

2. kapitola

Základné pojmy evolučných algoritmov - optimalizačný problém a kódovanie

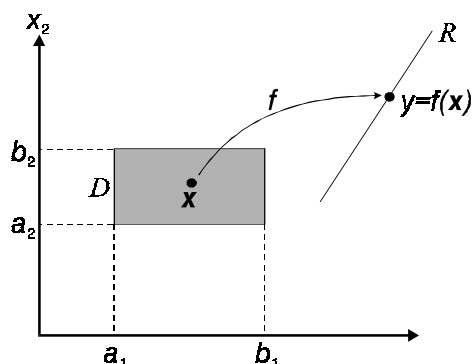
2.1 Optimalizačný problém

Nech funkcia

$$f: D \rightarrow R \quad (2.1a)$$

$$D = \prod_{i=1}^n [a_i, b_i] = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n] \quad (2.1b)$$

zobrazuje n -rozmernú kocku D (karteziánsky produkt uzavretých intervalov $[a_i, b_i]$) na reálne číslo $y \in R$, (pozri obr. 2.1).



Obrázok 2.1. Schematické znázornenie zobrazenia f , ktoré priradí každému n -rozmernému vektoru $x \in D$ reálne číslo $y \in R$

Táto funkcia je ohraničená dvoma podmienkami:

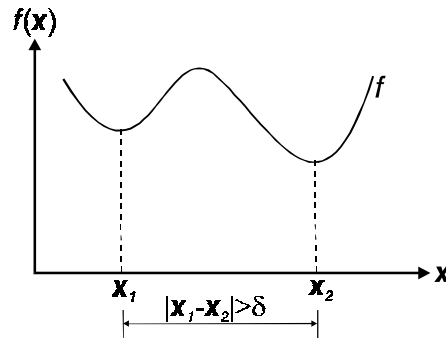
(1) Existuje taký algoritmus, ktorý funkciu f "vypočíta dostatočne rýchlo" s požadovanou presnosťou pre každé $x \in D$ (hovoríme, že funkcia f je *dobře vypočítateľná*).

(2) Pre každú dvojicu lokálnych miním $x_1, x_2 \in D$ vzdialenosť $|x_1 - x_2|$ je väčšia ako dané kladné číslo $\delta > 0$, $|x_1 - x_2| > \delta$ (pozri obr. 2.2). Podmienka ohraničuje zhora počet lokálnych miním funkcie f , ktoré sa vyskytujú na kocke D . Nie je možné, aby sa v ľubovoľnom okolí minima funkcie vyskytovalo iné minimum, pre určité malé okolie minima funkcie vyššie uvedená podmienka $|x_1 - x_2| > \delta$ by prestala platiť. Podmienka automaticky vylučuje z triedy prípustných funkcií tie funkcie, ktoré sú "fraktálového" typu, t.j. v každom okolí nejakého minima sa nachádza aspoň jedno iné minimum.

Globálne minimum funkcie f na kocke D je určené vzťahom

$$x_{opt} = \arg \min_{x \in D} f(x) \quad (2.2)$$

Nájdienie globálneho minima použitím klasických optimalizačných metód [1-3] (gradientových a negradientových) patrí medzi obtiažne numerické problémy pre funkcie, ktoré nie sú ohraničené ďalšími podmienkami (ako napr. že $f(x)$ je konvexná funkcia na oblasti D). Z týchto dôvodov sa v súčasnosti [4,5] pri riešení problému (2.2) často používajú tzv. *evolučné optimalizačné algoritmy*, ktoré poskytujú riešenia blízke globálnemu, alebo s ním totožné.

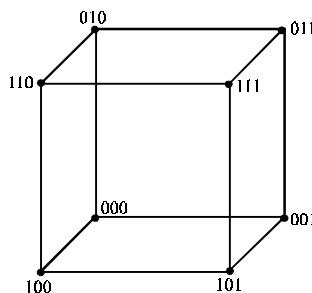


Obrázok 2.2. Ilustrácia druhej podmienky kladenej na funkciu f . Vzďalenosť medzi každou dvojicou miním x_1 a x_2 je väčšia ako predpísané kladné číslo δ , $|x_1 - x_2| > \delta$.

Príklad 2.1. Nech funkcia $f(x)$ má tvar

$$f(x) = e^{-0.01x^2} \sin(mx)$$

pre $m=2,5,10$ a $x \in [-10,10]$. Použitím programových systémov typu MAPLE [6] alebo MATHEMATICA [7] nakreslite priebeh týchto funkcií a nájdite niekoľko prvých najnižších miním (včítane globálneho). Zistite horné odhady konštánt δ (pozri podmienku 2 z kapitoly 2.1) funkcie $f(x)$ pre rôzne hodnoty parametrov m .



