

Riešenie praktických problémov pomocou bio-inšpirovaných výpočtových metód (BIOM) v MATLABe

Ivan Sekaj

*Fakulta elektrotechniky a informatiky, Slovenská
technická univerzita v Bratislave*

E-mail: ivan.sekaj@stuba.sk

Aké typy úloh vieme riešiť pomocou BIOM?

- **optimalizácia**
(minimalizácia / maximalizácia)
- **hľadanie riešení - prehľadávanie priestoru potenciálnych riešení**

Maximalizácia

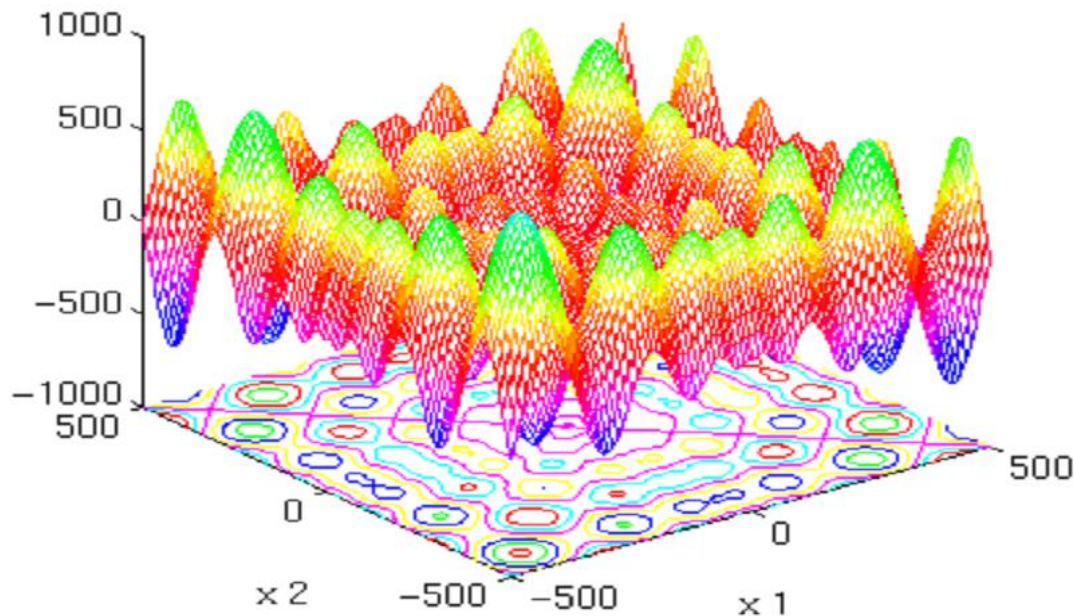
- **zisk**
- **výkon**
- **efektivita**
- **...**

Miminalizácia

- **spotreba, energia**
- **čas**
- **náklady**
- **chyba**
- **odchýlka od požadovaného stavu ...**

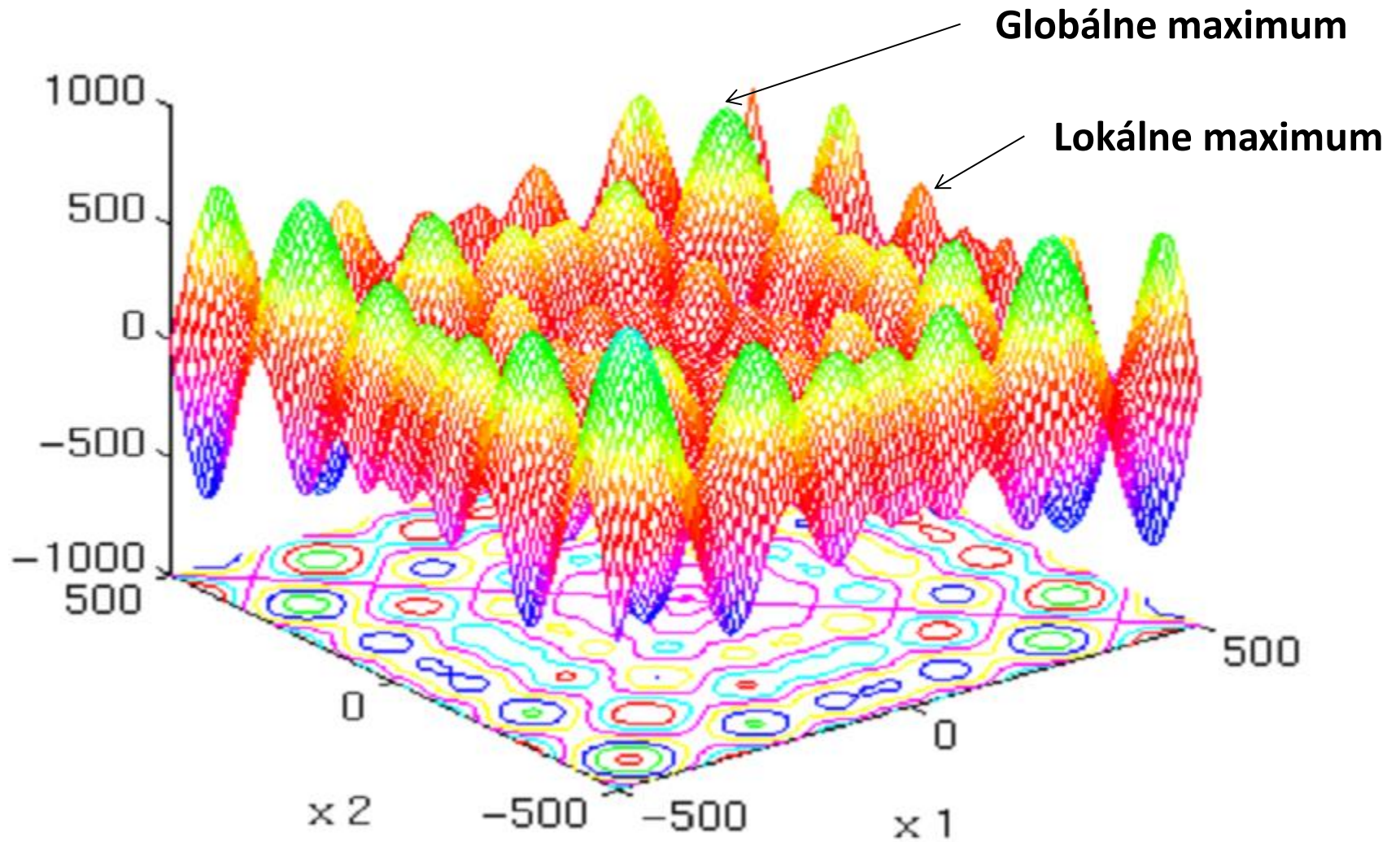
Optimalizačná úloha sa dá transformovať na minimalizáciu/maximalizáciu funkcie n-premenných

$$X^* = [x_1, x_2, \dots, x_n] = ? ; \quad F(X^*) = \min(F(X)); \quad (\text{resp. } \max)$$



účelová funkcia, kritériálna funkcia, „fitness“

Optimalizačná úloha



Náš toolbox (genetic toolbox, Matlab) je minimalizačný. Lepšie riešenia sú tie, ktoré dosahujú nižšiu hodnotu účelovej funkcie.

Prevedenie maximalizačnej úlohy na minimalizačnú:

$$\mathbf{x}_{\text{opt}} = \max(F(\mathbf{x})) \rightarrow \mathbf{x}_{\text{opt}} = \min(-F(\mathbf{x}))$$

Klasifikácia optimalizačných metód

- **exaktné (analytické)**
- **numerické (približné)**
 - **deterministické**
 - **stochastické (metaheuristické)**
 - **konvenčné (ne-evolučné)**
 - **evolučné**
 - **iné bio-inšpirované**

Pomocou optimalizačného prístupu možno riešiť aj úlohy, ktoré nie sú v prvom ponímaní maximalizačné/minimalizačné úlohy.

Prevedenie exaktnej úlohy - výpočet inverznej matice - na približnú minimalizačnú:

$$A^{-1}=? ; A * A^{-1}=I$$

$$A * A^{-1}-I=0$$

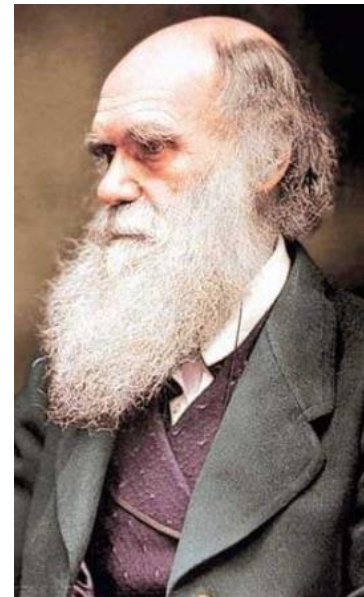
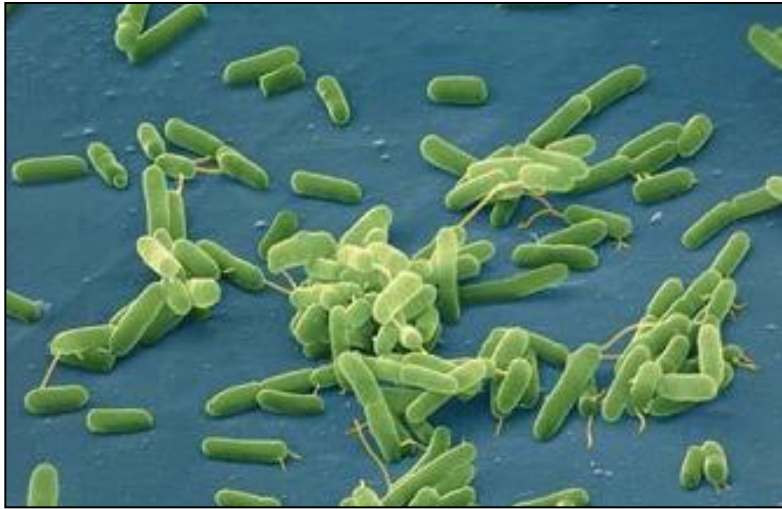
$$X=? ; (A * X-I) \rightarrow \min$$

