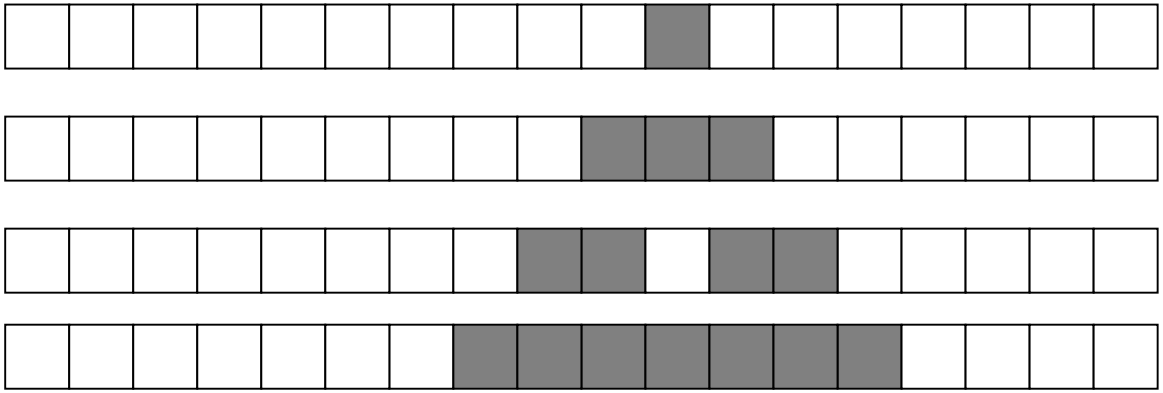
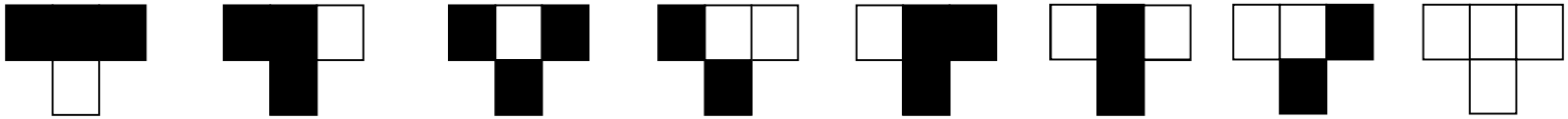
Genetické algoritmy

- Předpověděny J. Hollandem v 60. letech během jeho práce na *celulárních automatech*
- První „velká“ publikace o jejich použití k optimalizaci funkcí je kniha D. Goldberga z 80. let
- 2015: pro heslo „Genetic algorithms“ nalezen přibližný počet výsledků 2 920 000 (pro srovnání „Simulated annealing“1 430 000 výsledků)
- Za *standardní genetický algoritmus* je dnes považována binární verze z té doby

Celulární automata (CA)



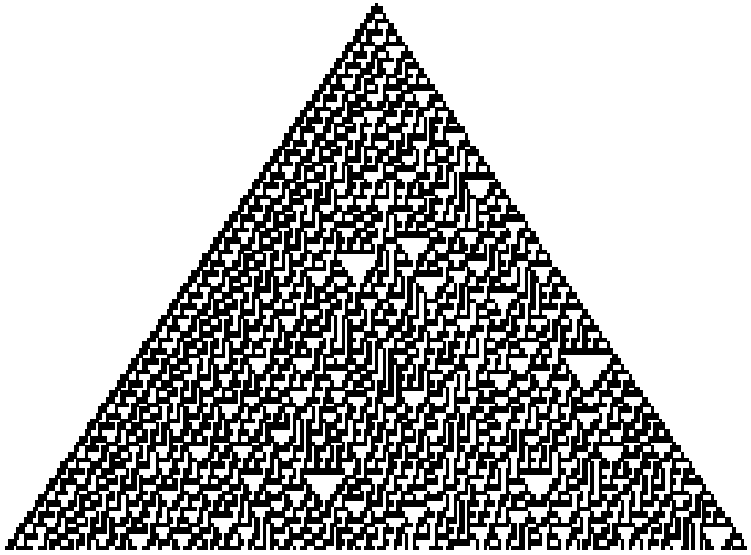


Pravidla:

- 3 Černé = Bílá
- 2 Černé = Černá
- 1 Černá = Černá
- 3 Bílé = Bílá

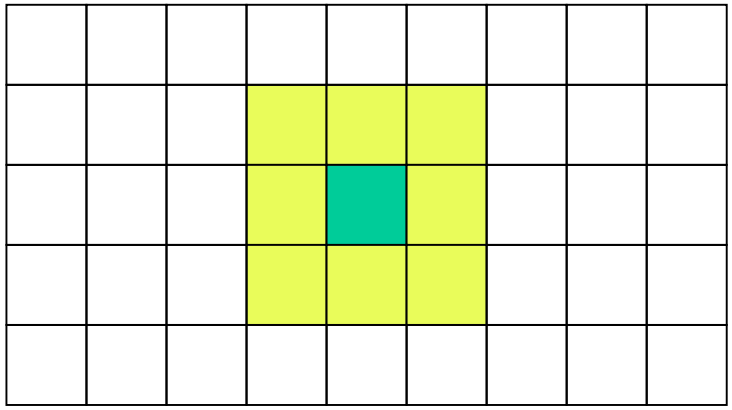
1D Celulární automata

1-dimensionální CA je založen na jedné řadě buněk (cells). Buňky se mění ze stavu do stavu podle daných pravidel.



2D Celulární automata

2-dimensionální CA je založen na konečné (nekonečné) síti buněk, kdy každá je v jednom z konečného počtu stavů. Čas je dán diskrétně a stav buňky v čase t je dán stavem sousedních buněk v čase t-1.

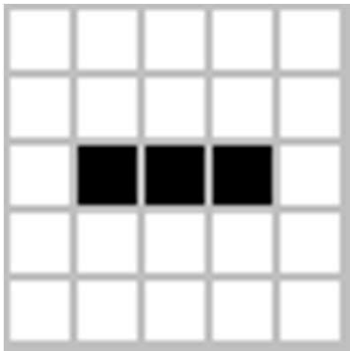


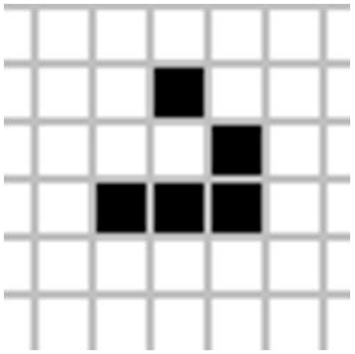
Př.: Conway's Game of Life **(Hra života)**

Dvourozměrná hrací plocha Hry života se znázorňuje jako čtvercová síť. Její políčka se nazývají buňky (proto název celulární neboli buněčný automat). Každá buňka může nabývat dvou stavů, může být živá či mrtvá. Buňka sousedí s dalšími buňkami, se kterými se dotýká hranou (ortogonálně) či vrcholem (diagonálně). Nový stav buňky určuje přechodová funkce :

