

Základy umělé inteligence

4. Evoluční výpočetní techniky

Jiří Kubalík
Katedra kybernetiky, ČVUT-FEL



<http://cw.felk.cvut.cz/doku.php/courses/y33zui/start>

Přírodní motivace EVT

:: Stochastické optimalizační algoritmy

- pracují na principu "vyšlechtení" kvalitního řešení postupnou evolucí populace potenciálních řešení založené na:
 - genetické dědičnosti (J.G. Mendel) a
 - zápasu o prežití, kde vítězí ti nejsilnější (Ch. Darwin), nejlépe adaptovaní v daném prostředí.

" V přírodní evoluci je základní úlohou biologického druhu vyhledávání výhodných adaptací vůči složitému a dynamicky se měnícímu prostředí. Znalost, která charakterizuje každý biologický druh, byla získána vývojem a je shrnuta v chromozómech každého jedince."

:: Názvosloví – gen, chromozóm, jedinec, populace, generace, křížení, mutace, ...

Historie EVT

- Fraser, Bremermann, Reed, (50-tá, 60-tá léta) – první pionýři
- L. Fogel, 1962 (San D., CA): Evolutionary Programming
- Rechenberg & H.-P. Schwefel, 1965 (Berlin, Germany): Evolution Strategies
- J. Holland (americký biolog, pokus o algoritmické vysvětlení diverzity druhů v přírodě), 1975 (Ann Arbor, MI) - Genetic Algorithms
- D. E. Goldberg, 1989 – Genetic Algorithms in Search, Optimisation and Machine Learning
- J. Koza, 1989 (Palo Alto, CA) - Genetic Programming
- Gene expression programming, Grammatical evolution, competent GAs, Estimation of Distribution Algorithms ...

D. Fogel: "*Měli bychom mít radost z toho, že evoluční výpočty vyšly asi z deseti nezávislých počátků v období let od 1953 do 1968. Je to klasický příklad konvergující evoluce.*"

Materiály: Literatura, Dema, Software

:: Literatura

- D. E. Goldberg: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.
- Z. Michalewicz: Genetic Algorithms + Data Structures = Evolution Programs, Springer, 1998.
- Z. Michalewicz: How to solve it? Modern heuristics. 2nd ed. Springer, 2004.
- K. A. DeJong: Evolutionary Computation, MIT Press, 2002.

:: Dema

- M. Obitko: Introduction to genetic algorithms with java applets,
<http://www.obitko.com/tutorials/genetic-algorithms/>

Materiály: Literatura, Dema, Software (2)

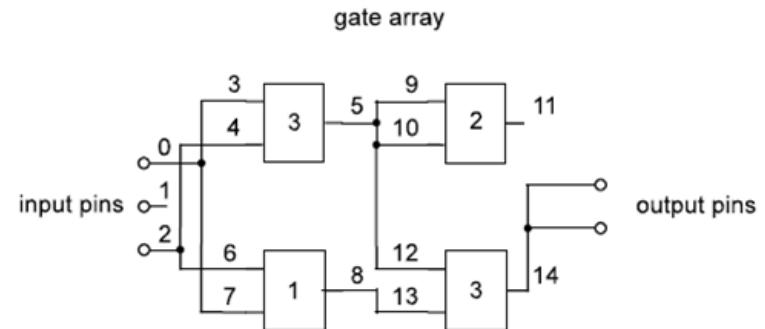
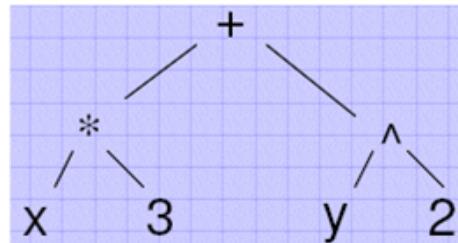
:: Software

- ECJ 16 – A Java-based Evolutionary Computation Research System
<http://cs.gmu.edu/eclab/projects/ecj/>
- Open BEAGLE 3.0.1 - An evolutionary computation framework in C++
<http://www.archivum.info/fm.announce/2005-10/msg01260.html>
- PISA – A Platform and Programming Language Independent Interface for Search Algorithms
<http://www.tik.ee.ethz.ch/sop/pisa/?page=selvar.php>
- GeneXproTools 4.0: Data mining software based on Gene Expression Programming
<http://www.gene-expression-programming.com/Downloads.asp>
- The Genetic Algorithm Utility Library (GAUL), an open source programming library written in C
<http://gaul.sourceforge.net/>

Reprezentace

:: Problém může být reprezentován jako

- binární řetězec – 1 0 1 1 0 1 1 0 0 1 0 1 1 0 1
- řetězec reálných hodnot – 3,24 1,78 -2,61
- řetězec znaků – D→E→A→C→B
- nebo jako **strom** nebo jako **graf**, ...



