

# Evoluční algoritmy

# Evoluční algoritmy

Třída optimalizačních metod inspirovaných principy biologické evoluce. Mezi nejvýznamnější přístupy patří:

- ▶ **Genetické algoritmy:** Simulují přirozený výběr a genetické operace, jako jsou křížení a mutace, s cílem nalézt optimální řešení.
- ▶ **Evoluční strategie:** Zaměřují se na adaptaci parametrů mutací a selekce, často za použití reálných čísel.
- ▶ **Genetické programování:** Evoluce programů nebo funkcí, které řeší specifické problémy.
- ▶ **Diferenciální evoluce:** Využívá rozdílů mezi vektory v populaci k iterativnímu zlepšování řešení.

Společným rysem je aplikace evolučních principů na populace potenciálních řešení daného problému.

# Inspirace z přírody



Charles Darwin

*On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life.*

(1859)

**Základní princip:** Výhodné vlastnosti jedinců vznikající **náhodnou variací** jsou zvýhodněny díky mechanismu nazývanému **přírozený výběr**.

**Propojení s evolučními algoritmy:** Evoluční algoritmy napodobují tento princip tak, že náhodně modifikují populaci řešení a vybírají ta nejúspěšnější pro další iterace.

# Mechanismy genetické variability

Nové nebo modifikované vlastnosti jedinců vznikají prostřednictvím následujících mechanismů:

- ▶ **Mutace:** Náhodné změny v DNA, které mohou ovlivnit fenotyp.
- ▶ **Křížení:** Výměna genetického materiálu mezi chromozomy během tvorby pohlavních buněk.
- ▶ **Přírozený výběr:** Jedinci s výhodnými vlastnostmi mají vyšší šanci na přežití a reprodukci, což vede k šíření těchto vlastností v populaci.

**Inspirace pro evoluční algoritmy:** Tyto biologické mechanismy slouží jako základ pro klíčové operace evolučních algoritmů: *mutaci, křížení a selekci.*

## Další principy evoluce

Kromě tří základních principů (mutace, křížení, přirozený výběr) se v biologické evoluci uplatňují i další mechanismy, které nacházejí využití v optimalizačních metodách:

- ▶ **Derivační drift:** Náhodné změny ve frekvenci alel v malé populaci.
- ▶ **Koevoluce:** Společná evoluce dvou nebo více druhů, které se navzájem ovlivňují.
- ▶ **Epistáze:** Interakce mezi geny, kde účinek jednoho genu závisí na přítomnosti jiných genů.
- ▶ **Paralelní evoluce:** Současná evoluce podobných znaků u nepříbuzných druhů.

### Doporučená literatura:



Rudolf Kruse *et al.*

*Computational Intelligence: A Methodological Introduction.*  
Springer, 2022 (3rd Edition)

# Umělá (simulovaná) evoluce

Principy biologické evoluce lze využít pro řešení optimalizačních problémů na základě představy, že udržujeme **populaci** potenciálních řešení daného problému a tato populace prochází iterativním procesem **evoluce**.

**Proces evoluce:** Proces inspirovaný biologickou evolucí (např. mutace, křížení, selekce) iterativně zlepšuje kvalitu řešení v populaci, dokud nenalezneme **uspokojivé řešení** problému.

# Základní schéma evolučního algoritmu

Abychom mohli specifikovat evoluční algoritmus, je třeba definovat:

- ▶ **Způsob kódování kandidátních řešení:**

- ▶ Klíčový krok, závisí na řešeném problému.
- ▶ **Populace** je množina kódů reprezentujících kandidátní řešení.
- ▶ *Příklad:* Celá čísla  $n \in \mathbb{N}$  lze kódovat jako binární řetězce.

- ▶ **Počáteční populace:**

- ▶ Typicky se generuje náhodně, přičemž respektuje omezení zadaná problémem.
- ▶ *Příklad:* Generování náhodných binárních řetězců délky  $k$ .

- ▶ **Fitness funkce ( $f$ ):**

- ▶ Ohodnocuje vhodnost jednotlivých řešení.
- ▶ Obvykle odpovídá optimalizované funkci, např.  $f(x)$  maximalizuje zisk nebo minimalizuje chybu.

**Poznámka:** Kódování a fitness funkce výrazně ovlivňují kvalitu výsledků algoritmu.

## Další klíčové aspekty evolučního algoritmu

### ► Genetické operátory:

- Slouží k modifikaci chromozomů (kódů kandidátních řešení).
- Mohou být generické (např. **mutace**, **křížení**) nebo vysoce závislé na problému.
- Výsledek musí respektovat omezení problému.

*Příklad:* V binárním řetězci změnit náhodný bit.

### ► Ukončovací podmínka:

- Definuje, kdy algoritmus skončí.

*Příklad:* Po dosažení stanoveného počtu generací, nebo pokud fitness nejlepšího řešení překročí požadovanou hodnotu.

### ► Parametry algoritmu:

- Ovlivňují výkon a konvergenci algoritmu.

*Příklady parametrů:* velikost populace, pravděpodobnost mutace (např. 0.01), počet jedinců použité k výběru pro křížení.

**Poznámka:** Správná volba parametrů a operátorů je klíčová pro efektivitu algoritmu.

# Schéma evolučního algoritmu

```
 $t \leftarrow 0;$ 
initialize-population( $t$ );
// Vytvoření počáteční populace;

evaluate-population( $t$ );
// Vyhodnocení fitness všech jedinců;

while not termination-criterion do
    // Hlavní evoluční smyčka;

     $t \leftarrow t + 1;$ 
    // Posun na další generaci;

    select-population( $t$ );
    // Výběr jedinců pro další generaci;

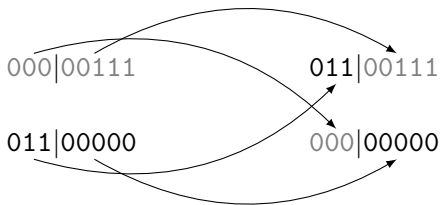
    genetic-alter-population( $t$ );
    // Aplikace mutací a křížení;

    evaluate-population( $t$ );
    // Vyhodnocení nové populace;
end
```

## Základní metoda kódování:

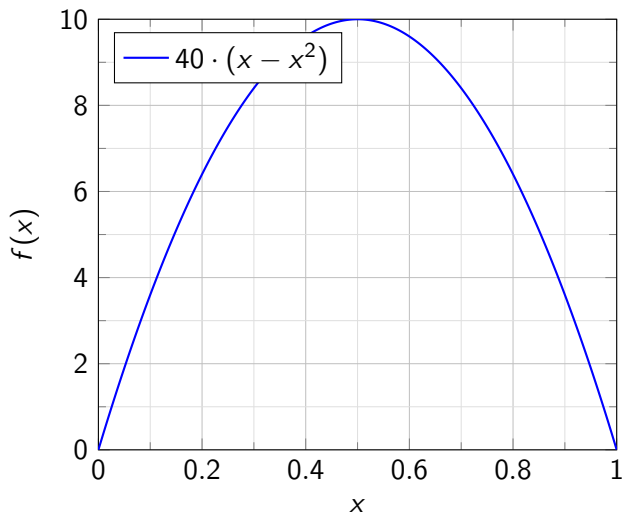
kandidátní řešení  $\rightarrow$  řetězce bitů.