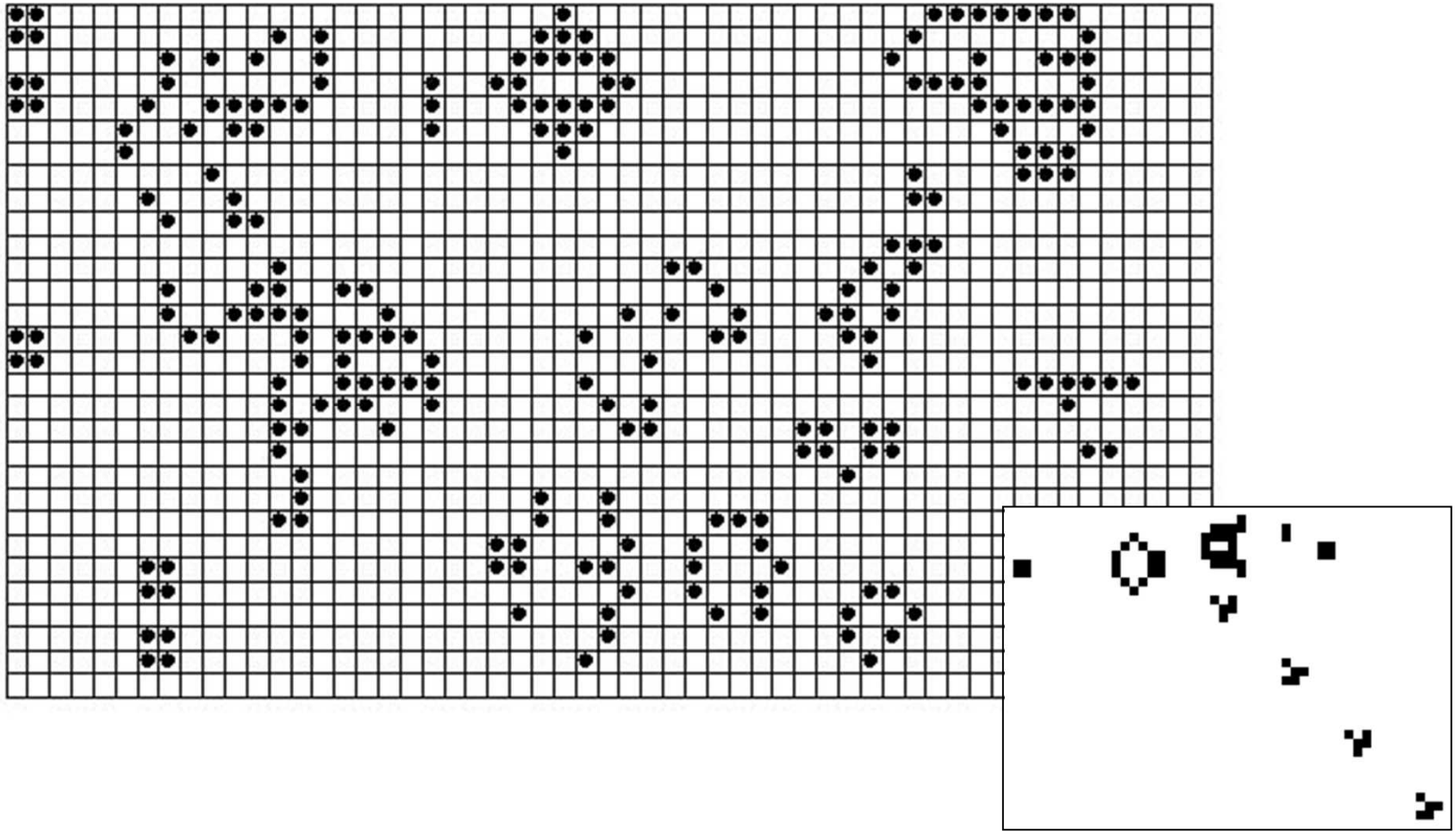
1. Každá živá buňka s méně než dvěma živými sousedy zemře.
2. Každá živá buňka se dvěma nebo třemi živými sousedy zůstává žít.
3. Každá živá buňka s více než třemi živými sousedy zemře.
4. Každá mrtvá buňka s právě třemi živými sousedy oživne.

[Wikimedia]





