Umělá inteligence
aneb co už není sci-fi

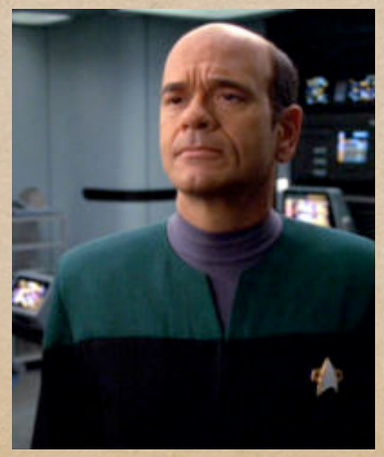
doc. Ing. Zuzana Komínková Oplatková, Ph.D.

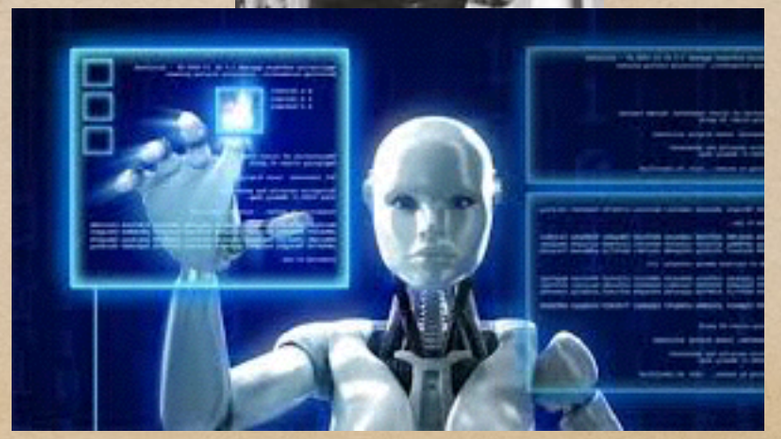
oplatkova@fai.utb.cz

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

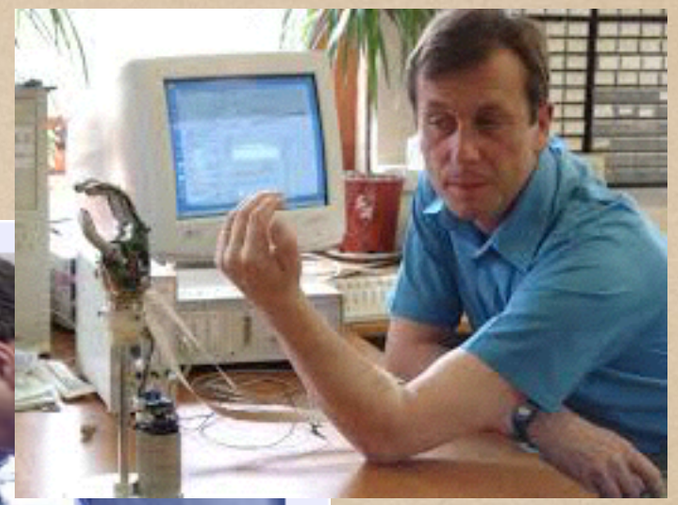
Umělá inteligence

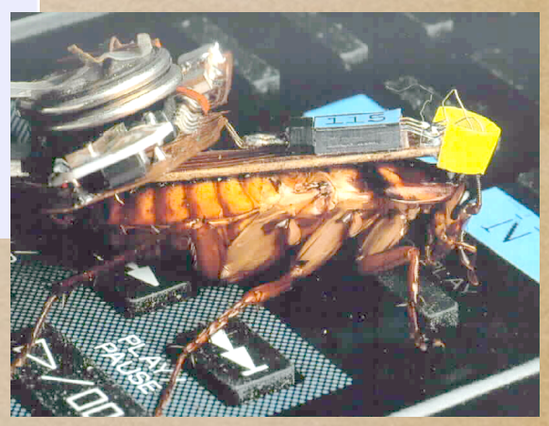
- člověk se snažil vždy vyrobit nějaký stroj nebo systém, který by mu usnadnil práci a případně byl i “inteligentní”
- snahy vytvořit takové systémy se nejčastěji objevovaly nejprve ve filmech, sci-fi, pak i v realitě



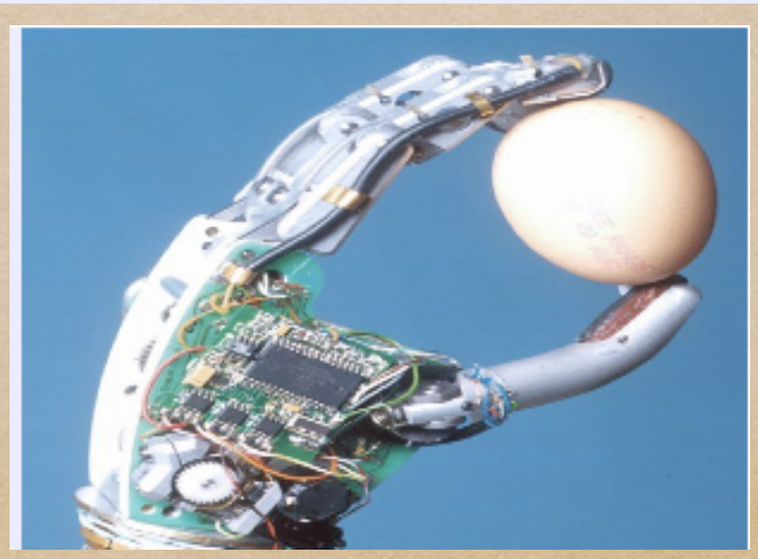


První kyborg





- prof. Kevin Warwick
- voperoval si RFID chip (1998) a 100 elektrodové pole (2002)



Kevin Warvick

- pohybovat se po budově - implantovaný silikonový chip (24. srpna 1998) umožňuje zapínat / zhasínat světla, ovládat dveře, topení
- druhý projekt - 100elektrodové pole - 14. března 2002
  - umožňuje posílat a přijímat signály mezi implantátem, nervovou soustavou a počítačem - ovládnutí ruky, invalidního vozíku
  - pokud úspěšné (bylo), i jeho žena dostane(la) chip a bude zkoumáno, jak lze posílat signály emocí z jedné osob do druhé a případně reagovat
- https://www.youtube.com/watch?v=Z8HeFNJjuj0
- https://www.youtube.com/watch?v=cx45D9aWEeY

HABIT - Visteon

* Human Bayesian Intelligence Technology - HABIT - společnost Visteon
    * předvídání situací a nastavení auta (klimatizace, rádio, muzika, jízdní systémy vozu), jak by je řidič mohl potřebovat (hlídá denní dobu, teploty, okolní vlivy, atd.. a “učí se”.