## Základní křížení:



# Příklad

Budeme hledat maximum funkce:



Náhodně vygenerujeme osm řetězců délky 8, které kódují čísla z intervalu  $\langle 0, 1 \rangle$  jako odpovídající binární číslo vydělené 255.

Ohodnotíme jejich kvalitu  $f$ :

- |    |          |                            |
|----|----------|----------------------------|
| 1. | 11110011 | $f(243/255) \approx 1.794$ |
| 2. | 01010101 | $f(85/255) \approx 8.889$  |
| 3. | 00011101 | $f(29/255) \approx 4.032$  |
| 4. | 00111001 | $f(57/255) \approx 6.943$  |
| 5. | 01100000 | $f(96/255) \approx 9.390$  |
| 6. | 10000011 | $f(131/255) \approx 9.992$ |
| 7. | 00000111 | $f(7/255) \approx 1.068$   |
| 9. | 11111000 | $f(248/255) \approx 1.068$ |

Naším cílem je, aby algoritmus na základě znalostí kvality řetězců vytvořil novou generaci, tak aby

- ▶ se na jejím složení více podílely silnější řetězce,
- ▶ a méně nebo vůbec slabší řetězce

Mechanismus tvorby nové generace se děje ve třech krocích:

## Reprodukce

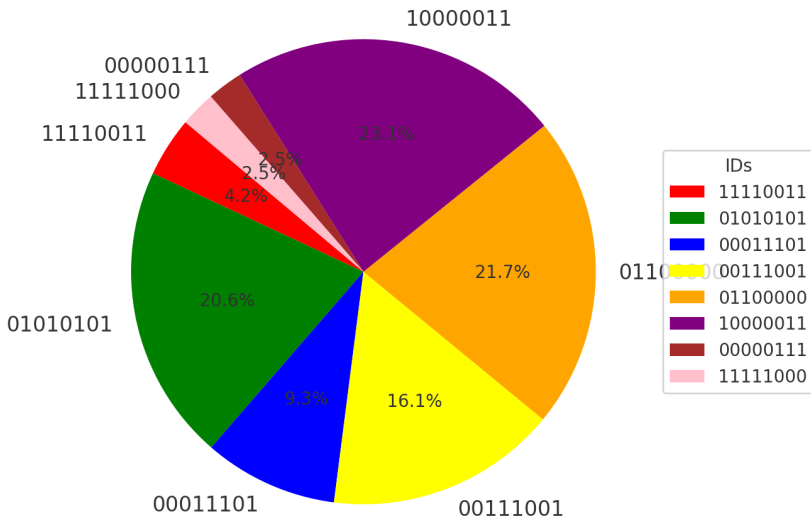
Do nové generace se kopírují řetězce ze staré generace.

Čím je řetězec lepší, tím větší je pravděpodobnost, že bude do nové generace okopírován.

**Základní způsob:** výběr pomocí vážené rulety

| řetězec  | $f(x)$ | pst   |
|----------|--------|-------|
| 11110011 | 1.794  | 4.16  |
| 01010101 | 8.889  | 20.59 |
| 00011101 | 4.032  | 9.34  |
| 00111001 | 6.943  | 16.08 |
| 01100000 | 9.390  | 21.75 |
| 10000011 | 9.992  | 23.14 |
| 00000111 | 1.068  | 2.47  |
| 11111000 | 1.068  | 2.47  |

# Pie Chart with Probabilities



Ruletu tedy  $8\times$  roztočíme a získáme tak novou generaci.  
Získaná generace bude vypadat třeba takto:

|    |          |              |                 |
|----|----------|--------------|-----------------|
| 1. | 00111001 | $f(57/255)$  | $\approx 6.943$ |
| 2. | 01010101 | $f(85/255)$  | $\approx 8.889$ |
| 3. | 01010101 | $f(29/255)$  | $\approx 8.889$ |
| 4. | 10000011 | $f(57/255)$  | $\approx 9.992$ |
| 5. | 01100000 | $f(96/255)$  | $\approx 9.390$ |
| 6. | 10000011 | $f(131/255)$ | $\approx 9.992$ |
| 7. | 00000111 | $f(7/255)$   | $\approx 1.068$ |
| 9. | 01100000 | $f(96/255)$  | $\approx 9.390$ |

Sice výrazný nárůst průměrné kvality ( $5.397 \rightarrow 8.069$ ), ale:

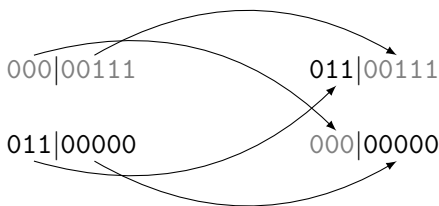
- ▶ Nemáme žádné nový řetězec (oproti výchozí populaci).
- ▶ Máme tam duplikované řetězce.

# Křížení (Crossover)

**Hlavní myšlenka:** Křížení umožňuje výměnu genetické informace mezi řetězci, což zajišťuje kombinaci vlastností obou rodičů.

## Postup:

- ▶ Z populace náhodně vybereme dva **řetězce** (rodiče).
- ▶ Tyto dva řetězce slouží jako **rodiče** pro vytvoření potomků.
- ▶ Každý **potomek** obsahuje část genetické informace od každého z rodičů.
- ▶ **Rozdělovací bod (crossover point):** Určuje, kde se genetická informace mezi rodiči dělí.



$$f(00000000) = 0$$

$$f(01100111) = 9.631$$

# Pravděpodobnost křížení

Ke křížení dvou vybraných řetězců nedochází vždy.

Pravděpodobnost křížení (*crossover rate*) určuje, jak často k této operaci dojde. Typická hodnota:  $p = 0.6$ .

## Postup:

- ▶ S pravděpodobností  $p$  ( $p = 0.6$ ) proběhne křížení: Potomci vzniklí z křížení se zařadí do nové generace.
- ▶ S pravděpodobností  $1 - p$  ( $1 - 0.6 = 0.4$ ): Do nové generace přecházejí **nezměněné** řetězce rodičů.

## Význam:

Náhodná pravděpodobnost křížení umožňuje:

- ▶ **Zachování kvalitních řešení:** Dobré řetězce zůstávají nezměněny.
- ▶ **Genetickou diverzitu:** Křížení kombinuje vlastnosti rodičů.

1. 00111011  $f(59/255) \approx 7.114$
2. 01010101  $f(85/255) \approx 8.889$
3. 01010000  $f(80/255) \approx 8.612$
4. 10000011  $f(57/255) \approx 9.992$
5. 01100101  $f(96/255) \approx 9.390$
6. 10000001  $f(129/255) \approx 9.999$
7. 00000000  $f(0/255) \approx 0.000$
9. 01100111  $f(103/255) \approx 9.631$

# Mutace

**Hlavní myšlenka:** Mutace probíhá s **nízkou pravděpodobností** (např.  $p = 0.001$ ) a slouží k zavedení genetické variability do populace.