Bio-inšpirované výpočtové metódy

evolučné / iné

Motivácia vzniku evolučných algoritmov (genetického algoritmu)

Biologická evolúcia



1 bunka - baktéria

6000 miliárd buniek - človek

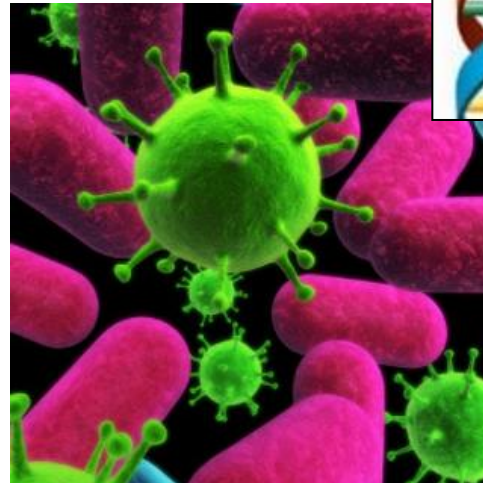
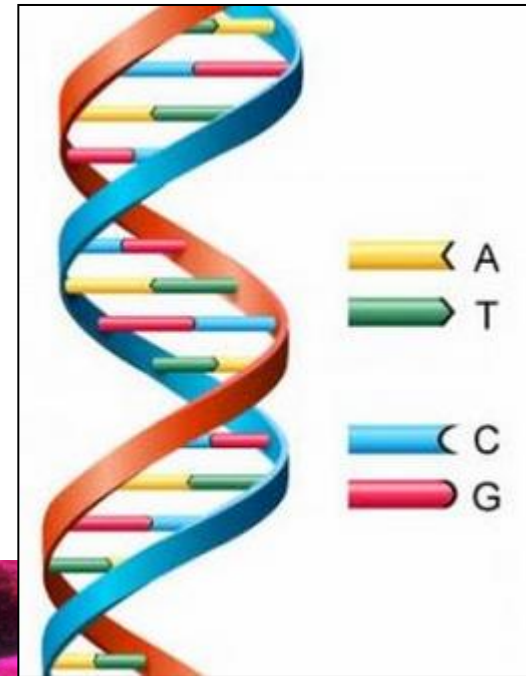
Reprezentanti bio-inšpirovaných prístupov

- Evolučné algoritmy
- ACO – kolónie mravcov
- PSO (krdľový alg.)
- Včelí algoritmus
- Kvetový algoritmus (opelovací)
- Svetluškový algoritmus
- Kukučkový algoritmus
- Iné ...



Reprezentanti evolučných algoritmov

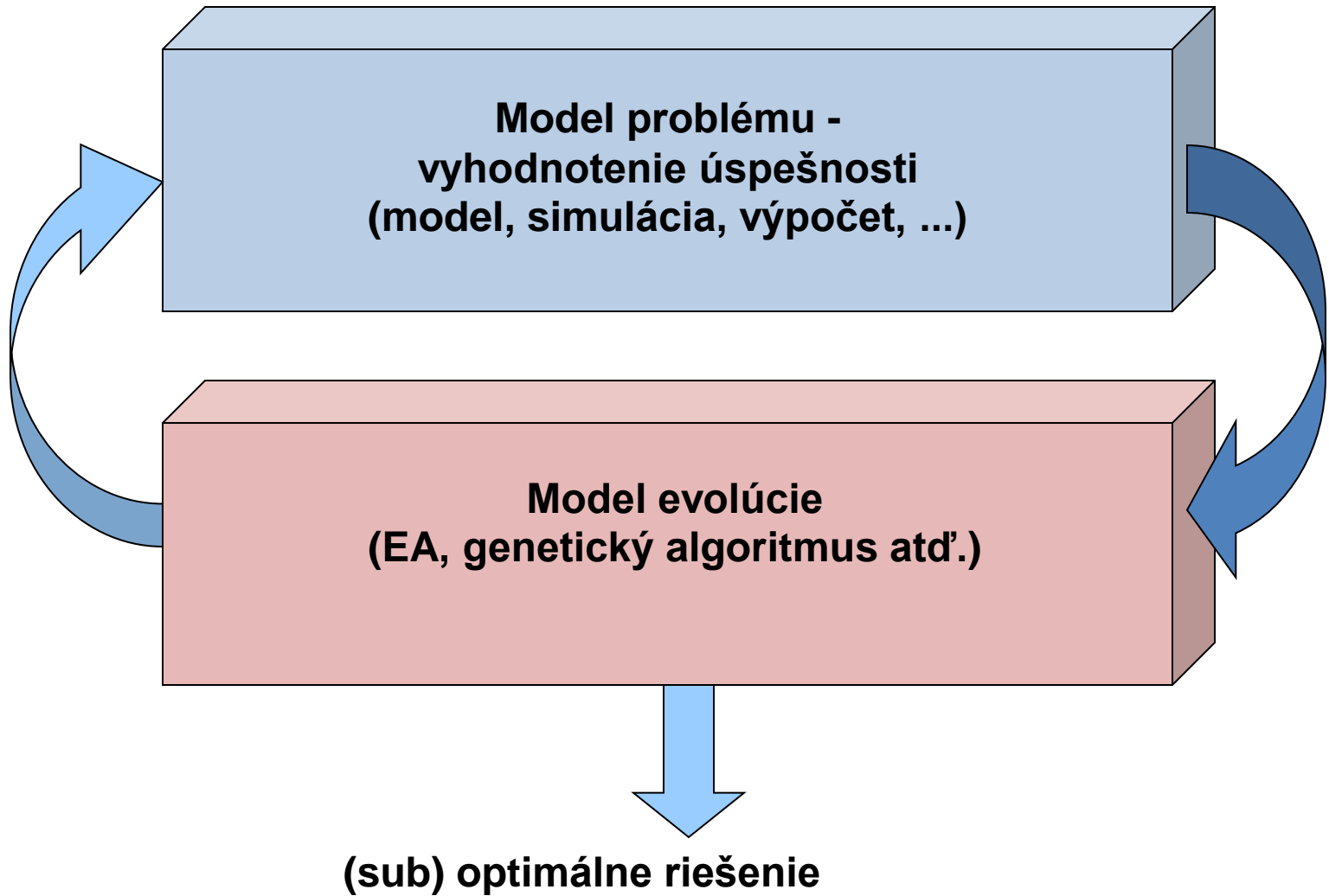
- Evolučné stratégie
- Evolučné programovanie
- Genetické algoritmy ←
- Genetické programovanie
- Diferenciálna evolúcia
- Umelý imunitný systém
- Gramatická evolúcia
- iné ...



Evolučný algoritmus

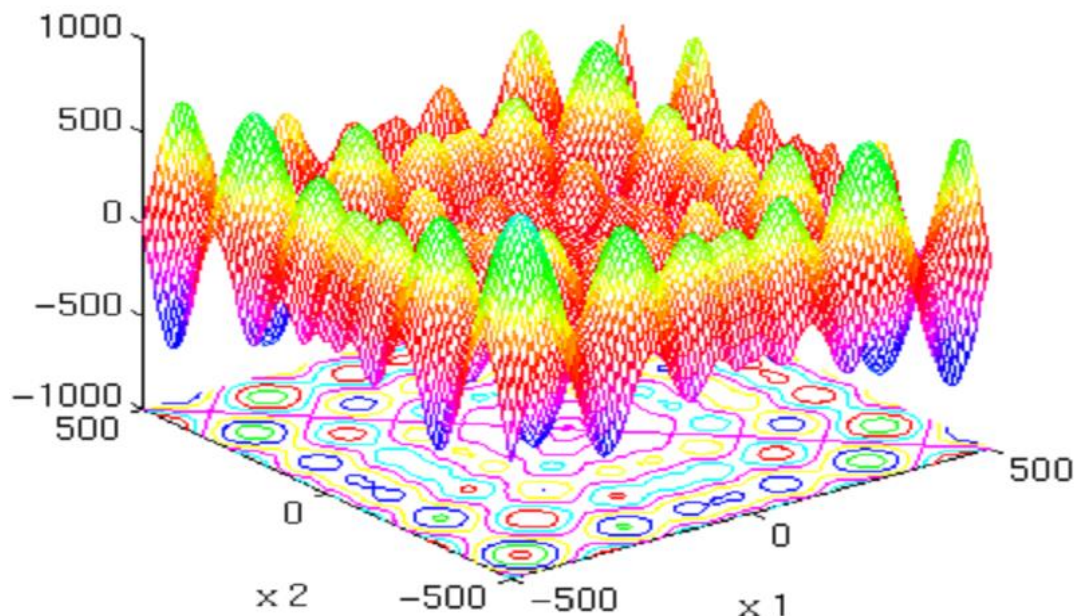
1. Inicializuj populáciu reťazcov
2. Vyhodnot' úspešnosť každého reťazca populácie
3. Vyber rodičov
(úspešnejší majú väčšiu pravdepodobnosť byť vybraní)
4. Modifikácia rodičov = potomkovia (mutácia, kríženie)
5. Vytvor novú populáciu z potomkov a prežívajúcich jedincov
6. Skok na bod 2

Riešenie problému pomocou EA



Optimalizácia funkcie n-premenných

$$X^* = [x_1, x_2, \dots, x_n] = ? ; \quad F(X^*) = \min(F(X)); \quad (\text{resp. } \max)$$