Google car

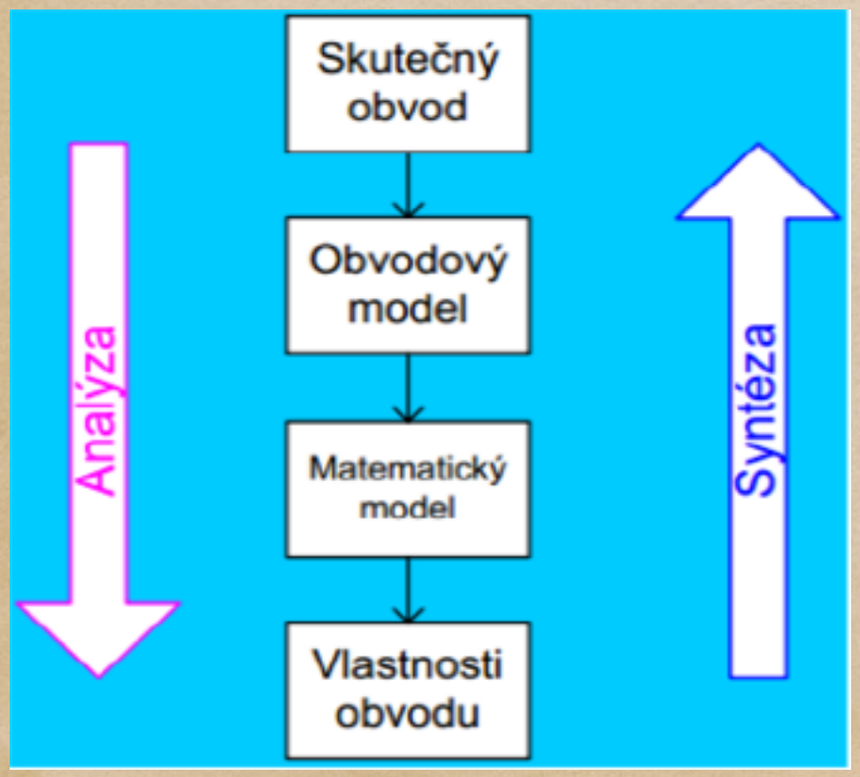
- autonomní systémy - auto nepotřebuje řidiče (např. Google car)
- orientuje se pomocí počítačových systémů - detekce okolí
    - radar
    - GPS
    - lidar (metoda dálkového měření vzdálenosti na základě výpočtu doby šíření pulsu laserového paprsku odraženého od snímaného objektu - vytvoření množství bodů -> interpolovat do podoby 3D digitálního modelu povrchu, budov, atd.)
    - objekty je třeba klasifikovat…metody pro počítačové vidění - image processing, neuronové sítě…

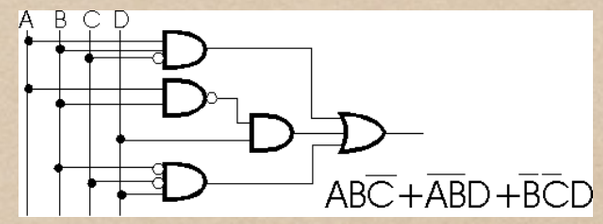
Výpočetní inteligence

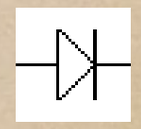
* součást umělé inteligence
* metody soft computingu
    * fuzzy množiny
* umělé neuronové sítě
* evoluční výpočetní techniky
* bayesovské sítě - založené na pravděpodobnostech
* …

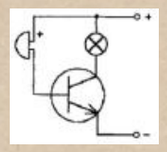
Syntéza elektronických (logických) obvodů

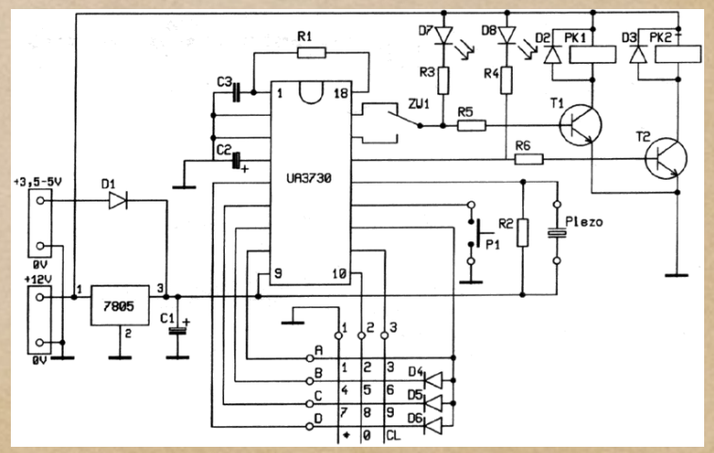
* známe vlastnosti obvodu - pravdivostní tabulku chování
* máme k dispozici prvky obvodu - AND, NAND, OR, NOR…
* chceme syntetizovat model obvodu











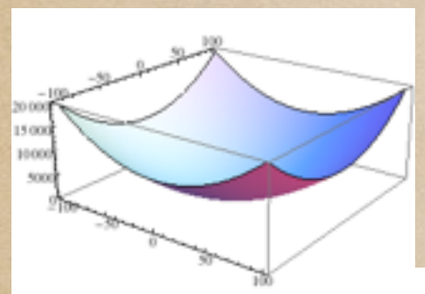
Evoluční výpočetní techniky

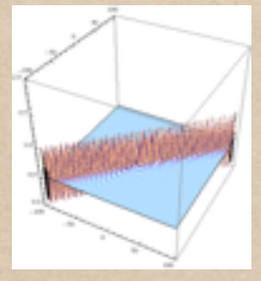
* patří do oblasti výpočetní inteligence a soft computingu
* slouží především k optimalizaci (evoluční algoritmy - genetické algoritmy, diferenciální evoluce, rojení částic (PSO), SOMA…)
* k syntéze analytických řešení (nadstavba evoluční algoritmů - Genetické programování, gramatická evoluce, analytické programování…)

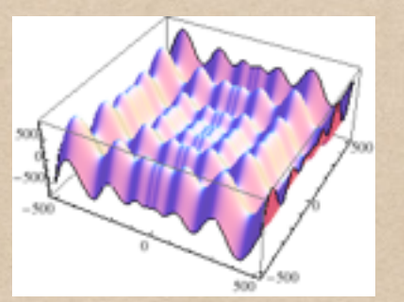
Optimalizace složitých funkcí

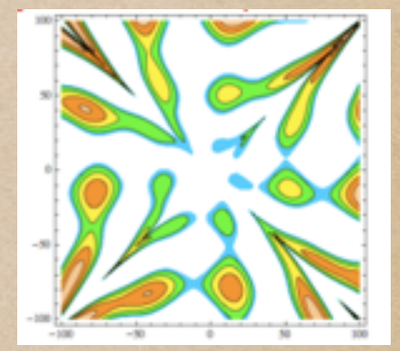
* evoluční algoritmy slouží k hledání optimálních hodnot složitých funkcí, účelových funkcí, které popisují daný problém