## Postup:

- ▶ Každý řetězec v generaci procházíme postupně, bit po bitu.
- ▶ Pro každou pozici (bit) s pravděpodobností  $p$  zaměníme 1 za 0 nebo 0 za 1.

## Význam mutace:

- ▶ Zabraňuje předčasné konvergenci k lokálnímu minimu.
- ▶ Umožňuje objevování nových oblastí v prostoru řešení.

# Schémata a kódování

**Schémata:** Schéma lze chápat jako **šablonu**, která popisuje podmnožinu všech možných řetězců dané délky.

## Vlastnosti schématu:

- ▶ Schéma má stejnou délku jako řetězce v populaci.
- ▶ Obsahuje symboly 0, 1, \*.
- ▶ Symbol \* zastupuje libovolnou hodnotu (0 nebo 1).

**Význam:** Schémata umožňují efektivně analyzovat a sledovat vlastnosti podmnožin řetězců během evolučního procesu.

# Příklad schématu

## Example

Schéma řetězce délky 8 může vypadat například takto: **\*\*1\*0010**.

**Podmínka shody:** Řetězec odpovídá schématu, pokud se shodují všechny **určené pozice**, tj. ty, kde se ve schématu nachází 0 nebo 1.

**Příklad:** Se schématem **\*\*1\*0010** se shodují následující řetězce:

00100010

00110010

01100010

10100010

01110010

10110010

11100010

11110010

# Kvalita schématu v generaci

**Definice: Kvalita schématu** v generaci je definována jako **průměr fitness všech řetězců** v generaci, které odpovídají danému schématu.

**Příklad:** Vezměme si generaci po křížení s následujícími řetězci a jejich fitness:

| # | Řetězec  | Normalizovaná hodnota | Fitness $f$ |
|---|----------|-----------------------|-------------|
| 1 | 00111011 | 59/255                | 7.114       |
| 2 | 01010101 | 85/255                | 8.889       |
| 3 | 01010000 | 80/255                | 8.612       |
| 4 | 10000011 | 57/255                | 9.992       |
| 5 | 01100101 | 96/255                | 9.390       |
| 6 | 10000001 | 129/255               | 9.999       |
| 7 | 00000000 | 0/255                 | 0.000       |
| 8 | 01100111 | 103/255               | 9.631       |

**Schéma:** Uvažujme schéma 0101\*\*\*\*.

V generaci jsou reprezentovány dvěma řetězci: 01010101  
a 01010000.

**Kvalita schématu:**

$$\text{Kvalita} = \frac{8.889 + 8.612}{2} = 8.751$$

**Význam:** Kvalita schématu umožňuje sledovat, jak efektivně dané podmnožiny řešení přispívají k optimalizaci.

# Změna optimalizované funkce

Uvažujme změnu funkce tak, že se její hodnota liší pouze v bodě 10000000 (odpovídajícím  $x = 128/255$ ):

$$y = \begin{cases} 12 & \text{pro } x = \frac{128}{255}, \\ 40(x - x^2) & \text{jinak.} \end{cases}$$

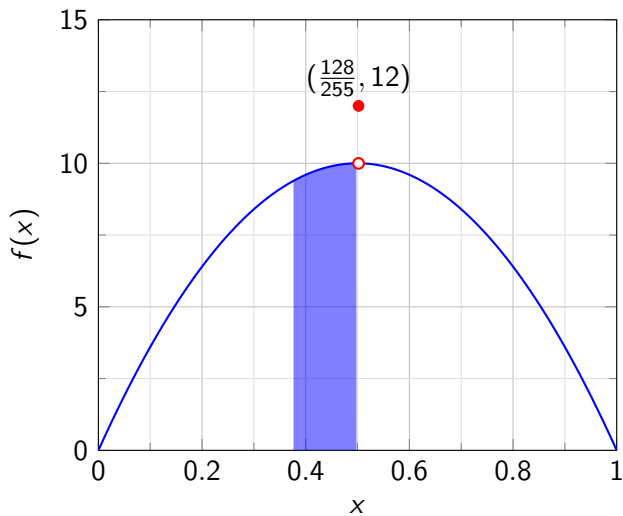
**Předchozí stav:** V původní funkci měly body 01111111 a 10000000 stejnou kvalitu a reprezentovaly maximum (v rámci diskretizace).

**Po změně:**

- ▶ Hodnota funkce pro 10000000 byla zvýšena na 12.
- ▶ 10000000 nyní představuje **nové globální maximum**.

**Důsledek:** Tato změna ovlivní průběh optimalizace, protože řetězce blízké 10000000 budou pravděpodobně preferovány evolučním algoritmem.

Naše situace:



## Dominance schématu 011\*\*\*\*\*

V našem kódování může nastat situace, kdy se většina řetězců v generaci shoduje se schématem 011\*\*\*\*\*.

**Proč?** Toto schéma reprezentuje **množinu velmi silných řetězců**:

- ▶ Nejslabší řetězec odpovídající tomuto schématu: 01100000, kde  $f(01100000) = 9.390$ .
- ▶ Průměrná kvalita všech řetězců odpovídajících schématu: 9.790.

**Problém:** Je velmi nepravděpodobné, že bychom z takové generace získali řetězec 10000000.

**Potřeba mutace:** Museli bychom zmutovat první 0 řetězce, čímž bychom získali řetězec odpovídající schématu 111\*\*\*\*\*.

Ale...

# Problém mutace směrem k 111\*\*\*\*\*

Pokud bychom zmutovali první 0, získali bychom řetězec odpovídající schématu 111\*\*\*\*\*.

## Problém:

- ▶ **Nejsilnější řetězec** ve schématu 111\*\*\*\*\*: kvalita 4.272.
- ▶ **Průměrná kvalita** řetězců v tomto schématu: 2.231.

## Důsledek:

- ▶ Řetězec odpovídající schématu 111\*\*\*\*\* **téměř nemá šanci přežít selekci** kvůli nízké kvalitě ve srovnání se silnějšími řetězci v generaci.
- ▶ Naopak, **řetězec** 01111111 je pravděpodobnější volbou, protože odpovídá silnému schématu 011\*\*\*\*\*.

## Pozorování:

- ▶ Nejlepší řetězec (10000000) a druhý nejlepší řetězec (01111111) jsou **sousední řetězce**.
- ▶ Evoluční algoritmus má potíže přejít z 011\*\*\*\*\* na 111\*\*\*\*\* kvůli silné dominanci schématu 011\*\*\*\*\*.

# Problém sousedních silných schémat

**Podstata problému:** V definičním oboru jsou za sebou dvě **silná schémata**, jejichž řetězce jsou zpravidla vysoce nadprůměrné.

Tato schémata jsou však **protichůdná**:

- ▶ V intervalu  $(0.376, 0.5)$  dominuje schéma 011\*\*\*\*\*.
- ▶ V intervalu  $(0.5, 0.624)$  dominuje schéma 100\*\*\*\*\*.