:: **Genotyp**

genetická informace o řešení



Fenotyp

konkrétní hodnoty parametrů řešení

ke každému genotypu musí být definován fenotyp

Ohodnocovací funkce

:: Fitness

- jediná informace o řešeném problému, kterou uživatel poskytne EA,
- musí být definována pro všechny možné chromozomy.

:: Fitness může být

- mnohorozměrná,
- nelineární,
- multimodální,
- vícekriteriální,
- diskrétní

:: Fitness nemusí být definována nalyticky

- výsledky simulace optimalizovaného systému,
- úspěšnost klasifikace.

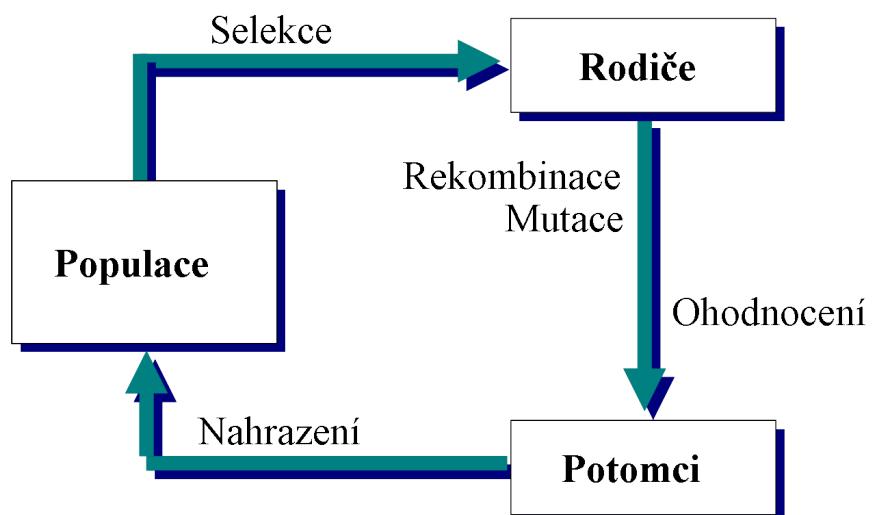
Příklad na kódování

:: Funkční optimalizace

- hledání maxima funkce $f(x, y) = x^2 + y^2$,
- na intervalu celých čísel $\langle 0, 31 \rangle$,
- x a y jsou kódovány na 5 bitech.

genotyp	fenotyp	fitness
0 0 0 0 0, 0 1 0 1 0	0, 10	100
0 0 0 0 1, 1 1 0 0 1	1, 25	625 + 1 = 626
0 1 0 1 1, 0 0 0 1 1	11, 3	121 + 9 = 130
1 1 0 1 1, 1 0 0 1 0	27, 18	729 + 324 = 1053

Evoluční cyklus



```
begin
    t:=0;
    Inicializace P(t);
    Ohodnocení P(t);
    Statistika P(t);
    while (not ukončovací podmínka) do
        begin
            t=t+1;
            Reprodukce P(t) z P(t-1);
            Rekombinace P(t);
            Ohodnocení P(t);
            Nahrazení P(t);
            Statistika P(t);
        end
    end
```

Inicializace počáteční populace

:: Náhodná

- náhodné navzorkování zvoleného počtu chromozomů (náhodný generátor nul a jedniček s p-stí 0,5),
- žádná apriorní znalost o podobě hledaného řešení,
- spoléhá pouze na "šťastné" pokrytí celého prohledávaného prostoru omezeným počtem vzorků.

:: Informovaná

- využívá apriorní znalost,
- může vést k nalezení lepších řešení,
- může zkrátit celkový výpočet,
- !!! může způsobit nevratné nasměrování GA k suboptimálnímu řešení.