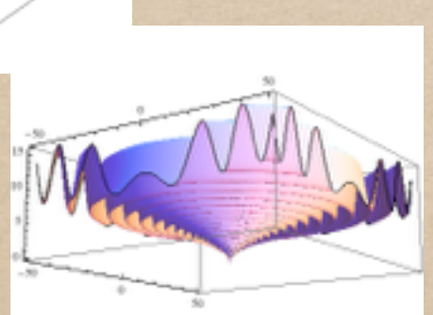
* případně se složitými podmínkami

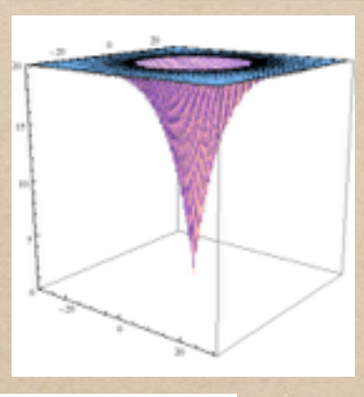


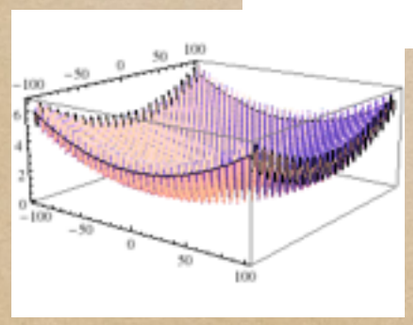


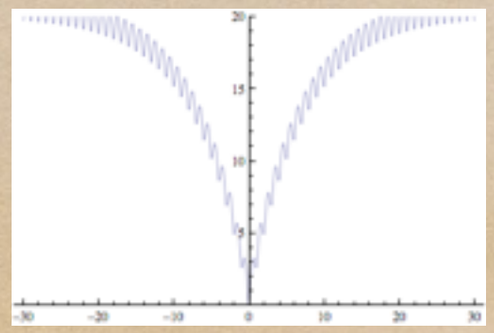


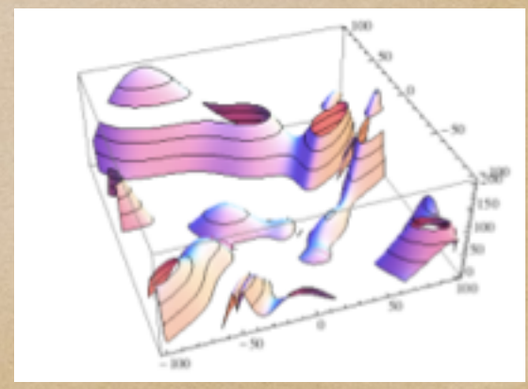






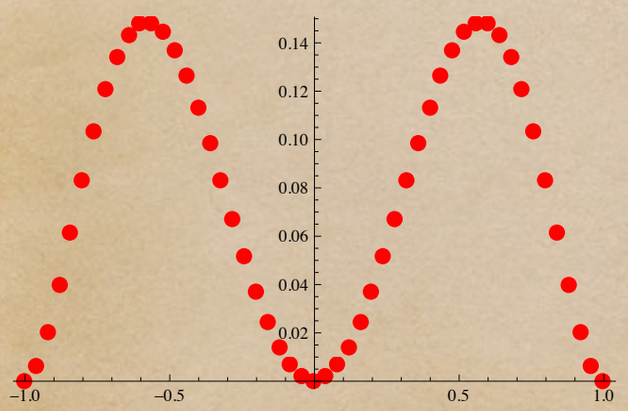


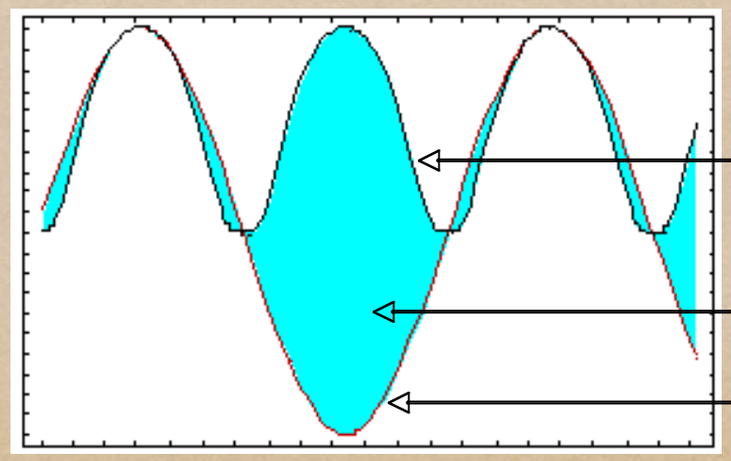




Účelová funkce

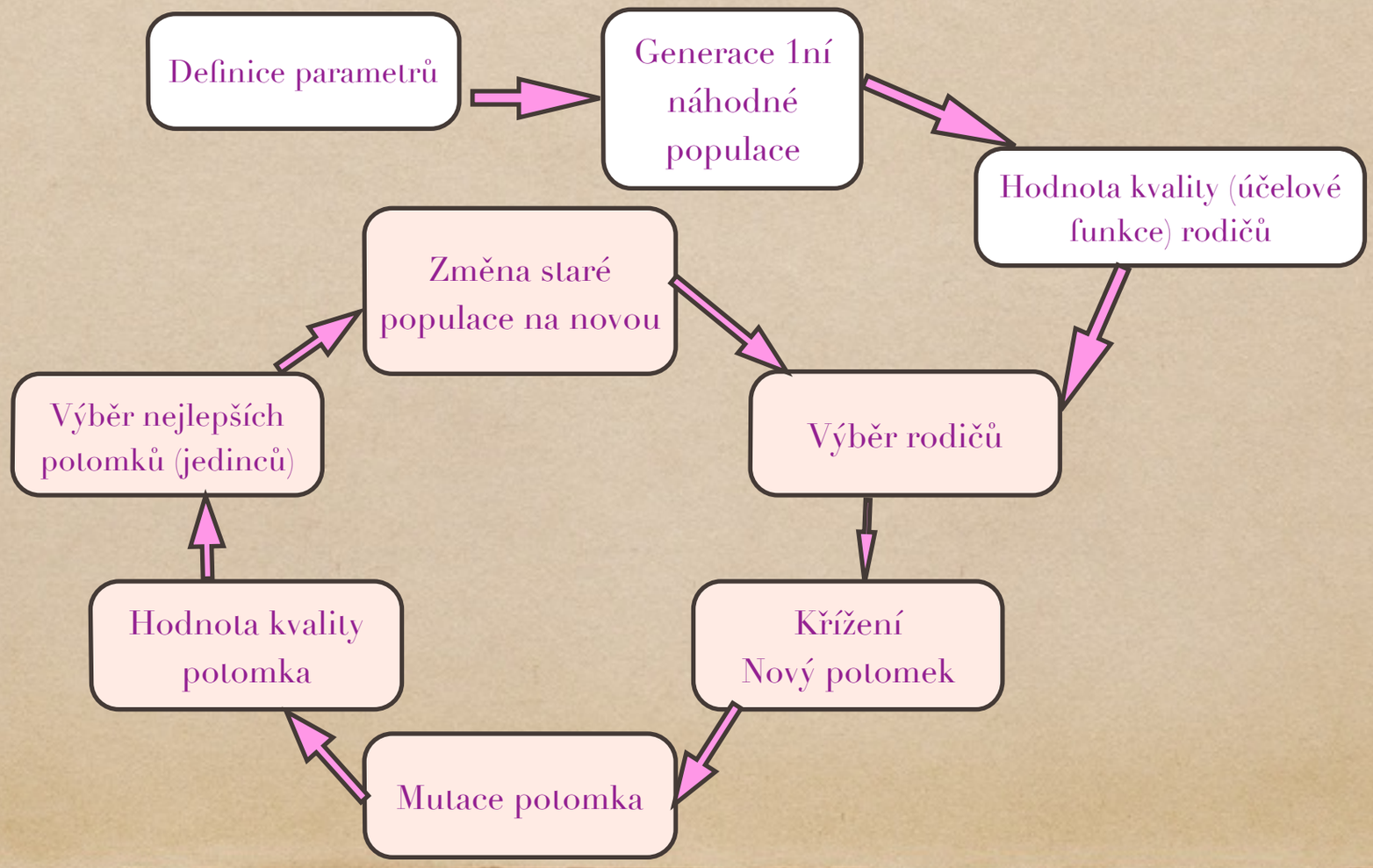
- neexistuje univerzální kuchařka, jak vytvořit účelovou funkci
- nicméně správné nastavení ovlivňuje kvalitu výsledků
- je potřeba vědět, z čeho se vychází a čeho se má dosáhnout
- účelová funkce F_{cost} se nastavuje jako rozdíl mezi žádaným a aktuálně získaným řešením (u syntetizování funkcí obecně, tedy např. u Analytického programování)
- nejlepší řešení je takové, kde hodnota účelové funkce je rovna nule
- cílem je tedy minimalizovat modrou plochu