**Důsledek:**

- ▶ Pokud se přibližujeme k nejvyšší hodnotě zleva: Generace bude převážně obsahovat řetězce odpovídající schématu 011\*\*\*\*\*, což znesnadní přechod k 100\*\*\*\*\*.
- ▶ Pokud se přibližujeme k nejvyšší hodnotě zprava: Generace bude převážně obsahovat řetězce odpovídající schématu 100\*\*\*\*\*, což **zvyšuje šanci na nalezení nejlepší hodnoty**.

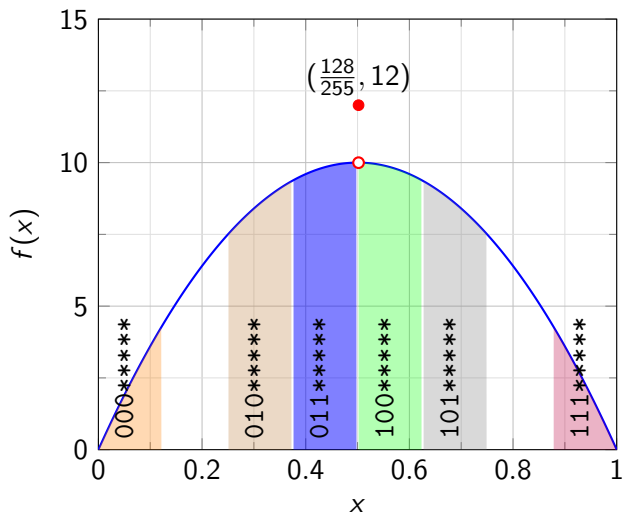
# Problém křížení rovnocenných schémat

**Situace:** Generace může dospět do stavu, kdy jsou řetězce odpovídající dvěma zkoumaným schématům (011\*\*\*\*\* a 100\*\*\*\*\* ) **rovnoměrně zastoupeny**.

## Důsledek křížení:

- ▶ Pokud zkřížíme dva řetězce (každý odpovídá jinému schématu) na první nebo druhé pozici:
  - ▶ Vzniknou dva nové řetězce.
  - ▶ Tito potomci budou **slabší než jejich rodiče**.
- ▶ Nové řetězce budou odpovídat schématům: 101\*\*\*\*\*, 010\*\*\*\*\*, 111\*\*\*\*\*, nebo 000\*\*\*\*\*.

**Důsledek pro optimalizaci:** Křížení mezi rovnocennými schématy vede ke **snížení průměrné kvality** generace, což zpomaluje nebo úplně zastavuje pokrok směrem k optimálnímu řešení.



# Grayův kód

**Vlastnost:** Grayův kód zajišťuje, že dva sousední body v prohledávaném prostoru se liší pouze na jediné pozici, což zmenšuje riziko výrazných změn při mutaci nebo křížení.

## Konstrukce:

- ▶ První bit Grayova kódu je stejný jako první bit binárního čísla.
- ▶ Každý další bit Grayova kódu je XOR předchozího bitu Grayova kódu a příslušného bitu binárního čísla.

## Příklad: Převod binárního čísla 1011 na Grayův kód:

- ▶ **Krok 1:** První bit Grayova kódu je stejný jako první bit binárního čísla.

$$G_1 = B_1 = 1$$

- ▶ **Krok 2:** Další bit Grayova kódu je XOR aktuálního a předchozího bitu binárního čísla:

$$G_2 = B_2 \oplus B_1 = 0 \oplus 1 = 1$$

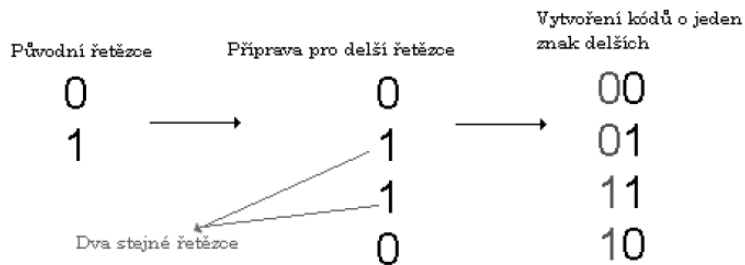
- ▶ **Krok 3:** Pokračujeme XOR dalších bitů:

$$G_3 = B_3 \oplus B_2 = 1 \oplus 0 = 1$$

$$G_4 = B_4 \oplus B_3 = 1 \oplus 1 = 0$$

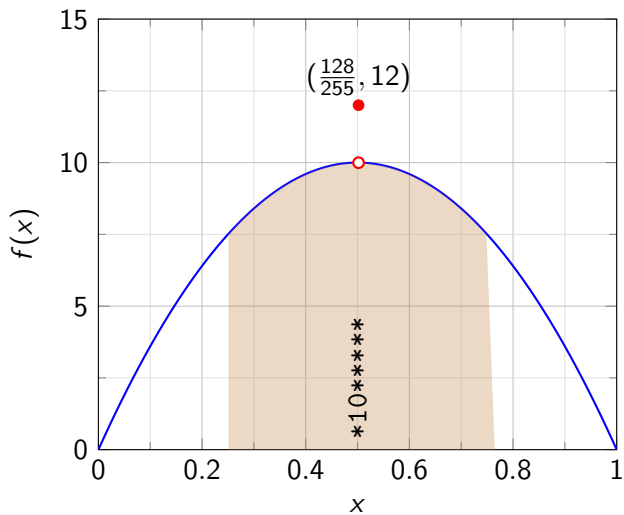
- ▶ **Výsledek:** Grayův kód pro 1011 je 1110.

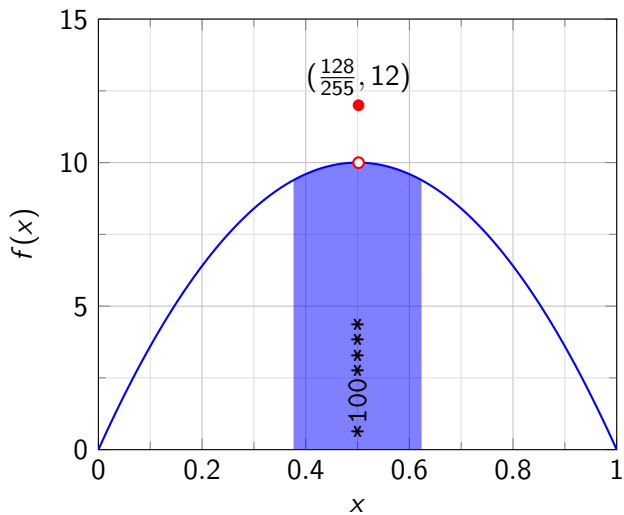
# Konstrukce



Rozdíl oproti binárnímu kódu:

|    |    |     |     | Grayův<br>kód | Binární<br>kód |
|----|----|-----|-----|---------------|----------------|
| 00 | 00 | 000 | 000 | 0000          | 0000           |
| 01 | 01 | 001 | 001 | 0001          | 0001           |
| 11 | 11 | 011 | 011 | 0011          | 0010           |
| 10 | 10 | 010 | 010 | 0010          | 0011           |
|    | 10 | 110 | 110 | 0110          | 0100           |
|    | 11 | 111 | 111 | 0111          | 0101           |
|    | 01 | 101 | 101 | 0101          | 0110           |
|    | 00 | 100 | 100 | 0100          | 0111           |
|    |    |     | 100 | 1100          | 1000           |
|    |    |     | 101 | 1101          | 1001           |
|    |    |     | 111 | 1111          | 1010           |
|    |    |     | 110 | 1110          | 1011           |
|    |    |     | 010 | 1010          | 1100           |
|    |    |     | 011 | 1011          | 1101           |
|    |    |     | 001 | 1001          | 1110           |
|    |    |     | 000 | 1000          | 1111           |





# Elitářství a škálování

**Problém:** Jedním z problémů v evolučních algoritmech je možnost **ztráty dosud nalezeného maxima**, což zahrnuje i potenciální ztrátu globálního maxima.