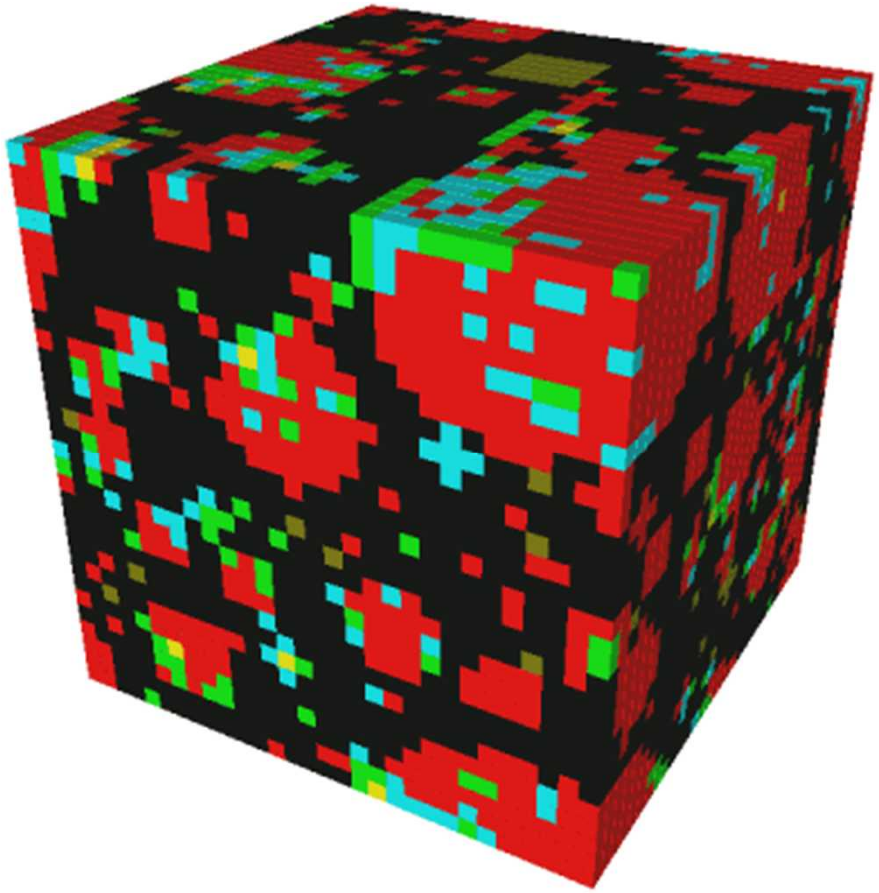
Př.: Conway's Game of Life



Př.: Conway's Game of Life

Reálný příklad:
Modelování vývoje mikrostruktury betonu

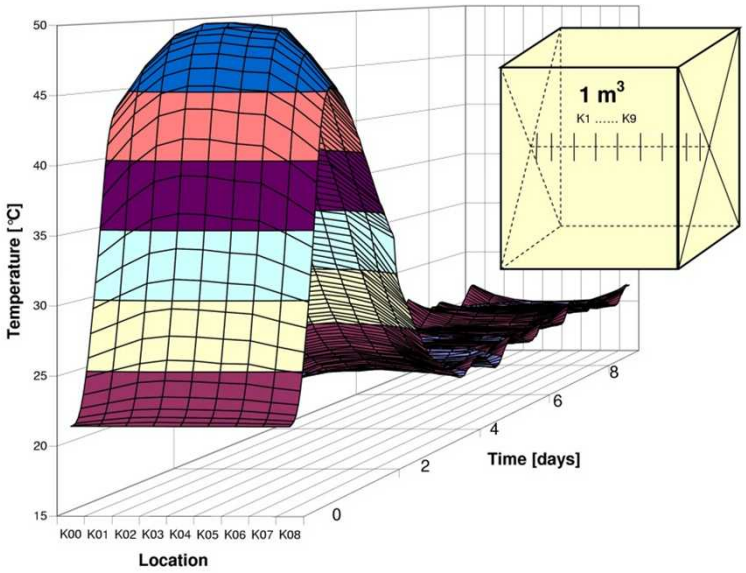
W/c = 0.4, RVE 30 x 30 x 30 μ m

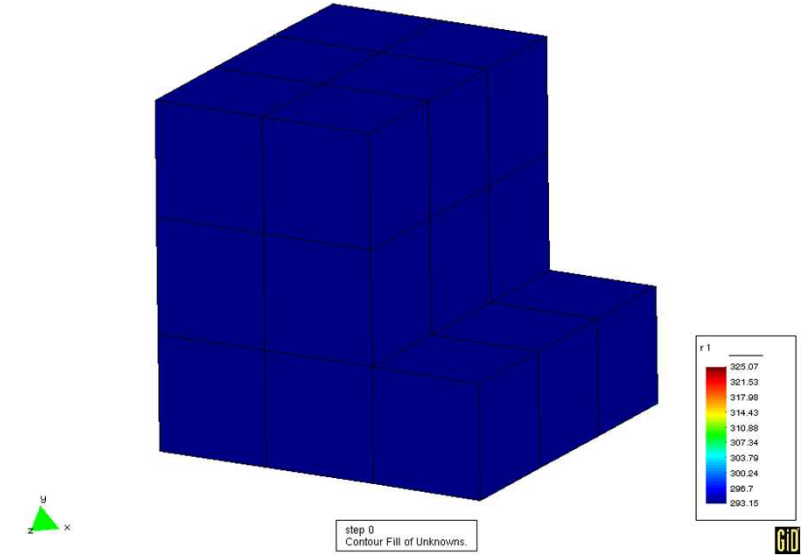


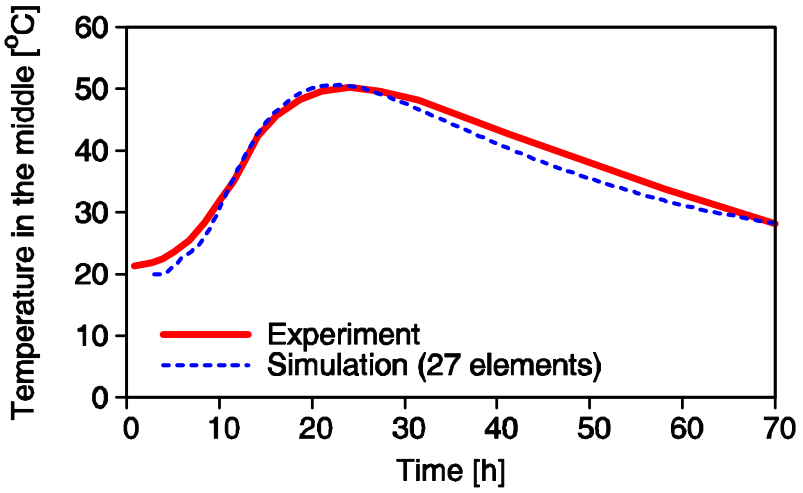
[Vít Šmilauer, Zdeněk Bittnar]