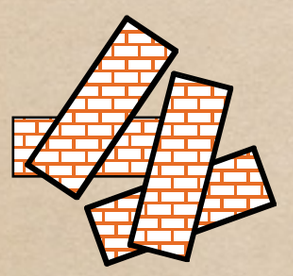


Evoluční algoritmy

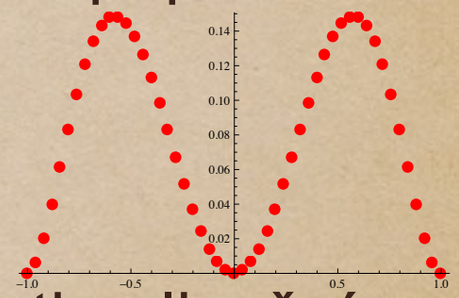
- obecné schéma
- jedinec - řešení dané účelové funkce
- důležitá je kvalita - hodnota účelové funkce
- parametry u různých algoritmů různé
- jedno “kolečko” - jedna “generace, mutace” - běží x generací, než se najde nejvhodnější (optimální) řešení



Symbolická regrese

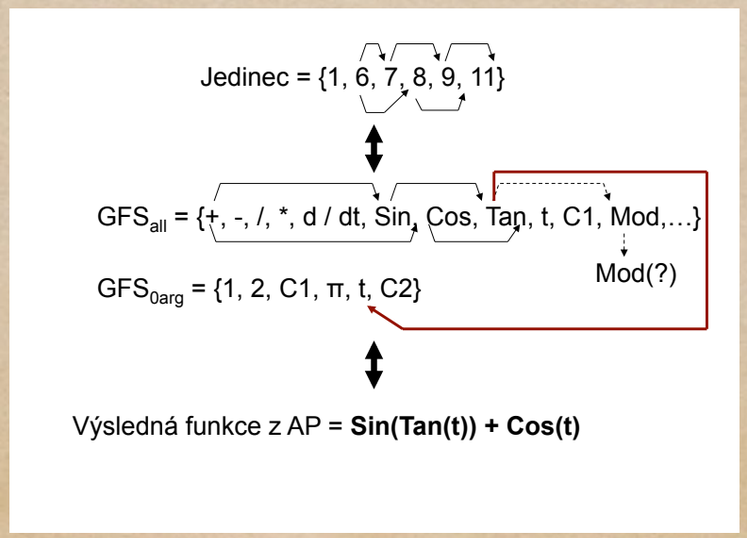


- používá se k syntéze analytického řešení podle zadaných kriterií
- symbolická regrese je proces, při kterém se z malých stavebních kamenů staví složitější struktura, která má popisovat zadané chování
- symbolická identifikace funkce
- k dosažení nejlepšího tvaru řešení je třeba použít optimalizační techniky - evoluční algoritmus



Analytické programování

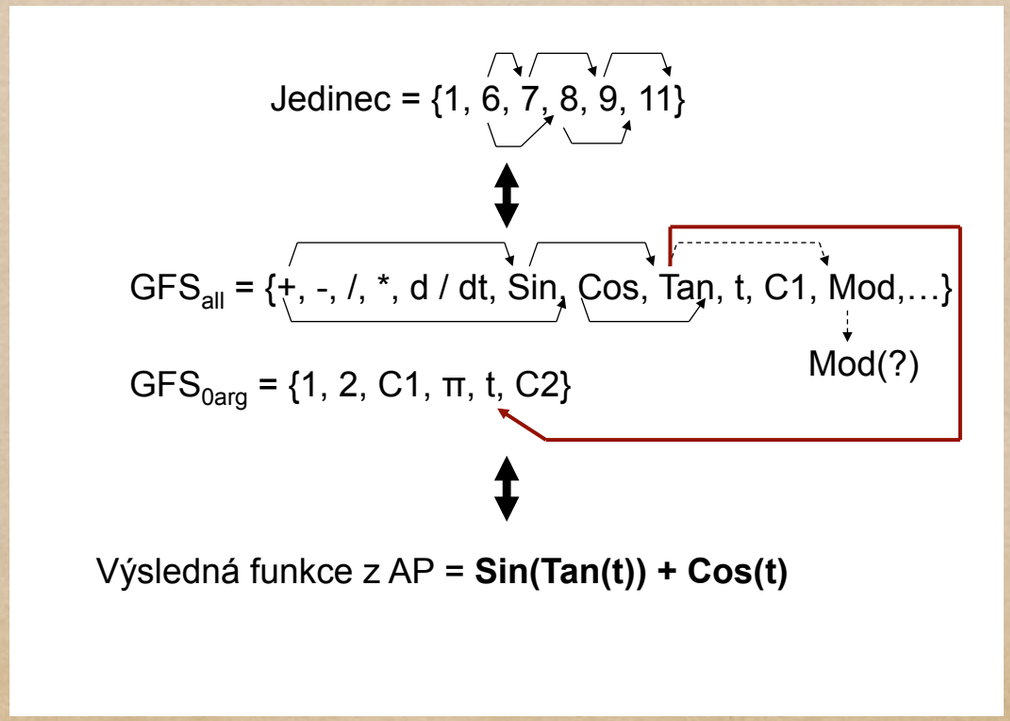
- prof. Ivan Zelinka
- algoritmus ve zjednodušené formě
- jedinec je řešení v dané účelové funkci
    - v případě AP obsahuje čísla, která jsou pointery do množiny operátorů (AND, NAND, plus, minus…) a proměnných (vstup 1, vstup 2, x…)
- je třeba získat kvalitu jedince - tedy nalézt strukturu funkce, do které se dosadí “požadované hodnoty vstupů a výstupu” a spočítat v rámci účelové funkce, jak moc se to požadovanému chování blíží
- GFS_{all} nebo GFS_{0arg} - je množina operátorů nebo proměnných - stanovuje uživatel
    - GFS_{0arg} - např. proměnná x - do proměnné nemůže už nic dalšího vstoupit, nelze ji rozvíjet
    - GFS_{2arg} - operátor Plus - lze jej rozvíjet, potřebuje 2 argumenty, něco plus něco



Analytické programování 2

* jedinec - první pozice 1
* první pozice v GFS_{all} je operátor **plus**

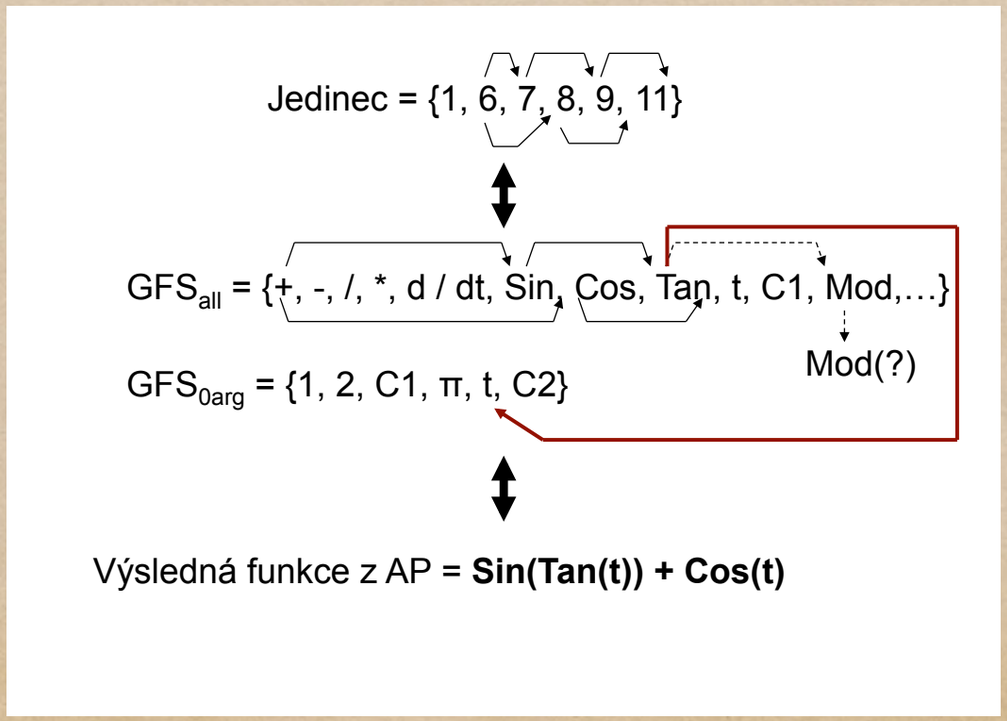
arg1 + arg2



Analytické programování 3

* jedinec - další dvě pozice - 6 a 7
* pozice v GFS_{all} - 6tá je funkce **Sin** a 7má **Cos**

