

Genetický algoritmus

1. Evolúcia v prírode

Biologická evolúcia je progresívna zmena genetického obsahu jedincov populácie v priebehu mnohých generácií. Evolučný proces obsahuje tieto dve zložky:

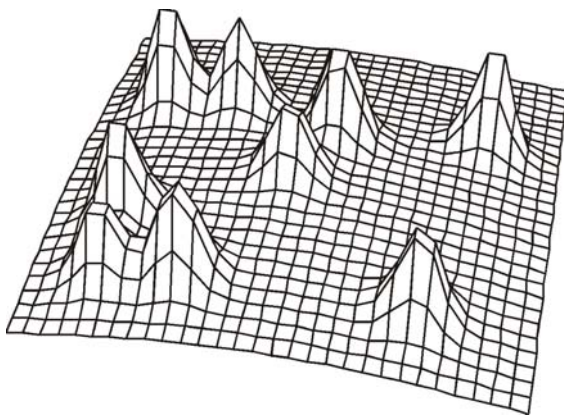
(1) **Prirodzený výber** (Darwin), proces v ktorom "silnejšie" jedinci populácie majú viac potomkov v nasledujúcej generácii, ako "slabší" jedinci.

(2) **Náhodný genetický drift**, náhodne udalosti v živote jedincov populácie. Takéto udalosti sú napr. náhodná mutácia alebo náhodná smrť zjavne "silného" jedinca predtým, ako dostal príležitosť reprodukcie. Stochastické efekty náhodného genetického driftu sú významné hlavne pre malá populácie.

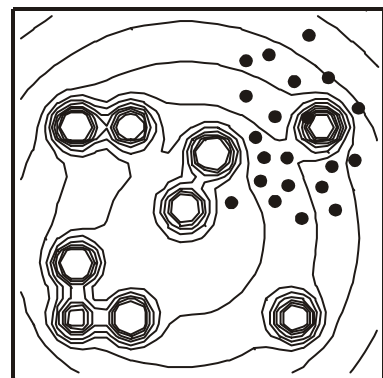
V biológii je "sila" (fitness) definovaná ako relatívna schopnosť prežitia a reprodukovania sa v danom **prostredí** a v danej **populácii**. "Sila" môže byť chápaná ako atribút genotypu.

Povrch "sily" (A) - vhodný prístup ako vizualizovať akty selekcie v populácii, ktorá je v evolúcii. Populácia je reprezentovaná ako "oblak" bodov (B) na povrchu "sily"

Potomkovia rodičov s väčšou silou budú viac početnejšie a "silnejšie" ako potomkovia menej "silnejších" rodičov.



A



B

2. Abstrakcia prirodzeho výberu

(1) Jedinci populácie sú reprezentované **chromozómami** - lineárne reťazce symbolov

$$P = \{x_1, x_2, \dots, x_p\}$$

$$x \in \{\alpha, \beta, \dots\}^n$$

(2) Každý chromozóm populácie je ohodnotený pozitívnym reálnym číslom, ktoré sa nazýva "**sila**" (fitness)

$$f: P \rightarrow R_+$$

(3) **Sexuálna reprodukcia** (proces množenia). Dva kvázináhodne vybraní jedinci z populácie (v závislosti od ich "sily", "silnejšie" jedince sú vybrané s väčšou pravdepodobnosťou) produkujú potomkov (deti). Hlavné časti reprodukcie sú:

- (1) **selekcia** rodičov,
- (2) **kríženie a mutácia**,
- (3) **návrat** potomkov do populácie.

$$(x_1^{new}, x_2^{new}) = O_{repro}(x_1^{old}, x_2^{old})$$

Pascalovský pseudokód prirodzeného výberu

```
P:=randomly generated population  
of chromosomes;  
t:=0;  
while t<tmax do  
begin Q:=∅;  
  while |Q|<|P| do  
    begin select quasirandomly two  
      parents x1,x2∈P;  
      (x1',x2'):=Orepro(x1,x2);  
      Q:=Q∪{x1',x2'};  
    end;  
    P:=Q;  
end;
```

Pascalovský pseudokód obecnějšího přístupu k prirodzenému výberu (Evolučné algoritmy)

