0**)$  platí  $o(\sigma)=3$ , pretože schéma obsahuje len tri "nedivoké" symboly. Dĺžka  $\Delta$  schémy  $\sigma$  je určená ako rozdiel medzi indexom posledného binárneho symbolu v  $\sigma$  a prvým binárnym symbolom v  $\sigma$

$$\Delta(\sigma) = \max\{i; \sigma_i \in \{0,1\}\} - \min\{i; \sigma_i \in \{0,1\}\} \quad (4.23)$$

Pre  $\sigma=(11**0**)$  platí  $\Delta(\sigma)=5-1=4$ . Veličiny rád a dĺžka schémy budú užitočné pre náš výpočet pravdepodobnosti prežitia schémy  $\sigma$  v populácii  $P$  pod aplikovaním mutácie alebo kríženia.

Pravdepodobnosť prežitia schémy  $\sigma$  po aplikovaní mutácie (určenej s pravdepodobnosťou  $P_{mut}$  jedno bitovej mutácie) je určená vzťahom

$$(1 - P_{mut})^{o(\sigma)} \quad (4.24a)$$

kde pravdepodobnosť prežitia jedného bitu je  $(1 - P_{mut})$ . Pretože schéma  $\sigma$  obsahuje práve  $o(\sigma)$  binárnych symbolov ( $k - o(\sigma)$  divokých symbolov), potom elementárna pravdepodobnosť prežitia jedného bitu musí byť povýšená na exponent  $o(\sigma)$ , tým sme dostali (4.24a). V mnohých praktických aplikáciách genetického algoritmu pravdepodobnosť  $P_{mut}$  je veľmi malá, preto (4.24a) môžeme zjednodušiť takto

$$1 - o(\sigma)P_{mut} \quad (4.24b)$$

Výpočet pravdepodobnosti prežitia schémy  $\sigma$  v populácii  $P$  pod aplikáciou kríženia je podstatne zložitejší ako predošlý postup. Schéma  $\sigma$  zaniká aplikovaním kríženia s pravdepodobnosťou  $P_{cross}\Delta(\sigma)/(k-1)$ . Naviac, každé kríženie s chromozómom  $\alpha \in P$ , ktorý je príkladom schémy  $\sigma$ ,  $\alpha \in \sigma$ , automaticky garantuje zachovanie tejto schémy. Nech  $N(\sigma) = |I(\sigma) \cap P|$  popisuje počet príkladov schémy  $\sigma$  v populácii  $P$ , potom len frakcia  $1 - N(\sigma)/|P|$  z populácie  $P$  môže spôsobovať deštrukciu schémy  $\sigma$  krížením. Potom, minimálna pravdepodobnosť prežitia schémy  $\sigma$  v populácii  $P$  je určená vzťahom

$$1 - P_{cross} \frac{\Delta(\sigma)}{k-1} \left( 1 - \frac{N(\sigma)}{|P|} \right) \quad (4.25a)$$

Táto formula sa dá podstatne zjednodušiť, ak sa zanedbá súčinový člen, dostaneme

$$1 - P_{cross} \frac{\Delta(\sigma)}{k-1} \quad (4.25b)$$

Upriamime našu pozornosť na problém výberu chromozómov vstupujúcich do reprodukcie. Zavedieme nasledujúce dve veličiny, *strednú silu schémy*  $\sigma$

$$F(\sigma) = \frac{1}{N(\sigma)} \sum_{\alpha \in I(\sigma) \cap P} F(\alpha) \quad (4.26)$$

a strednú silu chromozómov z populácie  $P$

$$\bar{F} = \frac{1}{|P|} \sum_{\alpha \in P} F(\alpha) \quad (4.27)$$

Pri prechode z generácie  $t$  na nasledujúcu generáciu  $t+1$  v genetickom algoritme, ktorý obsahuje len výber chromozómov podľa rulety a reprodukčný proces je plne ignorovaný, nastane nasledujúca zmena počtu výskytu schémy  $\sigma$  v populácii

$$N_{t+1}(\sigma) = N_t(\sigma) \frac{F_t(\sigma)}{\bar{F}_t} \quad (4.28)$$

kde dolný index popisuje index generácie populácie, pre ktorú boli dané veličiny počítané.

Nech  $\sigma$  je schéma, ktorej stredná hodnota sily vzhľadom k strednej hodnote chromozómov je určená vzťahom

$$F_t(\sigma) = \bar{F}_t \times (1 + c) \quad (4.29)$$

kde konštanta  $c$  je malé kladné alebo záporné číslo. Dosadením tohto vzťahu do (4.30) a opakovaným použitím výslednej formuly  $N_{t+1}(\sigma) = N_t(\sigma) \times (1 + c)$   $t$ -krát pre počiatkový stav  $t=0$  dostaneme "idealizovanú" rovnicu

$$F_t(\sigma) = F_0(\sigma) \times (1 + c)^t \quad (4.30)$$

To znamená, že pre proporcionálny výber (realizovaný ruletou) stredná hodnota síl schémy  $\sigma$  *exponenciálne rastie* (klesá) v priebehu mnohých generácií genetického algoritmu pre  $1+c > 1$  (resp.  $1+c < 1$ ).

Kombinovaním rovnice (4.28), ktorá vyjadruje zmenu výskytu schémy  $\sigma$  v dôsledku proporcionálneho výberu, s pravdepodobnosťami prežitia schémy  $\sigma$  v dôsledku mutácie a kríženia, dostaneme finálnu formulu pre odhad počtu príkladov schémy pre generáciu  $t+1$  populácie  $P$  za predpokladu, že reprodukčný proces obsahuje mutáciu, kríženie a výber pomocou rulety

$$N_{t+1}(\sigma) \approx N_t(\sigma) \frac{F_t(\sigma)}{\bar{F}_t} \left[ 1 - P_{cross} \frac{\Delta_t(\sigma)}{k-1} \times \left( 1 - \frac{N_t(\sigma)}{|P|} \right) \right] \times (1 - P_{mut})^{O_t(\sigma)} \quad (4.31a)$$

jej zjednodušená verzia podľa má tvar (zanedbajú sa "krížové" súčiny v hranatej zátvorke)

$$N_{t+1}(\sigma) \approx N_t(\sigma) \frac{F_t(\sigma)}{\bar{F}_t} \left[ 1 - P_{cross} \frac{\Delta_t(\sigma)}{k-1} - P_{mut} O_t(\sigma) \right] \quad (4.31b)$$

**Veta (Schema Theorem [3]).** Nech pre populáciu  $P_t$  platí, že  $N_t(\sigma)$  je počet chromozómov ktoré obsahujú schému  $\sigma$ ,  $F_t(\sigma)$  je stredná hodnota sily chromozómov obsahujúcich schému  $\sigma$  a  $\bar{F}_t$  je stredná hodnota sily všetkých chromozómov. Potom očakávaný počet chromozómov v nasledujúcej populácii, ktoré obsahujú schému  $\sigma$  je zdola ohraničený nerovnosťou

$$N_{t+1}(\sigma) \approx N_t(\sigma) \frac{F_t(\sigma)}{\bar{F}_t} (1 - P_{\text{zánik}}) \quad (4.32)$$

kde  $P_{\text{zánik}}$  je pravdepodobnosť zániku schémy v dôsledku mutácie a kríženia (pozri rovnicu (4.33)).

