Predmet: **Evolučné algoritmy**

Vypracoval: **Andrej Probst**, MFF UK, Informatika, Umelá inteligencia

Téma 4.
Genetický algoritmus pre funkciu

$$F(x) = 0{,}993851231 + e^{-0{,}001 * x * x} sin(10x)cos(8x)$$

Pre x z intervalu [-10,10]. Premenná x je binárne reprezentovaná tak v štandardnom kódovaní, ako aj v Grayovom kódovaní. Pokúste sa nájsť minimálnu veľkosť populácie a také hodnoty pravdepodobností P_{cross} a P_{mut}, aby ste skoro so 100% istotou dostali globálne minimum s predpísaným minimálnym počtom krokov.

Riešenie:

Teória optimalizačných problémov

Pri formulovaní teórie som vychádzal z prednášok z predmetu Evolučné Algoritmy, ktoré sa dajú nájsť na internetovskej stránke: ftp://math.chtf.stuba.sk/pub/vlado/evol_alg_mff.
Nech funkcia

$$F : D \rightarrow R$$

$$D = \prod_{i=1}^{n} [a_i, b_i]$$

zobrazuje n-rozmernú kocku D (karteziánsky produkt uzavretých intervalov [ai,bi]) na reálne číslo $y \in R$. Úlohou je nájsť globálne minimum funckie F. Na to, aby sme to dokázali, potrebujeme túto funkciu ohraničiť nasledovnými podmienkami:

1. Existuje taký algoritmus, ktorý funkciu f "vypočíta dostatočne rýchlo" s požadovanou presnosťou pre každé $x \in D$ (hovoríme, že funkcia f je dobre vypočítateľná).
2. Pre každú dvojicu lokálnych miním $x_1, x_2 \in D$ vzdialenosť $| x_1 - x_2 |$ je väčšia ako dané kladné číslo $\delta > 0$, $| x_1 - x_2 | > \delta$. 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 (1)

$$x_{opt} = \arg\min_{x \in D} F(x)$$

Nájdenie globálneho minima použitím klasických optimalizačných metód (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 pri riešení problému často používajú tzv. evolučné optimalizačné algoritmy, ktoré poskytujú riešenia blízke globálnemu, alebo s ním totožné.
My budeme pracovať s binárnymi funkciami. Potom bude funkcia F vyzerať nasledovne:

$$F : \{0,1\}^n \rightarrow R$$

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

$$y = F(\alpha)$$

Kardinalita množiny binárnych vektorov dĺžky k je určená vzťahom

$$|\{0,1\}^k| = 2^k$$

To znamená, že "dimenzia" priestoru všetkých možných binárnych vektorov dĺžky k rastie exponenciálne s dĺžkou k. Analóg optimalizačného problému (1) pre binárne vektory má tvar (2)

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

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

$$t_{CPU} \propto 2^k$$

Binárna reprezentácia reálnej premennej
Nech binárny vektor α dĺžky k

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

je interpretovaný ako celé číslo

$$int(\alpha) = \sum_{i=1}^{k} \alpha_i 2^{k-i} = \alpha_1 2^{k-1} + \alpha_2 2^{k-2} \ldots + \alpha_{k-1} 2 + \alpha_k$$

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

$$x \approx real(\alpha) = a + (b-a) / (2^k-1)\ int(\alpha)$$

V našom prípade prevodová funkcia z binárneho čísla α na reálne číslo x vyzerá nasledovne (3):

$$\mathbf{a = -10,\ b = 10}$$
$$\mathbf{x = -10 + 20/(2^{velkost} - 1)int(\alpha)}$$

Táto konštrukcia racionálneho čísla $a \le real(\alpha) \le b$ z binárneho reťazca a dĺžky k je formálne interpretovaná ako "transformácia" binárnej reprezentácie na "reálnu" reprezentáciu, kde zostrojené racionálne číslo real(a) 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. Príklad na binárneho čísla na reálne číslo:

Tab.1:

No.	α	Int(α)	Real(α)	α (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 (1) má tvar

$$int\,(\alpha) = \lceil (x-a)*(2^k-1)/(b-a) \rceil$$

kde symbol $\lceil x \rceil$ je celá časť reálneho čísla x (napr. $\lceil 1.1 \rceil=1$ α $\lceil 1.9 \rceil=1$). V dôsledku toho, že $\alpha \le x \le \beta$, zlomok (x - a) / (b - a) leží v uzavretom intervale [0,1], ak int(α)=0, potom x = a , ak int(α)=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. 1, kde komplementárne binárne reťazce a_1 =(011) a a_2=(100) sú interpretované ako celé čísla int(a_1)=4 resp. int(a_2)=5). Táto nevýhoda štandardného binárneho kódu je odstránená použitím tzv. Grayovho kódu. 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. 1, kde napr. binárne reťazce α'_1 =(010) a α'_2 =(110), ktoré sa líšia len v prvej polohe reťazca, odpovedajú susedným celým číslam int(a'_1)=4 resp. int(a'_2)=5. Konštrukcia racionálneho čísla z intervalu [a,b] (ktoré, ako už bolo povedané vyššie, aproximuje reálne číslo $a \le x \le b$) sa zakladá na tom, že v Grayovom kóde binárneho čísla najprv pretransformujeme do štandardného kódu a až potom sa použije vzťah

$$x \approx real(\alpha') = a + (b-a) / (2^k-1)\ int(\alpha')$$

kde $\alpha' = (\alpha'_1\ \alpha'_2\ \ldots, \alpha'_\kappa)$ je binárny reťazec zostrojený pomocou Grayovho kódu a int(α') je odpovedajúce celé číslo.

Prevod z Grayovho kódu do štandardného kódu

Populáciu v evolučnom algoritme reprezentujeme binárnym kódom. V mojej úlohe to bude štandardným aj Grayovým binárnym kódom. Potrebovať budem reviesť Grayov kód na štandardný binárny kód. Zoberme si binárne číslo v Grayovom kóde. Označme jeho bity postupne $b_1, b_2, \ldots, b_k$. Chceme previesť toto číslo do štandardného kódu $a_1, a_2, \ldots a_k$. Prevedieme ho pomocou nasledovných priradení:

$$a_1 := b_1$$
$$a_I := a_{i-1}\ XOR\ b_i \qquad pre\ i = 2,3, \ldots k$$

Transformácia spojitého optimalizačného problému na binárny optimalizačný problém

Našou hlavnou úlohou je riešiť spojitý optimalizačný problém pre globálne minimum funkcie F (x) na oblasti D. Pre ďalšie úvahy o použití evolučných algoritmov k riešeniu tohto problému je treba pretransformovať tento spojitý optimalizačný problém na binárny optimalizačný problém. 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. Nech funkcia F vyhovuje druhej podmienke pre danú kladnú konštantu δ. Budeme predpokladať, že dĺžka binárnej reprezentácie k je zvolená tak, že platí

$$\delta >> (b_i - a_i)/(2^k - 1) \qquad pre\ i = 1,2,\ldots,n$$

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 $x \in D$.

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

$$\Gamma : \{ 0,1\}^{kn} \to D$$
$$\boldsymbol{x} = \Gamma(\alpha)$$

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. Minimalizačný problém (1) pri použití binárnej reprezentácie n -rozmerných vektorov x sa teda realizuje na konečnej množine diskrétnych bodov. Označme a'_{opt} binárny vektor dĺžky kn , ktorý bol získaný riešením optimalizačného problému (2), pričom funkcia F tohto problému je totožná s funkciou F v optimalizačnom probléme (1)