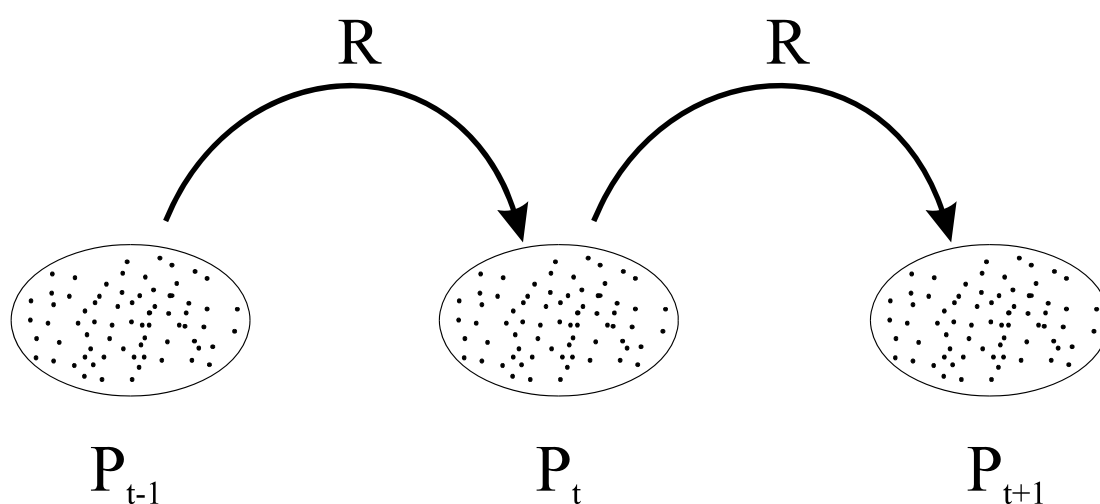
Funkcia (zobrazenie) R priradí **stochasticky** (vzhľadom

```
P0:=randomly generated population of  
solutions (chromosomes);  
t:=0;  
while t<tmax do  
begin all solutions of Pt are  
evaluated by fitness;  
t:=t+1;  
Pt:=R(Pt-1);  
end;
```

k "sile") aktuálnej populácii P_{t-1} nasledujúcu populáciu P_t .
Efektívnosť evolučných algoritmov je určená hlavne
realizáciou funkcie R .

Evolúcia v prírode

Rekurentná konštrukcia populácie z prechádzajúcej populácie



Genetický algoritmus (GA)

Objavený **John H. Holland** (1975)

Literatúra:

1. J.H. Holland: *Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. Michigan University Press, Ann Arbor 1975.
2. D.E. Goldberg: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Pub. Co., New York 1989.
3. L. Davis (Ed.): *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York 1991.
4. V. Kvasnička, J. Pospíchal, J. Tiňo: *Evolučné algoritmy*. STU, Bratislava, 2001.

Základné princípy GA

1. Populácia je zložená z daného počtu chromozómov - binárnych vektorov

$$P = \{\alpha_1, \alpha_2, \dots, \alpha_p\} \subseteq \{0,1\}^n$$

2. Nad množinou $\{0,1\}^n$ (táto množina obsahuje všetky binárne vektory dĺžky n) je definovaná funkcia F

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

ktorá priradí každému binárnemu vektoru $\alpha \in \{0,1\}^n$ reálne číslo $y \in R$.

3. Hľadáme maximum (minimum) funkcie F nad množinou $\{0,1\}^n$

$$\alpha_{opt} = \arg \max_{\alpha \in \{0,1\}^n} F(\alpha)$$

Vo všeobecnosti, jedná sa o NP-problém ($t \sim 2^n$)

4. Každý chromozóm - bitový vektor populácie P je ohodnotený "**silou**" (kladné reálne číslo)

$$f: P \rightarrow R_+$$

Toto zobrazenie vyhovuje podmienke monotónnosti

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

Renormalizovaná "sila" je určená vzťahom

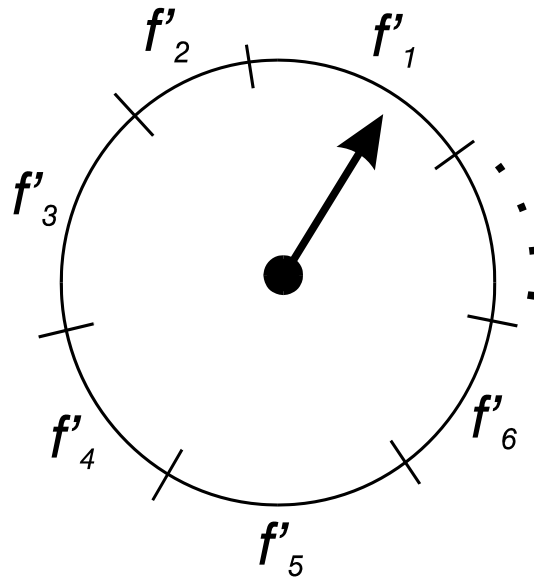
$$f'(\alpha) = \frac{f(\alpha)}{\sum_{\beta \in P} f(\beta)}$$