Táto veta hovorí, že ak platí  $F_t(\sigma) > \bar{F}_t$  ( $F_t(\sigma) < \bar{F}_t$ ), potom schéma  $\sigma$  sa vyskytuje v nasledujúcej generácii v chromozómoch s rastúcou (klesajúcou) frekvenciou. Tieto jednoduché závery sú založené na predpoklade, že pravdepodobnosť zániku

$$P_{\text{zánik}} = P_{\text{cross}} \frac{\Delta_t(\sigma)}{k-1} - P_{\text{mut}} o_t(\sigma) \quad (4.33)$$

je malé číslo. Toto je obvykle splnené pre krátke schémy (s malým  $\Delta(\sigma)$ ), nazývané stavebné kamene (building blocks). Možno konštatovať, že táto veta tvorí jeden z teoretických základov genetického algoritmu.

Teorém uvedený vyššie predstavuje jeden z hlavných teoretických výsledkov genetického algoritmu, ktorý bol dosiahnutý Hollandom v r. 1975 [3]. Naznačuje mechanizmus práce tohto algoritmu menovite v prvých iteračných krokoch, kde schémy - stavebné kamene hrajú dôležitú úlohu. Teoretické aspekty genetického algoritmu, ktoré súvisia s jeho schopnosťou nájsť globálne minimum účelovej funkcie aproximovanej reálnymi hodnotami z ortogonálnej mriežky bodov (pozri obr. 1.8) sú študované až v poslednej dobe [8,9]. Uvedieme teorém, ktorý špecifikuje za akých podmienok je genetický algoritmus potenciálne schopný poskytnúť globálne minimum s jednotkovou pravdepodobnosťou.

**Veta.** Nech v genetickom algoritme postupnosť populácií  $P_0, P_1, \dots$  je monotónna, t.j. pre každú generáciu  $t$  platí

$$\max_{\alpha \in P_{t+1}} F(\alpha) \geq \max_{\alpha \in P_t} F(\alpha) \quad (4.34)$$

potom genetický algoritmus asymptoticky pre  $t \rightarrow \infty$  dosiahne globálneho optima

$$\alpha_{\text{opt}} = \lim_{t \rightarrow \infty} \left( \arg \min_{\alpha \in P_t} F(\alpha) \right) \quad (4.35)$$

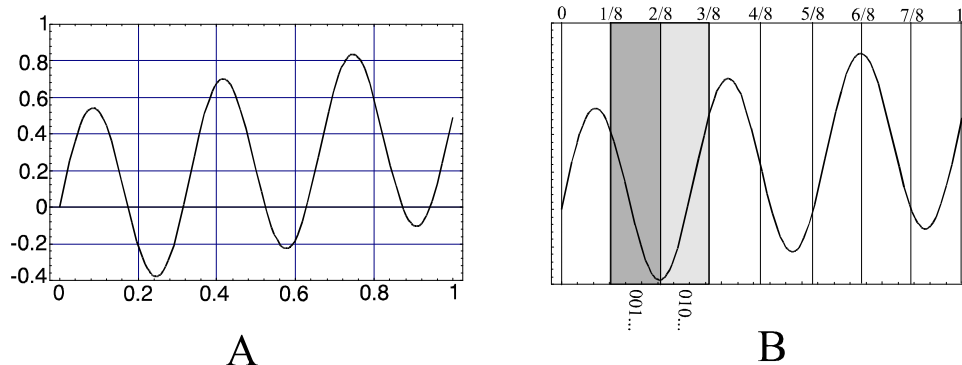
Ako dosiahnuť monotónnosť genetického algoritmu? Toto môže byť jednoducho dosiahnuté pomocou tzv. *elitizmu*, ktorý je založený na jednoduchej myšlienke, že najlepšie riešenie z populácie  $P_t$  je automaticky prenesené aj do nasledujúcej populácie  $P_{t+1}$ . Ináč povedané, najlepšie riešenie v populácii  $P_t$  nemá šancu byť náhodne vynechané výberom realizovaným ruletou.

## 4.5 Ilustračný príklad

Základné pojmy z predchádzajúcej kapitoly budú ilustrované jednoduchou implementáciou genetického algoritmu (pozri algoritmus 4.5) s nasledujúcou účelovou funkciou (pozri obr. 4.13)

$$f(x) = e^{-x^2/100} \times \sin(10x) \times \cos(9x) \quad (4.36)$$

pre  $x \in [0,1]$ . Našou úlohou bude detailne sledovať priebeh genetického algoritmu pri stanovení globálneho minima tejto funkcie.



**Obrázok 4.13.** (A) Priebeh funkcie definovanej (4.36). Táto 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, pozri obr. 4.14 a 4.15.

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

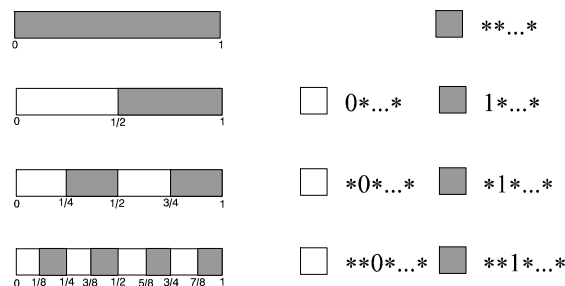
**Obrázok 4.14.** Delenie intervalu  $[0,1]$  podľa schém, ktorých prvé tri znaky sú špecifikované. Napríklad, schéma č. 20 tvaru  $1*0*$ ...určuje reálnu premennú z prieniku intervalov  $[1/2,5/8] \cup [6/8,7/8]$ .