:: Předzpracování jedinců počáteční populace

Selekce (reprodukce)

:: Modeluje přírodní princip "přežívání nejsilnějších"

- upřednostňuje zdatnější jedince před slabšími,
- každý jedinec má šanci přispět svým kódem do další generace.

:: Ruletové kolo

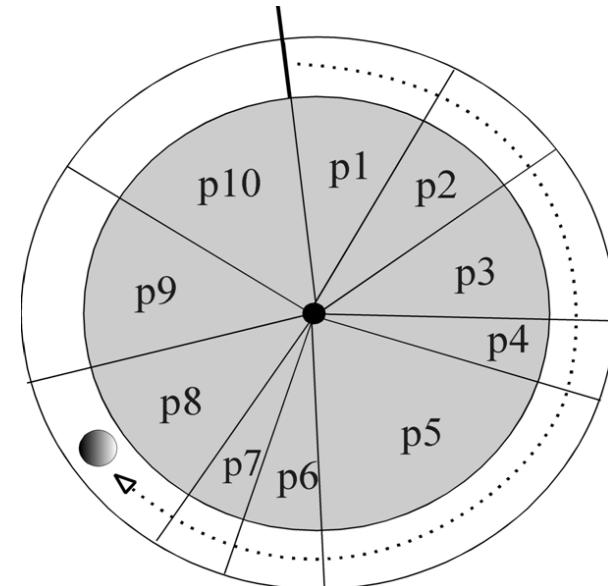
- pravděpodobnost výběru jedince je úměrná jeho fitness

$$P_i = f_i / \sum_{j=1}^{PopSize} f_j$$

- populární, ale nepřesná metoda.

:: Další metody

- stochastic universal sampling,
- reminder stochastic sampling,
- turnajová selekce, ...

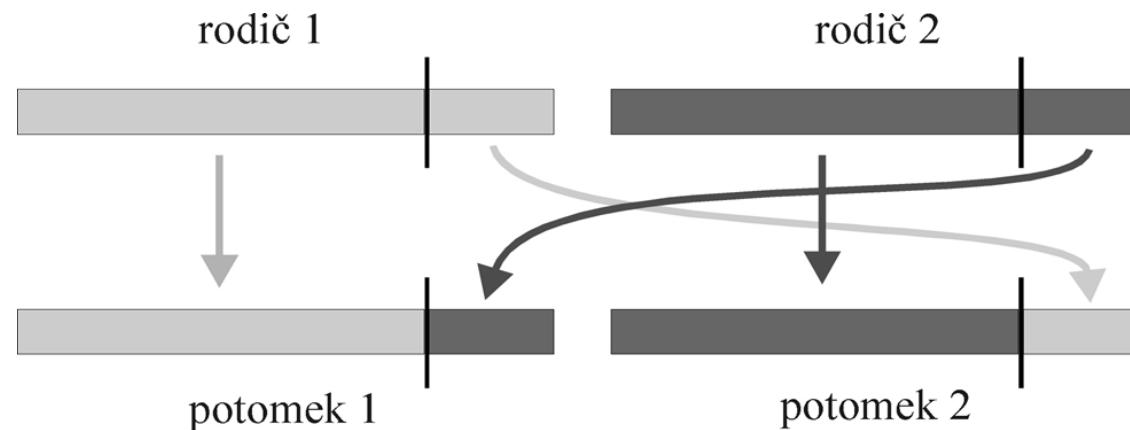


Genetické operátory

:: Křížení

- **myšlenka** – máme-li dvě dobrá řešení daného problému, pak jejich vhodným zkombinováním můžeme získat řešení, které bude ještě lepší.
- **úloha křížení** – vzorkování "exploration" prohledávaného prostoru.

Příklad: 1-bodové křížení (slepé křížení)



Genetické operátory 2

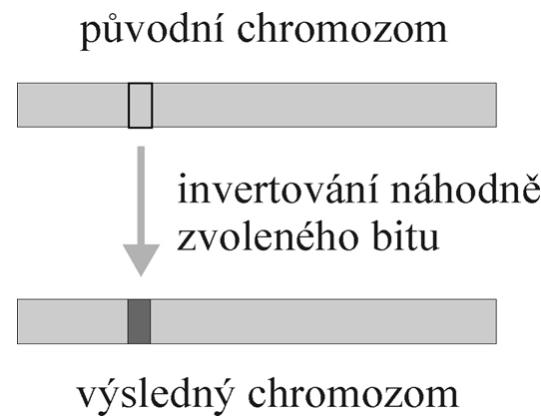
:: **Mutace** – částečná variace řešení

- udržení diverzity populace,
- minimalizace možnosti ztráty potenciálně užitečné části genetického kódu.