kde

$$\forall \alpha \in P: 0 \leq f'(\alpha) \leq 1$$

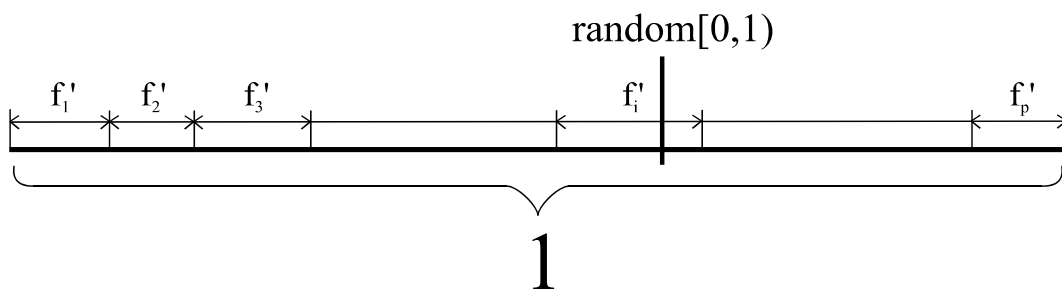
$$\sum_{\alpha \in P} f'(\alpha) = 1$$

5. Kvázináhodný výber "najsilnejších" chromozómov je realizovaný tzv. **ruletou**



i -tý chromozóm je vybraný vtedy ak

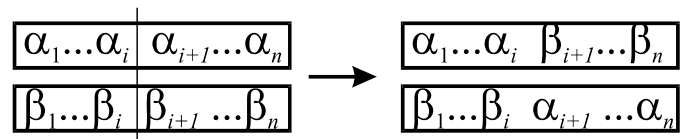
$$\sum_{k=1}^{i-1} f'_k \leq \text{random}[0,1) < \sum_{k=1}^i f'_k$$



6. **Kríženie** - priradí každej páre chro-mozómov iný pár chromozómov

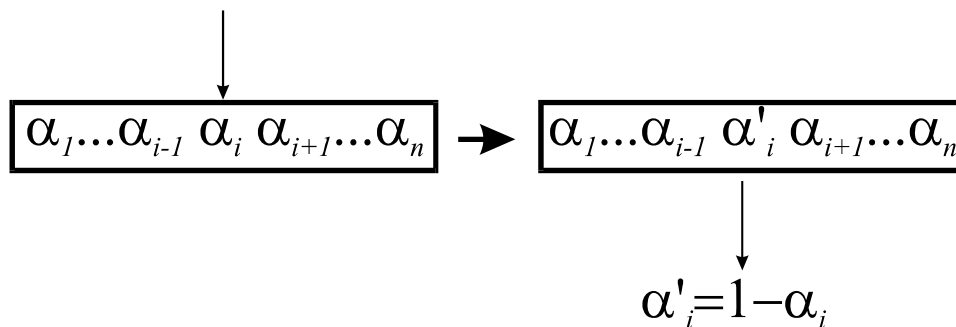
$$(\alpha', \beta') = O_{cross}(\alpha, \beta)$$