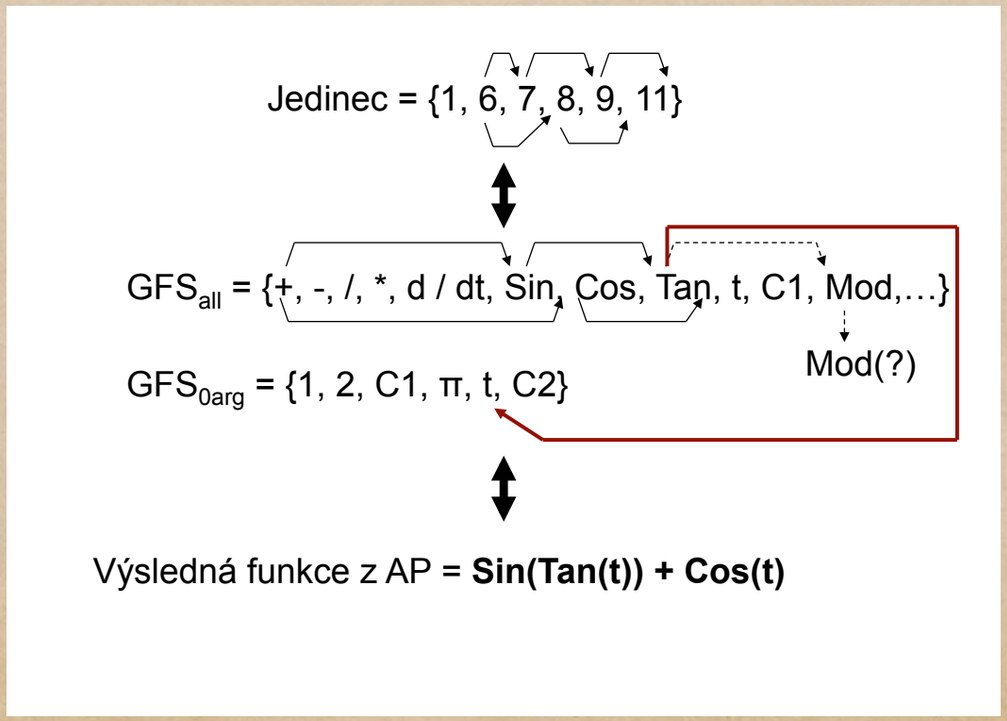
Sin(arg1) + **Cos**(arg2)



Analytické programování 4

* jedinec - pozice pro funkci **Sin** - 8 a pro funkci **Cos** 9
* pozice v GFS_{all} - 8má je funkce **Tan** a 9tá konstanta **t**

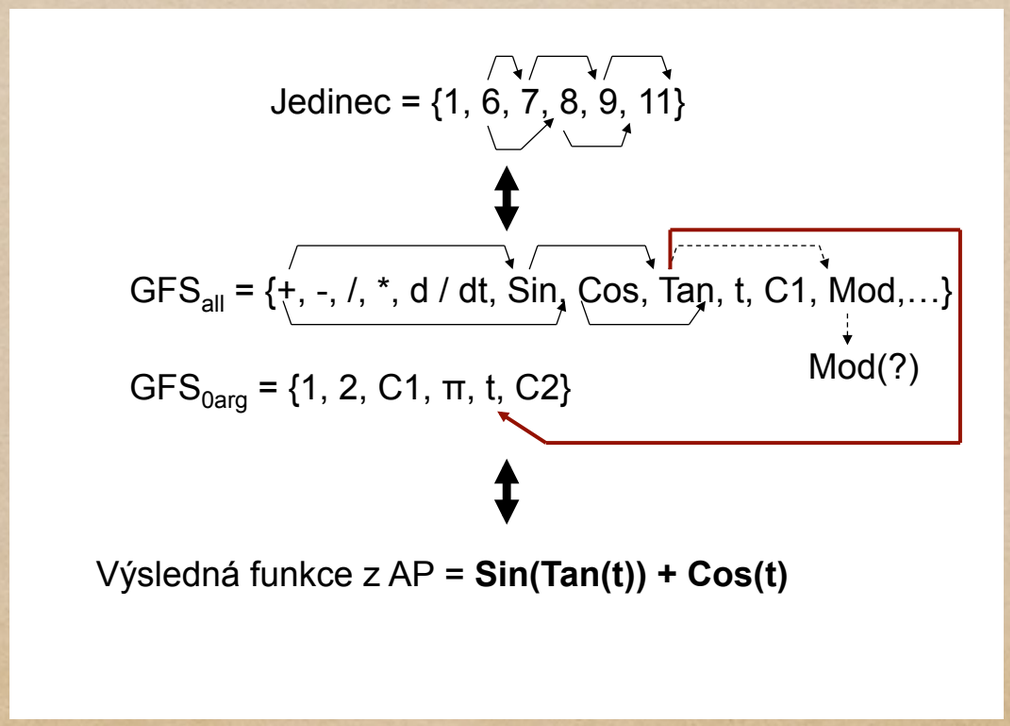
Sin(Tan(arg1**)) + Cos(t)**



Analytické programování 8

* jedinec - zbývá jedna pozice, část s funkcí Cos je uzavřena
* pozice v GFS_{all} - 11tá je funkce **Mod**, ta potřebuje další operátory, ale jedinec již nemá další pozice
* přeskočí se do GFS_{0arg} - 11tá pozice je konstanta **t**

Sin(Tan(t)) + Cos(t)



Problém sudé parity 1

Input 1	Input 2	Input 3	Output
True	True	True	False
True	True	False	True
True	False	True	True
False	True	True	True
True	False	False	False
False	True	False	False
False	False	True	False

$$f_{cost} = \sum_{i=1}^{2^n} |TT_i - P_i|$$

TT_i - i^{th} vystup pravdivostní tabulky

P_i - i^{th} odezva syntetizovaného programu

* pro účelovou funkci se za True a False zvolily hodnoty 1 a 0
* a jako vlastní účelová funkce se zvolila Hammingova vzdálenost mezi požadovaným chováním a aktuální odezvou syntetizovaného programu