:: **Selekce + Mutace**

- tato kombinace může být dostatečně silná i bez křížení,
- evoluční strategie.

Příklad: jednoduchá mutace



EA a prohledávání s omezeními

:: Prohledávaný prostor s omezeními – obsahuje přípustná i nepřípustná řešení.

Příklad: **Problém obchodního cestujícího** (Traveling Salesman Problem)

parent1: 4 8 1 | 3 5 2 7 6 offspring1: 4 8 1 | 1 3 6 5 4



parent2: 2 7 8 | 1 3 6 5 4 offspring2: 2 7 8 | 3 5 2 7 6

Žádný z potomků offspring1 a offspring2 není přípustným řešením – některá města budou v cestě chybí anebo se vyskytují dvakrát.

Tři základní způsoby, jak ošetřit omezení v EA

:: **Penalizace**

- nepřípustné řešení je penalizováno, čímž se mu zhorší p-st výběru,
- funguje jen s jednoduchými omezeními.

:: **Dekodéry a opravné algoritmy**

- speciální procedury, které dekódují chromozóm na přípustné řešení, nebo "opraví" nepřípustné řešení na nejbližší přípustné,
- může být velice náročné.

:: **Problémově závislá reprezentace a speciální operátory**

- slibný, ale náročný postup, ad hoc reprezentace.

:: **Transformace optimalizace s omezeními na vícekriteriální optimalizaci**

- míra porušení omezení představuje další (minimalizované) kritérium.

TSP: Hranový operátor (Edge-Recombination Operator)

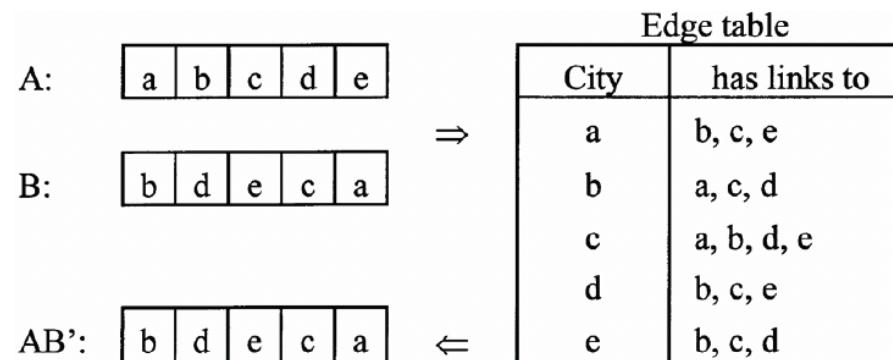
:: **Přímá reprezentace** (Path representation)

genotype: a e d b c
tour: a → e → d → b → c

:: Edge recombination crossover

1. Sestav tabulku sousedů – každé město v ní má seznam měst, se kterými sousedí v rodičovských cestách.
 2. Začni tvořit cestu z náhodně zvoleného města.
 3. Z aktuálního města jdi do některého z měst z jeho seznamu sousedů, které ještě není použito v cestě.

Když už takové město není, tak zvol náhodně některé ze zbývajících měst.



Nahrazovací strategie

:: Úloha nahrazovací strategie

- jak velká část populace bude nahrazena v každém generačním kroku,
- kteří jedinci budou v populaci nahrazeni.