1-bodové kríženie



7. **Mutácia**, priradí (stochasticky) každému chromozómu iný chromozóm

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



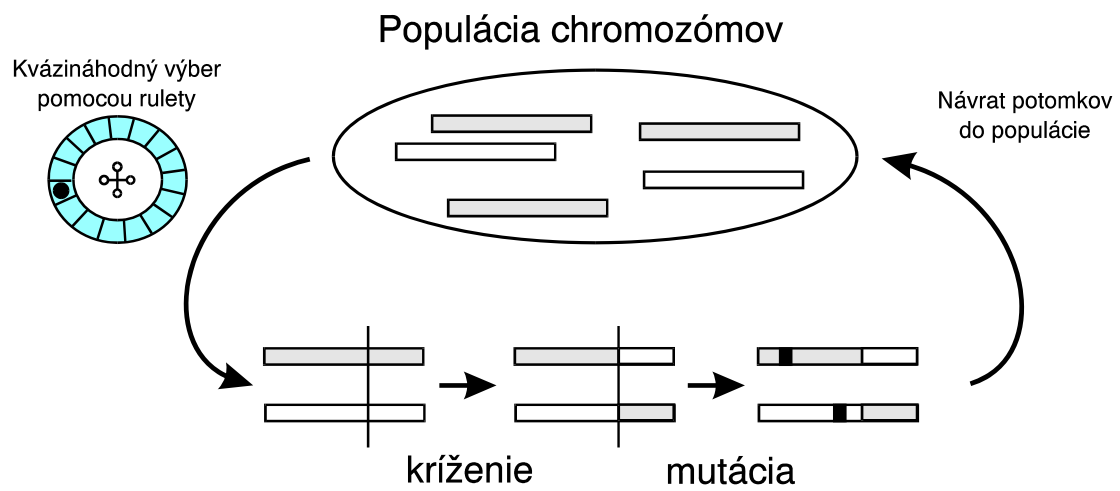
Mutácia jedného bitu chromozómu α je realizovaná s pravdepodobnosťou P_{mut} alebo $0 < P_{mut} \ll 1$

```

for i:=1 to n do
if random[0,1) < Pmut then
   $\alpha_i^{new} := 1 - \alpha_i^{old}$  else
   $\alpha_i^{new} := \alpha_i^{old}$ ;

```

Diagramatické znázornenie GA



8. **Výsledné riešenie** poskytnuté GA je také riešenie, ktoré má najväčšiu "silu"

$$\exists t_0 \forall t \geq t_0 : \tilde{\alpha} = \operatorname{argmax}_{\alpha \in P_t} f(\alpha)$$

Očakáva sa, že platí

$$\alpha_{opt} \approx \tilde{\alpha}$$

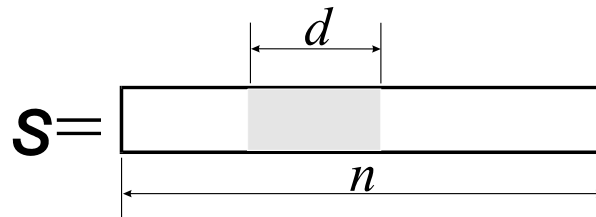
4. Ako pracuje GA?

Schéma - sekvencia bitov v chromozóme, ktorá je dôležitá pre ďalšou evolúciu populácie

$$s = (\#\#\#\#1101\#\#\#\#)$$

symbol # je nahradený buď 0 alebo 1.

Pravdepodobnosť výskytu schémy



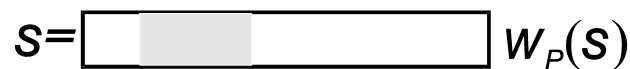
v populácii P je

$$w_P(s) = \frac{|P|}{2^{n-d}}$$

Napr. pre $|P|=10^3$, $n=30$, and $d=4$ dostaneme

$$w_P(s) \approx 6.0 \times 10^{-5}.$$

Majme dve neprekrývajúce sa schémy s and s'

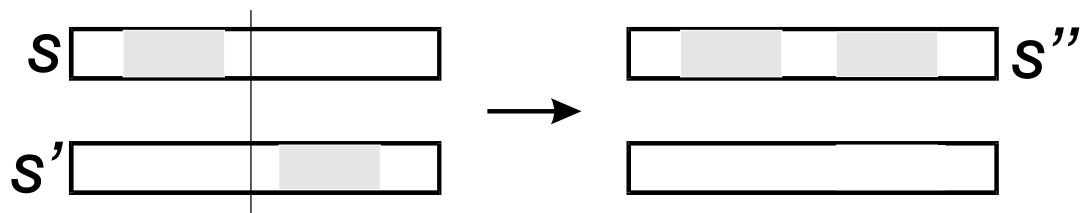


Naším cieľom je mať také schéma s'' , ktoré obsahuje obe schémy s a s'

$$s'' = \text{[shaded segment] [unshaded segment] [shaded segment]} \quad w_P(s'') = w_P(s) \cdot w_P(s')$$

Pravdepodobnosť výskytu tejto schémy je veľa malá. Pravdepodobnosť výskytu sa podstatne

zvýši, ak schéma s'' je vytvorená pomocou kríženia schém s a s'



Záver: Kríženie dramaticky zvyšuje pravdepodobnosť výskytu zložitejších schém, ktoré sú dôležité pre dosiahnutie správneho riešenia.