Např.: Vývin hydratačního tepla

- SCC, CEM I 42.5 R, 350 kg / m3

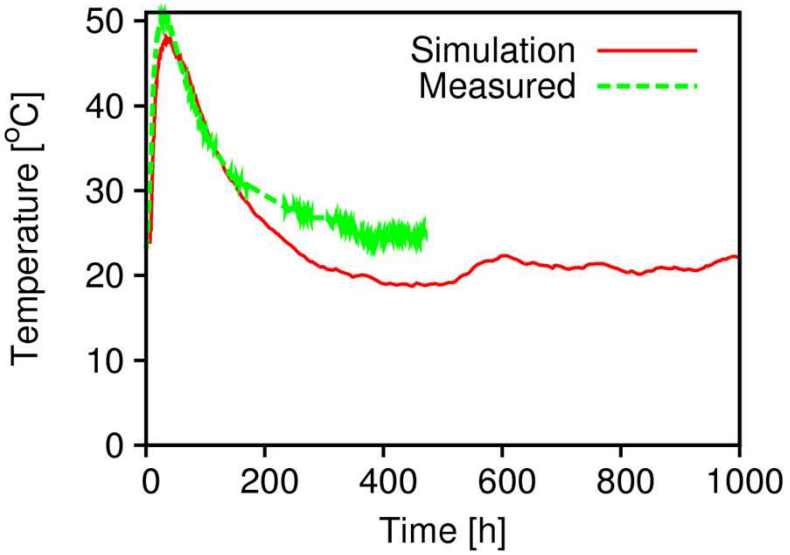




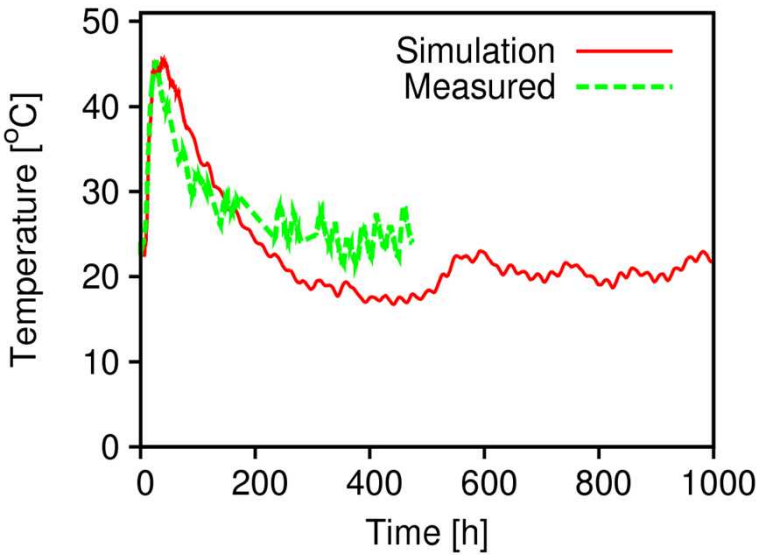


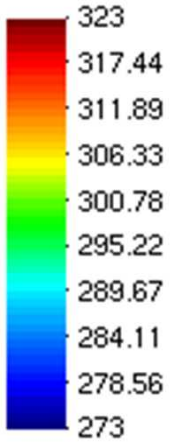
Např.: Vývin hydratačního tepla

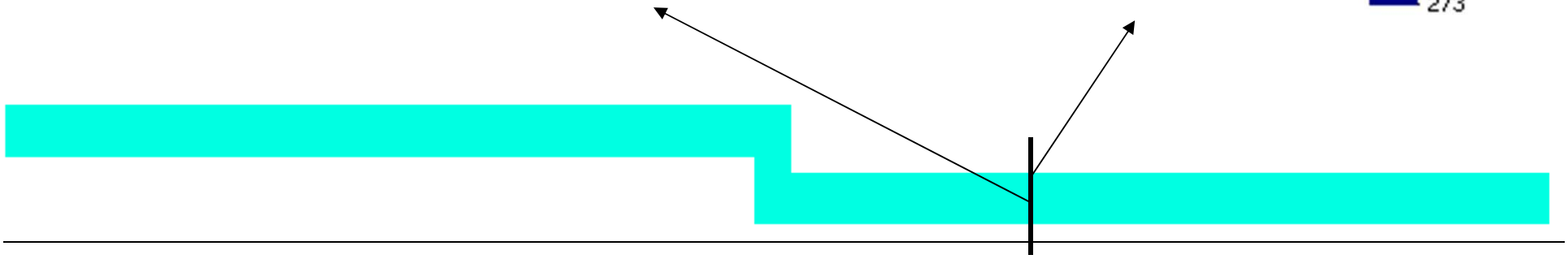
Middle



Top



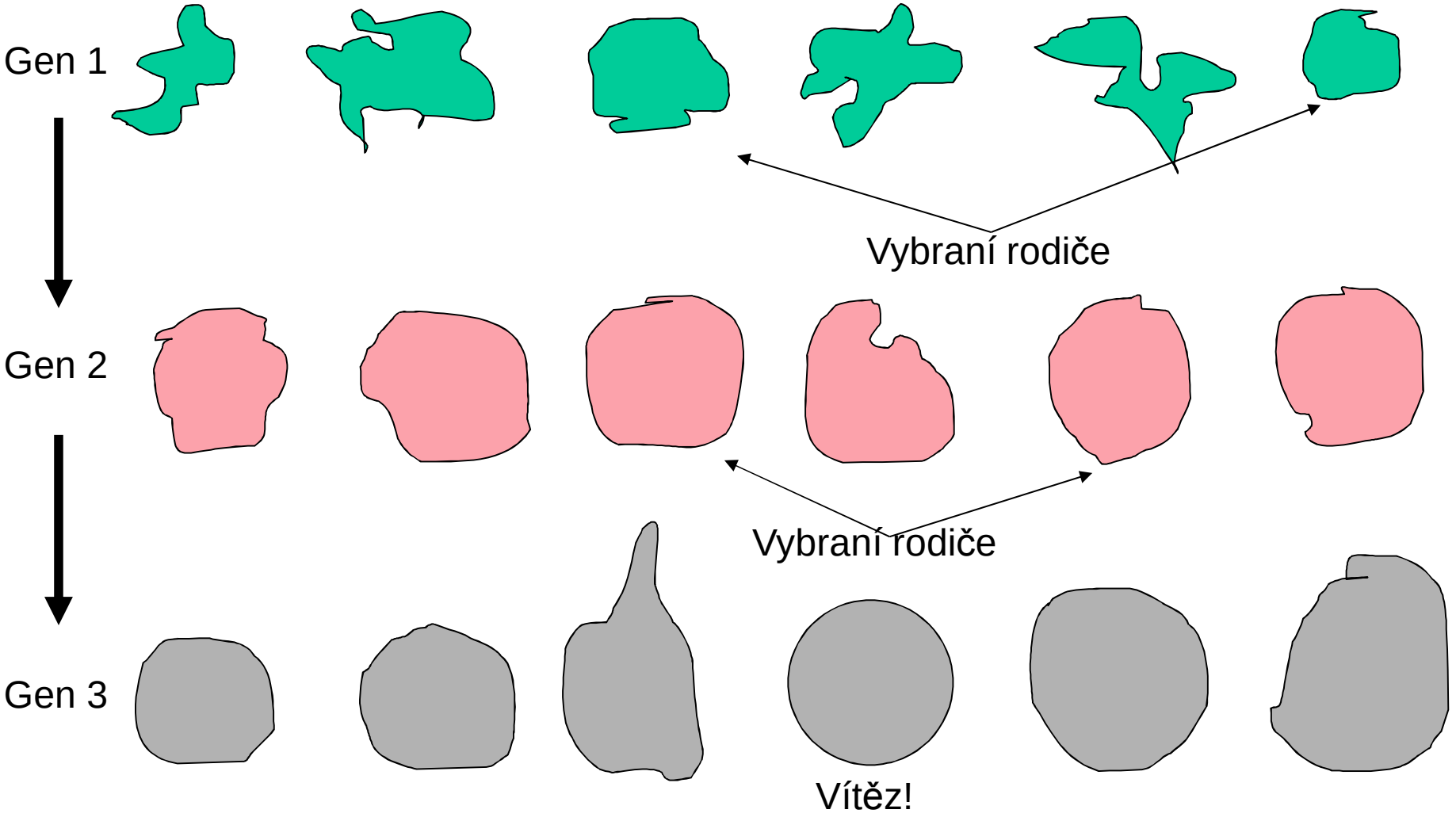




Vlastnosti GA

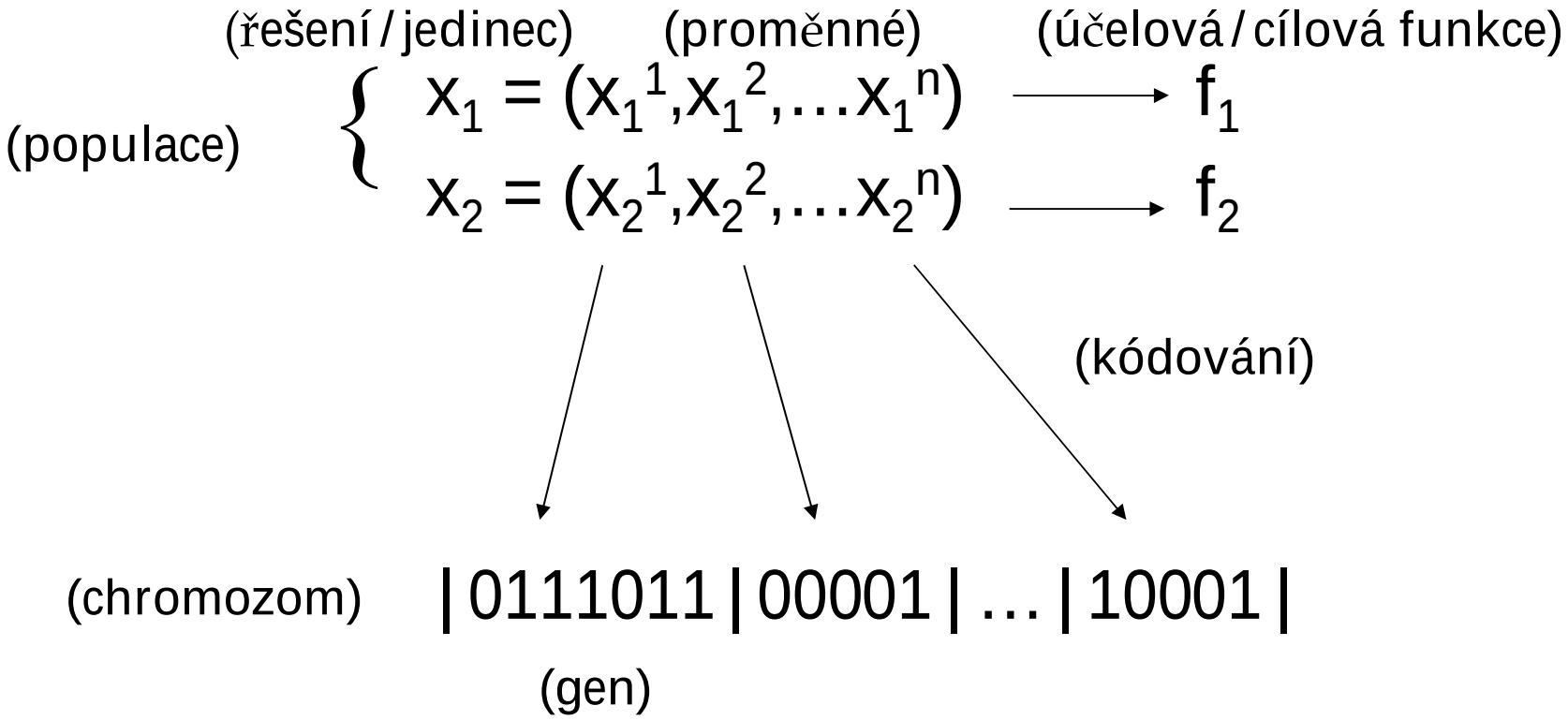
- Založeny na Darwinově myšlence „přežití nejsilnějších“
- Stochastické metody – vykazují náhodné chování
- Ne-gradientní metody- nevyžadují dokonce ani spojitost optimalizované funkce
- Důkaz konvergence založen na tzv. Schema teorému
- Většinou použity až jako poslední řešení, když ostatní metody selžou

Přežití nejkulatějšího



Genetické algoritmy

- Názvosloví z biologie



Genetické algoritmy

(genotyp)	(jedinec)	(fenotyp)	(fitness / síla)
B	x	$f(x) = x^2$	F = pořadí
|0000000000011|	3	9	F = 1
|0000000001001|	9	81	F = 2

Např.:

DNA → Svalovec → Rychlý běžec → Vítězství v závodě

Poznámka: V dnešních aplikacích obvykle účelová funkce a fitness splývají, nebo-li $f \equiv$ F.