:: Generační strategie

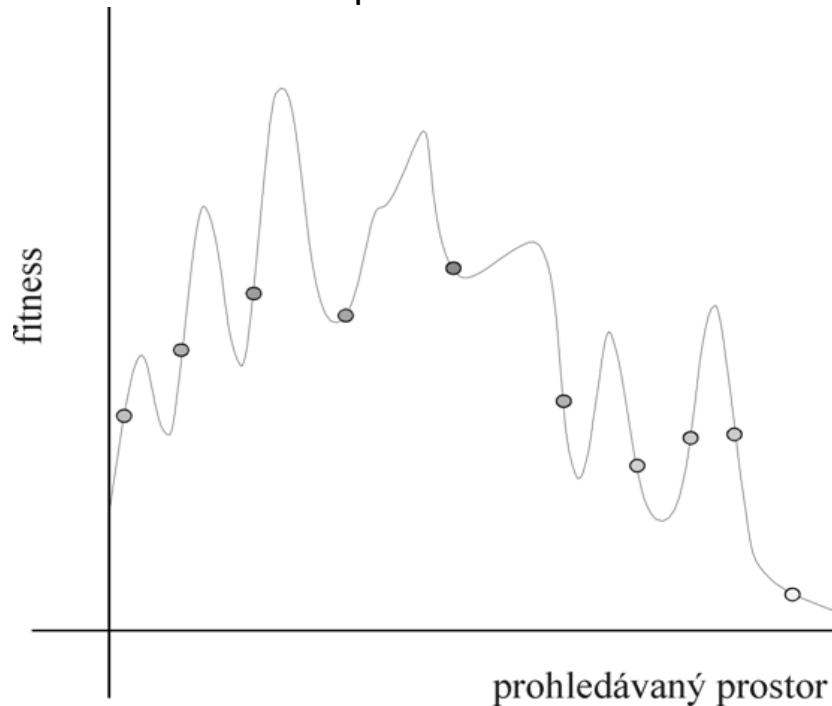
- stará populace je kompletně nahrazena novou populací,
- analogie se *short-lived species*.

:: Steady-state strategie

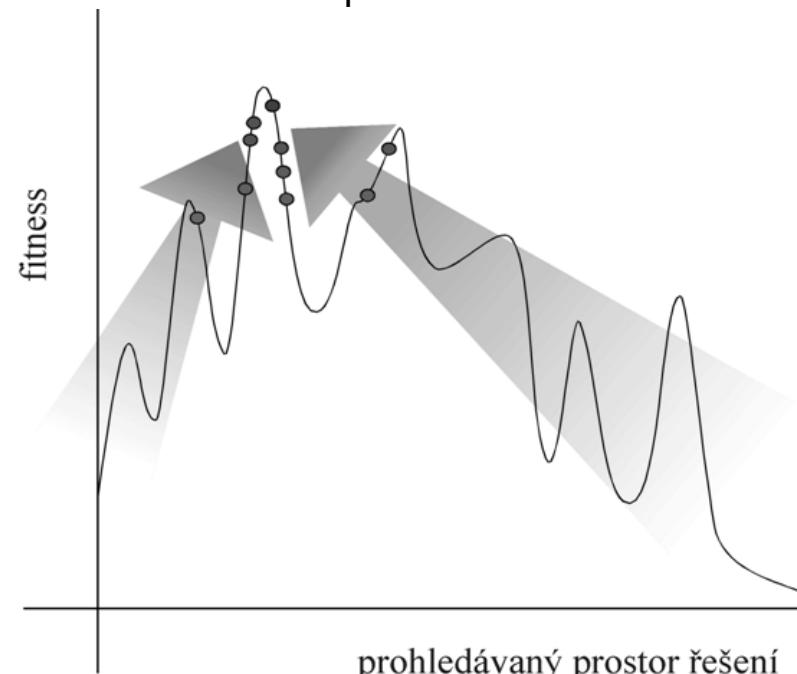
- pouze část populace je nahrazena, ostatní jedinci zůstávají,
- analogie s *long-lived species*.

Idealizovaný příklad vývoje populace

rovnoramné navzorkování prohledávaného prostoru



zaostření na slibné oblasti prohledávaného prostoru



Oblasti nasazení EA

:: EA jsou populární pro jejich

- snadnou implementaci,
- efektivnost,
- robustnost.

:: **J. Holland:** " Nejlepší uplatnění GA je v oblastech, kde nemáme žádnou představu o tom, jak by mělo řešení vypadat. Právě tam nás často překvapí s čím přijdou."

:: Aplikace

- | | |
|---------------------------|--------------------------|
| - řízení | - návrh neuronových sítí |
| - inženýrský návrh | - zpracování obrazu |
| - plánování a rozvrhování | - bankovnictví |
| - alokace zdrojů | - predikce časových řad |
| - layout planning | a další ... |

Vyhledávání podezřelých

:: Policejní databáze "hříšníků"



:: **Cílem** je vytvořit podpůrný systém pro vyhledání podezřelých v obrovské databázi fotek.