Kríženie v poèiatoènej etape GA pôsobí ako mutácia, avšak v neskorších štádiách evolúcie, kde určité chromozómy (riešenia) majú väčšiu silu ako ostatné, potom kríženie hrá dôležitú úlohu v "zosilnení" týchto doèasne "silných" riešení.

5. Ilustračný príklad

1. **Binárna reprezentácia** reálnych čísel. Študujeme bitový reťazec dĺžky n

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

Môžeme byť interpretovaný ako binárne číslo, dekadické celé číslo priradené tomuto binárnemu číslo má hodnotu

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

Reálne číslo $x \in [a,b]$ je aproximované číslom

$$\text{real}(\alpha) = a + \frac{b-a}{2^n - 1} \text{int}(\alpha)$$

Príklad. $x \in [0,1]$ a $n=3$

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

2. Optimalizačný problém funkcie s jednou reálnou premenou

$$F : [a, b] \rightarrow R$$

Hľadáme globálne minimum

$$x_{opt} = \arg \max_{x \in [a, b]} F(x)$$

Predpokladajme, že reálna premenná $x \in [a, b]$ je binárne reprezentovaná, potom vyššie uvedený optimalizačný problém je transformovaný na diskretný optimalizačný problém nad množinou binárnych reálnych čísel dĺžky n