Obrázok 2.3. Geometrická interpretácia binárnych vektorov dĺžky $k=3$ pomocou vrcholov kocky. Dva susedné vrcholy odpovedajú binárnym vektorom, ktorých Hammingova vzdialenosť je jednotková (t.j. vektory sa líšia len v jednej polohe).

Binárna verzia funkcie (2.1) má tvar

$$f: \{0,1\}^k \rightarrow R \tag{2.3a}$$

Táto funkcia je definovaná nad množinou binárnych vektorov dĺžky k , každému binárnemu vektoru zobrazenie f priradí reálne číslo z množiny R

$$y = f(\alpha) \quad (2.3b)$$

Kardinalita množiny binárnych vektorov dĺžky k je určená vzťahom

$$|\{0,1\}^k| = 2^k \quad (2.4)$$

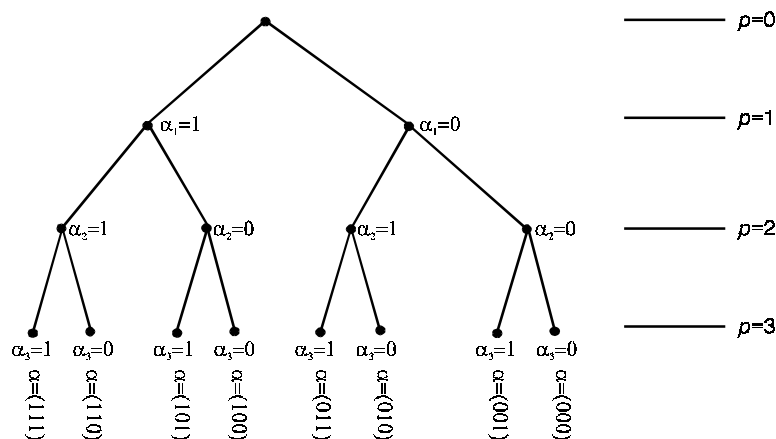
To znamená, že "dimenzia" priestoru všetkých možných binárnych vektorov dĺžky k rastie exponenciálne s dĺžkou k . Jednoduchá geometrická interpretácia binárnych vektorov dĺžky $k=3$ je znázornená na obr. 2.3.

Príklad 2.2. Nakreslite analóg kocky na obr. 2.3 pre $k=4$.

Príklad 2.3. Napíšte program (Pascal alebo C) na systematické prehľadávanie všetkých binárnych vektorov dĺžky k pomocou metódy spätného prehľadávania (backtracking) [8] (pozri Algoritmus 2.1).

```
procedure Backtrack_Searching(output :  $f_{opt}, \alpha_{opt}$ );
begin  $f_{opt} := \infty$ ;  $p := 1$ ;  $U_1 := 1$ ;
  while  $p \geq 1$  do
    if  $U_p \geq 0$  then
      begin  $\alpha_p := U_p$ ;  $U_p := U_p - 1$ ;
        if ( $p=k$ ) and ( $f(\alpha) < f_{opt}$ ) then
          begin  $f_{opt} := f(\alpha)$ ;  $\alpha_{opt} := \alpha$  end else
            if  $p < k$  then
              begin  $p := p+1$ ;  $U_p := 1$  end;
            end else  $p := p-1$ ;
      end
    end;
end;
```

Algoritmus 2.1. Implementácia metódy spätného prehľadávania pre riešenie optimalizačného problému (2.5). Pribeh algoritmu pre $k=3$ je znázornený na obrázku 2.4.



Obrázok 2.4. Diagramatické znázornenie algoritmu 2.1 spätného prehľadávania pomocou binárneho stromu.