Problém sudé parity 2

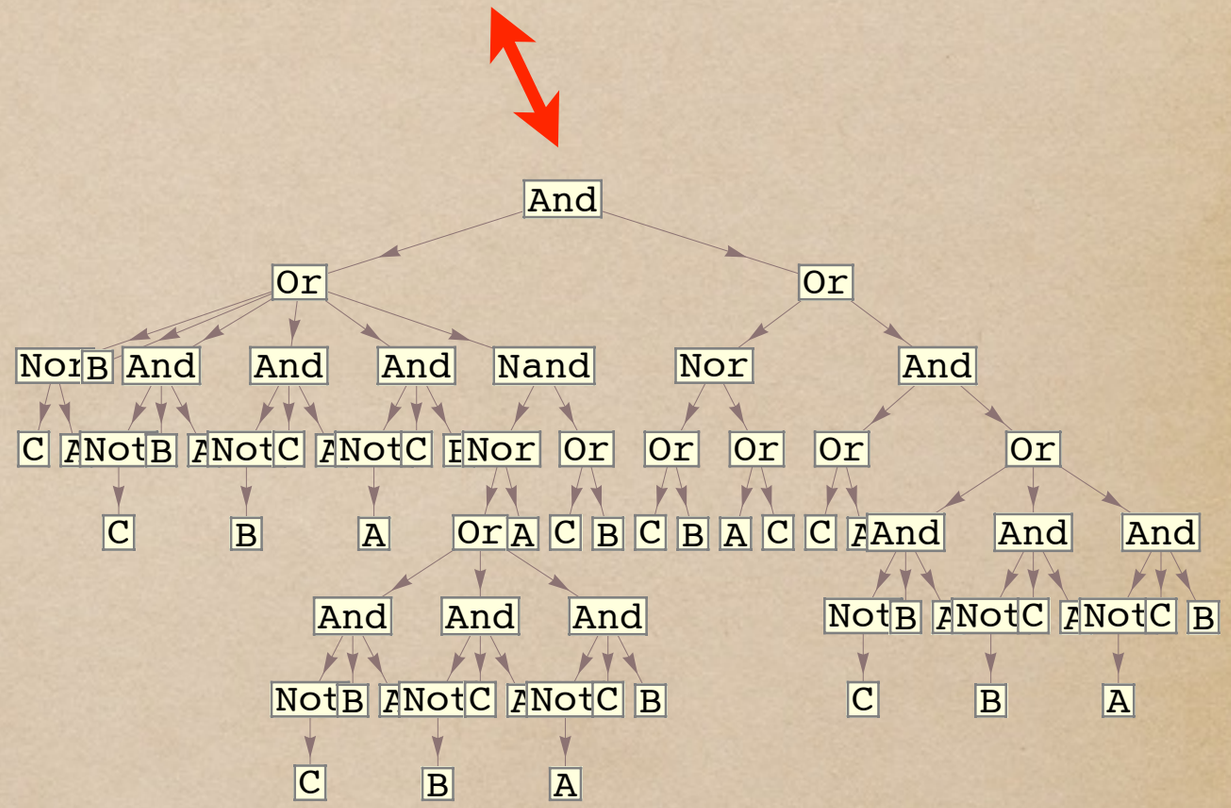
$$((C \,\overline{\nabla}\, A) \vee B \vee (\neg C \wedge B \wedge A) \vee (\neg B \wedge C \wedge A) \vee (\neg A \wedge C \wedge B) \vee (((\neg C \wedge B \wedge A) \vee (\neg B \wedge C \wedge A) \vee (\neg A \wedge C \wedge B)) \,\overline{\nabla}\, A) \,\overline{\wedge}\, (C \vee B))) \wedge (((C \vee B) \,\overline{\nabla}\, (A \vee C)) \vee ((C \vee A) \wedge ((\neg C \wedge B \wedge A) \vee (\neg B \wedge C \wedge A) \vee (\neg A \wedge C \wedge B))))$$

Operátory použité pro syntézu

GFS_{2arg} = {AND, NAND, OR, NOR}

GFS_{0arg} = {Vstup1, Vstup2, Vstup3}

AP_{basic} - bez ohodnocení konstant, žádné nejsou



- výslednou “analytickou funkci obvodu” lze překreslit jako stromovou strukturu
- délka programu je jedním “z problémů” návrhu, lze řešit penalizací účelové funkce
- např. když se objeví inputA AND inputA

Problém sudé parity 3

Řešení problému sudé-3-parity pomocí GA.

	CFE	Délka programu
Minimum	2923	220
Průměr	35386	2910
Maximum	130069	10157

Řešení problému sudé-3-parity pomocí SOMA.

	CFE	Délka programu
Minimum	4256	508
Průměr	23861	2950
Maximum	92542	10770

Řešení problému sudé-3-parity pomocí DE.

	CFE	Délka programu
Minimum	1371	58
Průměr	9456	126
Maximum	20789	186

* experiment byl proveden 50x s Analytickým programováním a různými evolučními algoritmy - Genetické algoritmy, Samoorganizující se migrační algoritmus (SOMA) a Diferenciální evoluce
* pokaždé našel řešení
* nicméně s různou délkou vlastního “programu” - logického obvodu a různým počtem ohodnocení účelové funkce (“jak byl algoritmus rychlý”)

Světelná signalizace 1

Operátory použité pro syntézu

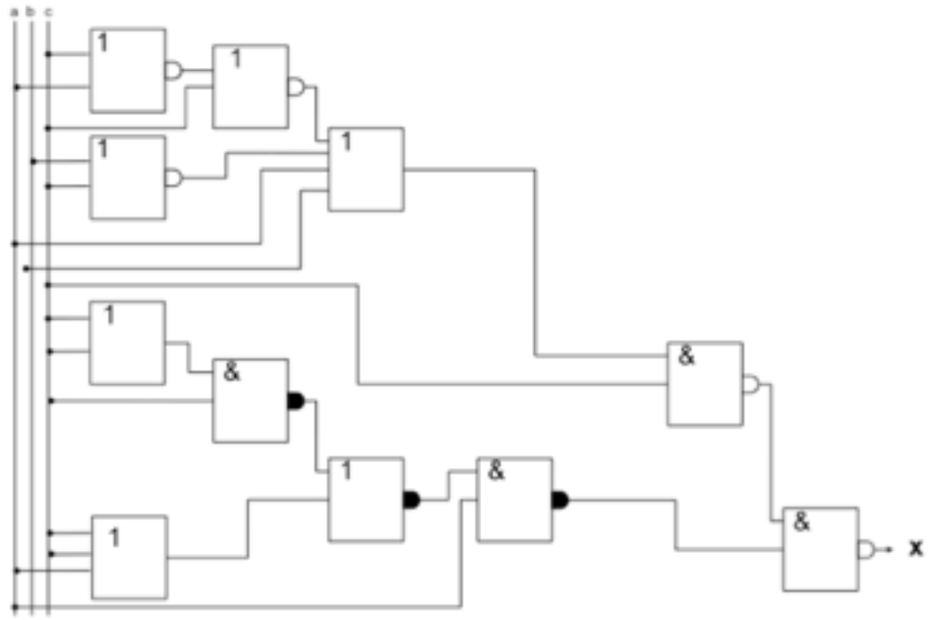
GFS_{2arg} = {AND, NAND, OR, NOR}

GFS_{0arg} = {Vstup1, Vstup2, Vstup3}

AP_{basic} - bez ohodnocení konstant, žádné nejsou

Vstup 1	Vstup 2	Vstup 3	Výstup
False	False	False	False
False	False	True	True
False	True	False	False
False	True	True	True
True	False	False	False
True	False	True	False
True	True	False	False

$(C \barwedge (((C \barvee A) \barvee C) \vee (B \barvee C) \vee A \vee B)) \barwedge (A \barwedge ((C \vee C \vee A) \barvee ((C \vee C) \barwedge C)))$



- průjezd na semafor - např. u závor,atd.
- ukázky z diplomové práce studenta Romana Strakoše