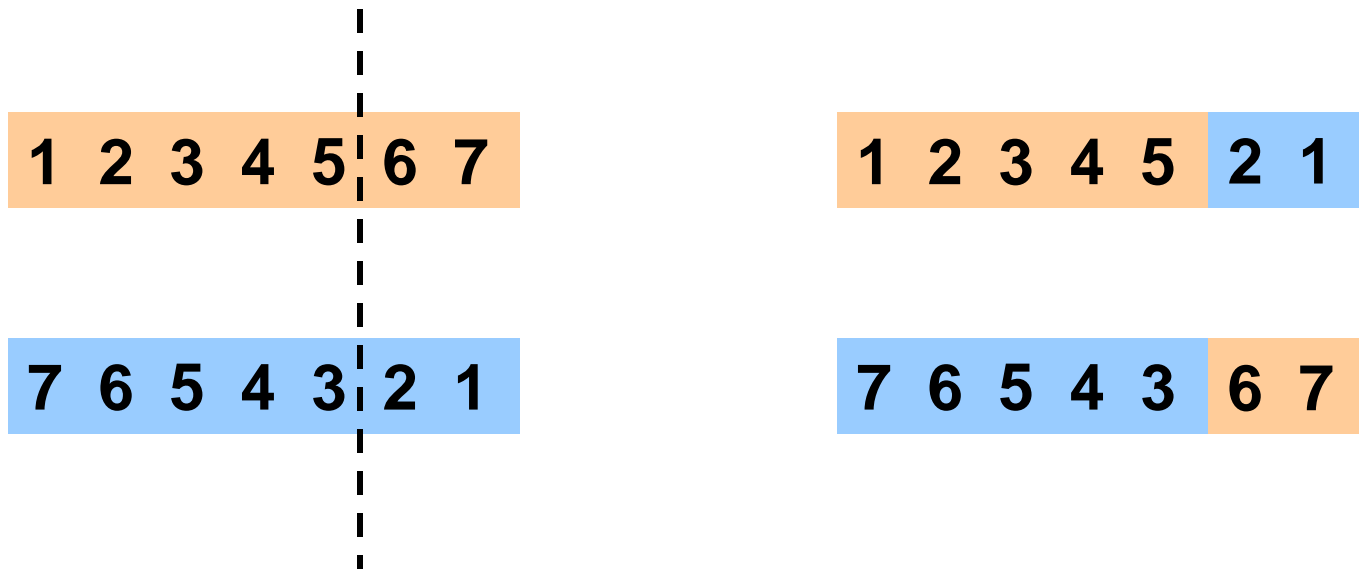
Chromozóm = $[x_1, x_2, \dots, x_n]$ potenciálne riešenie problému

Populácia

Množina „vhodného počtu“ chromozómov
(napr. 30-100)

Genetický algoritmus operuje nad populáciou
chromozómov.

Křížení chromozómov

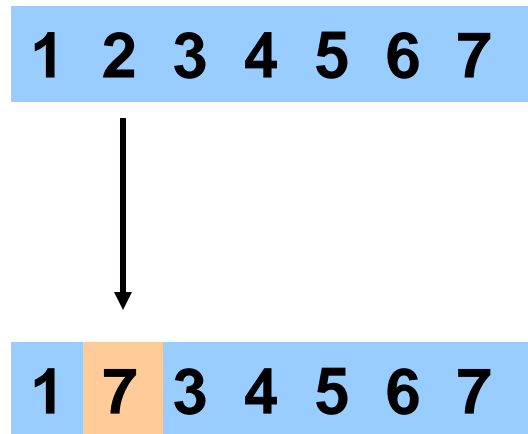


rodičovské reťazce

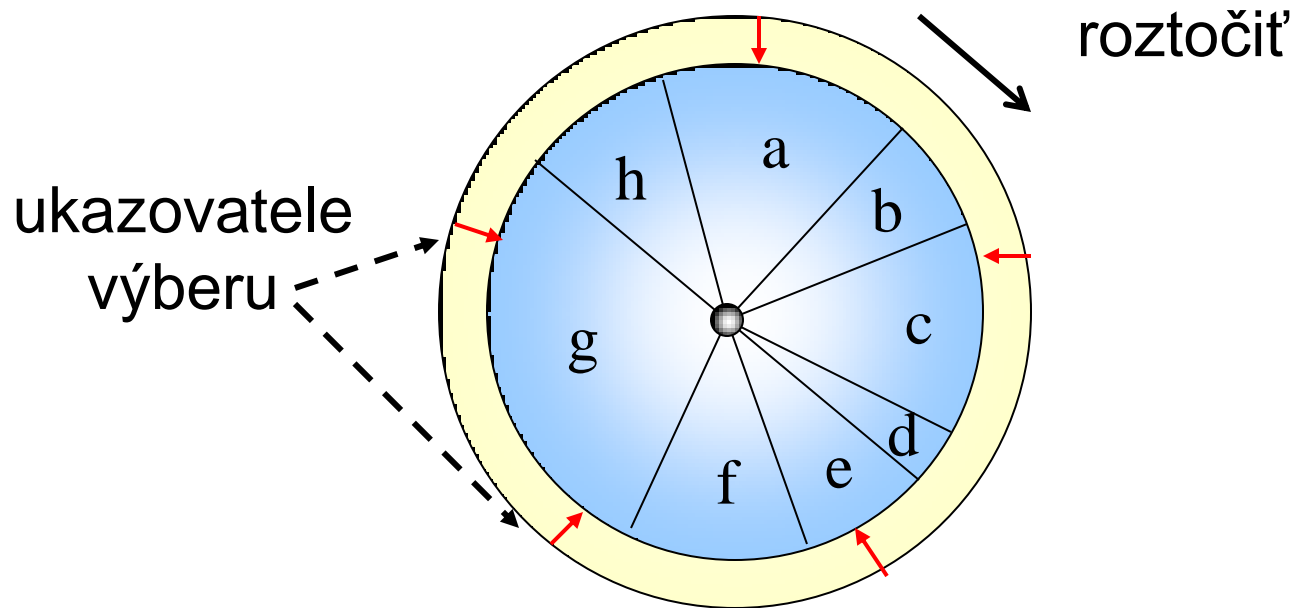
potomkovské reťazce

Príklad 1

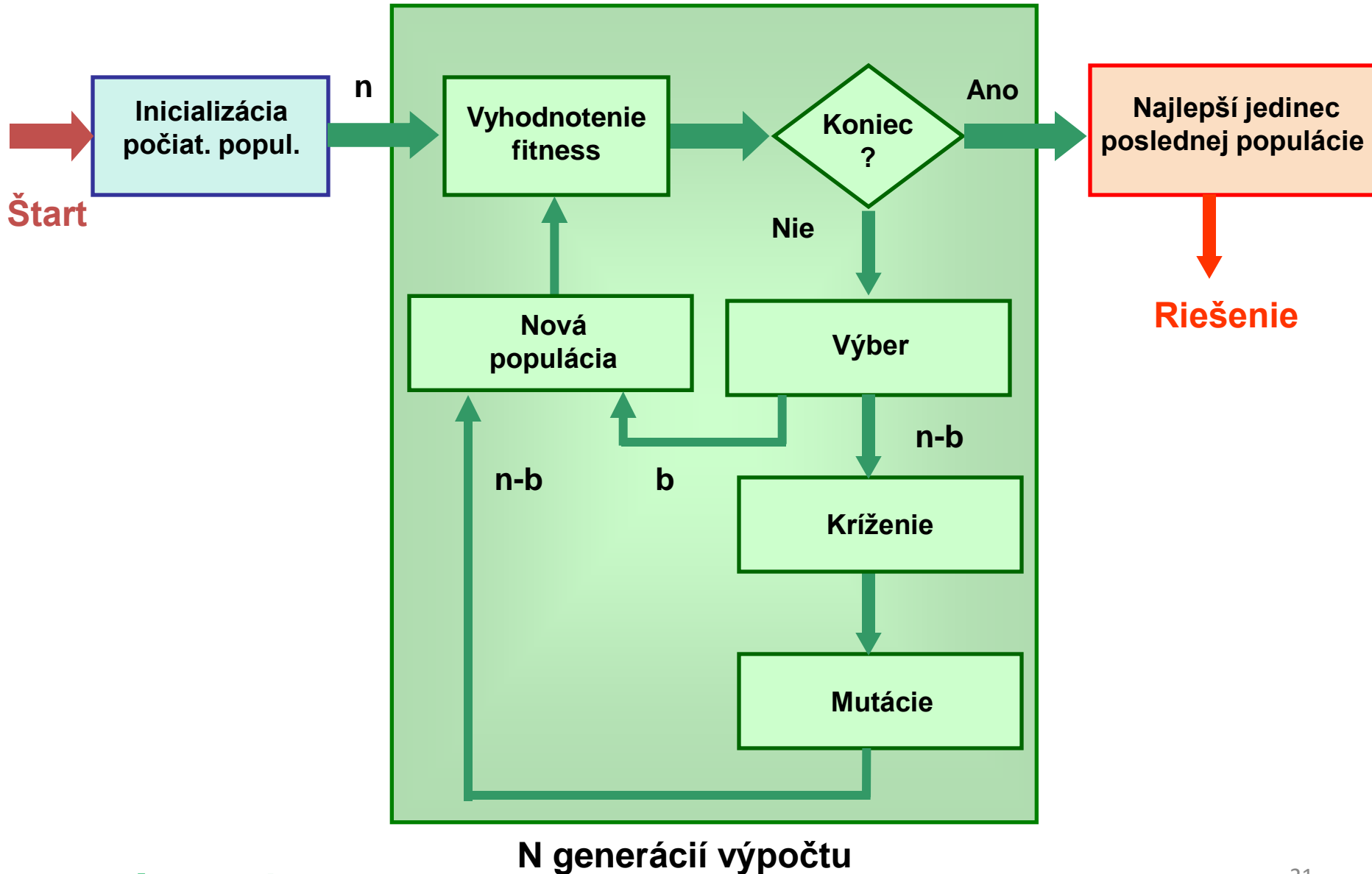
Mutácia chromozómu



Ruletový výber



Genetický algoritmus



Príklad 4

Genetic toolbox - Matlab (ÚRK FEI STU Bratislava, 2000)



2. Genetic toolbox - Mathworks
3. GA – iné zdroje, internet

Prehľad funkcií Genetic toolbox

among - medziľahlé kríženie
change - mutácia duplicitných reťazcov
crosgrp - kríženie medzi viacerými rodičmi
crossov - viacbodové kríženie dvoch rodičov
genrpop - vygenerovanie náhodnej reálnečíselnej populácie
invfit - doplnok účelovej funkcie
muta - aditívna mutácia
mutm - multiplikatívna mutácia
mutx - obyčajná mutácia
selbest - výber najlepších reťazcov
seldiv - výber podľa maximálnej diverzity
selrand - náhodný výber reťazcov
selsort - výber a zotriedenie reťazcov podľa úspešnosti
selsus - výber pomocou váhovaného ruletového kolesa
shake - náhodné premiešanie poradia reťazcov v populácii
swapgen - mutácia poradia génov v reťazci
swappart - vzájomná výmena poradia dvoch častí reťazcov
...

selsus - výber jedincov pomocou váhovaného ruletového kolesa

Charakteristika:

Funkcia vyberie zo vstupnej populácie do výstupnej populácie určený počet reťazcov pomocou výberu, ktorého ekvivalentom je otáčanie váhovaným ruletovým kolesom. Váhované ruletové koleso je rozdelené na kruhové výseky, z ktorých každý je priradený jednému reťazcu a jeho veľkosť je nepriamo úmerná hodnote jeho účelovej funkcie (resp. priamo úmerná jeho úspešnosti). Úspešnejšie reťazce majú väčšiu šancu byť vybrané.