Motivace

- Ukázkový příklad:

$$\max f(x) = x^2,\ x[\,0\,;\,33\,]$$

Řešení	Počáteční populace	Hodnota *x*	Fitness	Pravděpod.	Očekávaný počet	Aktuální počet
1	0 1 1 0 1	13	169	0,14	0,58	1
2	1 1 0 0 0	24	576	0,49	1,97	2
3	0 1 0 0 0	8	64	0,05	0,22	0
4	1 0 0 1 1	19	361	0,31	1,23	1
Suma			1170	1	4	4
Avg.			292,5	0,25	1	1
Max			576	0,49	1,97	2

Řešení	Mating pool	místo křížení	Potomci	Hodnota *x*	Fitness
1	0 1 1 0 | 1	4	0 1 1 0 0	12	144
2	1 1 0 0 | 0	4	1 1 0 0 1	25	625
2	1 1 | 0 0 0	2	1 1 0 1 1	27	729
4	1 0 | 0 1 1	2	1 0 0 0 0	16	256
Suma					1754
Avg.					438,5
Max					729

Řešení	Potomci	Po mutaci	Hodnota *x*	Fitness
1	0 1 1 0 0	1 1 1 0 0	26	676
2	1 1 0 0 1	1 1 0 0 1	25	625
2	0 1 0 1 1	0 1 0 1 1	27	729
4	1 0 0 0 0	1 0 0 1 0	18	324
Suma				2354
Avg.				588,5
Max				729

Standardní genetický algoritmus

1 $t = 0$

2 Vytvoř P_0, ohodnoť P_0

3 while (not zastavovací podmínka) {

4 $t = t+1$

5 Vyber M_t z P_{t-1} (aplikace výběru)

6 Změň M_t (použití genetických operátorů)

7 Vytvoř P_t z M_t a ohodnoť P_t (vytvoření nové populace)

8 }

Kde *P* je populace, nebo-li množina možných řešení, *M* je tzv. mating pool (pářecí rybníček), *t* je čítač iterací, zde nazvaných generace

Binární reprezentace proměnných

- Celá čísla jsou v počítači uložena v lehce dostupném binárním kódu, např. pomocí binárních operátorů v jazyce C.
- V případě neexistence transformace mezi celočíselným a binárním zápisem je transformace dána:

1. Délkou binárního řetězce *q*

$$q = \left\lfloor \frac{\ln(max_i - min_i) - \ln(p_i)}{\ln\ 2} \right\rfloor + 1\ ,$$

Binární reprezentace proměnných

2. Binárním řetězcem $\mathrm{B}^i \in \{0;1\}^q$, který je pro danou kladnou celočíselnou hodnotu z_i dán vztahem

$$B_j^i = z_i / 2^{j-1} \mod 2\,, \quad j = 1, \ldots, q\,.$$

3. S inverzním vztahem

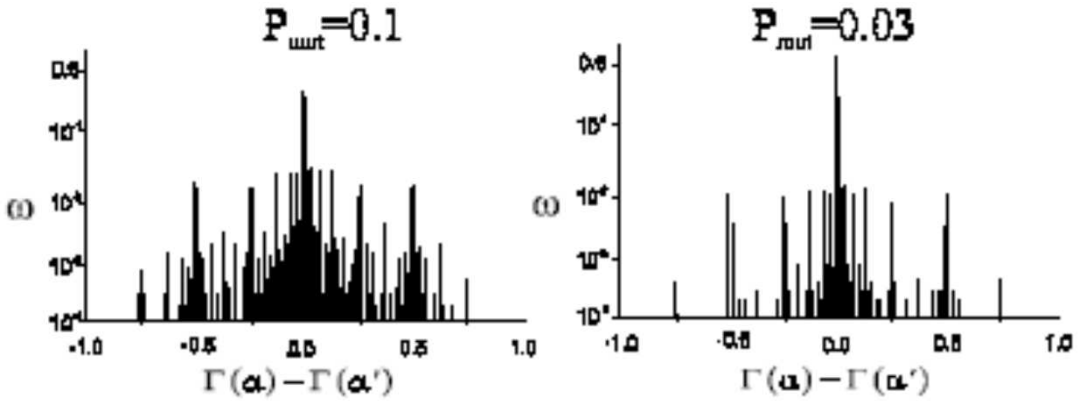
$$z_i = \sum_{j=1}^{q} B_j^i \, 2^{j-1}\,.$$

Hammingova bariéra

dvě čísla:

000000010000000 a
000000001111111

spolu sousedí, ale jsou v binárním kódu daleko od sebe
(mají velkou tzv. Hammingovskou vzdálenost)

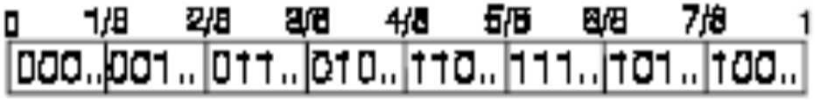


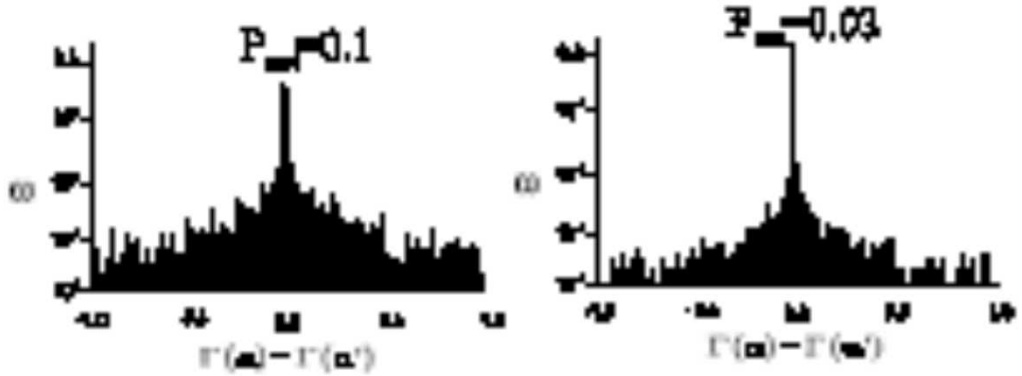
[Kvasnička et. al., 2000]

Hammingova bariéra: řešení

1) Operátor inverze

2) Grayovo kódování





Operátory křížení

Záleží na typu kódování chromozomů:

- Pro binární kódování včetně Grayova kódu:
  - jednobodové křížení

původní chromozomy:

111 | 1111111111
000 | 0000000000

nové chromozomy:

111 | 0000000000
000 | 1111111111

  - *k*-bodové křížení

původní chromozomy:

111 | 11111 | 11111
000 | 00000 | 00000

nové chromozomy:

111 | 00000 | 11111
000 | 11111 | 00000

Operátory křížení

Záleží na typu kódování chromozomů:

- Pro binární kódování včetně Grayova kódu:
  - uniformní křížení

původní chromozomy:

1111111111111
0000000000000

nové chromozomy:

1011000110101
0100111001010

Pozn. Jedno i *k*-bodové křížení zachovávají celistvost řešení, kdežto uniformní křížení schémata rozbíjí

Operátory křížení

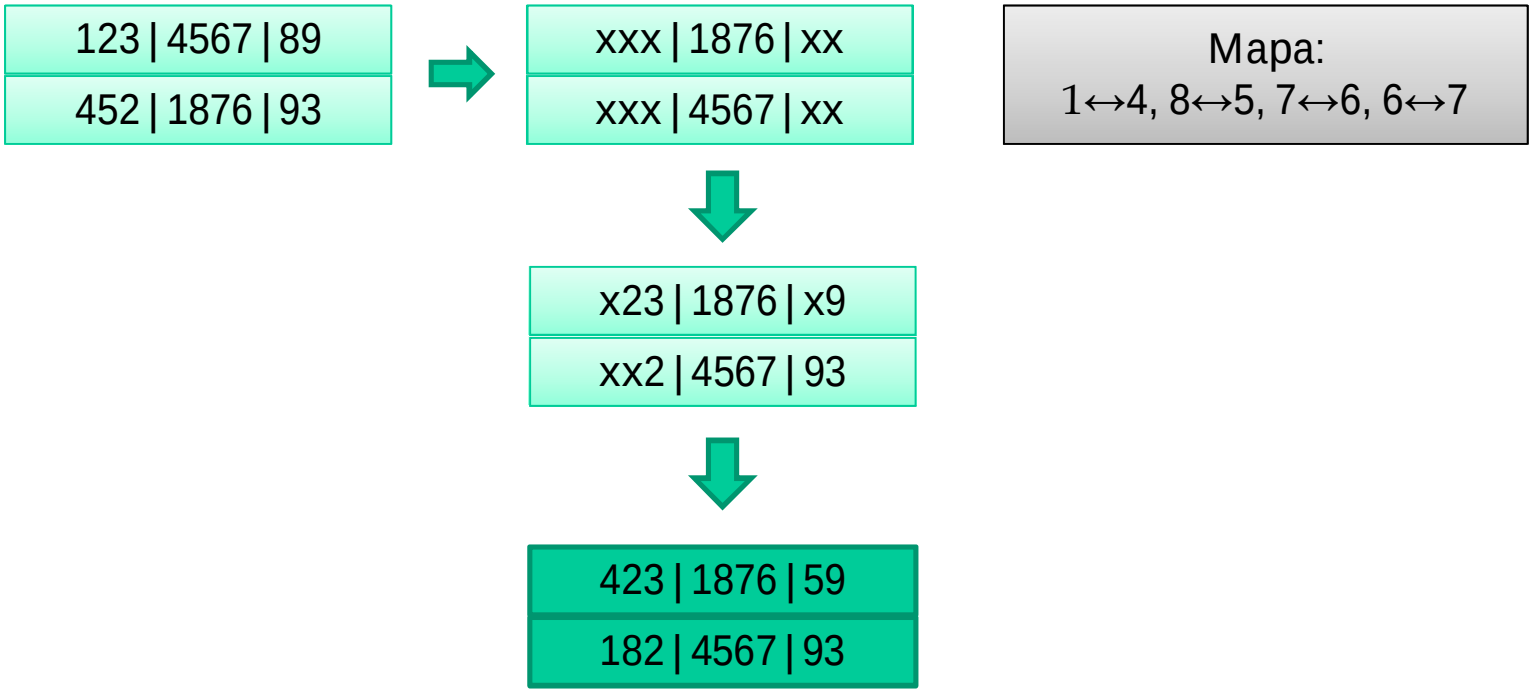
Záleží na typu kódování chromozomů:

- Pro reálné reprezentace chromozomů např.:
  - Aritmetický
    - Potomek 1: $C_1 = \alpha * P_1 + (1- \alpha) \cdot P_2$
    - Potomek 2: $C_2 = \alpha * P_2 + (1- \alpha) \cdot P_1$
    - α je náhodné číslo mezi 0 a 1
  - *blend crossover* BLX-α
    - Dva rodičové P_1 a P_2 se zadanými mezemi L a U
    - $R_1 = \max (L, P_1 - \alpha(P_2 - P_1))$
    - $R_2 = \min (U, P_2 + \alpha(P_2 - P_1))$
    - Potomci jsou vybráni náhodně mezi R_1 a R_2
    - α je většinou voleno jako 0,5

Operátory křížení

Speciální operátory:

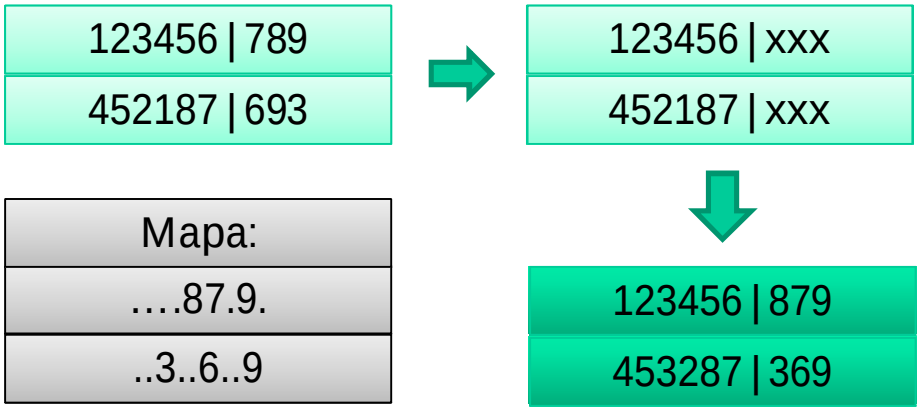
- Křížení s částečným přiřazením (PMX – *partially matched crossover*) – vhodné pro permutační úlohy



Operátory křížení

Speciální operátory:

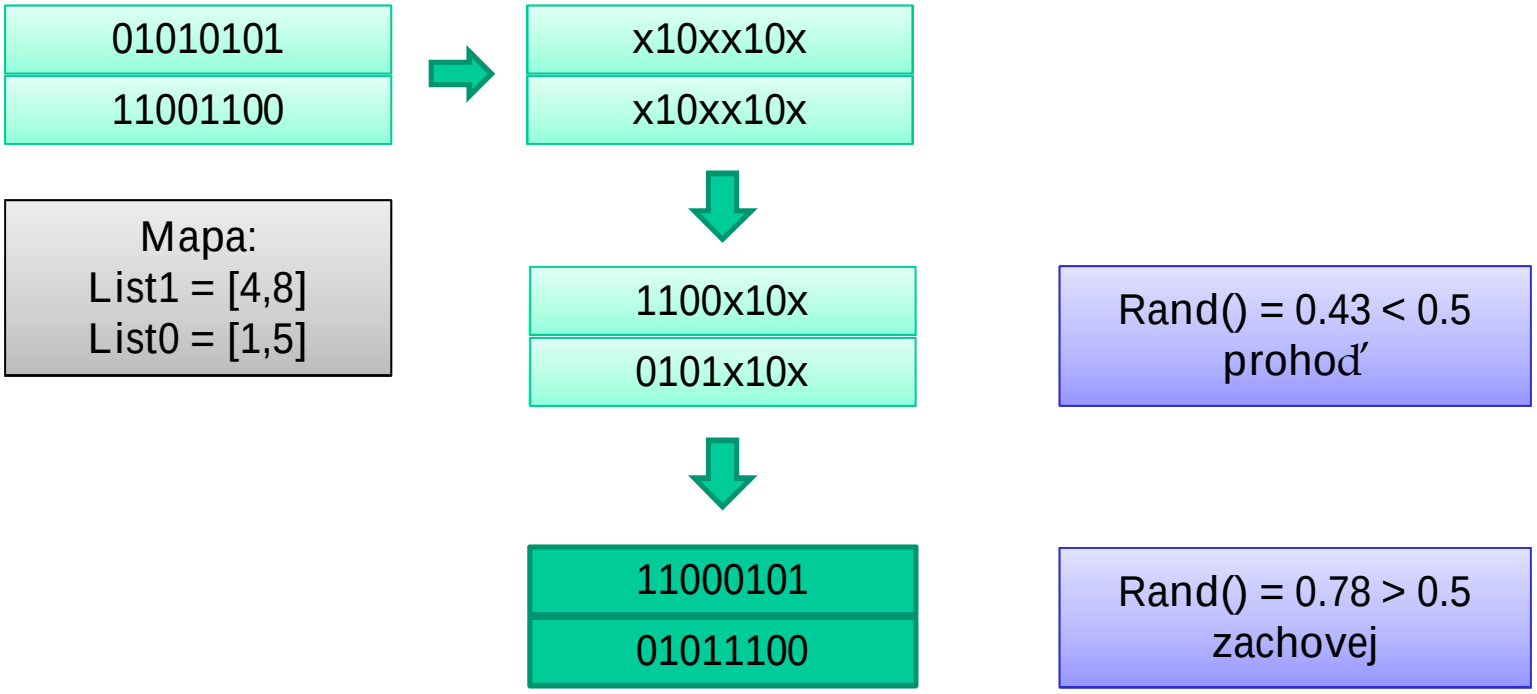
- Uniformní křížení se zachováním pořadí (*uniform order-based crossover*) – vhodné pro permutační úlohy



Operátory křížení

Speciální operátory:

- Křížení se zachováním počtu (CPC – *count preserving crossover*) – pro úlohy zachovávající počet 1 a 0



Operátory mutace

Záleží na typu kódování chromozomů:

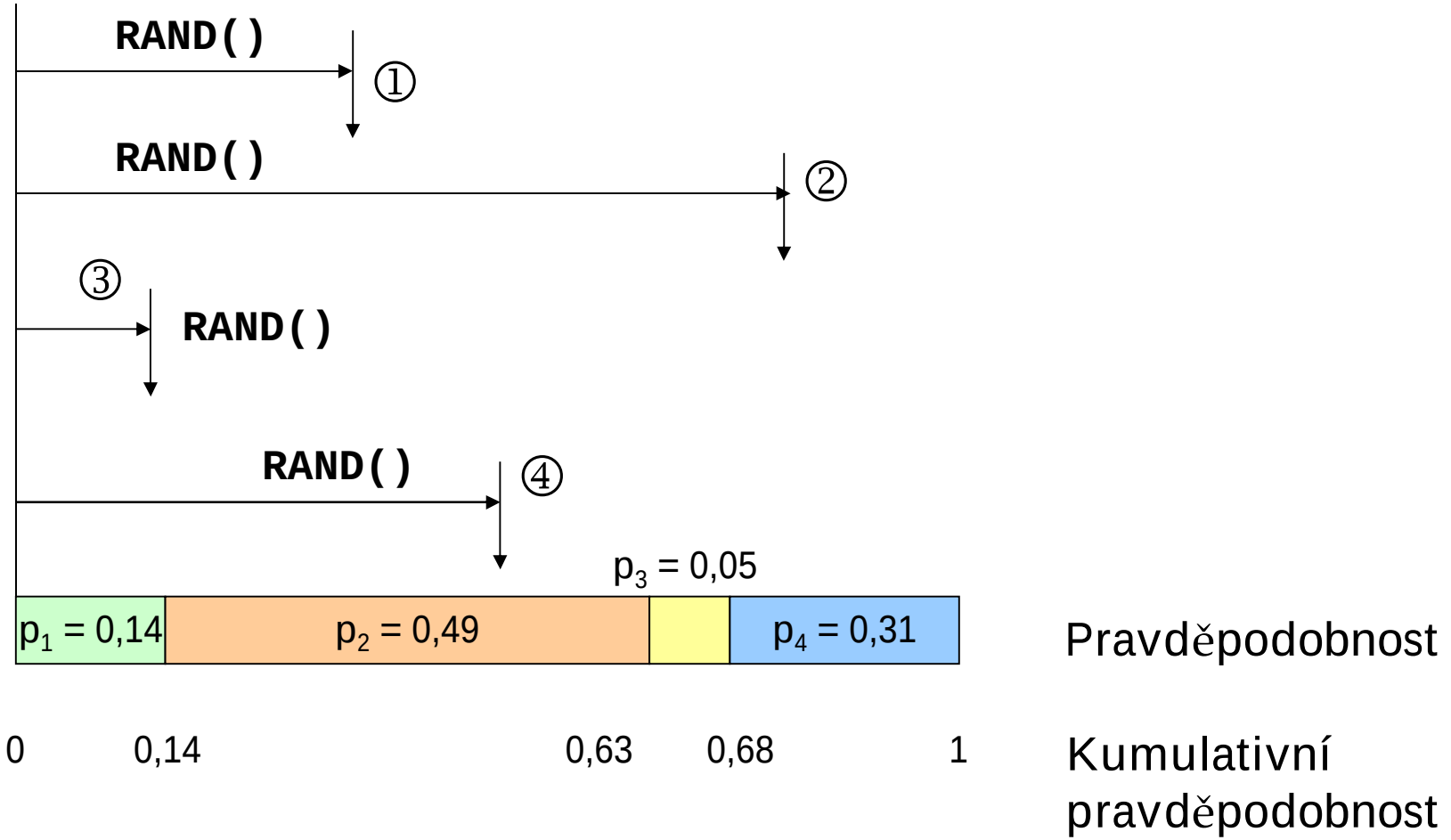
- Pro binární kódování včetně Grayova kódu:
  - **Jednobitová mutace**
  - **Vícebitová mutace** až **Uniformní mutace**
- Pro reálné kódování:
  - **Náhodná výměna** – alela se nahradí jinou
  - **Aditivní změna** – k původní alele se přičte náhodně vygenerovaná hodnota $c \in < -\varepsilon, \varepsilon >$
  - **Gaussovská mutace** – k původní alele se přičítá normálně rozdělené číslo s nulovou střední hodnotou a σ směrodatnou odchylkou
- Speciální operátory pro permutační i objem zachovávající úlohy
  - **Prohození** dvou alel

Výběr řešení

- Ve fázi tvorby *mating pool*, tzv. *parent selection*, nebo při tvoření nové *generace*, tzv. *survivor selection*

- Množství možných metod, zde uvedeme:
  - Roulette wheel (ruleta)
  - SUS – Stochastic Universal Sampling
  - Tournament selection (turnajový výběr)
  - Inverzní turnajový výběr
  - Elitismus
  - Ranking a linear scaling

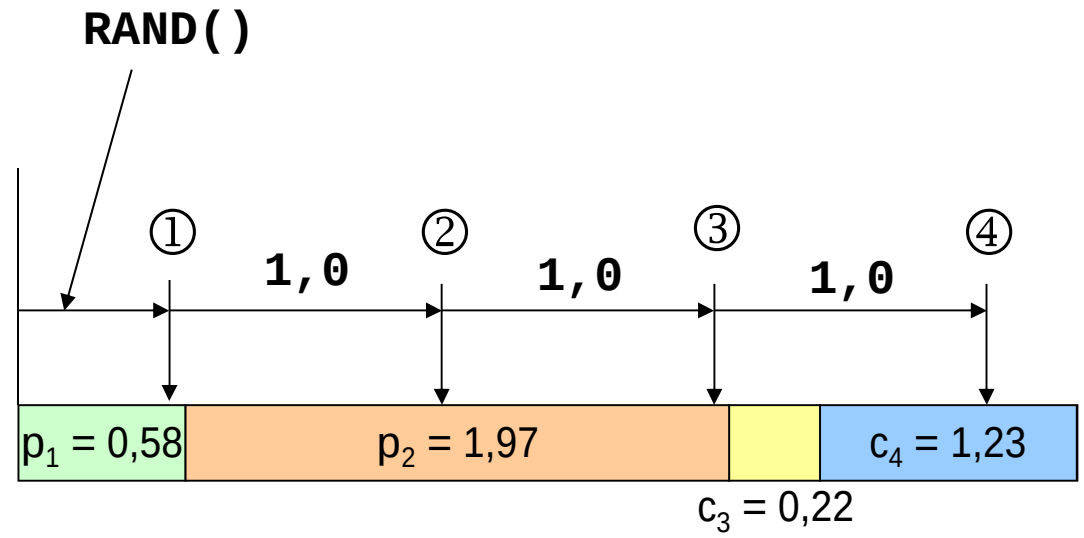
Ruleta


RAND( )
①
RAND( )
②
③
RAND( )
RAND( )
④
p3 = 0,05
p1 = 0,14
p2 = 0,49
p4 = 0,31
Pravděpodobnost
0
0,14
0,63
0,68
1
Kumulativní pravděpodobnost

Ruleta

- Tradiční nástroj
- Nezaručuje vybrání nejlepšího jedince, zejména pro velikosti populace v řádu desítek a výše
- Výpočetně náročná metoda, *n* x **RND( )**

SUS – Stochastic Universal Sampling



Očekávaný počet

Kumulativní očekávaný počet

SUS – Stochastic Universal Sampling

- Nezaručuje vybrání nejlepšího jedince, zejména pro velikosti populace v řádu desítek a výše, ale
- Zaručuje celočíselné očekávané počty, např. řešení s $c = 2{,}0001$ bude vždy vybráno 2x
- Výpočetně nejméně náročná metoda, pouze 1 x **RND()!**

Turnajový výběr

- Vybere z populace náhodně *k* „soutěžících“ (nejčastěji *k* = 2) a z nich vybere nejlepšího (turnaj) a tento postup se provede *n*-krát

- Nezaručuje vybrání nejlepšího jedince tím, že existuje pravděpodobnost, že nebude zařazen do turnaje

- Výpočetně náročné, *k* x *n* x **RND()**

Inverzní turnajový výběr

- Určená pro výběr *p* řešení, kde $p < n$
- Vybere z populace náhodně *k* „soutěžících“ (nejčastěji $k = 2$) a z nich „zahodí“ nejhoršího (inverzní turnaj) a tento postup se provede *n*-krát
- Jako jedna z mála metod zaručuje vybrání nejlepšího jedince
- Nejvíce „demokratická“ metoda – existuje pravděpodobnost vybrání nejhoršího řešení tím, že se nedostane do turnaje
- Výpočetně náročná metoda, *k* x *n* x **RND( )**

Elitismus

- Má dva významy:

1) Výběr nebo kopie nejlepšího řešení do další generace. Používá se nejčastěji z důvodu ztráty takovéhoto řešení vlivem náhody při výběru ruletou či turnajem
2) Vlastnost nebo postup, který upřednostňuje „lepší“ řešení před ostatními.