Světelná signalizace 2

- různá řešení
- řešení 1
- použité integrované obvody

  MC 4075 1x – tří vstupý OR

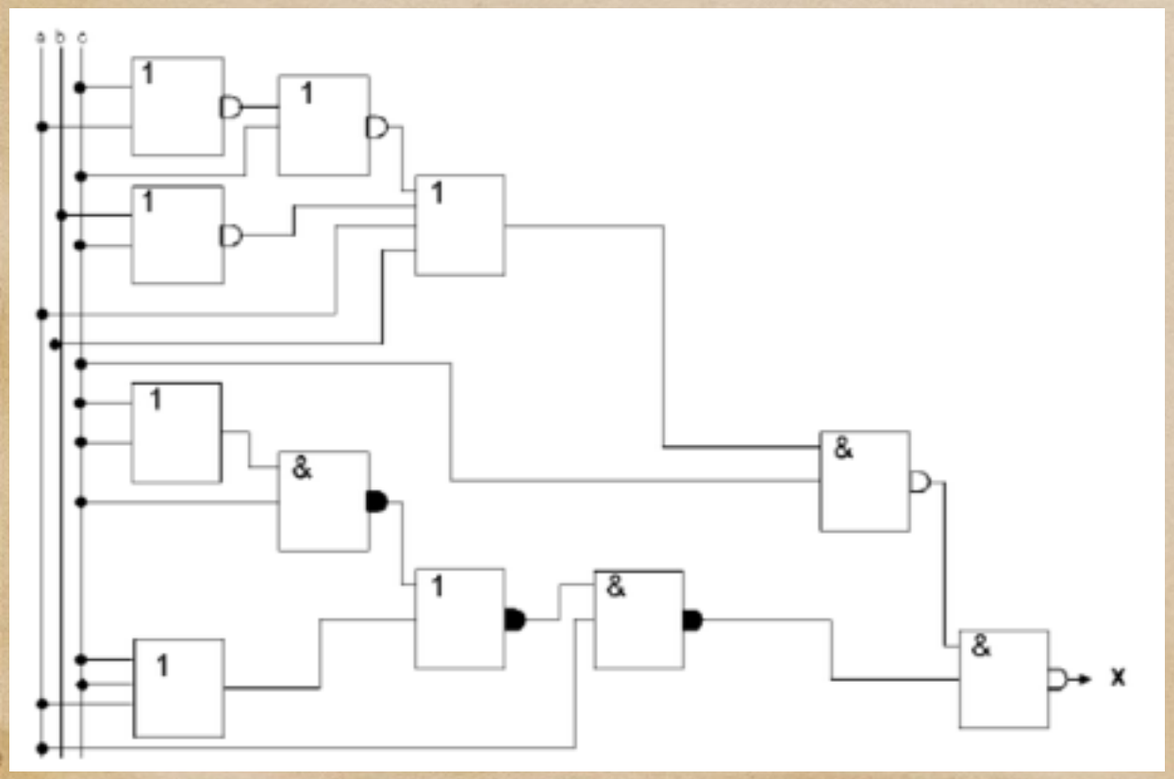
  MC 7400 1x – čtyř vstupý NAND

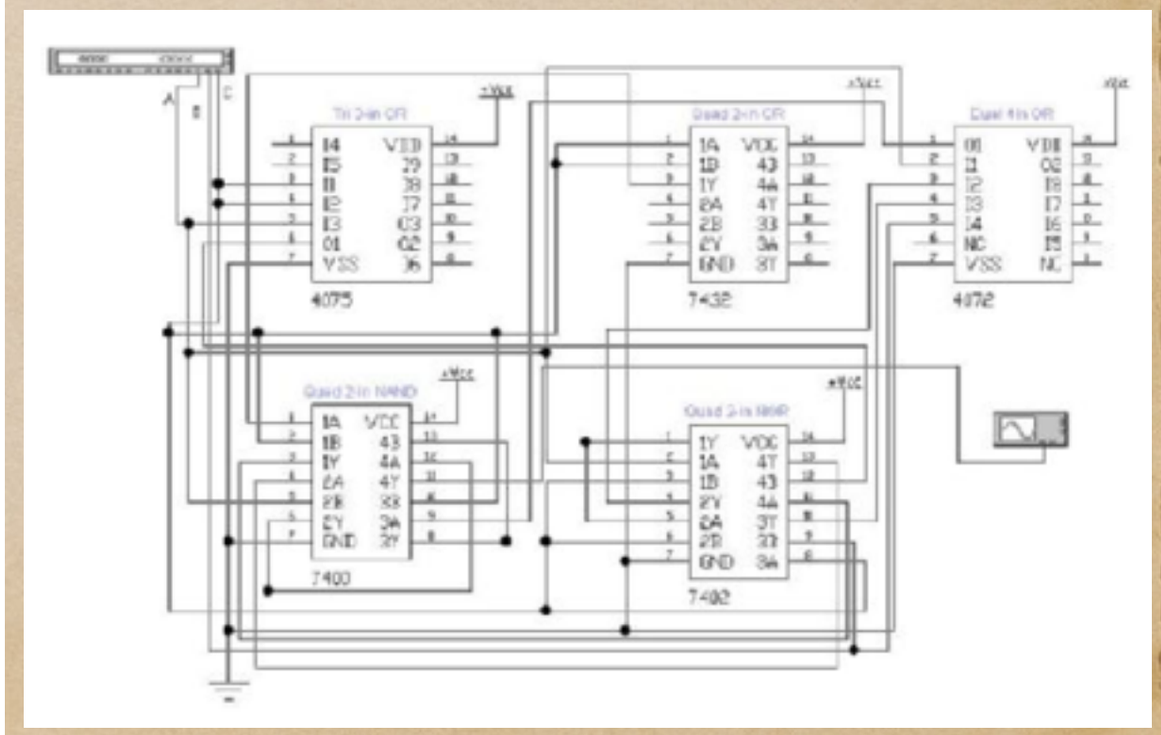
  MC 7432 1x – čtyř vstupý OR

  MC 7402 1x – čtyř vstupý NOR

  MC 4072 1x – osmi vstupý OR

$$(C \barwedge (((C \barvee A) \barvee C) \vee (B \barvee C) \vee A \vee B)) \barwedge (A \barwedge ((C \vee C \vee A) \barvee ((C \vee C) \barwedge C)))$$