Syntax:

Newpop=selsus(Oldpop,Objpop,Num);

Newpop - nová (výstupná) populácia

Oldpop - stará (vstupná) populácia

Objpop - vektor hodnôt účelovej funkcie starej populácie

Num – počet vybraných najúspešnejších reťazcov

crossov - kríženie

Charakteristika:

Funkcia vytvorí novú populáciu reťazcov, ktorá vznikne skrížením všetkých reťazcov starej populácie 1- až 4-bodovým krížením. Krížené sú všetky reťazce (ak je ich párny počet). Výber párov je buď náhodný alebo sú vybrané susedné reťazce v populácii podľa voľby parametra sel.

Syntax:

Newpop=crossov(Oldpop,num,sel)

Newpop - matica skríženej (výstupnej) populácie

Oldpop - pôvodná (vstupná) populácia

num - počet bodov kríženia (miest rozdelenia) od 1 do 4

sel - spôsob výberu dvojíc:

0 - náhodný

1 - susedné dvojice v populácii

mutx - obyčajná mutácia

Charakteristika:

Funkcia zmutuje populáciu reťazcov s intenzitou úmernou parametru rate (z rozsahu od 0 do 1). Mutovaných je len niekoľko génov v rámci celej populácie. Mutované hodnoty sú zmenené na náhodné hodnoty z priestoru definovaného ohraničeniami pomocou dvojriadkovej matice. Prvý riadok matice určuje dolné ohraničenia jednotlivých génov reťazcov a druhý riadok ich horné ohraničenia.

Syntax:

Newpop=mutx(Oldpop,rate,Space)

Newpop - nová, zmutovaná populácia

Oldpop - stará populácia

Space - matica obmedzení, ktorej 1.riadok je vektor minimálnych a 2. riadok je vektor maximálnych prípustných mutovaných hodnôt

rate - miera početnosti mutovania génov v populácii (od 0 do 1)

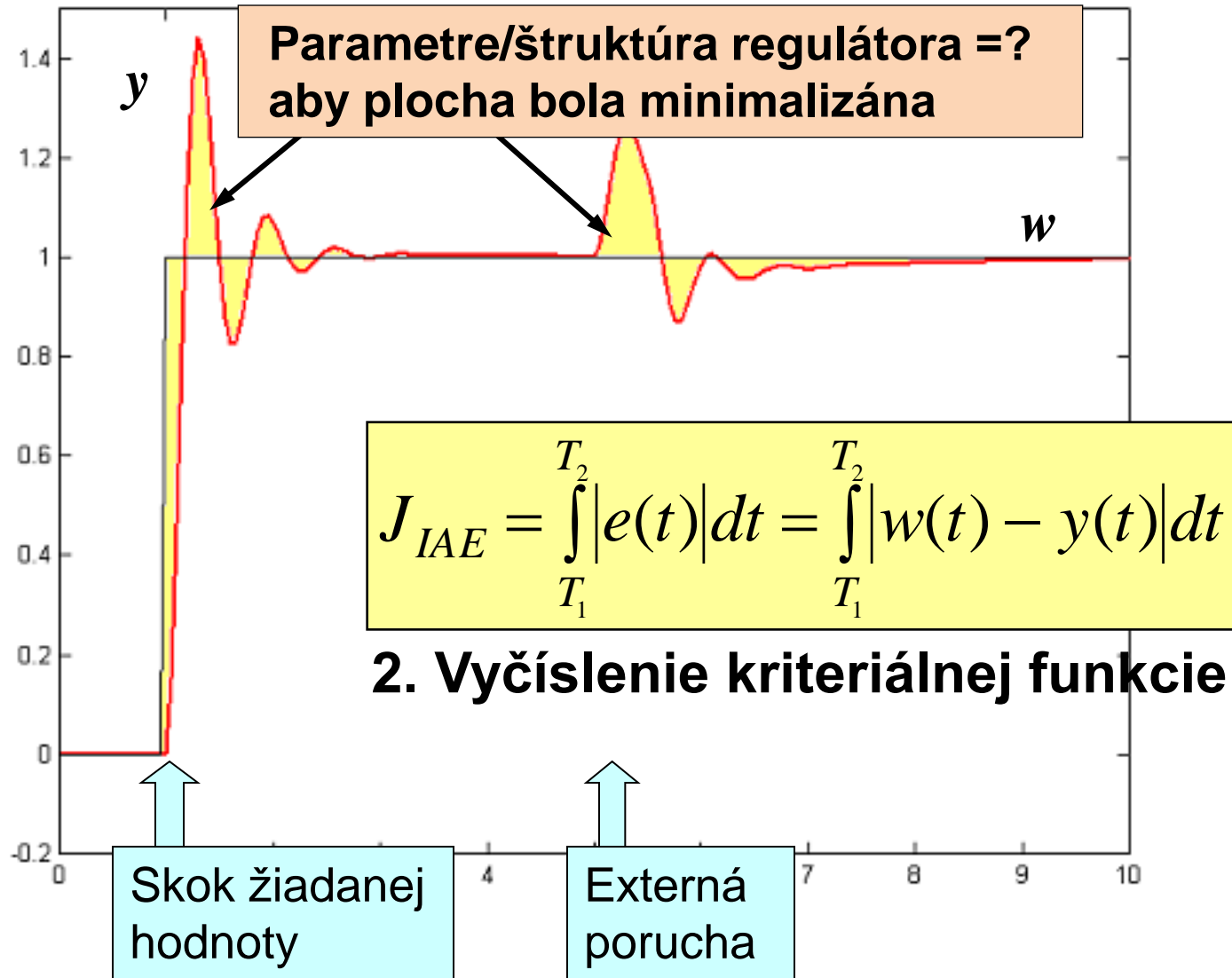
Aplikácie BIOM v praxi

Návrh regulačných obvodov

chromozóm = $[p_1, p_2, \dots, p_n]$
 p_i – parametre regulátora

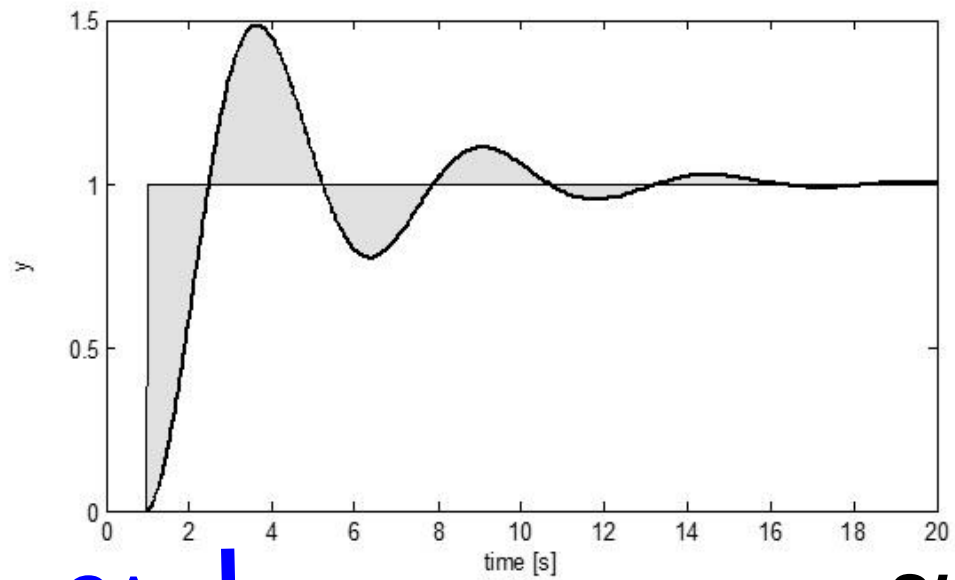
napr. PID regulátor:
chromozóm = $[P, I, D]$

účelová funkcia:



1.Simulácia (Simulink)