**Příklad:** V našem příkladu s Grayovým kódem:

- ▶ Všechny řetězce odpovídající schématu `*100****` kódují body z intervalu  $\langle 0.439, 0.561 \rangle$ .
- ▶ Kvalita těchto řetězců se pohybuje v úzkém intervalu  $\langle 9.852, 10 \rangle$ .

**Důsledek:**

- ▶ Všechny řetězce mají přibližně stejnou pravděpodobnost výběru při selekci.
- ▶ Pokud máme generaci o 100 řetězcích, pravděpodobnost vylosování nejlepšího z nich může být jen o málo vyšší než  $1/100$ .

## **Řešení: Elitářství**

Elitářství chrání nejlepší nalezená řešení tím, že zajišťuje jejich přežití do další generace bez ohledu na výsledek rulety.

To pomáhá předcházet ztrátě globálního maxima.

## Princip:

- ▶ Určíme počet  $m$  nejlepších řetězců (**elit**), které automaticky postoupí do další generace.
- ▶ Typicky je  $m = 1$ .
- ▶ Elity nejsou v tomto kroku ovlivněny křížením ani mutací.

## Postup:

- ▶ Ještě před zahájením reprodukce (losování nové generace) se do nové generace zkopíruje  $m$  nejlepších řetězců z aktuální generace.
- ▶ Poté probíhá standardní reprodukce, křížení a mutace zbývajících řetězců.

# Problém podobné kvality a fitness scaling

## Problém:

- ▶ Pokud řetězce kódují body v blízkém okolí maxima a mají **podobnou kvalitu**, algoritmus přestává hledat lepší řešení.
- ▶ Algoritmus se začne **náhodně pohybovat** ve vymezeném okolí maxima.
- ▶ Důvodem je zhruba **rovnoměrné rozdělení pravděpodobností na ruletě**.

**Potřeba řešení:** Abychom zvýšili šanci na výběr lepších řetězců, potřebujeme **více rozlišovat** mezi řetězci, jejichž kvalita se téměř neliší.

**Řešení: Škálování kvality (fitness scaling)** Škálování fitness upravuje hodnoty kvality tak, aby:

- ▶ Zvýraznilo rozdíly mezi blízkými hodnotami kvality.
- ▶ Zvýšilo šanci na výběr lepších řetězců.
- ▶ Zabránilo stagnaci v okolí maxima.

# Škálování kvality

**Idea:** Přepočítáme kvality řetězců tak, aby byly rozdíly mezi nimi výraznější, zatímco **zachováme pořadí** mezi lepšími a horšími řešeními.

**Lineární škálování:** Pro každý řetězec přepočítáme kvalitu podle vztahu:

$$\bar{f} = a + f \cdot b,$$

kde:

- ▶  $f$  je původní kvalita.
- ▶  $\bar{f}$  je naškálovaná kvalita.
- ▶  $a, b$  jsou koeficienty určené podle následujících pravidel:

**Podmínky pro  $a, b$ :**

$$\bar{f}_{\text{avg}} = f_{\text{avg}}, \quad (\text{průměrná kvalita zůstává zachována}),$$

$$\bar{f}_{\text{max}} = C_{\text{mult}} \cdot f_{\text{avg}}, \quad (\text{např. } C_{\text{mult}} = 1.2 \text{ nebo } C_{\text{mult}} = 2).$$

# Proč zachování průměru?

Zachování průměrné kvality v generaci zajišťuje stabilitu pravděpodobnosti výběru řetězců a počtu jejich očekávaných potomků.

## Důvody:

- ▶ Řetězec s kvalitou **rovnou průměru** ( $f_{\text{avg}}$ ) má pravděpodobnost výběru:

$$P = \frac{1}{n}, \quad \text{kde } n \text{ je počet řetězců v generaci.}$$

- ▶ **Očekávaný počet potomků** takového řetězce v další generaci je 1.
- ▶ Zachování průměru tedy zajišťuje, že řetězce s průměrnou kvalitou si udržují svou důležitost a nejsou neúměrně zvýhodněny ani potlačeny.

Z rovník

$$f_{\text{avg}} = a + f_{\text{avg}} \cdot b$$

$$C_{\text{mult}} \cdot f_{\text{avg}} = a + f_{\text{max}} \cdot b$$

dostaneme

$$a = f_{\text{avg}} \cdot \frac{f_{\text{max}} - C_{\text{mult}} \cdot f_{\text{avg}}}{f_{\text{max}} - f_{\text{avg}}}$$

$$b = \frac{(C_{\text{mult}} - 1) \cdot f_{\text{avg}}}{f_{\text{max}} - f_{\text{avg}}}$$

V našem příkladu:

Nechť je generace ve stádiu, kdy se všechny řetězce shodují se schématem \*100\*\*\*\*.

| řetězec  | kvalita | úsek na ruletě (procenta) |
|----------|---------|---------------------------|
| 01001000 | 9.852   | 12.401                    |
| 01001111 | 9.932   | 12.501                    |
| 01000011 | 9.996   | 12.582                    |
| 11000111 | 9.991   | 12.563                    |
| 11000101 | 9.974   | 12.554                    |
| 11001010 | 9.904   | 12.466                    |
| 11001000 | 9.852   | 12.401                    |
| 11001100 | 9.956   | 12.532                    |

Koeficienty pro škálování:

$$f_{\text{avg}} = 9.931 \quad f_{\text{max}} = 9.996 \quad a = -292.94 \quad b = 30.5 \quad C_{\text{mult}} = 1.2$$

| řetězec  | kvalita | úsek na ruletě (procenta) |
|----------|---------|---------------------------|
| 01001000 | 7.525   | 9.472                     |
| 01001111 | 9.965   | 12.543                    |
| 01000011 | 11.917  | 15.000                    |
| 11000111 | 11.460  | 14.425                    |
| 11000101 | 11.246  | 14.155                    |
| 11001010 | 9.111   | 11.468                    |
| 11001000 | 7.525   | 9.472                     |
| 11001100 | 10.697  | 13.465                    |

Může se stát, že po škálování dostaneme záporné kvality některých řetězců.

### **Možnosti:**

- ▶ Neškálovat.
- ▶ Určit  $a$ ,  $b$  tak, aby nejmenší kvalita byla 0.