Nech  $x \in [0,1]$  je reálna premenná, ktorá je reprezentovaná binárnym reťazcom dĺžky  $k$  (zatiaľ táto konštanta nebude bližšie špecifikovaná). Na obr. 4.14 sú uvedené jednoduché schémy, ktoré obsahujú v prvých troch polohách 0 alebo 1. Vzájomnou kombináciou týchto prípadov dostaneme  $27$  ( $3^3$ ) možností, v ktorých prvé tri pozície sú obsadené symbolmi 0, 1, alebo \*, pozri obr. 4.15.

Podľa diagramu B obr. 4.13 je globálne minimum funkcie (4.36) 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



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 0011111... .



**Obrázok 4.15.** Postupné delenie intervalu  $[0,1]$  na polovicu, štvrtinu a osminu. V pravej časti obrázku sú uvedené oblasti, ktoré odpovedajú uvedeným schémam.

Tabuľka 4.1 obsahuje prvých šesť iterácií genetického algoritmu pre účelovú funkciu (4.36), pričom populácia obsahuje 20 chromozómov dĺžky  $k=10$ . Sily chromozómov sú počítané pomocou formuly (4.12). Pravdepodobnosti reprodukcie a mutácie sú  $P_{repro}=0.5$  a  $P_{mut}=0.01$ . Jednotlivé kroky reprodukcie budú ilustrované pri prechode z 1. populácie do 2. populácie, rodičovské chromozómy sú vyznačené v 1. populácii tieňovaním, podobne následnícke chromozómy v 2. populácii sú tiež tieňované. Jednoduchými úvahami zistíme, že kríženie bolo uskutočnené od polohy 4, potom prebehla mutácia v druhom chromozóme v polohe osem.

Rodičovské chromozómy:

0	0	1	1	1	0	0	0	0	1
1	0	0	0	0	1	1	1	0	0

Kríženie:

0	0	1	0	0	1	1	1	0	0
1	0	0	1	1	0	0	0	0	1

Mutácia:

(výsledné chromozómy)

0	0	1	0	0	1	1	1	0	0
1	0	0	1	1	0	0	1	0	1

Algoritmus bol inicializovaný (0. populácia) náhodne generovanou populáciou 20 chromozómov - binárnych reťazcov dĺžky 10. Ku každému chromozómu je uvedená hodnota účelovej funkcie  $f$  a hodnota renormalizovanej sily počítanej podľa (4.12). Ruletou je kvázináhodne vybraných 10 rodičovských dvojíc pre reprodukciu, ktorá sa uskutočňuje s pravdepodobnosťou  $P_{repro}=0.5$ . Vidíme, že pre tento jednoduchý ilustračný príklad genetický algoritmus pomerne rýchlo konverguje, už v šiestej iterácii populácia obsahuje 35% chromozómov, ktoré odpovedajú globálnemu minimu ( $f=-0.3776$ ).

**Tabuľka 4.1.** Prvých šesť iterácií - epoch genetického algoritmu pre účelovú funkciu (4.36).

No.	0. populácia			reprodukcia	1. populácia			reprodukcia	2. populácia		
	$f$	$F'$	chromozóm		$f$	$F'$	chromozóm		$f$	$F'$	chromozóm
1	-0.0402	0.0714	0100111101		0.2414	0.0235	0111111011		-0.3213	0.0864	0011100001
2	-0.0323	0.0667	1000011100		-0.3170	0.0751	0011100000		-0.3776	0.0909	0011111100
3	0.6869	0.0143	0110011101		0.5392	0.0094	0001011101		0.1968	0.0182	0010011100
4	-0.2858	0.0857	0100011100		0.7347	0.0047	1100011110		-0.1799	0.0636	1001100101
5	0.2834	0.0238	1101010111		0.0222	0.0469	0010110000		-0.3213	0.0864	0011100001
6	-0.2061	0.0810	1001011101		-0.3213	0.0798	0011100001		-0.3170	0.0682	0011100000
7	-0.3735	0.0952	0011110100		0.1761	0.0329	1010010111		-0.3213	0.0864	0011100001
8	0.0293	0.0524	0000000011		0.1030	0.0376	1010001111		-0.3213	0.0864	0011100001
9	0.8315	0.0048	1100000001		-0.3735	0.0939	0011110100		-0.0323	0.0545	1000011100
10	0.7309	0.0095	1011011000		-0.2061	0.0610	1001011101		0.0222	0.0455	0010110000
11	0.2795	0.0286	0010010010		-0.0323	0.0563	1000011100		-0.0534	0.0591	1000011111
12	0.1761	0.0333	1010010111		-0.3170	0.0751	0011100000		0.0766	0.0273	1010001100
13	-0.1665	0.0762	0100101110		0.0293	0.0423	0000000011		-0.0323	0.0545	1000011100
14	0.0168	0.0571	1111000111		-0.2858	0.0657	0100011100		0.2621	0.0091	0000011100
15	-0.3170	0.0905	0011100000		-0.3735	0.0939	0011110100		0.4961	0.0045	0111100000
16	0.0134	0.0619	0010110001		-0.0323	0.0563	1000011100		0.0293	0.0409	0000000011
17	0.5062	0.0190	0111011011		-0.3721	0.0845	0100000011		0.1030	0.0227	1010001111
18	0.0813	0.0476	1101110000		0.5287	0.0141	0001001011		0.0293	0.0409	0000000011
19	0.0913	0.0429	0101001011		0.2942	0.0188	0101100000		0.0293	0.0409	0000000011
20	0.1030	0.0381	1010001111		0.1968	0.0282	0010011100		0.2414	0.0136	0111111011

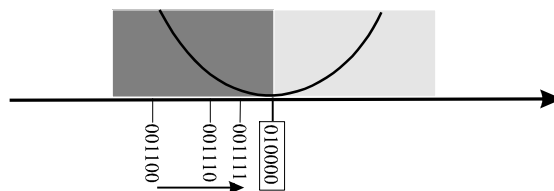
**Tabuľka 4.1.** (Pokračovanie).