   Např. turnaj s $k = 3$ je více *elitistický* než turnaj s $k = 2$

Ranking a linear scaling

- Metody na úpravu fitness, na podporu nebo naopak na potlačení *elitismu*
- Ranking přiděluje fitness nikoliv na základě velikosti účelové funkce, ale podle pořadí daného řešení vzhledem k populaci. Např. nejlepší řešení bude mít očekávaný počet kopií 3, druhé nejlepší 2 a ostatní do výběru po 1 a zbytek (nejhorší řešení) 0.

Ranking a linear scaling

- Linear scaling určuje fitness tak, aby fitness populace měla lineární průběh, byl zachován průměr hodnot účelových funkcí a aby maximální fitness byl v intervalu 1,2-2,0 násobku průměru (voleno uživatelem)

Důkaz konvergence: „Schema theorem“

dvě řešení:

0101001010010101010111010101010101 a
0101101010010111000111011101010111

jsou pokryty schematem:

0101#010100101#10#011101#10101#1

Očekávané množství výskytu určitého schematu v čase t+1

$$m(H,t+1) \geq m(H,t) \cdot \frac{f(H)}{\bar{f}} \cdot \left[1 - P_c \frac{\delta(H)}{L-1} - P_m o(H) \right]$$

Viz Doc. Lažanský

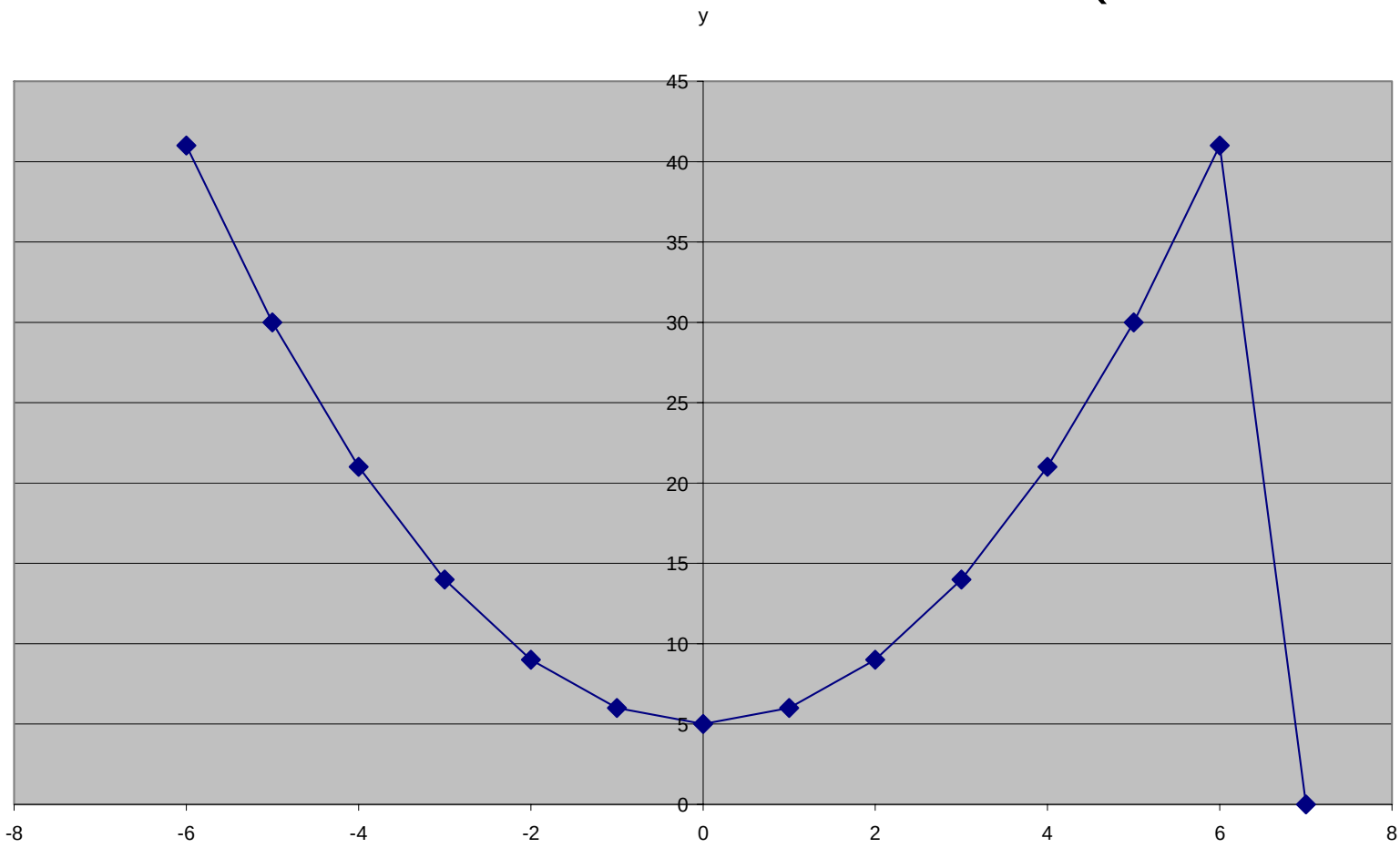
Premature Convergence

(předčasná konvergence)

- V případě, že jeden genotyp obsadí celou populaci, je populace *konvergovaná*
- Konvergence je *předčasná*, pokud nebylo nalezeno požadované řešení
- Řešení:
  - Udržování diversity v populaci
  - Multi-modální řešení

Deceptive functions

(klamavé funkce)



Viz. Robust design optimization

Reference

[1] Holland, J. H. (1975). Adaptation in Natural and Artificial Systems. MIT Press.

[2] Goldberg, D. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

[3] Michalewicz, Z. (1992). Genetic Algorithms + Data Structures = Evolution Programs. AI Series. Springer-Verlag, New York.

[4] Wolfram, S. (2002). Cellular Automata and Complexity. Perseus Publishing.

[5] A. E. Eiben, J. E. Smith (2003). Introduction to Evolutionary Computing. Springer.

[6] Dreo J, Petrowsky A, Siarry P and Taillard E. (2006). Metaheuristics for hard optimization. Methods and case studies. Berlin Heidelberg: Springer

Reference

[7] P. Convey, R. Highfield: Mezi chaosem a řádem. Mladá fronta, 2003.

[8] Dawkins, Richard: Sobecký gen. Mladá fronta, edice: Kolumbus, 2003.

[9] Zrzavý, Jan; Storch, David; Mihulka, Stanislav: Jak se dělá evoluce. Paseka, edice: Fénix, 2004.

[10] Mařík, V. a kol: Umělá inteligence I-IV, Academia, 1996—2003.

[11] V. Kvasnička, J. Pospíchal, P Tiňo (2000). Evolučné algoritmy. STU Bratislava.

Prosba. V případě, že v textu objevíte nějakou chybu nebo budete mít námět na jeho vylepšení, ozvěte se prosím na **matej.leps@fsv.cvut.cz**.

Oprava 28.3.2008: Strana 37,34, Opravena reference [4] a přidána refernce [6]
Oprava 28.3.2008: Strana 18, Opravena reference
Oprava 9.11.2008: Strana 12 a 14, Oprava a zvýraznění
Oprava 9.11.2008: Vymazány stránky s celočíselným a reálným kódováním
Oprava 9.11.2008: Prohozena reference [4] a [5]

Datum poslední revize: 9.11.2008
Verze: 002