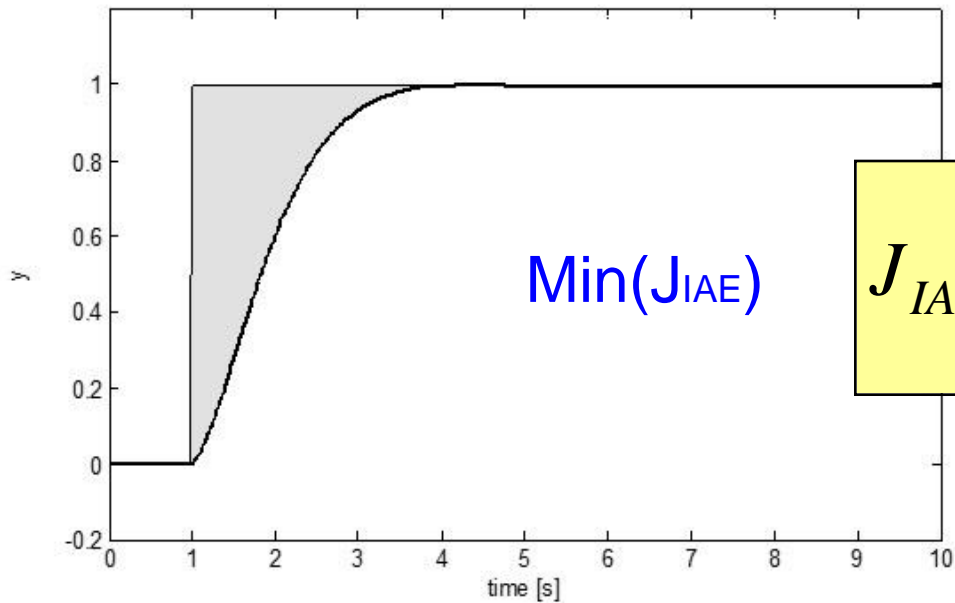
Príklad 5



Optimalizácia - GA



Simulink

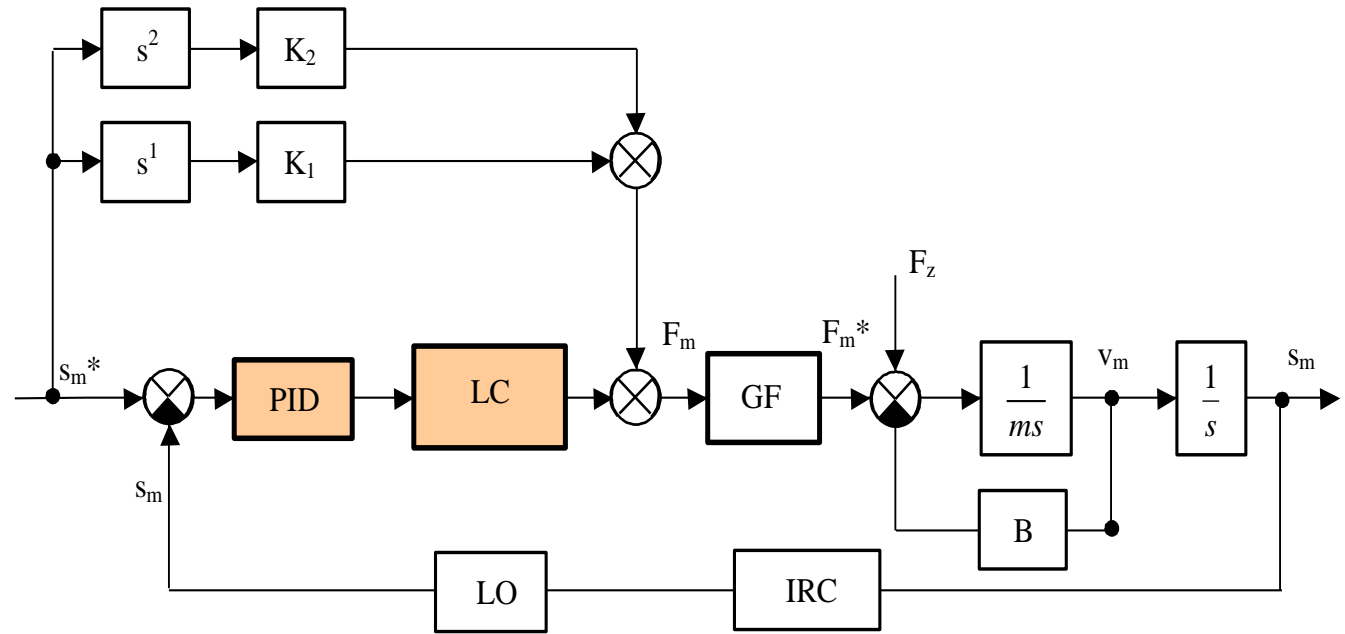


Min(J_{IAE})

$$J_{IAE} = \int_{T_1}^{T_2} |e(t)| dt = \int_{T_1}^{T_2} |w(t) - y(t)| dt$$

Matlab

Návrh riadenia polohového systému s PID regulátorom, lead kompenzátorom a Luenbergerovým pozorovateľom



chromozóm:

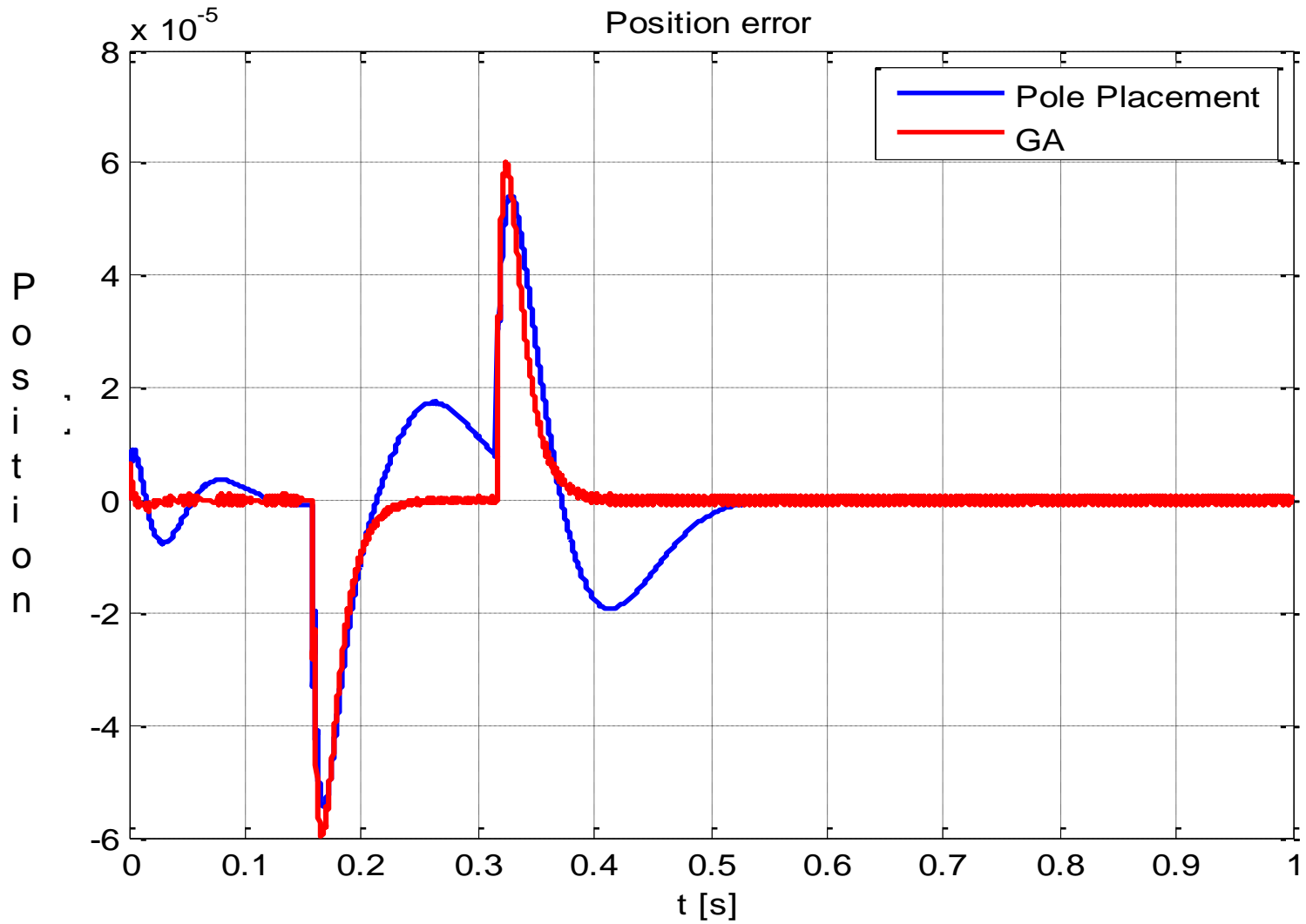
$$ch = \{P, I, D, a_1, a_0, b_0\}$$

Polohový systém s PID regulátorom, lead kompenzátorom (LC), Luenbergerovým pozorovateľom (LO), generátor sily (GF).