No.	3. populácia			reprodukcia	4. populácia			reprodukcia	5. populácia		
	$f$	$F'$	chromozóm		$f$	$F'$	chromozóm		$f$	$F'$	chromozóm

1	0.2711	0.0041	1010100001		-0.3746	0.0539	0011110101		0.8255	0.0040	1011110001
2	-0.2024	0.0290	0011001100		-0.1487	0.0041	1001101100		0.5061	0.0081	000110110
3	-0.3776	0.0830	0011111100		-0.3776	0.0830	0011111100		-0.3746	0.0484	0011110101
4	-0.3213	0.0622	0011100001		-0.3776	0.0830	0011111100		-0.3674	0.0282	0011110000
5	0.1968	0.0083	0010011100		-0.3776	0.0830	0011111100		-0.3692	0.0363	0011110001
6	-0.3213	0.0622	0011100001		-0.3776	0.0830	0011111100		-0.3674	0.0282	0011110000
7	-0.3213	0.0622	0011100001		-0.1837	0.0083	1001100100		-0.1487	0.0121	1001101100
8	-0.3213	0.0622	0011100001		-0.3773	0.0830	0011111101		-0.3674	0.0282	0011110000
9	-0.3776	0.0830	0011111100		-0.2313	0.0249	0011100001		-0.3776	0.0806	0011111100
10	-0.3170	0.0332	0011100000		-0.2313	0.0249	0011100001		-0.3674	0.0282	0011110000
11	-0.0323	0.0166	1000011100		-0.3692	0.0498	0011110001		-0.3776	0.0806	0011111100
12	-0.0534	0.0207	1000011111		-0.3586	0.0373	0011101100		-0.3776	0.0806	0011111100
13	0.0222	0.0124	0010110000		-0.3776	0.0830	0011111100		-0.3692	0.0363	0011110001
14	-0.1799	0.0249	1001100101		-0.2313	0.0249	0011100001		-0.3773	0.0806	0011111101
15	-0.3213	0.0622	0011100001		-0.3586	0.0373	0011101100		-0.3746	0.0484	0011110101
16	-0.3213	0.0622	0011100001		-0.3674	0.0415	0011110000		-0.3776	0.0806	0011111100
17	-0.3213	0.0622	0011100001		-0.3692	0.0498	0011110001		-0.3776	0.0806	0011111100
18	-0.3776	0.0830	0011111100		-0.3586	0.0373	0011101100		-0.3746	0.0484	0011110101
19	-0.3776	0.0830	0011111100		-0.2313	0.0249	0011100001		-0.3776	0.0806	0011111100
20	-0.3776	0.0830	0011111100		-0.3776	0.0830	0011111100		-0.3776	0.0806	0011111100

## 4.6 Hammingova bariéra

Uvažujme genetický algoritmus v ktorom sú reálne premenné účelovej funkcie binárne kódované štandardným spôsobom (pozri kapitolu 1.2). Dva chromozómy  $\alpha=(00100\dots0)$  a  $\beta=(000111\dots1)$  reprezentujú dve susedné celé čísla (ktorým sa priradia dve "susedné" reálne premenné). Čo je zaujímavé na týchto chromozómoch, je skutočnosť, že aj keď reprezentujú dve "susedné" čísla, ich Hammingovská vzdialenosť (t.j. počet rôznych bitových premenných) je veľká (v limitnom prípade sa môže rovnať až dĺžke binárneho reťazca). Predpokladajme, že optimálne riešenie odpovedá binárnemu reťazcu  $\alpha=(00100\dots0)$  a že populácia obsahuje chromozómy, ktoré sú blízke alebo totožné s binárnym reťazcom  $\beta=(000111\dots1)$ . Aká môže byť v tomto prípade evolúcia chromozómov populácie? Chromozómy populácie ( $00011\dots\dots$ ) 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 ( $00011\dots\dots$ ) mutáciou na ( $00100\dots\dots$ ) je veľmi malá, preto sa v populácii budú s najväčšou pravdepodobnosťou vyskytovať len chromozómy odpovedajúce schéme ( $00011\dots\dots$ ). To znamená, že evolúcia populácie chromozómov, ktorá bola inicializovaná chromozómami typu ( $00011\dots\dots$ ), má len veľmi malú šancu byť smerovaná do výslednej populácie, ktorá by obsahovala optimálne riešenie  $\alpha=(00100\dots0)$ , a s najväčšou pravdepodobnosťou skončí u riešenia  $\beta=(000111\dots1)$  (pozri obr. 4.16). 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.



**Obrázok 4.16.** Známenie evolúcie riešenia typu 0011\*\* k presnému riešeniu 010000. V priebehu evolúcie genetického algoritmu populácia bude obsahovať stále viac a viac chromozómov 001111, ktoré sú blízke optimálnemu riešeniu 010000.

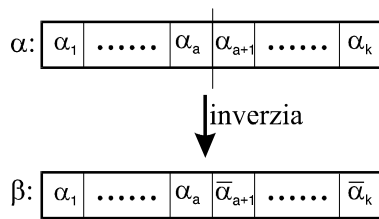
Ako sa vyhnúť Hammingovej bariére v genetickom algoritme? Tento problém môže byť riešený dvoma prístupmi. Prvý prístup je založený na použití ďalšieho operátora transformácie chromozómov, a to tzv. *inverzného operátora*. Tento operátor zmení binárny reťazec  $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$  na iný binárny reťazec  $\beta = (\beta_1, \beta_2, \dots, \beta_k)$

$$\beta = O_{inv}(\alpha) \quad (4.37)$$

Táto operácia je definovaná tak, že pre náhodne vygenerované celé číslo  $1 < a < k$  (bod inverzie), komponenty binárneho vektora pre indexy  $i > a$  sú zamenené za ich komplementy (pozri obr. 4.17 a algoritmus 4.6)

$$\beta_i = \begin{cases} \alpha_i & (\text{pre } i \in [1, a]) \\ 1 - \alpha_i & (\text{pre } i \in [a + 1, k]) \end{cases} \quad (4.38)$$

Stochastický operátor inverzie odstraňuje vznikajúce Hammingove bariéry a dáva šancu genetickému algoritmu evolúciou zostrojiť populáciu, ktorá obsahuje chromozómy zodpovedajúce optimálnemu riešeniu. Kde implementovať operátor inverzie v genetickom algoritme? Najjednoduchšie je ho prezentovať ako časť reprodukčného operátora. Algoritmus 4.4 bude rozšírený o dve operácie inverzie, ktoré sa vykonávajú s pravdepodobnosťou  $P_{inv}$ . V prípade, že chromozóm binárne reprezentuje  $n$  reálnych premenných účelovej funkcie, pričom každá premenná je kódovaná binárnym reťazcom dĺžky  $k$ , operácia inverzie sa vykoná zvlášť pre každý podreťazec odpovedajúci jednotlivým premenným účelovej funkcie, pozri obr. 4.18.