# Teorie genetických algoritmů

**Hlavní otázka:** Jak genetický algoritmus dokáže najít dobrá řešení problému bez přímé znalosti struktury zkoumaného problému?

Genetický algoritmus pracuje se **schématy**, která popisují podmnožiny řešení, a zachovává kvalitní schémata prostřednictvím selekce, křížení a mutací.

# Schémata a určené pozice

**Vliv určených pozic:** Čím více má schéma **určených pozic**, tím méně řetězců se s ním může shodovat.

Počet řetězců odpovídajících schématu závisí také na **rozložení určených pozic**.

## Příklad:

- ▶ Pokud křížíme řetězec odpovídající schématu 1\*\*\*\*\*1 s jiným řetězcem, který:
  - ▶ Neshoduje se ani s 1\*\*\*\*\*, ani s \*\*\*\*\*1,
  - ▶ **Výsledek:** Žádný z potomků se neshoduje se schématem 1\*\*\*\*\*1.
- ▶ Pokud křížíme řetězec odpovídající schématu \*\*\*\*\*11 s jiným řetězcem, který:
  - ▶ Neshoduje se ani s \*\*\*\*\*1, ani s \*\*\*\*\*1\*,
  - ▶ **Výsledek:** Pravděpodobnost shody potomka se schématem \*\*\*\*\*11 je  $\frac{5}{6}$ .

# Délka a řád schématu

## Definice:

- ▶ **Délka schématu** ( $\delta(H)$ ): Vzdálenost mezi **první** a **poslední** určenou pozicí schématu.
- ▶ **Řád schématu** ( $o(H)$ ): Počet **určených pozic** (pozic obsahujících 0 nebo 1) ve schématu.

## Příklad:

- ▶ Mějme schéma  $H = 01**1***$ .
- ▶ První určená pozice: 1 (obsahuje 0).
- ▶ Poslední určená pozice: 5 (obsahuje 1).
- ▶ Počet určených pozic: 3 (0, 1, 1).

## Výpočty:

$$\delta(H) = 5 - 1 = 4, \quad o(H) = 3$$

# Přežití schématu v generaci

**Otázka:** Jak se bude v generaci měnit počet řetězců odpovídajících schématu  $H$  během jednoho kroku genetického algoritmu?

**Označení:**

- ▶  $m(H, t)$ : Počet řetězců odpovídajících schématu  $H$  v generaci po  $t$ -tém kroku.
- ▶  $A(t)$ : Populace po  $t$ -tém kroku genetického algoritmu.

**Cíl:** Zkoumat očekávaný počet řetězců  $m(H, t + 1)$ , které odpovídají schématu  $H$ , v generaci  $(t + 1)$ .

**Hlavní vlivy:** Počet  $m(H, t + 1)$  závisí na:

- ▶ **Selekci:** Pravděpodobnost výběru řetězců odpovídajících  $H$ .
- ▶ **Křížení:** Pravděpodobnost, že řetězce odpovídající  $H$  zůstanou zachovány po křížení.
- ▶ **Mutaci:** Pravděpodobnost, že řetězec odpovídající  $H$  nebude mutací zničen.

## Reprodukce a očekávaný počet řetězců

**Pravděpodobnost výběru řetězce:** Pravděpodobnost výběru  $i$ -tého řetězce při reprodukci je dána vztahem:

$$p_i = \frac{f_i}{\sum f_i},$$

kde:

- ▶  $f_i$  je kvalita  $i$ -tého řetězce,
- ▶  $\sum f_i$  je součet kvalit všech řetězců v generaci.

**Očekávaný počet řetězců shodujících se se schématem  $H$ :**

Po reprodukci z generace  $A(t)$  je očekávaný počet řetězců odpovídajících schématu  $H$ :

$$m(H, t + 1) = m(H, t) \cdot n \cdot \frac{f(H)}{\sum f_j}$$

$$m(H, t + 1) = m(H, t) \cdot n \cdot \frac{f(H)}{\sum f_j},$$

kde:

- ▶  $m(H, t)$  je počet řetězců odpovídajících  $H$  v generaci  $t$ ,
- ▶  $n$  je celkový počet řetězců v generaci,
- ▶  $f(H)$  je průměrná kvalita řetězců odpovídajících schématu  $H$ ,
- ▶  $\sum f_j$  je součet kvalit všech řetězců v generaci.

**Význam:** Tento vzorec určuje, jak reprodukce ovlivňuje zachování kvalitních schémat mezi generacemi.

**Příklad:** Pokud máme:

- ▶  $m(H, t) = 10$ ,  $n = 100$ ,  $f(H) = 8$ ,  $\sum f_j = 500$ ,

pak očekávaný počet v generaci  $t + 1$  je:

$$m(H, t + 1) = 10 \cdot 100 \cdot \frac{8}{500} = 16.$$

# Změna počtu řetězců během reprodukce

**Průměrná kvalita řetězce:** Průměrnou kvalitu v generaci lze vypočítat jako:

$$\bar{f} = \frac{\sum f_j}{n},$$

kde  $\sum f_j$  je součet kvalit všech řetězců a  $n$  je velikost generace.

**Přepsaný vzorec:** Pomocí průměrné kvality lze přepsat vzorec pro očekávaný počet řetězců:

$$m(H, t + 1) = m(H, t) \cdot \frac{f(H)}{\bar{f}},$$

kde:

- ▶  $f(H)$  je průměrná kvalita schématu  $H$  v generaci,
- ▶  $\bar{f}$  je průměrná kvalita všech řetězců v generaci.

## Význam:

- ▶ Schémata s **nadprůměrnou kvalitou** ( $f(H) > \bar{f}$ ) budou mít v nové generaci **více** jedinců.
- ▶ Schémata s **podprůměrnou kvalitou** ( $f(H) < \bar{f}$ ) budou mít v nové generaci **méně** jedinců.
- ▶ Reprodukce postupně **odstraňuje schémata s nízkou kvalitou**.

## Příklad: Pokud:

- ▶  $m(H, t) = 10$ ,  $f(H) = 9.0$ ,  $\bar{f} = 8.0$ ,

pak:

$$m(H, t+1) = 10 \cdot \frac{9.0}{8.0} = 11.25 \quad (\text{více jedinců díky nadprůměrné kvalitě}).$$

## Růst a pokles počtu řetězců

**Otázka:** Jak rychle roste počet řetězců odpovídajících **silným schémátům** a klesá počet řetězců odpovídajících **slabým schémátům**?

**Předpoklad:** Mějme schéma  $H$ , jehož kvalita v generaci je:

$$f(H) = \bar{f} + \bar{f} \cdot c,$$

kde:

- ▶  $\bar{f}$  je průměrná kvalita v generaci,
- ▶  $c$  je konstanta, která vyjadřuje relativní odchylku kvality schématu od průměru ( $c > 0$  pro silná schémata,  $c < 0$  pro slabá schémata).