Analóg optimalizačného problému (2.2) pre binárne vektory má tvar

$$\alpha_{opt} = \arg \min_{\alpha \in \{0,1\}^k} f(\alpha) \quad (2.5)$$

Vo všeobecnosti, globálne optimum α_{opt} sa nájde po preskúšaní všetkých možných binárnych vektorov dĺžky k . Obrázne povedané, problém (2.5) vyriešime tak, že sa určitým systematickým spôsobom pohybujeme po vrcholoch k -rozmernej hyperkocky (pozri obr. 2.3) tak, že navštívime všetkých 2^k vrcholov. Algoritmicky tento prístup môže byť implementovaný pomocou metódy spätného prehľadávania [8], pozri algoritmus 2.1. CPU čas potrebný na riešenie optimalizačnej úlohy (2.5) je potom úmerný kardinalite priestoru riešení

$$t_{CPU} \propto 2^k \quad (2.6)$$

2.2 Binárna reprezentácia reálnej premennej

Nech binárny vektor α dĺžky k

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k) \in \{0,1\}^k \quad (2.7)$$

je interpretovaný ako celé číslo

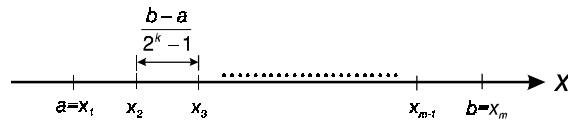
$$\begin{aligned} \text{int}(\alpha) &= \sum_{i=1}^k \alpha_i 2^{k-i} = \\ &= \alpha_1 2^{k-1} + \alpha_2 2^{k-2} + \dots + \alpha_{k-1} 2 + \alpha_k \end{aligned} \quad (2.8)$$

K tomuto celému číslu jednoduchým spôsobom priradíme reálne číslo, ktoré môže byť chápané ako aproximácia reálneho čísla $x \in [a,b]$

$$x \approx \text{real}(\alpha) = a + \frac{b-a}{2^k - 1} \text{int}(\alpha) \quad (2.9)$$

Táto konštrukcia racionálneho čísla $a \leq \text{real}(\alpha) \leq b$ z binárneho reťazca α dĺžky k je formálne interpretovaná ako "transformácia" binárnej reprezentácie na "reálnu" reprezentáciu, kde zostrojené racionálne číslo $\text{real}(\alpha)$ aproximuje požadované reálne

číslo x s presnosťou $(b-a)/2^k-1$. Interval $[a,b]$ obsahuje $m=2^k$ bodov $x_1=a$, $x_2=a+(b-a)/(2^k-1)$, ..., $x_i=a+(i-1)(b-a)/(2^k-1)$, ..., $x_n=b$, pozri obr. 2.5 a tab. 2.1.



Obrázok 2.5. Reálna premenná x z uzavretého intervalu $[a,b]$ v binárnej reprezentácii je aproximovaná bodmi x_1, x_2, \dots, x_m , kde počet bodov $m=2^k$. Presnosť tejto aproximácie je určená dĺžkou binárnych reťazcov k , vzdialenosť medzi dvoma susednými bodmi je $(b-a)/(2^k-1)$.

Tabuľka 2.1.

No.	α	int(α)	real(α)	$\tilde{\alpha}$ (Gray)
1	000	0	0	000
2	001	1	1/7	001
3	010	2	2/7	011
4	011	3	3/7	010
5	100	4	4/7	110
6	101	5	5/7	111
7	110	6	6/7	101
8	111	7	1	100

Inverzná transformácia k formule (2.9) má tvar

$$\text{int}(\alpha) = \left\lceil \frac{x-a}{b-a} (2^k - 1) \right\rceil \quad (2.10)$$

kde symbol $\lceil x \rceil$ je celá časť reálneho čísla x (napr. $\lceil 1.1 \rceil = 1$ a $\lceil 1.9 \rceil = 1$). V dôsledku toho, že $a \leq x \leq b$, zlomok $(x-a)/(b-a)$ leží v uzavretom intervale $[0,1]$, ak $\text{int}(\alpha)=0$, potom $x=a$, ak $\text{int}(\alpha)=2^k-1$, potom $x=b$.

Vyššie uvedená binárna reprezentácia má jednu podstatnú nevýhodu. Dvojica binárnych reťazcov, ktoré sú odlišné vo všetkých polohách bitových premenných môže odpovedať dvom susedným celým číslam (pozri napr. tab. 2.1, kde komplementárne binárne reťazce $\alpha_1=(011)$ a $\alpha_2=(100)$ sú interpretované ako celé čísla $\text{int}(\alpha_1)=4$ resp. $\text{int}(\alpha_2)=5$). Táto nevýhoda štandardného binárneho kódu je odstránená použitím tzv. Grayovho kódu [9]. Jeho základná myšlienka spočíva v tom, že kóduje binárne čísla tak, že dve susedné celé čísla sú binárne reprezentované reťazcami, ktoré sú rôzne len v jednej polohe binárneho reťazca (pozri tab. 2.1, kde napr. binárne reťazce $\tilde{\alpha}_1=(010)$ a $\tilde{\alpha}_2=(110)$, ktoré sa líšia len v