$$Fitness = \int_0^T (\alpha |e(t)| + \beta |e'(t)| + \gamma |e''(t)|) dt$$

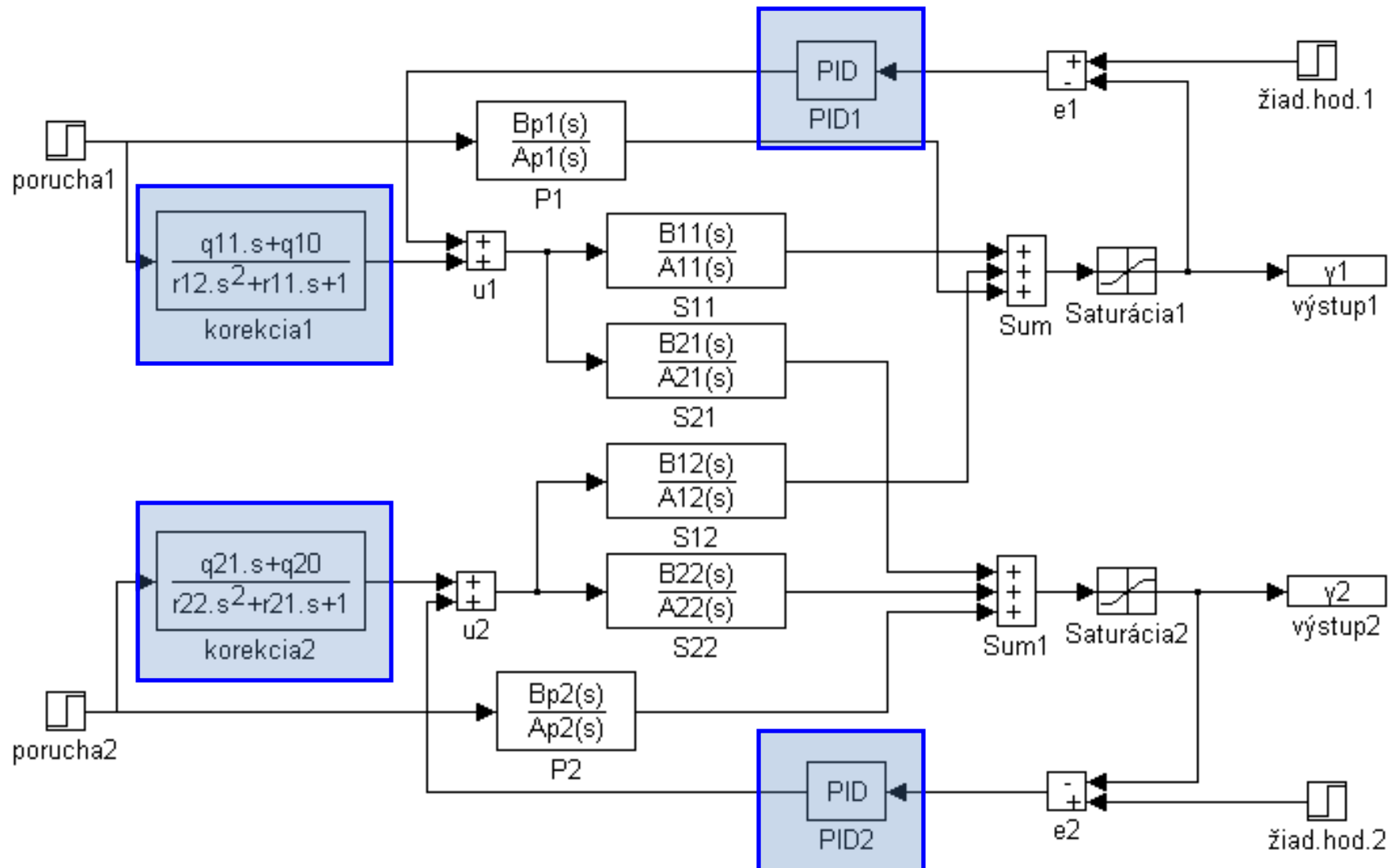
$$\alpha=1, \beta=0.5, \gamma=0$$

$$LC = \frac{a_1 z + a_0}{z + b_0}$$

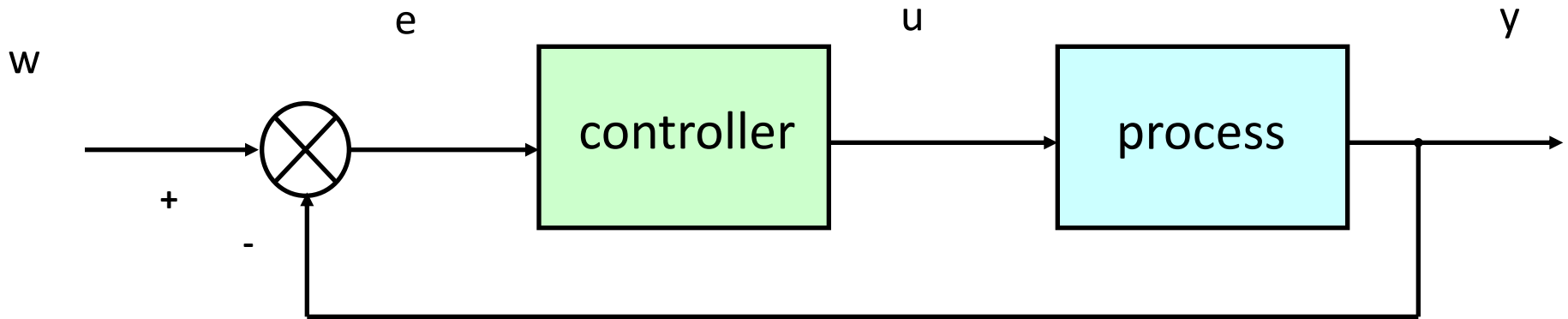


	P	I	D	a₁	a₂	B₂
Pole-placement	4737.4	99220	75.3882	20	-19.8425	-0.8425
GA	45100	4.8020	780.2274	1.5315	-1.4123	-0.4219

Návrh riadenia MIMO systému



Fix (known) structure of the controller



Chromosome: $ch = [p_1, p_2, \dots, p_i, \dots, p_n]$

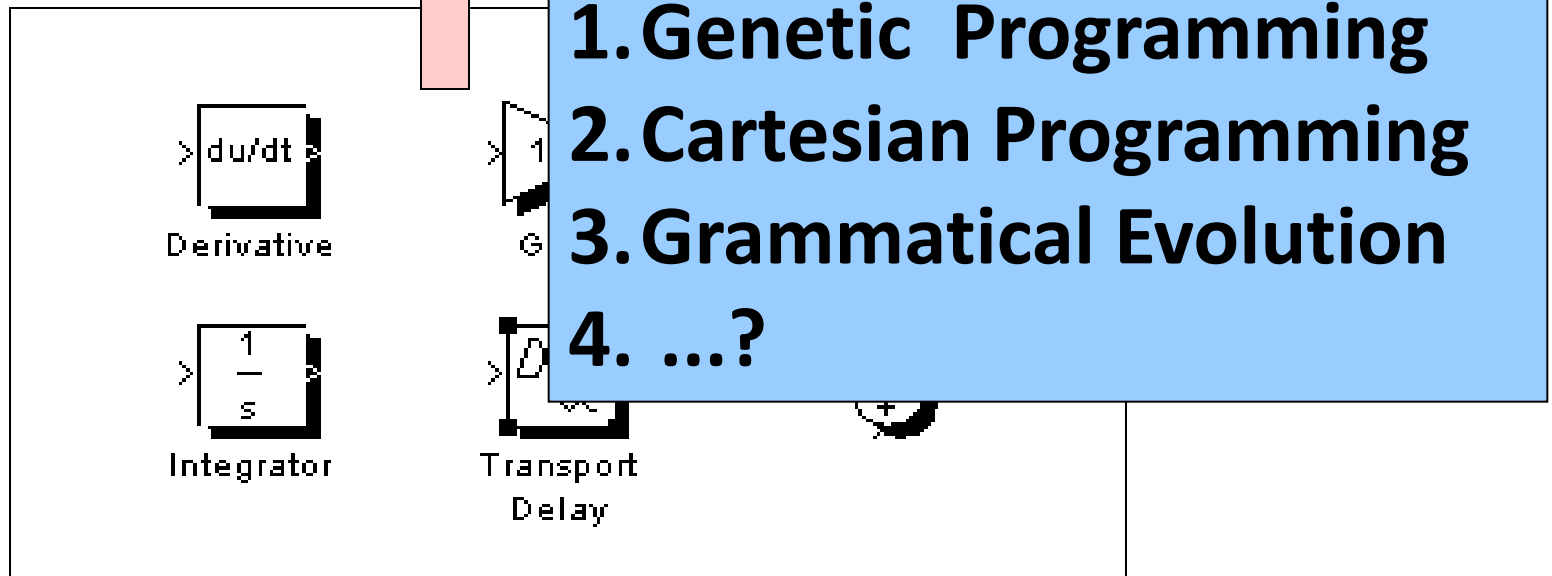
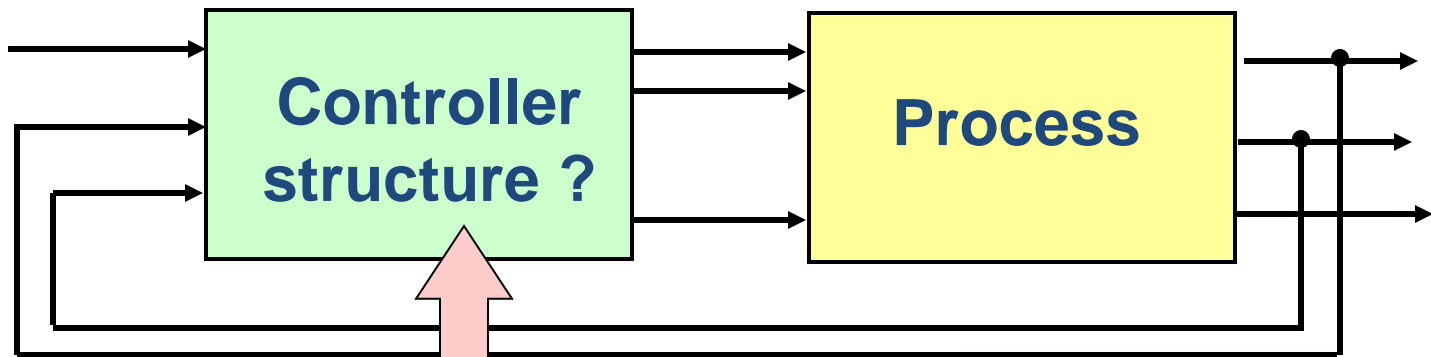
PID controller : C_{PI}

General linear cont

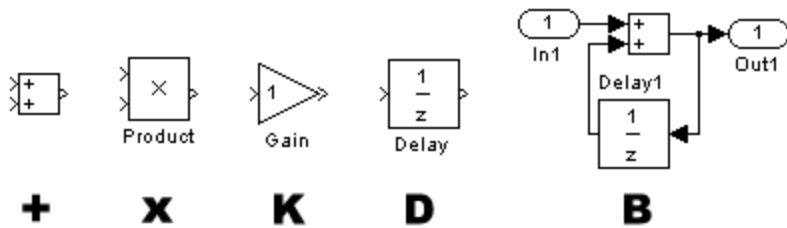
Fix number of

1. Genetic Algorithms (GA)
2. Differential Evolution
3. PSO
4. Artificial Immune System
5. Evolutionary Strategies
6. other numerical optimisers ...