Za tohoto předpokladu lze očekávaný počet řetězců přepsat jako:

$$m(H, t + 1) = m(H, t) \cdot \frac{\bar{f} + \bar{f} \cdot c}{\bar{f}} = (1 + c) \cdot m(H, t).$$

**Obecný vztah:** Pokud začneme s  $m(H, 0)$  řetězcí v čase  $t = 0$ , dostáváme:

$$m(H, t) = m(H, 0) \cdot (1 + c)^t,$$

kde:

- ▶  $c > 0$  znamená exponenciální růst (silná schémata),
- ▶  $c < 0$  znamená exponenciální pokles (slabá schémata).

# Růst řetězců shodujících se se schématem

**Geometrická řada:** Očekávaný počet řetězců shodujících se se schématem  $H$  po  $t$  krocích:

$$m(H, t) = m(H, 0) \cdot (1 + c)^t,$$

je **geometrická řada** s kvocientem  $(1 + c)$ . Při  $c > 0$  pozorujeme **exponenciální růst**.

## Omezení růstu:

- ▶ Růst počtu řetězců odpovídajících schématu nemůže být neomezený, protože kapacita generace je fixní.
- ▶ Pokud schéma zaplní většinu generace, jeho **průměrná kvalita** se začne blížit průměrné kvalitě celé generace.
- ▶ Jakmile  $f(H) \approx \bar{f}$ , koeficient  $c$  klesá k nule, což zastaví exponenciální růst.



# Křížení a jeho vliv na schémata

**Vliv křížení:** Křížení má **destruktivní vliv** na schémata, protože může narušit určené pozice, které schéma definují.

Pokud křížíme:

- ▶ řetězec  $A$ , který odpovídá schématu  $H$ ,
- ▶ řetězec  $B$ , který neodpovídá schématu  $H$ ,

pak může nastat situace, kdy žádný z potomků neodpovídá schématu  $H$ .

**Podmínky, kdy se schéma  $H$  nezachová:**

- ▶ Křížení probíhá v místě, které má na **obou stranách určené pozice** schématu  $H$ .
- ▶ Části řetězce  $B$  (na levé i pravé straně od místa křížení) se neshodují s určenými pozicemi schématu  $H$ .

**Ilustrace:** Schéma  $H = 1***0***1$ :

- ▶ Řetězec  $A = 10100111$  odpovídá schématu  $H$ .
- ▶ Řetězec  $B = 01111010$  neodpovídá schématu  $H$ .
- ▶ Křížení vytvoří potomky, kteří neodpovídají  $H$ .

# Náchylnost schémat ke ztrátě při křížení

**Vliv délky schématu:** Některá schémata jsou **více náchylná ke ztrátě** při křížení, zejména pokud mají **větší délku**.

## Příklad:

- Mějme schémata:

$$H_1 = *1***0**, \quad H_2 = ***01***,$$

a řetězec  $A = 11001010$ , který odpovídá oběma schématům.

- Řád obou schémat je stejný:

$$o(H_1) = o(H_2) = 2.$$

## Křížení:

- Schéma  $H_2$ : K jeho ztrátě může dojít pouze při křížení mezi čtvrtou a pátou pozicí.
- Schéma  $H_1$ : K jeho ztrátě může dojít při křížení na **čtyřech různých místech**.

**Důsledek:** Schéma  $H_1$  je náchylnější ke ztrátě, protože jeho **velká délka** ( $\delta(H_1) = 7$ ) zvyšuje pravděpodobnost narušení určených pozic.

**Závěr:** Délka schématu ( $\delta(H)$ ) má zásadní vliv na jeho pravděpodobnost přežití při křížení.

# Pravděpodobnost ztráty schématu při křížení

**Definice:** Máme řetězec  $A$ , který odpovídá schématu  $H$ . Zkoumáme pravděpodobnost  $p_d$ , že po křížení  $A$  s jiným řetězcem nevznikne žádný potomek, který by odpovídal schématu  $H$ .

**Vzorec:**

$$p_d \leq \frac{\delta(H)}{\ell - 1},$$

kde:

- ▶  $\delta(H)$  je délka schématu (vzdálenost mezi první a poslední určenou pozicí),
- ▶  $\ell$  je délka řetězce.

Pravděpodobnost, že aspoň jeden potomek odpovídá schématu  $H$ , je:

$$1 - p_d.$$

# Pravděpodobnost přežití schématu

## Pravděpodobnost přežití:

Uvažujme, že ke křížení dochází s pravděpodobností  $p_c$ .  
Pravděpodobnost, že schéma  $H$  přežije v aspoň jednom z potomků, je:

$$p_s \geq 1 - p_c \cdot \frac{\delta(H)}{\ell - 1}.$$

## Interpretace:

- ▶ Čím menší je  $\delta(H)$  (délka schématu), tím vyšší je pravděpodobnost přežití schématu.
- ▶ Pravděpodobnost přežití klesá s rostoucí pravděpodobností křížení  $p_c$ .

# Odhad počtu řetězců v generaci

**Očekávaný počet řetězců:** Počet řetězců odpovídajících schématu  $H$  v generaci  $A_c(t + 1)$  lze odhadnout:

$$m(H, t + 1) \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \cdot \left[ 1 - p_c \cdot \frac{\delta(H)}{\ell - 1} \right],$$

kde:

- ▶  $\frac{f(H)}{\bar{f}}$ : Exponenciální růst díky nadprůměrné kvalitě schématu.
- ▶  $1 - p_c \cdot \frac{\delta(H)}{\ell - 1}$ : Pravděpodobnost přežití schématu při křížení.

# Závěr: Schémata s nadprůměrnou kvalitou a malou délkou

## Takže:

- ▶ Schémata s nadprůměrnou kvalitou mají díky reprodukci **exponenciální růst**.
- ▶ Pravděpodobnost přežití schématu při křížení klesá s rostoucí délkou  $\delta(H)$ .
- ▶ Kombinace nadprůměrné kvality a relativně malé délky zajišťuje úspěšnost schématu v generaci.

Schémata s **nadprůměrnou kvalitou** a **relativně malou délkou** jsou nejúspěšnější při reprodukci a křížení.

# Mutace a její vliv na schémata

## Princip:

- ▶ Každá pozice v řetězci je změněna mutací s pravděpodobností  $p_m$ .
- ▶ Pravděpodobnost, že ke změně nedojde, je  $1 - p_m$ .

## Vliv na schéma:

- ▶ Mějme řetězec  $A$ , který odpovídá schématu  $H$ .
- ▶ Počet určených pozic schématu  $H$  je  $o(H)$ .
- ▶ Pokud dojde ke změně v některé z těchto  $o(H)$  pozic, řetězec  $A$  již nebude odpovídat schématu  $H$ .

**Pravděpodobnost přežití schématu:** Pravděpodobnost, že řetězec  $A$  bude po mutaci stále odpovídat schématu  $H$ , je:

$$p_s = (1 - p_m)^{o(H)},$$

kde:

- ▶  $1 - p_m$ : Pravděpodobnost, že jedna pozice nebude změněna.
- ▶  $o(H)$ : Počet určených pozic schématu.