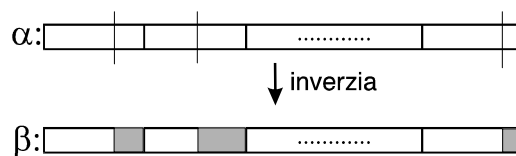
**Obrázok 4.17.** Schematické znázornenie operácie inverzie. Pre náhodne vybraný bod inverzie  $a$  vykonáme inverziu binárneho vektora  $\alpha$  tak, že od bodu  $a+1$  až do konca binárneho vektora nahradíme jeho zložky komplementami,  $\bar{\alpha}_i = 1 - \alpha_i$ .

```

procedure Inversion(input: $\alpha$ ; output: $\beta$ );
begin if random <  $P_{inv}$  then
  begin  $a := 1 + \text{random}(k-1)$ ;
    for  $i := 1$  to  $a$  do  $\beta_i := \alpha_i$ ;
    for  $i := a+1$  to  $k$  do  $\beta_i := 1 - \alpha_i$ ;
  end else for  $i := 1$  to  $k$  do  $\beta_i := \alpha_i$ ;
end;

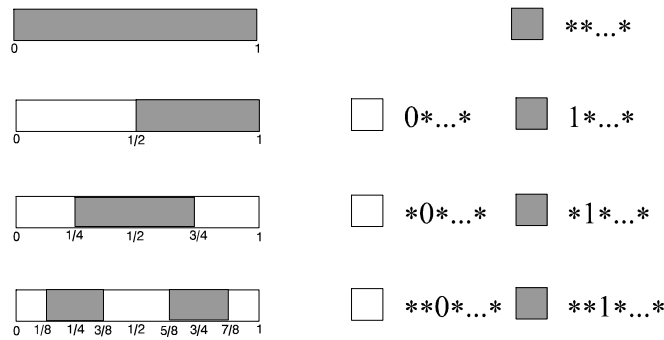
```

**Algoritmus 3.6.** Pseudopascalovská implementácia inverzie binárneho vektora dĺžky  $k$ . Pravdepodobnosť  $P_{inv}$  (malé kladné číslo, napr.  $P_{inv} = 0.01 - 0.001$ ) určuje, či sa vykoná inverzia alebo nie.

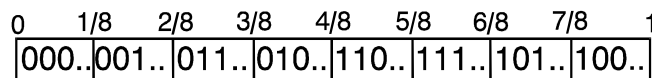


**Obrázok 4.18.** Realizácia operácie inverzie pre binárny vektor reprezentujúci  $n$  reálnych premenných, pre každý podreťazec sa vykoná operácie inverzie separátne.

Druhý prístup na odstránenie Hammingovej bariéry je použitie Grayovho kódu pre binárnu reprezentáciu reálnych premenných (pozri kapitolu 1.2). Delenie jednotkového intervalu  $[0,1]$  na podintervaly v rámci Grayovho kódovania binárnych reťazcov je znázornené na obr. 4.18 a obr. 4.19. Čo je dôležité v tomto prípade, susedné intervaly sú kódované binárnymi reťazcami, ktoré nie sú diametrálne odlišné medzi sebou, ako to môže byť vtedy, keď je použité štandardné kódovanie. Pre lepšie pochopenie tejto myšlienky sa vráťme k obr. 4.12, diagram B. Na tomto obrázku je znázornený priebeh účelovej funkcie (4.36), pričom globálne minimum tejto funkcie je blízko zľava v bode  $1/4$ . V štandardnom kódovaní intervaly  $[1/8, 1/4]$  a  $[1/4, 3/8]$  sú reprezentované reťazcami 001... resp. 010... , pričom optimálne riešenie má tvar 001111... . Vidíme, že keď sa blížíme k optimálnemu riešeniu z pravého intervalu, tak potom v určitom momente musí nastať dramatická zmena zloženia binárneho reťazca, a to z 010000... na 001111... . Genetický algoritmus s normálne nastavenou pravdepodobnosťou mutácie ( $P_{mut} = 0.001 - 0.01$ ) nemá šancu prekonať túto "Hammingovu bariéru". V prípade, že použijeme Grayovo kódovanie, situácia sa podstatne zlepšuje v tom zmysle, že nedochádza k tak dramatickým zmenám zloženia chromozómov pri prechode z jedného podintervalu na druhý (pozri obr. 4.19).



**Obrázok 4.18.** Postupné delenie intervalu  $[0,1]$  pre binárnu reprezentáciu pomocou Grayovho kódovania pre rôzne schémy. V pravej časti obrázku sú uvedené oblasti, ktoré odpovedajú uvedeným schémam.



**Obrázok 4.19.** Delenie intervalu  $[0,1]$  na osem podintervalov použitím Grayovho kódovania binárnych reťazcov.

V tomto momente musíme odpovedať na otázku, čo je lepšie, či štandardné kódovanie alebo Grayovo kódovanie binárnych reťazcov v genetickom algoritme. Dať odpoveď na túto otázku nie je jednoduché. V štandardnej reprezentácii existuje rýchla transformácia binárneho reťazca na reálne číslo, avšak za cenu potenciálnej existencie Hammingových bariér. V prvej časti tejto podkapitoly bolo ukázané, že tento problém štandardnej reprezentácie môže byť odstránený zavedením operátora inverzie do genetického algoritmu ako súčasť reprodukčného procesu. Grayova reprezentácia odstraňuje problém s Hammingovou bariérou, táto je jednoducho odstránená tým, že v tomto kódovaní susedné intervaly reálnych čísel sú kódované podobnými binárnymi reťazcami (na rozdiel od štandardného kódovania). Použitie Grayovho kódovania v genetickom algoritme naráža na problém numerickej náročnosti. Na rozdiel od štandardného kódovania, Grayovo kódovanie pri konštrukcii reálnych čísel požaduje ešte medzitransformáciu binárneho reťazca do štandardného kódovania. Táto medzitransformácia nám môže podstatne predĺžiť CPU čas genetického algoritmu, pretože transformácia binárneho reťazca na reálne číslo patrí medzi najčastejšie volanú procedúru algoritmu, čiže už malé zvýšenie jej časovej náročnosti môže mať dramatický dopad na celkový CPU čas.

## 4.7 "Messy" chromozómy (M-chromozómy)

V r. 1989 sa Goldberg so spolupracovníkmi [10,11] pokúšali prekonať niektoré obtiaže genetického algoritmu pri hľadaní globálneho minima tzv. *deceptívnych účelových funkcií* (multimodálne funkcie s mnohými minimami ale len s jedným globálnym minimom, ktoré je oddelené od ostatných minim bariérou) pomocou "messy" genetického algoritmu. Hlavná myšlienka tejto modifikácie genetického algoritmu spočíva okrem iného v použití m-chromozómov, ktoré majú premennú dĺžku a ich komponenty sú špecifikované súčasne tak indexom ako aj hodnotou. Na rozdiel od štandardných chromozómov, špecifikovaných ako binárne vektory fixnej dĺžky, m-chromozómy môžu mať premennú dĺžku a byť tak "preurčené", ako aj "podurčené". Táto flexibilita m-chromozómov obvykle podstatne zvyšuje efektivitu a robustnosť evolučných algoritmov, ktoré ich využívajú [10,11].