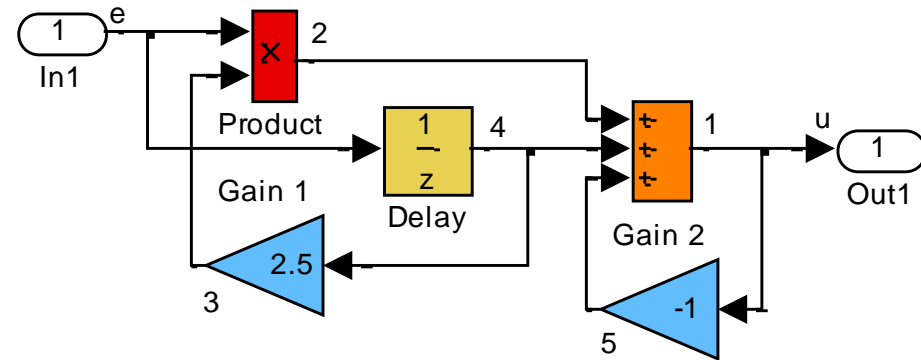
Unknown structure of the controller (unknown number of parameters)



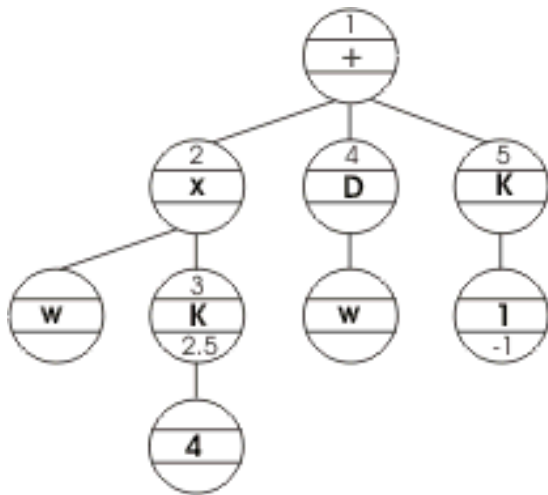
Evolúcia regulátora – Genetické programovanie



Stavebné prvky



Regulátor v Simulinku - fenotyp



Stromová reprezentácia

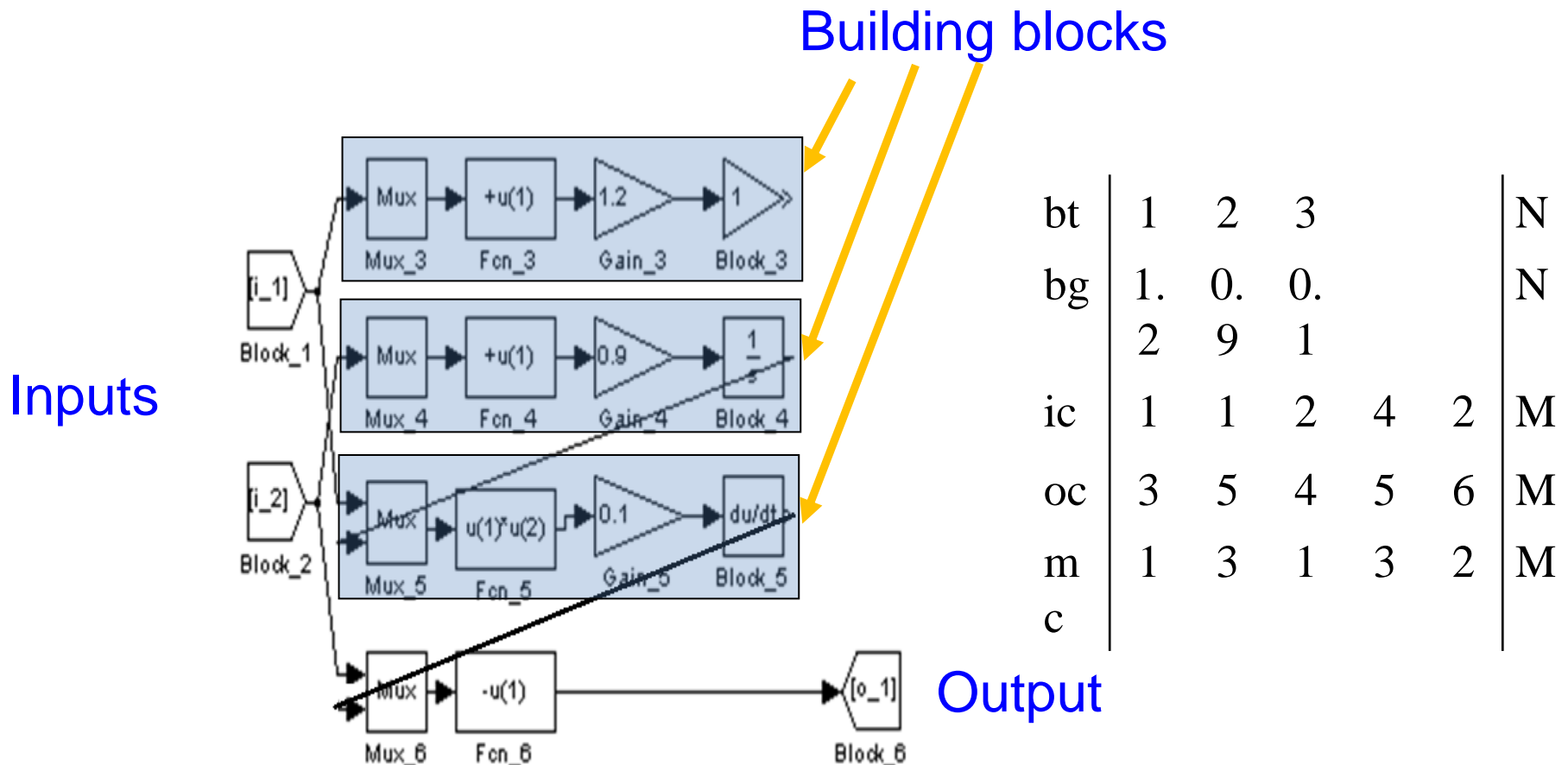
$$\begin{pmatrix}
 3 & 2 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
 + & x & w & K & 4 & D & w & K & 1 \\
 1 & 2 & & 3 & & 4 & & 5 & \\
 & & & 2.5 & & & & -1 &
 \end{pmatrix}$$

Tabulková reprezentácia - genotyp

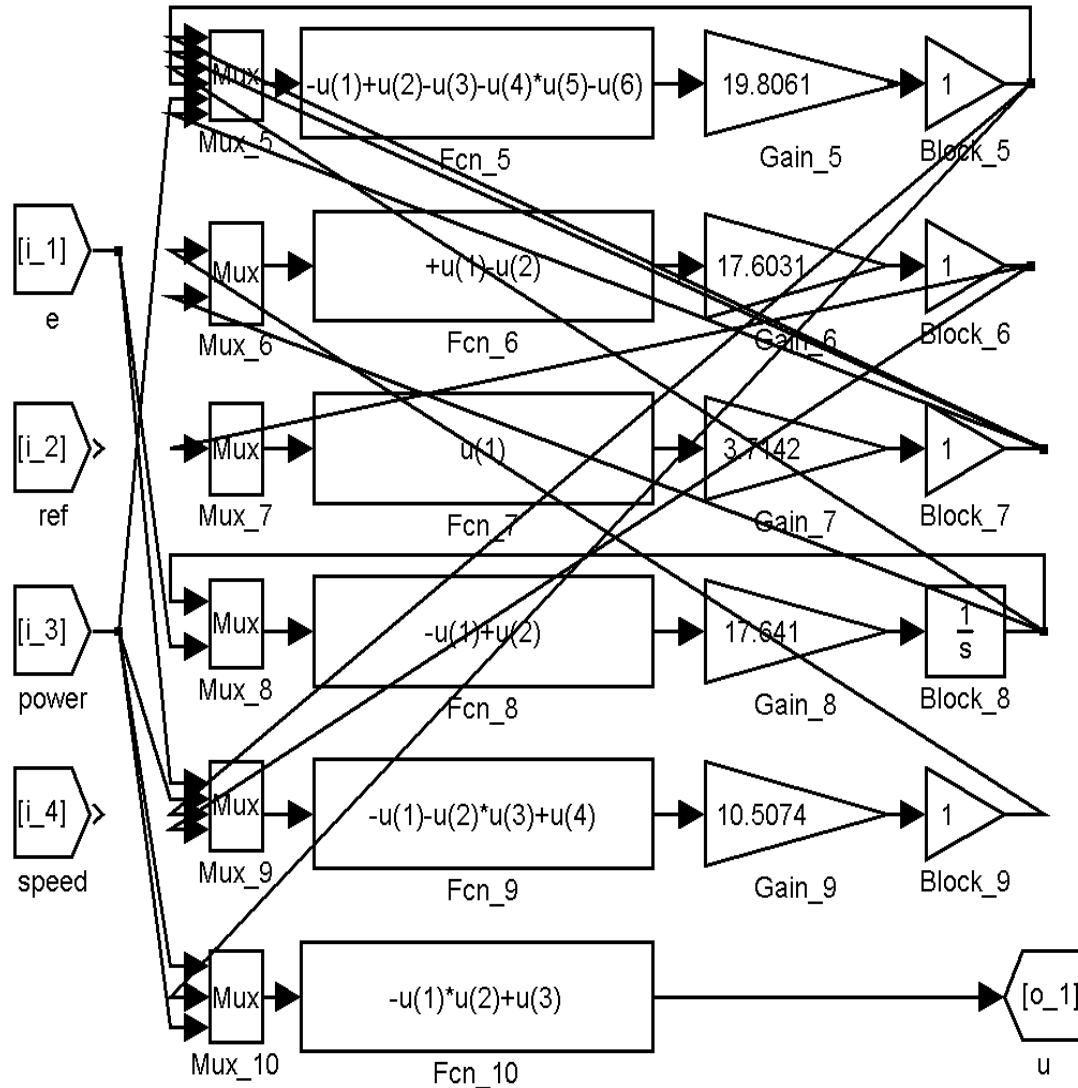
Controller design - Kartesian genetic programming

Individual coding:

[1 2 3 1.2 0.9 0.1 1 1 2 4 2 3 5 4 5 6 1 3 1 3 2]