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

Vektor reálnych premenných priradený tomuto binárnemu vektoru je $\mathbf{x}'_{opt} = \Gamma(\mathbf{a}'_{opt})$. Budeme predpokladať, že presnosť binárnej reprezentácie (t.j. počet bitov k rezervovaných pre každú reálnu premennú) je taká, že vektor $\mathbf{x}'_{opt}$ je blízky presnému riešeniu $\mathbf{x}_{opt}$ spojitého optimalizačného problému (1) funkcie F nad oblasťou D

$$\boldsymbol{x}_{opt} \approx \boldsymbol{x}'_{opt} = \Gamma(\boldsymbol{a}'_{opt})$$

Presnosť riešenia optimalizačného problému (1) 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) 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 byt pomerne veľká. Ortogonálna mriežka bodov nad oblasťou D , ktoré sú generované zvolenou binárnou reprezentáciou reálnych premenných, musí byť dostatočne jemná, aby sa postihli a odlíšili blízko seba ležiace minimá funkcie F .

Genetické programovanie pre hľadanie globálneho minima funkcie F

Nech P je množina (multimnožina, t.j. P môže obsahovať α_i, α_j také, že $\alpha_i = \alpha_j$ a zároveň $i \neq j$) - populácia riešení tvaru

$$P = \{ \alpha_1, \alpha_2, ..., \alpha_{pocet} \} \subseteq \{0,1\}^{velkost}$$

Kde **pocet** značí veľkosť populácie P a **velkost** značí kardinalitu binárneho čísla, ktoré reprezentuje jedinca buď v štandardnom binárnom kóde alebo v Grayovom kóde. Každý element z populácie P je ohodnotený funkčnou hodnotou $F(\alpha)$ (fitness jedinca α). V našom prípade je fitness nasledovná:

$$\underline{\mathbf{F(x) = 0{,}993851231 + e^{-0{,}001 * x * x} sin(10x)cos(8x)}}$$

Funckiu F aplikujeme na prevedené binárne číslo α zo vzťahu (3). Fitness v mojom prípade je inek ponímaná ako je bežne zaužívané. Čím menšie F(x), tým je x lepšie. Z populácie P vyberieme podmnožninu - podpopuláciu rodičov $Q_1 \subseteq P$ a mutantov $Q_2 \subseteq P$, ktorých kardinalita $|Q_1 + Q_2|$ je menšia alebo nanajvýš rovná kardinalite pôvodnej populácie P , $|Q_1 + Q_2| \leq |P|$. Riešenia z podpopulácie Q_1 sú transformované mutačným operátorom O_{krizuj} na podpopuláciu potomkov a riešenia z Q_2 operátorom Q_{mutuj}:

$$Q_1' = \{ \alpha' = O_{krizuj}(\alpha) \mid a \in Q \}$$
$$Q_2' = \{ \alpha' = O_{mutuj}(\alpha) \mid a \in Q \}$$

pričom kardinality Q_1 a Q_1' sú rovnaké, $|Q_1|=|Q_1'|$ to isté platí aj pre Q_2. Potomkovia z podpopulácie Q_i' sú ohodnotené funkčnými hodnotami F (α).

```
procedure Evolutionary_Programmning(output: αopt);
begin
      P:=náhodne vybratá populácia;
      stop_criterion:=false;
      while not stop_criterion do
      begin
```

```
            Q1:=kvázináhodne vybratá podpopulácia P;
            Q1':=Okrizuj(Q1);
            Q2:=náhodne vybratá podpopulácia (P \ Q1);
            Q2':=Omutuj(Q2);
            P:=(P \ Q2)∪Q2';
            S:=kvázináhodne vybratá podpopulácia (P \ Q1);
            P:=(P \ S)∪Q1';
            if convergence criteria are fulfilled then
                  stop_criterion:=true;
      end;
      α opt = arg min F(α)
                  α∈ P
end;
```

V procedúre sa tri krát používa náhodný výber. Napriek tomu je každý spomenutý náhodný výber iný. Pri výbere jedincov do podpopulácie Q_1 majú väčšiu šancu prežiť jedince α, ktoré majú vysoké F (α). Veľkosť populácie Q_1 je Round (| P | * P_{cross}). P_{cross} vyjadruje, relatívne množstvo jedincov, ktorí sa bude krížiť. Do podpopulácie Q_2 sa vyberie náhodne Round (| P | * P_{mut}) jedincov, ktorý zmutujú. Znova P_{mut} vyjadruje relatívny počet jedincov na mutáciu. Do podpopulácie S sa vyberú jedinci tak, že najväčšiu pravdepodobnosť výberu majú jedinci α s malou F (α). Musí pri tom platiť: | Q_1 | = | S |.