$$\alpha_{opt} = \arg \max_{\alpha \in \{0, 1\}^n} F(\text{real}(\alpha))$$

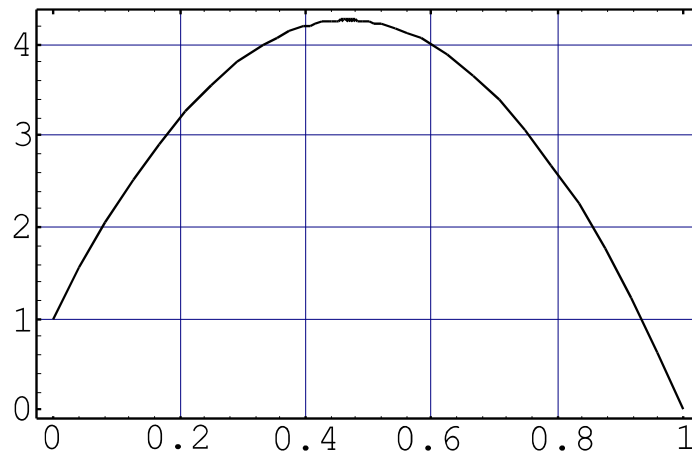
kde

$$x_{opt} \approx \text{real}(\alpha_{opt})$$

3. Študujme kvadratickú funkciu

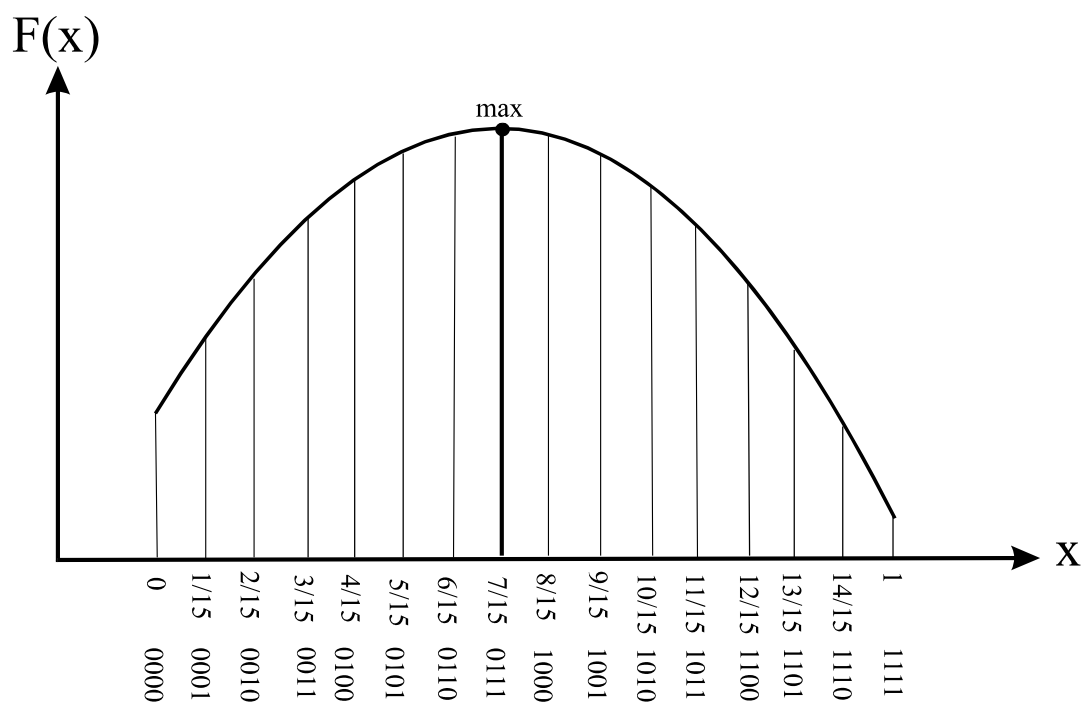
$$F(x) = 14x - 15x^2$$

pre $x \in [0,1]$. Táto funkcia má globálne maximum pre $x_{\text{opt}} = 7/15$.



Reálna premenná x je aproximovaná binárnou reprezentáciou špecifikovanou $n=4$. Optimálne binárne riešenie je $\alpha_{\text{opt}} = (0111)$.

Grafický priebeh funkcie $F(x)$ a binárna reprezentácia jednotlivých bodov z intervalu $[0,1]$



Prvé tri generácia genetického algoritmu

Epocha # 0

0111	(4.2667)
1111	(0.0000)
1101	(1.8667)
0111	(4.2667)
0001	(1.8667)
1100	(2.6000)
0000	(1.0000)
0010	(2.6000)
1101	(1.8667)
1001	(4.0000)

Epocha # 1

1100	(2.6000)
1101	(1.8667)
0001	(1.8677)
1001	(4.0000)
1101	(1.8667)
1001	(4.0000)
0111	(4.2667)
0010	(2.6000)
0111	(4.2667)
0001	(1.8667)

Epocha # 2

0100	(3.6667)
1111	(0.0000)
0001	(1.8677)
0001	(1.8677)
0001	(1.8667)
0111	(4.2667)
0111	(4.2667)
0111	(4.2667)
0111	(4.2667)
0111	(4.2667)

Epocha # 2

0100	(3.6667)
1111	(0.0000)
0001	(1.8677)
0001	(1.8667)
0001	(1.8667)
0111	(4.2667)
0111	(4.2667)
0111	(4.2667)
0111	(4.2667)
0111	(4.2667)

Epocha # 3

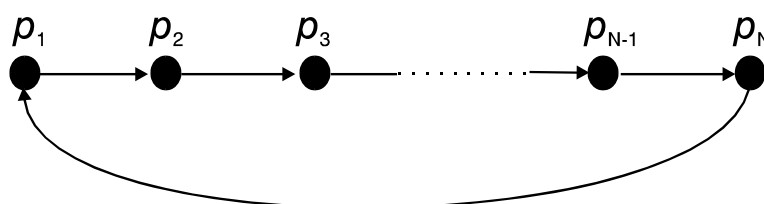
0111	(4.2667)
0111	(4.2667)
0111	(4.2667)
0111	(4.2667)
0111	(4.2667)
1001	(4.0000)
0111	(4.2667)
0111	(4.2667)
0111	(4.2667)
0111	(4.2667)

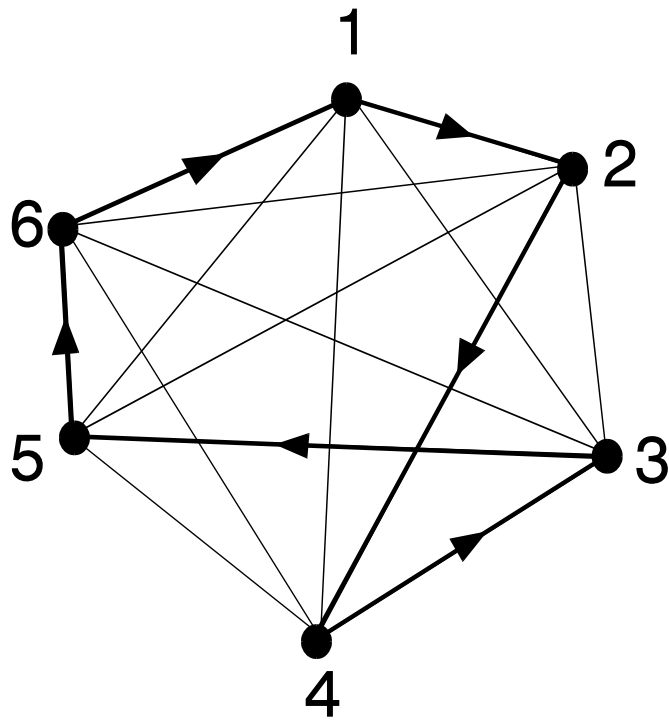
Problém obchodného cestujúceho (TSP)