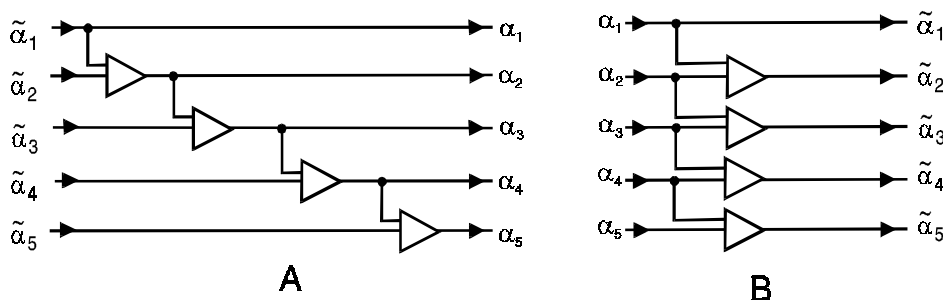
prvej polohe reťazca, odpovedajú susedným celým číslam $\text{int}(\tilde{\alpha}_1)=4$ resp. $\text{int}(\tilde{\alpha}_2)=5$. Konštrukcia racionálneho čísla z intervalu $[a,b]$ (ktoré, ako už bolo povedané vyššie, aproximuje reálne číslo $a \leq x \leq b$) sa zakladá na tom, že v Grayovom kóde binárneho čísla najprv pretransformujeme do štandardného kódu a až potom sa použije vzťah (2.9)

$$x \approx \text{real}(\tilde{\alpha}) = a + \frac{b-a}{2^k-1} \text{int}(\tilde{\alpha}) \quad (2.11)$$

kde $\tilde{\alpha} = (\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_k)$ je binárny reťazec zostrojený pomocou Grayovho kódu a $\text{int}(\tilde{\alpha})$ je odpovedajúce celé číslo priradené reťazcu $\tilde{\alpha}$

$$\text{int}(\tilde{\alpha}) = \text{int}(\alpha) \quad (2.12)$$

kde jednotlivé komponenty reťazca $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$ sú určené pomocou zložiek reťazca $\tilde{\alpha} = (\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_k)$ (pozri obr. 2.6)



Obrázok 2.6. Transformácie Grayovho kódu na štandardný kód (A, pozri (2.13)) resp. štandardného kódu na Grayov kód (B, pozri (2.14)) použitím algebraickej operácie "xor" reprezentovanej trojuholníkmi.

$$\begin{aligned} \alpha_1 &= \tilde{\alpha}_1 \\ \alpha_2 &= \tilde{\alpha}_1 \oplus \tilde{\alpha}_2 = \alpha_1 \oplus \tilde{\alpha}_2 \\ \alpha_3 &= \tilde{\alpha}_1 \oplus \tilde{\alpha}_2 \oplus \tilde{\alpha}_3 = \alpha_2 \oplus \tilde{\alpha}_3 \\ &\dots\dots\dots \\ \alpha_k &= \tilde{\alpha}_1 \oplus \tilde{\alpha}_2 \oplus \dots \oplus \tilde{\alpha}_k = \alpha_{k-1} \oplus \tilde{\alpha}_k \end{aligned} \quad (2.13)$$

kde binárna operácia " \oplus " je definovaná ako "xor" súčet binárnych čísel, $0 \oplus 0=0$, $0 \oplus 1=1$, $1 \oplus 0=1$ a $1 \oplus 1=0$. Tieto vzťahy (2.13) môžu byť jednoducho verifikované na príkladoch z tabuľky 2.1. Posledný stĺpec tejto tabuľky (Grayov kód) je použitím formúl (2.13) pretransformovaný na druhý stĺpec (štandardný kód). Inverzná transformácia vzhľadom k (2.13) (t.j. konštrukcia Grayovho kódu zo štandardného kódu) sa jednoducho určí ako riešenia rovníc (2.14)

$$\begin{aligned} \tilde{\alpha}_1 &= \alpha_1 \\ \tilde{\alpha}_2 &= \alpha_1 \oplus \alpha_2 \\ \tilde{\alpha}_3 &= \alpha_2 \oplus \alpha_3 \\ &\dots\dots\dots \\ \tilde{\alpha}_k &= \alpha_{k-1} \oplus \alpha_k \end{aligned} \quad (2.14)$$

Príklad 2.4. Napíšte program, ktorý transformuje binárne vektory dĺžky k na reálne čísla z intervalu $[-10,10]$, pričom sa využíva tak štandardné, ako aj Grayovo kódovanie. Súčasťou programu nech je aj inverzná transformácia vypočítaného reálneho čísla na binárny tvar. Tento zostrojený binárny vektor musí byť totožný s binárnym vektorom, ktorý vstupoval do programu. Zreprodukuje tabuľku 2.1 a zostrojte pomocou programu jej rozšírenú verziu pre $k=4$.