Kríženie

Každý jedinec α je reprezentovaný binárne:

$$\alpha_\iota \subseteq \{0,1\}^{velkost}$$
$$\alpha_i = b_{velkost}\, b_{velkost-1} \dots b_2\, b_1 ,\ kde\ b_i \in \{0,1\}$$

Kríženie prebehne nasledovne. Zvolí sa náhodné číslo od 1 po **velkost** a na tomto mieste sa chromozómy jedincov prekrížia týmto spôsobom.
Pred krížením:

$$\alpha_i = b_{velkost}\, b_{velkost-1} \dots b_2\, b_1 ,\ kde\ b_i \in \{0,1\}$$
$$\alpha_j = c_{velkost}\, c_{velkost-1} \dots c_2\, c_1 ,\ kde\ c_i \in \{0,1\}$$

Nech k je index kríženia potom kríženie bude vyzerať nasledovne:

$$\alpha_i = b_{velkost}\, b_{velkost-1} \dots b_k\, c_{k-1} \dots c_2\, c_1 ,\ kde\ b_i,\ c_i \in \{0,1\}$$
$$\alpha_j = c_{velkost}\, c_{velkost-1} \dots c_k\, b_{k-1} \dots b_2\, b_1 ,\ kde\ b_i,\ c_i \in \{0,1\}$$

Mutácia

Mutácia prebieha tak, že sa postupuje po bitoch jedinca a s vopred danou pravdepodobnosťou sa bit zmení na opačnú hodnotu.

Kvázináhodný výber s uprednostnením jedincov s vyššou resp. nižšou fitness.

Náhodný výber som vytvoril nasledovným spôsobom. Zosumoval som vhodne upravené funčkné hodnoty F (α) všetkých jedincov populácie. Potom som nechal vygenerovať náhodné číslo z intervalu (0, Suma). Vybral sa toho jedinca, na ktorého ukázalo vygenerované náhodné číslo. Tento výber kvázináhodného výberu vychádza z princípu rulety:

Podmienka ukončenia genetického algoritmu

Algoritmus môžme ukončiť viacerými spôsobmi.

1. Môžme ohraničiť počet generácií zhora nejakou konštantou.
2. Stop_criterion bude „príbuznosť„ jedincov. Vo svojem programe som „príbuznosť„ vyjadril ako rozdiel medzi priemerom fitnesov všetkých jedincov a fitnesom najsilnejšieho jedinca.

Program

Program je pripojený k tejto case-study. Naprogramovaný je v Pascale. Po spustení programu užívateľ zadá konštanty, ktorými sa dá vplývať na priebeh výpočtu. Medzi voliteľné konštanty patrí: veľkosť populácie, veľkosť jedinca reprezentovaného binárnym kódom, pravdepodobnosť mutácie a kríženia a nakoniec výber medzi štandardným a Grayovým kódovaním. Program potom nájde globálne minimum spôsobom takým, že prehľadá celý priestor a nájde jedinca (jeho chromozóm) s najlepším fitnesom. Následne na to sa začnú generovať jedntolivé generácie až kým nebude splnená podmienka Stop_criterion. V tomto programe nastane vtedy, ak užívateľ stlačí ľubovoľnú klávesu. Po celý čas priebehu výpočtu, program zobrazuje poradie práve spracovávanej generácie, jej najlepšieho jedinca spolu s fitness funkciuo a priemer fitnesov jedincov v populácii. Užívateľ môže takto priamo sledovať vývoj populácie.