:: Struktura chromozómu

skull_shape	hair_cut	spectacles	beard
0 0 1 0 1	0 1 1 0 1	0 1 1 0 0	1 0 0 0 1
hair_colour	eye_colour		
0 1 0 1 0	...	1 1 0 1 0	

:: **Svědek slouží jako "vyhodnocovač" kvality potenciálních řešení.**

? UK Home Office, Police Systems Research and Development Group ?

Genetické programování (GP)

- :: **GP** je založeno na stejných postupech jako GA – simulovaný princip přežití těch nejsilnějších jedinců a analogie genetických operací.
- :: **GP se liší od GA** reprezentací, genetickými operátory a oborem vhodných aplikačních oblastí.
- :: **GP jako rozšíření konvenčních GA** – struktury, které jsou vyvíjeny, jsou stromy proměnné velikosti a tvaru, reprezentující programy.

:: **Applikace**

- konstrukce programů – spíše podprogramů,
- učení klasifikátorů – rozhodovací stromy,
- učení rozhodovacích a klasifikačních pravidel,
- modelování – symbolická regrese,
- učení strategií,
- ...

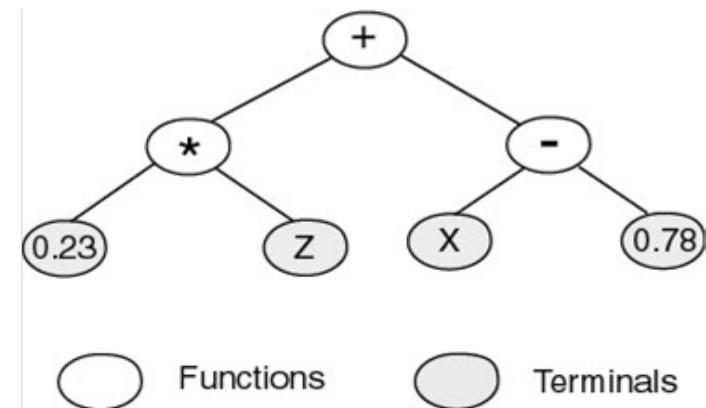
GP: Reprezentace

:: Stromy se skládají z funkcí (vnitřní uzly) a terminálů (listové uzly) vhodně navržených pro daný problém.

- **Terminály** – vstupy vyvíjených programů
(nezávislé proměnné), reálné, celočíselné nebo **Příklad:** Strom reprezentující LISP-ovský logické konstanty, výkonné akce.
S-výraz $0.23 * Z + X - 0.78$

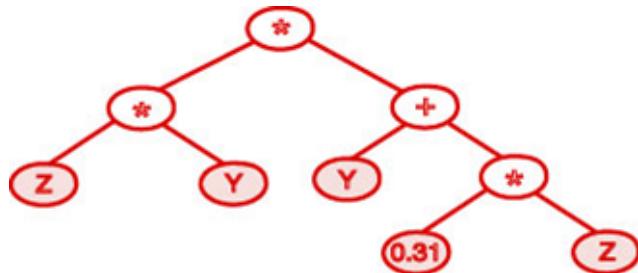
■ Funkce

- aritmetické operace (+, -, *, /),
 - algebraické funkce (sin, cos, exp, log),
 - logické funkce (AND, OR, NOT),
 - podmíněné op. (If-Then-Else, cond?true:false),
 - a jiné.

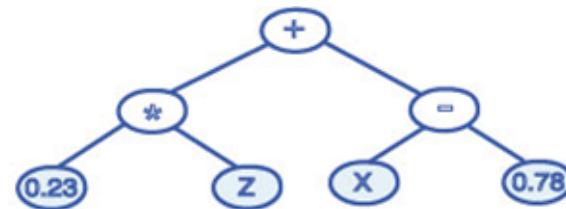


:: **Podmínka uzavřenosti** – každá funkce musí být schopna akceptovat jako svůj vstupní argument výstupní hodnotu libovolné funkce nebo terminálu.