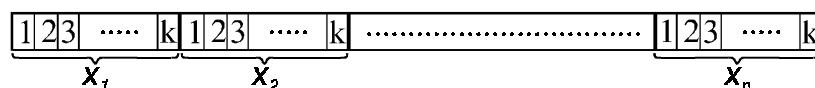
Príklad 2.5. Aká musí byť dĺžka binárnej reprezentácie premennej funkcie $f(x)$ z príkladu 2.1, ak chceme získať výsledky s presnosťou 10^{-2} ? Je výsledok kompatibilný so získanými hodnotami konštant δ z príkladu 2.1?

2.3 Transformácia spojitého optimalizačného problému na binárny optimalizačný problém

Našou hlavnou úlohou je riešiť spojité optimalizačný problém (2.2) pre globálne minimum funkcie $f(x)$ na oblasti D . Pre ďalšie úvahy o použití evolučných algoritmov k riešeniu tohto problému bude užitočné pretransformovať tento spojité optimalizačný problém (2.2) na binárny optimalizačný problém (2.5). Predpokladajme, že každá z n premenných vektora $x \in D$ je vyjadrená v binárnej reprezentácii bitovým vektorom dĺžky k . To znamená, že vektor $x \in D$ je v binárnej reprezentácii vyjadrený bitovým vektorom dĺžky kn , pozri obr. 2.7. Nech funkcia f vyhovuje druhej podmienke z odseku 2.1 pre danú kladnú konštantu δ . Budeme predpokladať, že dĺžka binárnej reprezentácie k je zvolená tak, že platí

$$\delta \gg \frac{(b_i - a_i)}{(2^k - 1)} \quad (\text{pre } i = 1, 2, \dots, n) \quad (2.15)$$

Táto podmienka požaduje, aby minimálna vzdialenosť medzi dvoma minimami funkcie $f(x)$ na oblasti D bola o mnoho väčšia ako "presnosť" binárnej reprezentácie pre každú premennú vektora $x \in D$.



Obrázok 2.7. Schematické znázornenie vektora $x \in D$ (obsahujúceho n komponent) a jeho binárnej reprezentácie pomocou bitových vektorov dĺžky k . Výsledný bitový vektor má dĺžku kn .

Prechod od binárneho vektora $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_{kn}) \in \{0,1\}^{kn}$ k spojitému vektoru $x = (x_1, x_2, \dots, x_n) \in D$ môže byť formálne chápaný ako transformácia

$$\Gamma : \{0,1\}^{kn} \rightarrow D \quad (2.16a)$$

$$x = \Gamma(\alpha) \quad (2.16b)$$

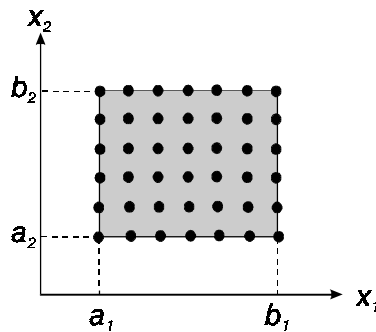
ktorá zobrazuje množinu binárnych vektorov dĺžky kn na body - n -tice reálnych čísel z kocky D . Ináč povedané, konečná množina (2^{kn}) binárnych vektorov dĺžky kn je reprezentovaná pomocou zobrazenia Γ bodmi, ktoré môžu byť na oblasti D

usporiadané do ortogonálnej mriežky, pozri obr. 2.8. Minimalizačný problém (2.2) pri použití binárnej reprezentácie n -rozmerných vektorov \mathbf{x} sa teda realizuje na konečnej množine diskretných bodov. Označme $\tilde{\alpha}_{opt}$ binárny vektor dĺžky kn , ktorý bol získaný riešením optimalizačného problému (2.5), pričom funkcia f tohto problému je totožná s funkciou f v optimalizačnom probléme (2.2)

$$\tilde{\alpha}_{opt} = \arg \min_{\alpha \in \{0,1\}^{kn}} f(\Gamma(\alpha)) \quad (2.17a)$$