Grafovo-teoretická formulácia: Nech G je úplný graf obsahujúci N vrcholov (miest) a $N(N-1)/2$ hrán (spojov, ciest). Každá hrana (i,j) , spájajúca vrcholy i a j , je ohodnotená vzdialenosťou $d(i,j)$.

Cycká cesta obchodného cestujúceho, ktorý musí navštíviť všetkých N miest (práve jeden-krát) a vrátiť sa do počiatočného mesta, je určená permutáciou N objektov

$$P = (p_1, p_2, \dots, p_{N-1}, p_N)$$





$$P=(1,2,4,3,5,6)$$

Každá cesta $P=(p_1,p_2,\dots,p_N)$ je ohodnotená vzdialenosťou

$$d(P) = d(p_1, p_N) + \sum_{i=2}^N d(p_{i-1}, p_i)$$

TSP: Hľadá takú cestu P , ktorá poskytuje **minimálnu** vzdialenosť $d_{\min} = d(P_{\min})$

$$P_{\min} = \arg \min_P d(P)$$

Toto je tzv. NP-obtiažny problém, jeho CPU čas rastie rýchlejšie ako polynomiálne pre $N \rightarrow \infty$

$$t_{CPU} \approx N!$$

Použitie presných (enumeratívnych) metód k riešeniu TSP je beznádejne obtiažny numerický problém pre väčšie hodnoty N .

Táto skutočnosť je hlavným dôvodom prečo TSP je riešený GA (alebo iným evolučným algoritmom).

Ako modifikovať GA pre účely TSP?

1. Populácia je tvorená množinou permutácií N objektov.
2. Každý chromozóm P (permutácia) je ohodnotený "silou" $f(P)$

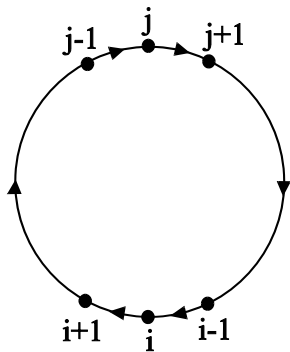
$$d(P_1) \leq d(P_2) \Rightarrow f(P_1) \geq f(P_2)$$

3. Mutácia chromozómov je realizovaná transpozíciou dvoch náhodne vybraných komponent

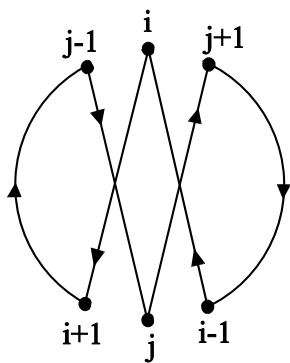
$$P = (p_1, \dots, p_i, \dots, p_j, \dots, p_N)$$