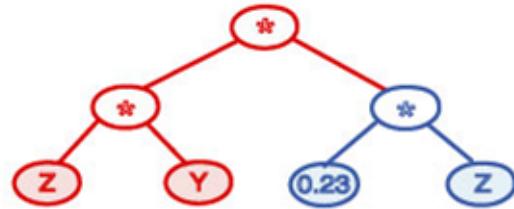
GP: Křížení



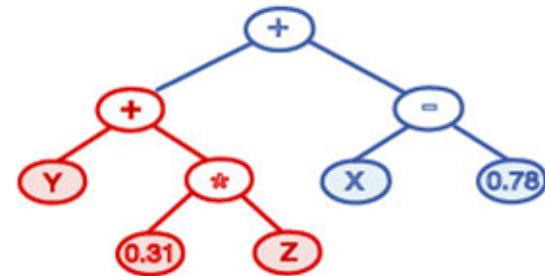
Parent 1: $Z * Y * (Y + 0.31 * Z)$



Parent 2: $0.23 * Z + X - 0.78$



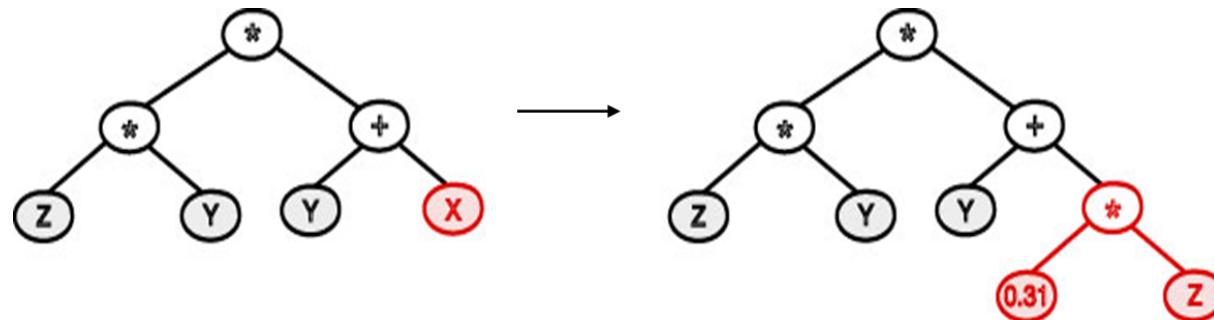
Child 1: $0.23 * Y * Z^2$



Child 2: $Y + 0.31 * Z + X - 0.78$

GP: Mutace a jiné operátory

:: **Mutace** nahrazuje náhodně zvolený uzel a jeho podstrom jiným náhodně vygenerovaným podstromem.



:: Other operators::

- permutace,
 - editace,
 - zapouzdření,
 - decimace,

GP: Trigonometrická identita

:: **Cílem** je nalézt ekvivalentní výraz ke $\cos(2x)$.

:: **GP implementace:**

- **Množina terminálů** $T = \{x, 1.0\}$.
- **Množina funkcí** $F = \{+, -, *, \%, \sin\}$.
- **Trénovací množina:** 20 párů (x_i, y_i) , kde x_i jsou hodnoty rovnoměrně navzorkované v intervalu $(0, 2\pi)$.
- **Fitness:** Suma absolutních odchylek požadovaných hodnot y_i od hodnot vrácených generovaným výrazem.
- **Ukončovací kritérium:** Řešení s chybou menší než 0.01.

GP: Trigonometrická identita (2)

:: 1. výpočet, 13. generace

$$(-(-1(*(\sin x)(\sin x))))(*(\sin x)(\sin x)))$$

což se po úpravách rovná $1 - 2 * \sin^2 x$

:: 2. výpočet, 34. generace

$$(-1(*(\sin x)(\sin x))2))$$

což je pouze jiný zápis předchozího výrazu.

:: 3. výpočet, 30. generace

$$\begin{aligned} & (\sin (-(-2(*x2))) \\ & (\sin (\sin (\sin (\sin (\sin (\sin (*(\sin (\sin 1)) \\ & (\sin 1)) \\))))))))) \end{aligned}$$

Pozn.: Výraz na druhém a třetím řádku se poměrně přesně blíží hodnotě $\pi/2$, takže objevená identita je

$$\cos(2x) = \sin(\pi/2 - 2x).$$