Výsledky

Presnosť výsledkov podľa kardinality binárneho čísla reprezentujúceho jedinca z populácie P

K	Real (α)	F (Real (α))
5	5.4838709677	2.6526754387E-01
6	-4.7619047620E-01	5.6697941918E-03
7	-3.9370078740E-01	2.8082589435E-01
8	-8.2352941176E-01	1.1451031830E-01
9	-8.0234833660E-01	2.3497448497E-02
10	-7.9178885631E-01	3.4253203166E-03
11	-7.8651685393E-01	1.2020833128E-04
12	-7.8388278389E-01	1.6425077320E-04
13	-7.8500793554E-01	7.0671276262E-06
14	-7.8557040835E-01	5.8547111621E-06
15	-7.8524124881E-01	3.0480532587E-07
16	-7.8538185701E-01	5.1471488405E-07
17	-7.8529957046E-01	7.1759131970E-10

V tabuľke je dobre vidno ako presnosť narastá s podrobnejšou reprezentácie chromozómu jedinca. Dá sa tušiť, že globálnym minimum zadanej funkcie je hodnota nula. Rozhodol som sa reprezentovať jedincov binárnym kódom s kardinalitou 10, nakoľko vyššia kardinalita mi veľa na presnosti nepridá. Výber kardinality binárneho čísla závisí na používateľovi, aká presnosť mu vyhovuje.

Tabuľka 2: pre k=10, kódovanie je štandardné, veľkosť populácie 50, P_{cross}=0.2, P_{mut}=0.12.

Poradie generácie	Chromozóm jedinca α	Real (α)	F (Real (α))
5	0111010111	-7.9178885631E-01	3.4253203166E-03
10	0111010111	-7.9178885631E-01	3.4253203166E-03
15	0111010110	-8.1133919844E-01	5.4254236609E-02
20	0111010110	-8.1133919844E-01	5.4254236609E-02
25	0111010111	-7.9178885631E-01	3.4253203166E-03
30	0111010111	-7.9178885631E-01	3.4253203166E-03
35	0111010111	-7.9178885631E-01	3.4253203166E-03
40	0111010111	-7.9178885631E-01	3.4253203166E-03
45	0111010111	-7.9178885631E-01	3.4253203166E-03
50	0111010111	-7.9178885631E-01	3.4253203166E-03
...	...	...	...
100	0100010111	-5.1808406647E-01	3.4253203166E-03

Tabuľka 3: pre k=10, kódovanie je štandardné, veľkosť populácie 100, P_{cross}=0.2, P_{mut}=0.12.

Poradie generácie	Chromozóm jedinca α	Real (α)	F (Real (α))
5	0111010100	-8.5043988269E-01	3.08321676180E-01
10	0100111001	-3.8807429130E+00	2.7611618375E-01
15	0111011001	-7.5268817204E-01	2.7611618375E-01
20	0111011000	-7.7223851417E-01	1.38479049583E-02
25	0111010111	-7.9178885631E-01	3.4253203166E-03
30	0111010111	-7.9178885631E-01	3.4253203166E-03
35	0111010111	-7.9178885631E-01	3.4253203166E-03
40	0111011000	-7.7223851417E-01	1.38479049583E-02
45	0100010111	-5.1808406647E-01	3.4253203166E-03
50	0100010111	-5.1808406647E-01	3.4253203166E-03
...	...	...	...
100	0100010111	-5.1808406647E-01	3.4253203166E-03

Tabuľka 4: pre k=10, kódovanie je štandardné, veľkosť populácie 150, P_{cross}=0.2, P_{mut}=0.12.