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

$$\chi = ((\mu_1, v_1), (\mu_2, v_2), \dots, (\mu_l, v_l)) \in Q^l \quad (4.39)$$

Tento m-chromozóm je dekódovaný na jednoduchý binárny vektor dĺžky  $k$  nasledujúcim postupom (pozri obr. 4.21): Idúc postupne zľava doprava, prvá dvojica  $(\mu_1, v_1)$  špecifikuje hodnotu  $v_1$  v polohe

$\mu_1, \alpha_{\mu_1} = v_1$ . Druhá dvojica  $(\mu_2, v_2)$ , ak  $\mu_1 \neq \mu_2$ , špecifikuje hodnotu  $v_2$  v polohe  $\mu_2, \alpha_{\mu_2} = v_2$ . Vo všeobecnosti platí

$$\alpha_{\mu_i} = v_i \quad (\text{ak } \forall j \leq i-1: \alpha_j \neq \alpha_i) \quad (4.40)$$

To znamená, že problém preurčenia je riešený pravidlom, že určujúca je tá dvojica, ktorá sa vyskytuje prvá, idúc cez  $m$ -chromozóm zľava doprava. Problém podurčenia (t.j. niektorá poloha v zostrojenom binárnom vektore nie je špecifikovaná) sa rieši pomocou templátu

$$\mathbf{t} = (t_1, t_2, \dots, t_k) \in \{0,1\}^k \quad (4.41)$$

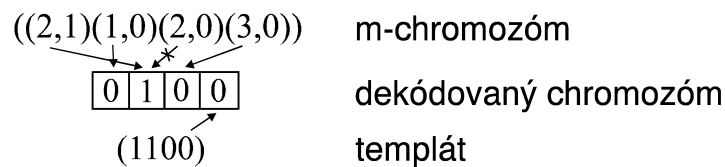
Ak  $p$ -ta poloha nie je špecifikovaná (4.40), t.j.  $p \neq \alpha_i$  pre  $\forall i \leq l$ , potom táto poloha je určená  $p$ -tou polohou v templáte  $\mathbf{t}$

$$\alpha_p = t_p \quad (\text{pre každú nešpecifikovanú polohu}) \quad (4.42)$$

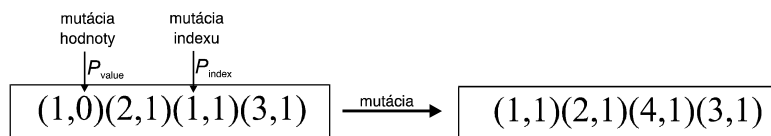
Problém dekódovania  $m$ -chromozómu  $\chi$  pre daný templát  $\mathbf{t}$  môže byť vo všeobecnosti chápaný ako zobrazenie  $G_t: Q^l \rightarrow \{0,1\}^k$

$$\alpha = G_t(\chi) \quad (4.43)$$

Pre lepšie pochopenie vyššie uvedeného procesu dekódovania  $m$ -chromozómu pozri obr. 4.22, kde je uvedený jednoduchý ilustračný príklad.



**Obrázok 4.21.** 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.



**Obrázok 4.22.** Mutácia  $m$ -chromozómu má dve časti, mutáciu indexu a mutáciu binárnej hodnoty. Každá takáto mutácia sa vykonáva s pravdepodobnosťou  $P_{index}$  resp.  $P_{value}$ .

$M$ -chromozóm  $\chi$  je transformovaný na iný  $m$ -chromozóm  $\chi'$  operáciou mutácie (pozri obr. 4.22)

$$\chi' = O_{mut}(\chi) \quad (4.44)$$

kde  $\chi = ((\mu_1, v_1), (\mu_2, v_2), \dots, (\mu_l, v_l))$  a  $\chi' = ((\mu'_1, v'_1), (\mu'_2, v'_2), \dots, (\mu'_l, v'_l))$ . Stochastický operátor mutácie  $O_{mut}$  sa realizuje tak, že idúc zľava doprava postupne náhodne meníme usporiadané dvojice, t.j. indexy a binárne hodnoty

$$\mu'_i = \begin{cases} \hat{\mu}_i & (\text{ak } random < P_{index}) \\ \mu_i & (\text{ostatné prípady}) \end{cases} \quad (4.45a)$$

kde  $1 \leq \hat{\mu}_i \leq k$  je náhodné celé číslo s rovnomernou distribúciou pravdepodobnosti, pričom  $\hat{\mu}_i \neq \mu_i$ . Pravdepodobnosť  $P_{index}$  určuje stochastičnosť zmeny indexovej časti  $m$ -chromozómu. Podobným spôsobom sa mení aj binárna časť  $m$ -chromozómu

$$v'_i = \begin{cases} 1 - v_i & (\text{ak } random < P_{value}) \\ v_i & (\text{ostatné prípady}) \end{cases} \quad (4.45b)$$

Pravdepodobnosť  $P_{value}$  určuje stochastičnosť zmeny binárnej časti m-chromozómu. Operácia mutácie je ilustrovaná príkladom na obr. 4.22. Pseudopascalovská implementácia mutácie je uvedená v algoritme 4.7.

```

procedure M_Mutation(input:  $\chi$ ; output:  $\chi'$ );
begin for i:=1 to l do
  begin  $\mu'_i := \mu_i$ ;  $v'_i := v_i$ ;
    if random <  $P_{index}$  then  $\mu'_i := 1 + \text{random}(k)$ ;
    if random <  $P_{value}$  then  $v'_i := 1 - v_i$ ;
  end;
end;

```

**Algoritmus 4.7.** Procedúra mutácie m-chromozómu.

Operácia kríženia dvoch rodičovských m-chromozómov  $\chi = ((\mu_1, v_1), (\mu_2, v_2), \dots, (\mu_{l_\chi}, v_{l_\chi}))$  a  $\kappa = ((\xi_1, \zeta_1), (\xi_2, \zeta_2), \dots, (\xi_{l_\kappa}, \zeta_{l_\kappa}))$  poskytuje dva nové m-chromozómy - potomkov  $\chi' = ((\mu'_1, v'_1), (\mu'_2, v'_2), \dots, (\mu'_{l_\chi}, v'_{l_\chi}))$  a  $\kappa' = ((\xi'_1, \zeta'_1), (\xi'_2, \zeta'_2), \dots, (\xi'_{l_\kappa}, \zeta'_{l_\kappa}))$

$$(\chi', \kappa') = O_{cross}(\chi, \kappa) \quad (4.46)$$

Táto operácia sa realizuje tak, že v prvom kroku sa náhodne generuje "bod kríženia"  $1 < a < \min\{l_\chi, l_\kappa\}$  a potom sa tvoria nové m-chromozómy tak, že až po "bod kríženia" a sa jednoducho okopírujú rodičovské chromozómy, za týmto bodom si nové m-chromozómy vymenia zvyšné časti, pozri obr. 4.22 a algoritmus 4.8.