Světelná signalizace 3

- řešení 2
- použité integrované obvody

  MC 7408 1x – čtyř vstupý AND

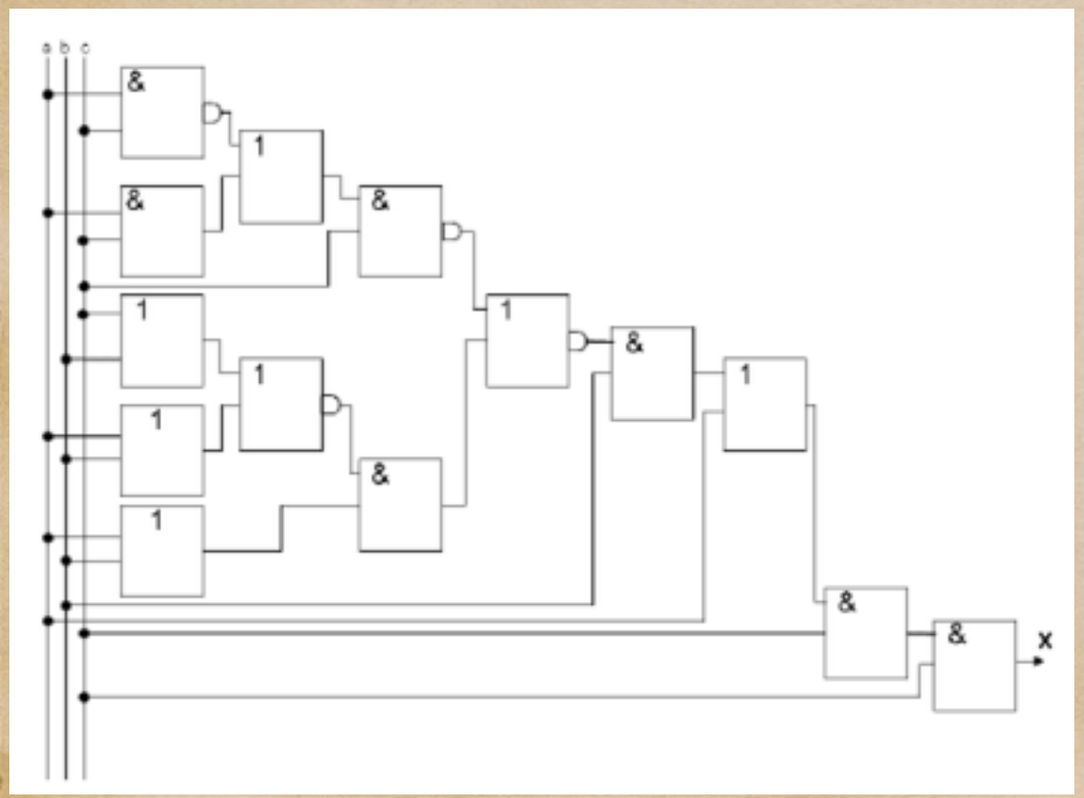
  MC 7400 1x – čtyř vstupý NAND

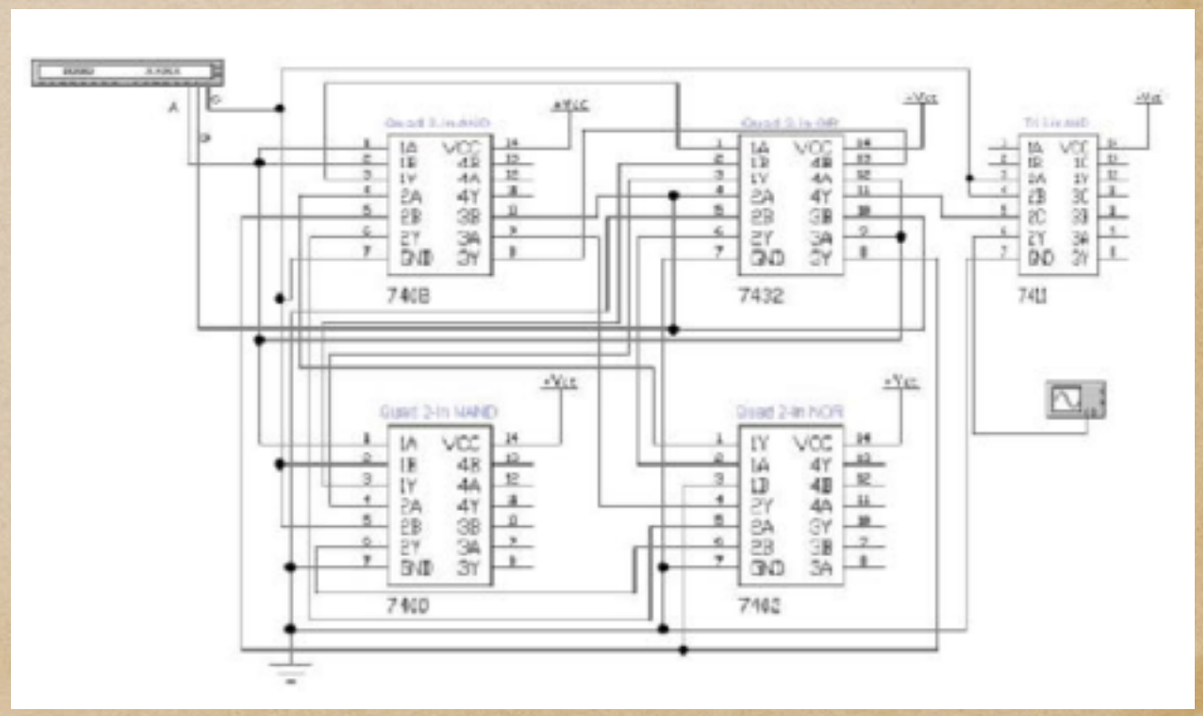
  MC 7432 1x – čtyř vstupý OR

  MC 7402 1x – čtyř vstupý NOR

  MC 7411 1x – tří vstupý AND

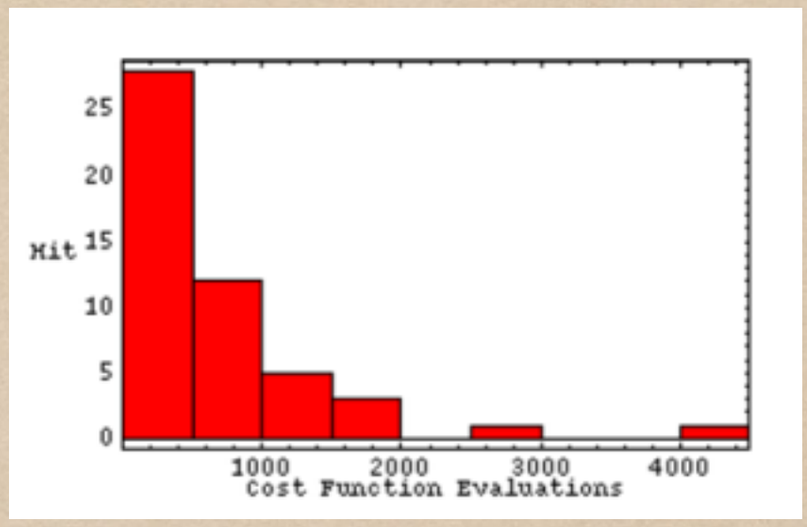
$$(A \vee (B \wedge ((((A \barwedge C) \vee (A \wedge C)) \barwedge C) \bar{\vee} (((C \vee B) \bar{\vee} (A \vee B)) \wedge (A \vee B))))) \wedge C \wedge C$$



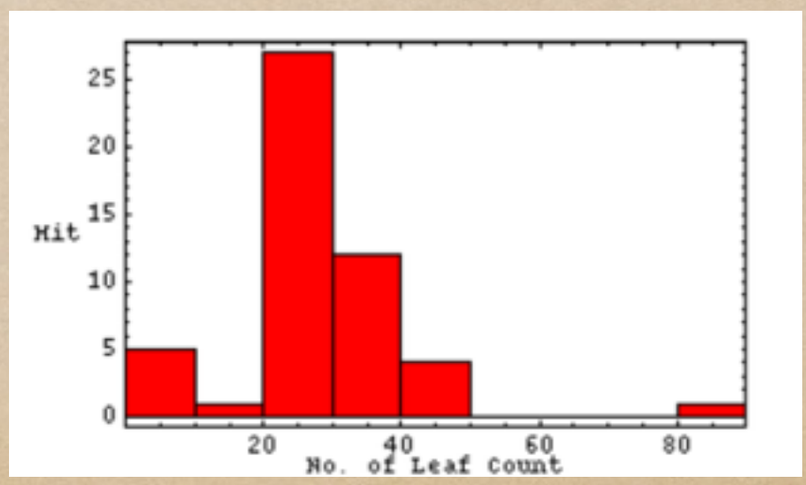


Světelná signalizace 4

- různý počet ohodnocení účelové funkce CFE
- Průměr = 636.2
- MIN = 3
- MAX = 4154



- různý počet objektů finálního obvodu
- Průměr = 28.84
- MIN = 5
- MAX = 82



Závěrem

- metody výpočetní inteligence lze s úspěchem použít v mnoha oblastech
- na naší fakultě (Fakulta aplikované informatiky, Univerzita Tomáše Bati ve Zlíně) máme tým kolegů, kteří se touto oblastí zabývají a nejnovější poznatky tohoto rychle se rozvíjejícího odvětví začleňují pravidelně do svých přednášek a cvičení
- 30ti letá historie v oblasti řízení systémů, potažmo aplikované informatiky (1986 - Katedra automatizovaných systémů řízení technologických procesů, 2006 - Fakulta aplikované informatiky)
  - dnes 7 ústavů (kateder) + 2 “jednotky” z projektů podpořených evropskými fondy
    - Ústav elektroniky a měření
    - Ústav matematiky
    - **Ústav informatiky a umělé inteligence**
    - Ústav počítačových a komunikačních systémů
    - Ústav automatizace a řídicí techniky
    - Ústav řízení procesů
    - Ústav bezpečnostního inženýrství

    - Regionální výzkumné centrum Cebia – tech
    - Vědecko – technický park Informační a komunikační technologie

FAI - UTB - studijní programy