Poradie generácie	Chromozóm jedinca α	Real (α)	F (Real (α))
5	0111010111	-7.9178885631E-01	3.4253203166E-03
10	1001111000	2.3558162268E+00	4.7849077780E-02
15	0111011000	-7.7223851417E-01	1.3847950710E-02
20	0111010110	-8.1133919844E-01	5.4254236609E-02
25	0111010110	-8.1133919844E-01	5.4254236609E-02
30	0111011000	-7.7223851417E-01	1.3847950710E-02
35	0111010111	-7.9178885631E-01	3.4253203166E-03
40	0111010111	-7.9178885631E-01	1.38479049583E-02
45	0111010111	-7.9178885631E-01	3.4253203166E-03
50	0111010111	-7.9178885631E-01	3.4253203166E-03
...	...	...	...
100	0111010111	-7.9178885631E-01	3.4253203166E-03

Tabuľka 5: pre k=10, kódovanie je Grayovo, veľkosť populácie 50, P_{cross}=0.2, P_{mut}=0.12.

Poradie generácie	Chromozóm jedinca α	Real (α)	F (Real (α))	Priemer
5	0100111101	-8.1133919844E-01	5.4254236609E-02	9.0119940900E-01
10	0100111101	-8.1133919844E-01	5.4254236609E-02	7.1375115771E-01
15	0100011111	-4.2033235582E-01	1.4340476803E-01	7.0370581935E-01
20	0100100101	-1.1241446725E+00	1.2394154204E-01	5.5158079424E-01
25	0100100101	-1.1241446725E+00	1.2394154204E-01	6.0462897766E-01
30	0100100101	-1.1241446725E+00	1.2394154204E-01	4.9204184739E-01
35	0100011101	-4.3988269794E-01	1.1128660521E-01	3.3531162485E-01
40	0100111101	-8.1133919844E-01	5.4254236609E-02	4.2261993245E-01
45	0100111101	-8.1133919844E-01	5.4254236609E-02	4.4949575495E-01
50	0100111100	-7.9178885631E-01	3.4253203166E-03	4.7993586954E-01
...	...	...	...	...
100	0100111100	-7.9178885631E-01	3.4253203166E-03	3.3399340548E-01

Tabuľka 6: pre k=10, kódovanie je Grayovo, veľkosť populácie 100, P_{cross}=0.2, P_{mut}=0.12.

Poradie generácie	Chromozóm jedinca α	Real (α)	F (Real (α))	Priemer
5	0100110100	-7.7223851417E-01	1.3847950710E-02	8.7244641990E-01
10	0100011101	-4.3988269794E-01	1.1128660521E-01	8.6872749590E-01
15	1101001100	2.3362658847E+00	7.7480878502E-02	7.5178225137E-01
20	0100011101	-4.3988269794E-01	1.1128660521E-01	7.8315873224E-01
25	0100110101	-7.5268817204E-01	8.4279049583E-02	7.3176211020E-01
30	0100111101	-8.1133919844E-01	5.4254236609E-02	6.4802230291E-01
35	0100111101	-8.1133919844E-01	5.4254236609E-02	6.0090048181E-01
40	0100111100	-7.9178885631E-01	3.4253203166E-03	5.6560550313E-01
45	0100111101	-8.1133919844E-01	5.4254236609E-02	6.0529449390E-01
50	0100111100	-7.9178885631E-01	3.4253203166E-03	5.9360907022E-01
55	0100110100	-7.7223851417E-01	1.3847950710E-02	5.7389181754E-01
60	0100111100	-7.9178885631E-01	3.4253203166E-03	5.9786967339E-01
65	0100110100	-7.7223851417E-01	1.3847950710E-02	5.8329069309E-01
70	0100111100	-7.9178885631E-01	3.4253203166E-03	5.6881784142E-01
75	0100110100	-7.7223851417E-01	1.3847950710E-02	5.8828002603E-01
80	0100110100	-7.7223851417E-01	1.3847950710E-02	5.0034970616E-01
85	0100111100	-7.9178885631E-01	3.4253203166E-03	4.5594462843E-01
90	0100111100	-7.9178885631E-01	3.4253203166E-03	5.0196300232E-01
95	0100111100	-7.9178885631E-01	3.4253203166E-03	6.4071592240E-01
100	0100111100	-7.9178885631E-01	3.4253203166E-03	6.5390919393E-01

Tabuľka 7: pre k=10, kódovanie je Grayovo, veľkosť populácie 150, P_{cross}=0.2, P_{mut}=0.12.