# Vliv mutace na nerovnost přežití schématu

**Aproximace pro malé hodnoty  $p_m$ :** Pro velmi malé hodnoty pravděpodobnosti mutace  $p_m$  lze výraz:

$$(1 - p_m)^{o(H)} \approx 1 - o(H) \cdot p_m.$$

**Upravená nerovnost:** Přidáním vlivu mutace se nerovnost pro očekávaný počet řetězců mění na:

$$m(H, t + 1) \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \cdot \left[ 1 - p_c \cdot \frac{\delta(H)}{\ell - 1} \right] \cdot [1 - o(H) \cdot p_m].$$

$$m(H, t + 1) \geq m(H, t) \cdot \frac{f(H)}{\bar{f}} \cdot \left[ 1 - p_c \cdot \frac{\delta(H)}{\ell - 1} \right] \cdot [1 - o(H) \cdot p_m].$$

Kde:

- ▶  $\frac{f(H)}{\bar{f}}$ : Exponenciální růst díky nadprůměrné kvalitě schématu.
- ▶  $1 - p_c \cdot \frac{\delta(H)}{\ell - 1}$ : Pravděpodobnost přežití schématu při křížení.
- ▶  $1 - o(H) \cdot p_m$ : Pravděpodobnost přežití schématu při mutaci.

Nadprůměrná schémata s:

- ▶ **malou délkou**  $\delta(H)$  (nižší riziko ztráty při křížení),
- ▶ **malým řádem**  $o(H)$  (nižší riziko ztráty při mutaci),

mají největší šanci na přežití a exponenciální růst v populaci.

# Věta o schématech

**Závěr:** Z předchozích úvah vyplývá následující důležité tvrzení:

## Theorem (Věta o schématech)

*Krátká, nadprůměrná schémata s nízkým řádem získávají exponenciální nárůst počtu řetězců v následujících generacích.*

### Klíčové pojmy:

- ▶ **Krátká:** Schémata s malou délkou  $\delta(H)$  (méně náchylná ke ztrátě při křížení).
- ▶ **Nadprůměrná:** Schémata s vysokou relativní kvalitou  $\frac{f(H)}{\bar{f}}$ .
- ▶ **Nízký řád:** Schémata s malým počtem určených pozic  $o(H)$  (nižší riziko ztráty při mutaci).

**Stavební bloky:** Taková schémata se nazývají **stavební bloky** (**building blocks**) a tvoří základ pro vytváření kvalitnějších řešení v genetických algoritmech.

**Příklad stavebního bloku:** Schéma  $H = *1*0*$  je krátké ( $\delta(H) = 4$ ), má nízký řád ( $o(H) = 2$ ), a pokud je nadprůměrné, bude se exponenciálně šířit v populaci.

## Problém dvou-rukého a $k$ -rukého bandity

**Problém:** Stanovili jsme tvrzení o nárůstu krátkých nadprůměrných schémat, ale zatím jsme nevysvětlili, **proč je tento exponenciální nárůst užitečný.**

**Relevance:** Odpověď na tuto otázku nám poskytne analogie s problémem **dvou-rukého a  $k$ -rukého bandity.**

**Zadání problému dvou-rukého bandity:**

- ▶ Máme stroj se dvěma pákami.
- ▶ Po zatažení za páku získáme průměrnou výhru:

$$m_1 \quad (\text{první páka}), \quad m_2 \quad (\text{druhá páka}).$$

- ▶ Hodnoty  $m_1$  a  $m_2$  jsou neznámé.
- ▶ Máme  $N$  pokusů.

**Cíl:** Navrhnout strategii tahání za páky, která maximalizuje celkovou výhru.

**Výzva:** Musíme **vyvážit průzkum (exploration)** nových možností a **exploataci (exploitation)** páky, která se zdá být výnosnější.

# Strategie průzkumu a exploatace

**Předpoklad:** Pokud  $m_1 > m_2$ , tedy průměrný zisk z první páky je větší než z druhé, je pro nás výhodnější tahat za první páku.

**Výzva:** Na začátku hry nevíme, která páka je lepší. Musíme zvolit strategii, která vyváží:

- ▶ **Průzkum (exploration):** Vyzkoušet obě páky, abychom zjistili jejich průměrné zisky.
- ▶ **Exploataci (exploitation):** Soustředit se na lepší páku pro maximalizaci zisku.

**Příklad strategie:** Rozdělíme naše pokusy:

- ▶ Prvních  $2 \cdot \ell$  pokusů použijeme pro průzkum:  $\ell$  pokusů na každé páce.
- ▶ Zbývajících  $N - 2 \cdot \ell$  pokusů použijeme na páku, která dosud dala vyšší průměrnou výhru.

**Klíčová otázka:** Jak zvolit hodnotu  $\ell$ ?

- ▶ **Menší  $\ell$ :** Rychle přejdeme k exploataci, ale riskujeme, že určíme lepší páku nesprávně.
- ▶ **Větší  $\ell$ :** Získáme přesnější informace o páce, ale více pokusů s jistotou věnujeme horší páce.

**Cíl:** Najít optimální hodnotu  $\ell$ , která vyváží průzkum a exploataci, aby maximalizovala celkový zisk.

# Teoretické výsledky pro rozdělení pokusů

## Rozdělení pokusů:

- ▶  $n^*$ : Počet pokusů na horší páce.
- ▶  $N - n^*$ : Počet pokusů na lepší páce.

**Vzorec:** Optimální počet pokusů na lepší páce lze přibližně vyjádřit:

$$N - n^* \approx \sqrt{8\pi b^4 \ln(N^2)} \cdot e^{\frac{n^*}{2b^2}},$$

kde:

- ▶  $b$ : Parametr související s rozptylem výher páky.
- ▶  $\ln(N^2)$ : Závislost na celkovém počtu pokusů  $N$ .

## Zjednodušená forma:

$$N - n^* \approx a \cdot e^{c \cdot n^*},$$

kde:

- ▶  $a = \sqrt{8\pi b^4 \ln(N^2)}$  (konstanta závislá na  $b$  a  $N$ ),
- ▶  $c = \frac{1}{2b^2}$  (rychlost růstu v závislosti na  $n^*$ ).

# Genetický algoritmus jako mnoho $k$ -rukých banditů

**Pozorování:** Genetický algoritmus lze interpretovat jako řešení mnoha problémů  $k$ -rukého bandity, kde každé schéma odpovídá jedné „páče“.

## Zadání $k$ -rukého bandity:

- ▶ Máme stroj s  $k$  pákami.
- ▶ Po zatažení za  $i$ -tou páku obdržíme průměrnou výhru  $m_i$ .
- ▶ Hodnoty  $m_i$  jsou neznámé.
- ▶ Máme  $n$  pokusů.

**Cíl:** Maximalizovat celkovou výhru taháním za páky.

## Relevance pro genetické algoritmy:

- ▶ Každé **schéma** odpovídá jedné „páče“.
- ▶ Průměrná výhra  $m_i$  odpovídá **kvalitě schématu**.
- ▶ Počet tahů za páku odpovídá **počtu řetězců odpovídajících schématu**.

**Exponenciální nárůst:** I u  $k$ -rukého bandity je žádoucí dosáhnout **exponenciálního nárůstu** pokusů u nejlepších pák, protože to odpovídá šíření kvalitních schémat v genetickém algoritmu.