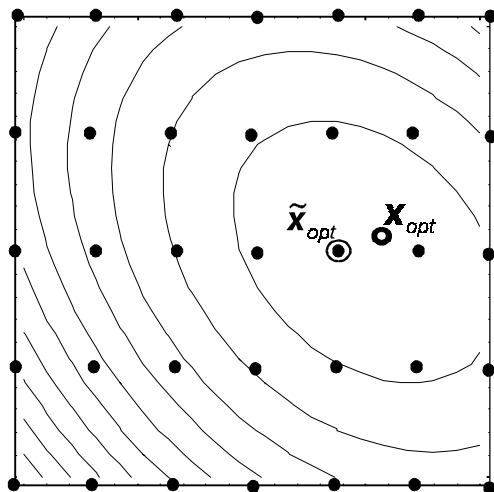
Vektor reálnych premenných priradený tomuto binárnemu vektoru je $\tilde{\mathbf{x}}_{opt} = \Gamma(\tilde{\alpha}_{opt})$. Budeme predpokladať, že presnosť binárnej reprezentácie (t.j. počet bitov k rezervovaných pre každú reálnu premennú) je taká, že vektor $\tilde{\mathbf{x}}_{opt}$ je blízky presnému riešeniu \mathbf{x}_{opt} spojitého optimalizačného problému (2.2) funkcie f nad oblasťou D , pozri obr. 2.8.

$$\mathbf{x}_{opt} \approx \tilde{\mathbf{x}}_{opt} = \Gamma(\tilde{\alpha}_{opt}) \quad (2.17b)$$



Obrázok 2.8. Oblasť D je aproximovaná ortogonálnou mriežkou bodov, ktoré majú binárnu reprezentáciu vyjadrenú pomocou kn -rozmerných vektorov.

Presnosť riešenia optimalizačného problému (2.2) pri prechode zo spojitej reprezentácie k binárnej reprezentácii závisí na konštante k , ktorá určuje dĺžku binárnych vektorov reprezentujúcich jednotlivé reálne premenné. Ak funkcia f obsahuje málo miním, ktoré sú dostatočne navzájom izolované (konštanta δ je veľká, pozri obr. 2.2) a "široké", konštanta k nemusí byť veľká. Avšak, ak funkcia f obsahuje množstvo miním, ktoré ležia blízko seba (konštanta δ musí byť malá), potom konštanta k musí byť pomerne veľká. Ortogonálna mriežka bodov nad oblasťou D , ktoré sú generované zvolenou binárnou reprezentáciou reálnych premenných, musí byť dostatočne jemná, aby sa postihli a odlišili blízko seba ležiace minimá funkcie f .



Obrázok 2.9. Kontúrový graf funkcie f nad oblasťou D , ktorá je v binárnej reprezentácii aproximovaná ortogonálnou mriežkou bodov. Presné minimum funkcie f nad oblasťou D je označené \mathbf{x}_{opt} , minimum funkcie f nad bodmi ortogonálnej mriežky je označené $\tilde{\mathbf{x}}_{opt}$. Vychádzajúc z podmienky, že parametre binárnej reprezentácie boli zvolené tak, aby vhodne approximovali oblasť D (dĺžka k binárnych vektorov je dostatočná), potom vektor $\tilde{\mathbf{x}}_{opt}$ dobre aproximuje presné riešenie \mathbf{x}_{opt} .

Literatúra

- [1] L. Lukšan: Metody s proměnnou metrikou. Academia, Praha 1990.
- [2] M. Mañas: Optimalizační metody. SNTL, Praha 1979.
- [3] A. Brunovská: Malá optimalizácia. Metódy, programy, príklady. Alfa, Bratislava 1990.
- [4] T. Bäck: Evolutionary Algorithms in Theory and Practice. Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press, New York 1996.
- [5] D.B. Fogel: Evolutionary Computation. Toward a new Philosophy of Machine Intelligence. IEEE Press, New York 1995.
- [6] B. Char, K.O. Geddes, G.H. Gonnet, B.L. Leong, M.B. Monagan, S. Watt: Maple V. Springer Verlag, Berlin 1992.7
- [7] S. Wolfram: Mathematica. A System for Doing Mathematics by Computers. Addison Wesley, Reading, MA 1993.
- [8] L. Kučera: Kombinatorické algoritmy. SNTL, Praha 1983.
- [9] R. Hamming: Coding and Information Theory. Prentic-Hall Inc. Englewood Cliffs, NJ 1980.