Poradie generácie	Chromozóm jedinca α	Real (α)	F (Real (α))	Priemer
5	0100110100	-7.7223851417E-01	1.3847950710E-02	9.1786790734E-01
10	0100110100	-7.7223851417E-01	1.3847950710E-02	8.2768525696E-01
15	0100110100	-7.7223851417E-01	1.3847950710E-02	7.1020509864E-01
20	0100110100	-7.7223851417E-01	1.3847950710E-02	7.5872459736E-01
25	0100110100	-7.7223851417E-01	1.3847950710E-02	6.5479004685E-01
30	0100111100	-7.9178885631E-01	3.4253203166E-03	6.0345658938E-01
35	0100111100	-7.9178885631E-01	3.4253203166E-03	5.4023422817E-01
40	0100111100	-7.9178885631E-01	3.4253203166E-03	5.6634625532E-01
...	...	...	...	...
100	0100111100	-7.9178885631E-01	3.4253203166E-03	4.4225066568E-01

Tabuľka 8: Veľkosť populácie je 100. Kardinalita je rovná 10. Pcross=0.20, Pmut=0.15, použilo sa štandardné kódovanie.

Poradie generácie	Chromozóm jedinca α	Real (α)	F (Real (α))	Priemer
5	0111010111	-7.9178885631E-01	3.4253203166E-03	8.4739879151E-01
10	1001111000	2.3558162268E+00	4.7849077780E-02	7.5956544920E-01
15	1001111001	2.3753665689E+00	7.6921465769E-02	7.7806403100E-01
20	1001111001	2.3753665689E+00	7.6921465769E-02	8.4884526057E-01
25	0111000111	-1.1045943304E+00	1.7294512517E-01	8.4385537073E-01
30	0111101000	-4.5943304007E-01	1.4080256357E-01	8.3611588022E-01
35	0111000101	-1.1436950147E+00	1.3502139145E-01	8.2264494858E-01

40	0111011000	-7.7223851417E-01	1.3847950710E-02	7.1656204622E-01
45	0111011000	-7.7223851417E-01	1.3847950710E-02	7.1069796494E-01
50	0111010111	-7.9178885631E-01	3.4253203166E-03	8.0223729196E-01
55	0111010111	-7.9178885631E-01	3.4253203166E-03	7.3380128917E-01
60	0111010111	-7.9178885631E-01	3.4253203166E-03	6.4552466921E-01
65	0111010111	-7.9178885631E-01	3.4253203166E-03	7.4960136943E-01
70	0111000110	-1.1241446725E+00	1.2394154204E-01	7.8617500012E-01
75	1001111000	2.3558162268E+00	4.7849077780E-02	7.5667537239E-01
80	0111010110	-8.1133919844E-01	5.4254236609E-02	7.6179526775E-01
85	0111010110	-8.1133919844E-01	5.4254236609E-02	7.6563557158E-01
90	1001110111	2.3362658847E+00	7.7480878502E-02	5.3038944200E-01
95	0111010111	-7.9178885631E-01	3.4253203166E-03	5.6203867373E-01
100	0111010111	-7.9178885631E-01	3.4253203166E-03	5.3722258747E-01

Tabuľka 9: Veľkosť populácie je 100. Kardinalita je rovná 10. Pcross=0.30, Pmut=0.10, použilo sa štandardné kódovanie.