**Obrázok 4.23.** Operácia kríženia dvoch m-chromozómov, tieto si vymenia svoje časti nasledujúce za náhodne generovaným bodom kríženia. Operácia kríženia mení, vo všeobecnosti, dĺžku m-chromozómov.

Pomocou procedúr M\_Mutation a M\_Crossover (pozri algoritmy 4.7 a 4.8) môžeme jednoducho modifikovať algoritmus 4.4 (procedúra Reproduction) realizujúci reprodukciu dvoch rodičovských chromozómov na dva nové chromozómy - potomkov, stačí vymeniť procedúry pre mutáciu a kríženie za podobné procedúry pre m-chromozómy.

```

procedure M_Crossover(input:  $\chi, \kappa$ ; output:  $\chi', \kappa'$ );
begin a:=1+random(min( $l_\chi, l_\kappa$ )-1);
  for i:=1 to a do
    begin  $\mu'_i := \mu_i$ ;  $v'_i := v_i$ ;
       $\xi'_i := \xi_i$ ;  $\zeta'_i := \zeta_i$ ;
    end;
   $l_{\chi'} := l_\kappa$ ;  $l_{\kappa'} := l_\chi$ ;
  for i:=a+1 to  $l_{\chi'}$  do begin  $\mu'_i := \xi_i$ ;  $v'_i := \zeta_i$  end;
  for i:=a+1 to  $l_{\kappa'}$  do begin  $\xi'_i := \mu_i$ ;  $\zeta'_i := v_i$  end;
end;

```

**Algoritmus 4.8.** Procedúra kríženia dvoch rodičovských m-chromozómov  $\chi$  a  $\kappa$  na dva nové m-chromozómy - potomkov  $\chi'$  a  $\kappa'$ .



```

procedure M_Genetic_Algorithm(input:timemax; output:αopt);
begin 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 χ, κ ∈ P;
              if random < Prepro then
                Reproduction(χ, κ, χ', κ') else
                  begin χ' := χ; κ' := κ end;
                Q := Q ∪ {χ', κ'};
            end;
          P := Q; t := arg minχ ∈ P (Gt(χ));
          if convergence criteria are fulfilled then
            stop_criterion := true;
        end;
      αopt := t;
end;

```

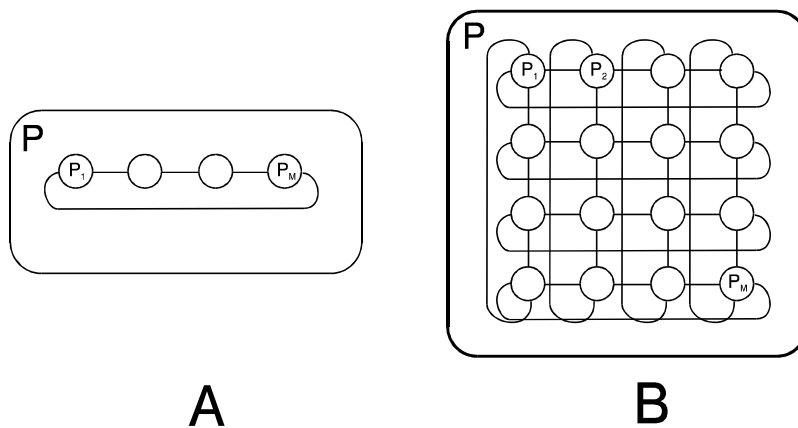
**Algoritmus 4.9.** Pseudopascalovská implementácia genetického algoritmu (pozri algoritmus 4.5) pre m-chromozómy. Algoritmus je inicializovaný náhodným generovaním populácie m-chromozómov a binárneho vektora - templátu. Templát sa obnovuje v každom iteračnom kroku pomocou binárnej interpretácie najlepšieho chromozómu z predchádzajúcej iterácie. Na záver algoritmu sa templát použije ako výstupné riešenie získané genetickým algoritmom s m-chromozómami.

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 (pozri (4.43) a obr. 4.19). 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).

**Príklad 4.7.** Implementujte genetický algoritmus podľa algoritmu 4.5, pričom miesto chromozómov sa používajú "messy" chromozómy z kapitoly 4.7. Porovnajte účinnosť takto modifikovaného "messy" genetického algoritmu s obyčajnou verziou genetického algoritmu.

## 4.8. Paralelný genetický algoritmus

Genetický algoritmus tak ako bol prezentovaný v kapitole 4.3 je sekvenčný algoritmus. V literatúre existuje mnoho verzií tzv. **paralelného genetického algoritmu** (PGA) [12], 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.



**Obrázok 4.23.** Schematické znázornenie (disjunktného) rozkladu populácie  $P$  na podpopulácie  $P_1, P_2, \dots, P_M$ ,  $P = P_1 \cup P_2 \cup \dots \cup P_M$  a  $P_i \cap P_j = \emptyset$ , pre  $i \neq j$ . Diagram A znázorňuje taký rozklad, kde každá podpopulácia má práve dvoch susedov, "pravého" a "ľavého". Diagram B znázorňuje taký rozklad populácie, že každá podpopulácia má práve štyroch susedov, "pravého" a "ľavého" a "dolného" a "horného". Zvolené "toroidovské" topológie 1D (diagram A) a 2D (diagram B) boli zvolené tak, aby neexistovali "okrajové" efekty, t.j. každá podpopulácia mala rovnaký počet susedov.