$$P' = (p_1, \dots, p_j, \dots, p_i, \dots, p_N)$$

Diagramatická vizualizácia mutácie

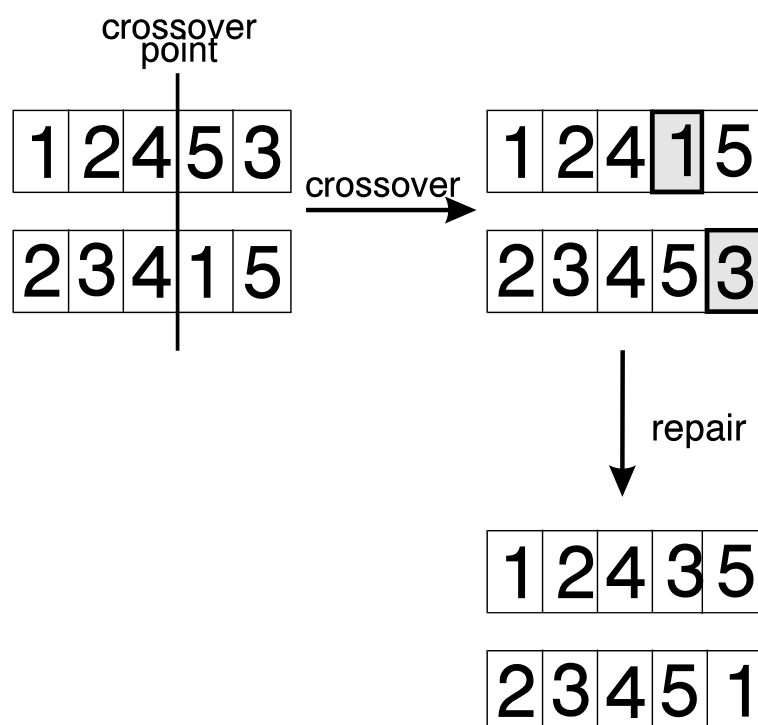


$$P = (p_1 \dots p_{i-1} p_i p_{i+1} \dots p_{j-1} p_j p_{j+1} \dots p_N)$$

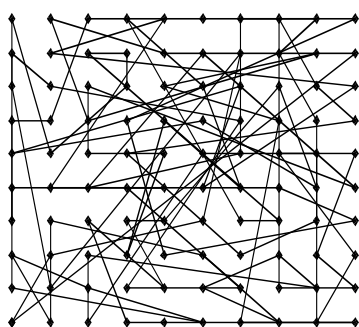


$$P = (p_1 \dots p_{i-1} p_j p_{i+1} \dots p_{j-1} p_i p_{j+1} \dots p_N)$$

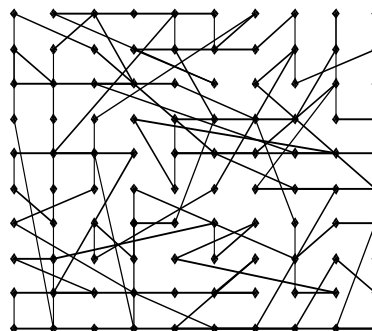
4. Kríženie dvoch chromozómov je realizované výmenou dvoch náhodne vybraných častí chromozómov. Vytvorené chromozómy nemusia byť permutáciami, preto opravný proces je aplikovaný na vytvorené chromozómy.



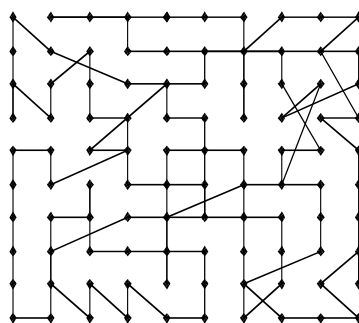
Štvorcová mriežka typu 10×10
(100 vrcholov)
Veľkosť populácie=200 chromozómov



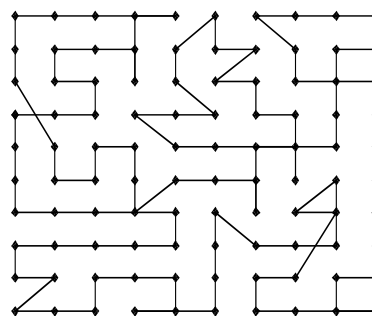
Epoch=0



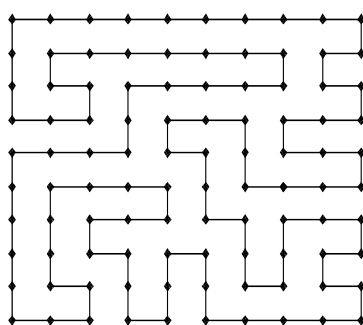
Epoch=100



Epoch=200



Epoch=300



Epoch=1500

Závery

1. GAs je **robustný** a **efektívny** numerický prostriedok pre nájdenie globálneho minima funkcií, ktoré sú vysoko multimodálne.
2. V praktických aplikáciách je doporuèované používať tzv. **hybridný GA**, ktoré kombinujú GA s klasickými optimalizačnými metódami. Obvykle, ich efektívnosť sa dramaticky zvýši. Prinajhoršom, doporučuje sa použiť GA ako **generátor počiatočných riešení** pre klasické optimalizačné metódy.
3. Binárna reprezentácia premenných v GA môže byť zamenená za "reálnu" reprezentáciu, t.j. chromozóm je vektor reálnych premenných.
4. Implementácia GA v C++ alebo Pascale je ve³/₄mi jednoduchá.