Poradie generácie	Chromozóm jedinca α	Real (α)	F (Real (α))	Priemer
5	0111101010	-4.2033235582E-01	1.4340476803E-01	7.3530523524E-01
10	0111101001	-4.3988269794E-01	1.1128660521E-01	7.2538755814E-01
15	1001110111	2.3362658847E+00	7.7480878502E-02	6.2329147952E-01
20	1001110111	2.3362658847E+00	7.7480878502E-02	6.4606258436E-01
25	1001110111	2.3362658847E+00	7.7480878502E-02	5.1244723950E-01
30	0111010111	-7.9178885631E-01	3.4253203166E-03	6.0685024948E-01
35	0111010111	-7.9178885631E-01	3.4253203166E-03	5.5795047884E-01
40	0111011000	-7.7223851417E-01	1.3847950710E-02	4.7989332702E-01
45	0111010111	-7.9178885631E-01	3.4253203166E-03	4.0431900988E-01
50	0111010111	-7.9178885631E-01	3.4253203166E-03	3.6751642547E-01
55	0111010111	-7.9178885631E-01	3.4253203166E-03	4.4706672560E-01
60	0111010111	-7.9178885631E-01	3.4253203166E-03	3.9914158039E-01
65	0111010111	-7.9178885631E-01	3.4253203166E-03	3.6539696867E-01
70	0111010111	-7.9178885631E-01	3.4253203166E-03	3.5816068935E-01
75	0111010111	-7.9178885631E-01	3.4253203166E-03	2.7598702867E-01
80	0111010111	-7.9178885631E-01	3.4253203166E-03	2.9473119642E-01
85	0111010111	-7.9178885631E-01	3.4253203166E-03	9.4509836879E-02
90	0111010111	-7.9178885631E-01	3.4253203166E-03	1.1962996424E-01
95	0111010111	-7.9178885631E-01	3.4253203166E-03	8.7215824216E-02
100	0111010111	-7.9178885631E-01	3.4253203166E-03	3.1648112253E-02

Čo vyplýva z tabuliek:

Parametre Pcross a Pmut treba voliť opatrne. Nesmú byť blízko pri sebe, pretože vtedy začína prevažovať náhoda pri zmene a výbere jedincov do ďalšej generácie. Preto je lepšie voliť Pcross väčšie ako Pmut. Mne sa ukázalo po viacerých simuláciách, že vhodné je nastaviť paramatre nasledovne Pcross = 0.30 a Pmut = 0.10. Algoritmus pracuje lepšie, čím je väčšia populácia. Vtedy existuje väčšia pravdepodobnosť, že sa v populácii vyskytnú jedince s dobrou fitness. Ďalší faktor vplývajúci na presnosť algoritmu je počet generácií. Čím dlhiš necháme algoritmus pracovať, tým má väčšiu šancu dostať sa do globálneho minima. Väčšine pokusov, ktoré som uskutočnil, stačilo ísť po stú generáciu. V stej generácii sa väčšinou našlo globálne minimum a zostalo „kvázi„ stabilné, čo znamená, že občas sa môžu objaviť aj iné hodnoty, ale časom sa populácia vráti k svojmu najlepšiemu jedincovi.

Ak by som mal porovnať štandardné kódovanie a Grayovo kódovanie, tak dôjdem k nasledujúcim záverom. Pri Grayovom kódovaní sa väčšinou výsledky jednotlivých generácií líšia o jeden bit, čo znamená, že tieto jedince sú „blízko pri sebe„. Dobre to dokumentuje tabuľka 6. Nevýhodou však je, ak sa program dostane do nejakého lokálneho maxima, tu sa zdrží dlhú chvíľu, kým sa podarí odtiaľ dostať. Naopak pri štandardnom kódovaní má algoritmus schopnosť viac „skákať„ a tak sa výsledky jednotlivých generácií niekedy až dramaticky líšia.
Odporúčané nastavenie:

Použiť Grayovo kódovanie
Veľkost populácie: 150
Kardinalita binárneho čísla: 12
Pcross: 0.30
Pmut: 0.10
Stop_criterion: 200. generácia
Použitie elitizmu zaručí monotónnosť algoritmu a lepšie výsledky ako algoritmus bez použitia tejto metódy. Elitizmus znamená, že do ďalšej generácie sa určite dostane najlepší jedince z predchádzajúcej generácie.