Nech populácia  $P$  je rozdelené na vzájomne disjunktné podpopulácie

$$P = P_1 \cup P_2 \cup \dots \cup P_M \quad (4.47)$$

Obvykle sa k tomuto rozkladu definuje ešte tzv. topológia, ktorá popisuje susedov danej podpopulácie. Formálne, táto topológia rozkladu (4.47) je definovaná pomocou neorientovaného súvislého grafu, ktorý obsahuje  $M$  vrcholov - podpopulácií. Ak dva vrcholy sú spojené hranou, potom hovoríme, že podpopulácie sú susedné (pozri obr. 4.23). V prípade, že neuvažujeme susednosť medzi podpopuláciami, budeme formálne predpokladať, že graf je úplný, t.j. každý vrchol susedí s ostatnými vrcholmi.

```

procedure One_generation GA(input  $P_{ini}$ ; output  $P_{fin}$ );
begin  $P_{fin} := \emptyset$ ;
  while  $|P_{fin}| < |P_{ini}|$  do
    begin select by roulette wheel two
      chromosomes  $\alpha_1, \alpha_2 \in P_{ini}$ ;
      if random  $< P_{repro}$  then
        Reproduction( $\alpha_1, \alpha_2, \alpha'_1, \alpha'_2$ ) else
          begin  $\alpha'_1 := \alpha_1$ ;  $\alpha'_2 := \alpha_2$  end;
           $P_{fin} := P_{fin} \cup \{\alpha'_1, \alpha'_2\}$ ;
    end;
end;

```

**Algoritmus 4.10.** Procedúra pre realizáciu jedného kroku genetického algoritmu (pozri Algoritmus 4.11). Vstupná populácia  $P_{ini}$  sa zmení na výstupnú populáciu  $P_{fin}$ .

```

procedure Parallel_GA(input  $t_{\max}$ ; output  $\alpha_{\text{opt}}$ );
begin  $t:=0$ ; stop_criterion:=false;
  for  $i:=1$  to  $M$  do
     $P_i:=\{\text{randomly generated subpopulation of chromosomes}\}$ ;
    while ( $t < t_{\max}$ ) and (not stop_criterion) do
      begin  $t:=t+1$ ;
        for  $i:=1$  to  $M$  do
          begin One_generation GA( $P_i, P'_i$ );
             $P_i:=P'_i$ ;
          end;
          if random  $< P_{\text{exch}}$  then
            begin randomly are selected two
              subpopulations  $P_a$  and  $P_b$ ;
              exchange( $P_a, P_b$ );
            end;
          if convergence criteria are satisfied then
            stop_criterion:=true;
        end;
       $\alpha_{\text{opt}}:=\text{best chromosome of all subpopulations}$ 
         $\{P_1, P_2, \dots, P_M\}$ ;
    end;
end;

```

**Algoritmus 4.11.** Pseudopascalovská implementácia paralelného genetického algoritmu. Algoritmus je inicializovaný náhodnou generáciou podpopulácií  $P_1, P_2, \dots, P_M$ . Vonkajší while-cyklus sa opakuje predpísaný počet generácií ( $t_{\max}$ ) alebo pokiaľ boolovská premenná stop\_criterion prestane mať hodnotu "false". Vo vnútri tohto cyklu, separátne pre každú podpopuláciu sa realizuje proces jednej generácie, v ktorom aktuálna podpopulácia sa reprodukčným procesom (pozri Algoritmus 4.10) zmení na novú podpopuláciu. Stochasticky sa realizuje (s pravdepodobnosťou  $P_{\text{exch}}$ ) výmena chromozómov medzi dvoma náhodne vybranými podpopuláciami. V tomto bode existujú veľké odchýlky medzi rôznymi implementáciami paralelného genetického algoritmu. V prípade, že rozklad populácie na podpopulácie je "štruktúrovaný" (pozri obr. 4.22), potom podpopulácie  $P_a$  a  $P_b$  sú susedné. Aktivácia procedúry exchange( $P_a, P_b$ ) má za výsledok, že podpopulácie si vymenia chromozómy.

Jednoduchá verzia paralelného genetického algoritmu spočíva v tom, že pre každú podpopuláciu prebieha nezávislá evolúcia, pričom s určitou malou pravdepodobnosťou si podpopulácie vymenia chromozómy (pozri Algoritmus 4.11). Výmena chromozómov je v algoritme 4.11 implementovaná pomocou procedúry exchange( $P_a, P_b$ ). Výsledok jej aktivácie nie je bližšie popísaný, ako príklad uvedieme túto možnú realizáciu: Podpopulácie  $P_a$  a  $P_b$  sú náhodne vybrané tak, aby boli susedné. V populácii  $P_a$  kvázináhodne vyberieme najlepší chromozóm, potom tento vytesní v populácii  $P_b$  kvázináhodne vybraný najslabší chromozóm. Akt výmeny chromozómov medzi dvoma podpopuláciami je plne stochastický a uskutočňuje sa s pravdepodobnosťou  $P_{\text{exch}}$ .

## Literatúra

- [1] Ch. Darwin: Darwinův původ druhů v ilustracích. Panorama, Praha 1989.
- [2] S. Wright: Evolution of Mendelian Populations. Genetics **16** (1931) in 97.
- [3] J.H. Holland: Adaptation in Natural and Artificial Systems. The University of Michigan Press, Ann Arbor, MI, 1975.
- [4] D.E. Goldberg: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, MA, 1989.
- [5] M. Mitchel: An introduction to Genetic Algorithms. MIT Press, Boston, MA, 1996.
- [6] N. Wirth: Algoritmy a štruktúry údajov. Alfa, Bratislava 1988.
- [7] T. Bäck: Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York 1996.
- [8] R.F. Hartl: A Global Convergence Proof for a Class of GAs. Technische Universität Wien, 1990.
- [9] A.E. Eiben, E.H.L. Aarts, K.M. Van Hee: Global Convergence of GAs. An Infinite Markov Chain Analysis. In Parallel Problem Solving from Nature I, Lecture Notes in Computer Science, Vol. 496, Springer Verlag, Berlin, 1991. (Ed. by H.P. Schwefel and R. Männer).
- [10] D.E. Goldberg, B. Korb, K. Deb: Messy Genetic Algorithms. Motivations, Analysis, and First Results. Complex Systems **3** (1989) 493.

- [11] D.E. Goldberg, K. Deb, B. Korb: Messy Genetic Algorithms Revisited. Nonuniform Size and Scale. *Complex Systems* **4** (1990) 415.
- [12] E. Cantu-Paz: A Survey of Parallel GAs (1997) Illinois Genetic Algorithms Laboratory. University of Illinois at Urbana-Champaign. Available at anonymous ftp server `ftp-illigal.ge.uiuc.edu/pub/papers/Illigals/97003.ps.Z`