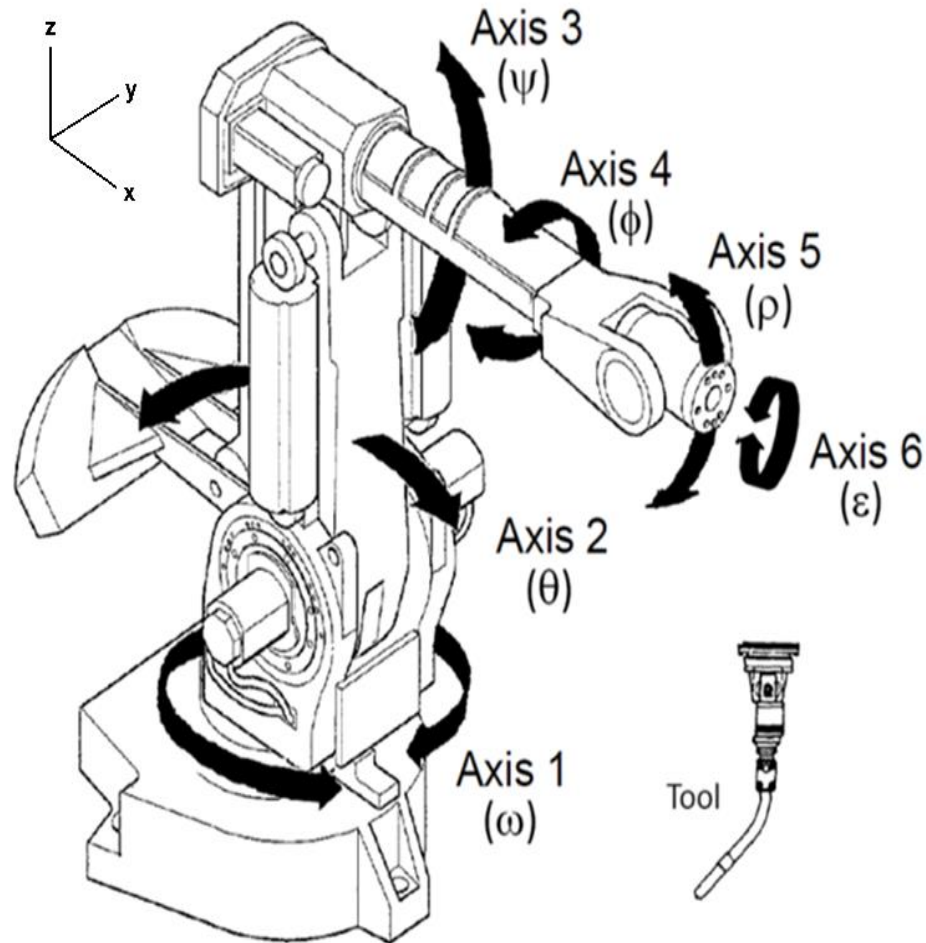


Hydro-turbine controller obtained after evolution



Optimalizácia trajektórie robotického ramena – príklad (Robot ABB IRB 6400)

http://www.youtube.com/watch?v=p2wCSyl_f6M



Presnosť polohovania

$$D_{tr} = \sum_{i=1}^N \sqrt{[x_{w,i} - x_{GA,i}]^2 + [y_{w,i} - y_{GA,i}]^2 + [z_{w,i} - z_{GA,i}]^2}$$

Ďalšie kritériá

Energia z bodu do bodu: $E_{p2p} = \sum_{i=1}^n [(\alpha_{b,i} - \alpha_{a,i})] * EP_{r,i}$

a) Energia celého pohybu: $E_{tr} = \sum_{j=1}^N \sum_{i=1}^n [(\alpha_{b,i,j} - \alpha_{a,i,j})] * EP_{r,i}$

$EP_{r,i}$ - energia i-teho kľbu na 1° rotácie

b) Čas operácie celého cyklu:

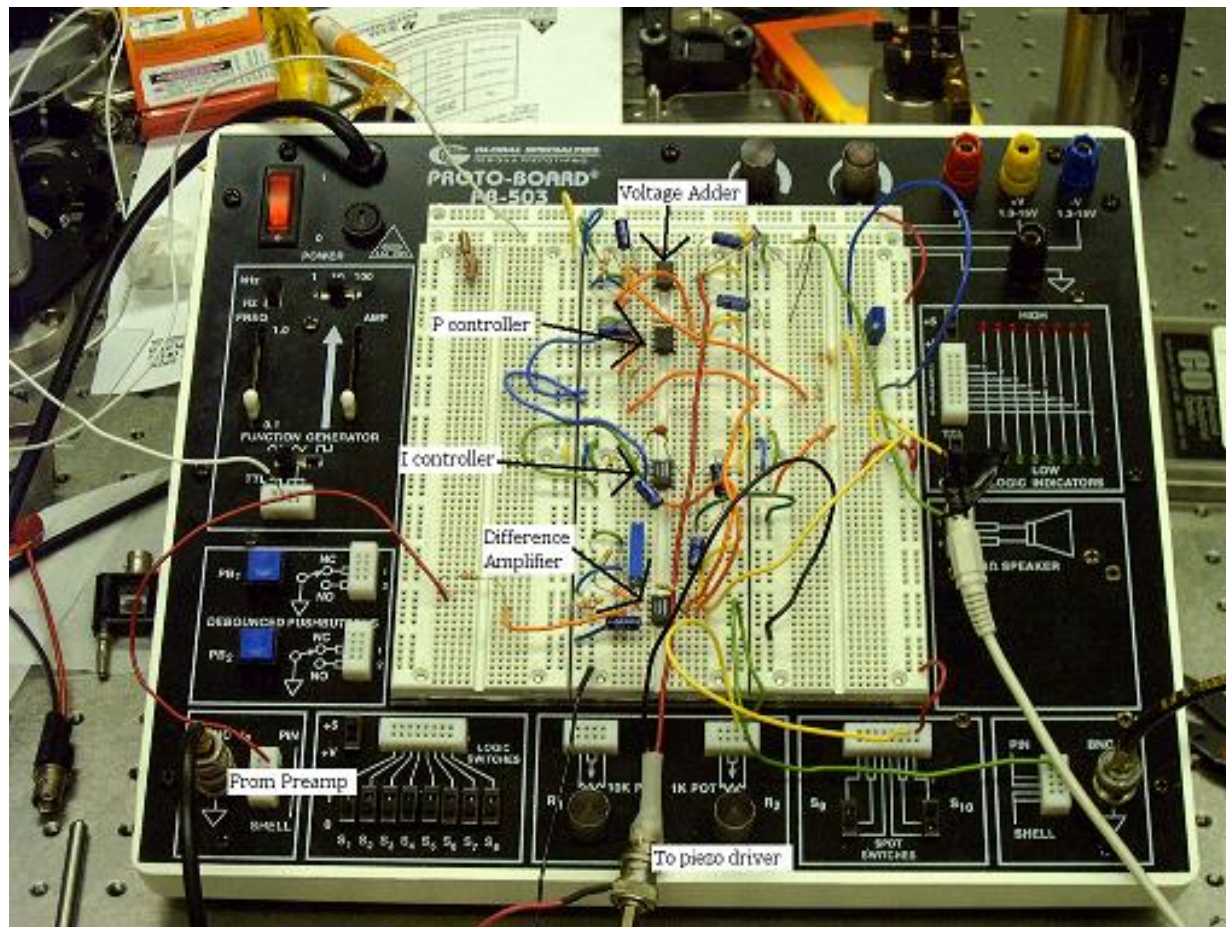
$$T_{tr} = \sum_{j=1}^N \max_{i=1..n} [(\alpha_{b,i,j} - \alpha_{a,i,j})] * TP_{r,i}$$

$TP_{r,i}$ je čas pohybu i-teho kľbu na 1° rotácie

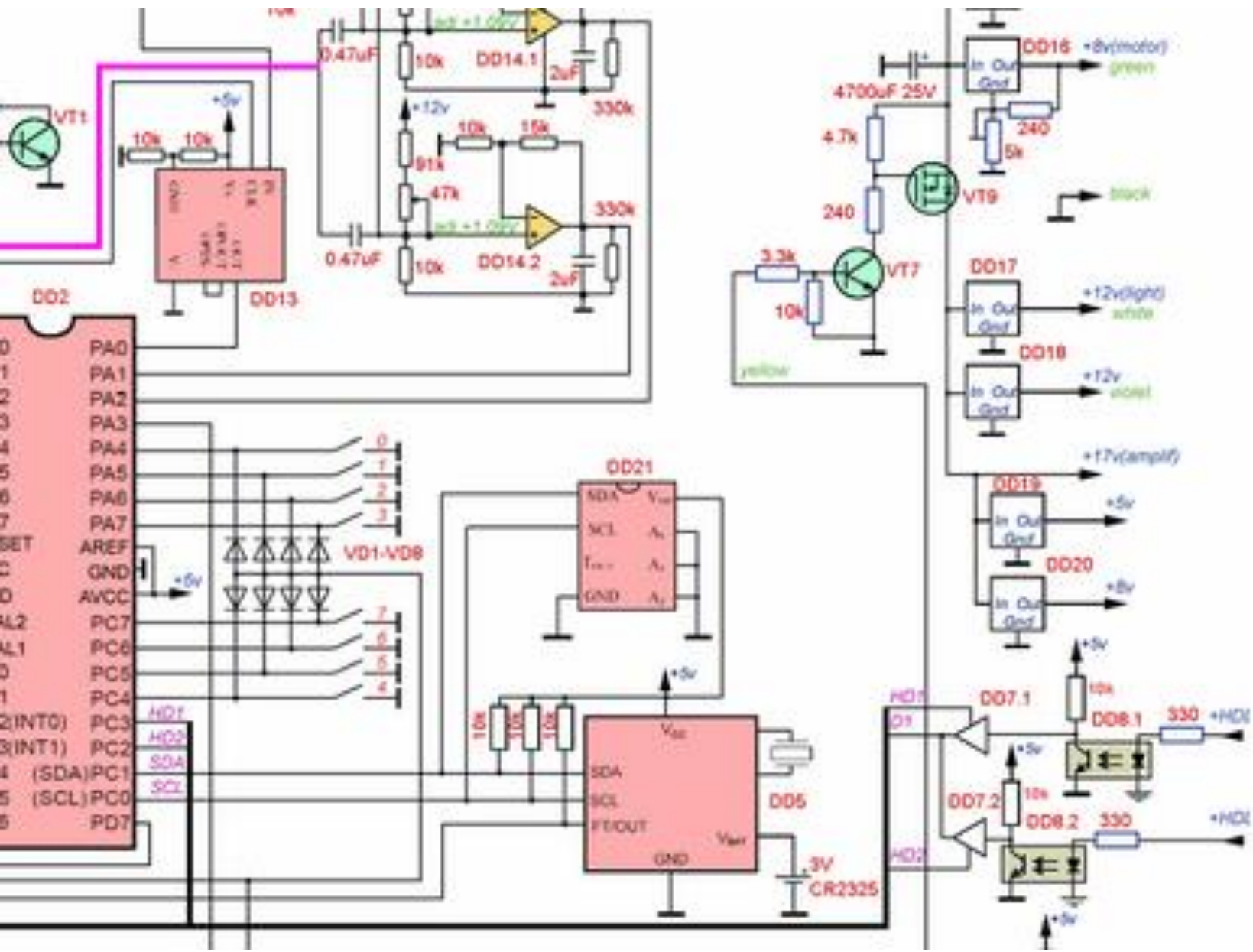
c) Sumárna rotácia všetkých kľbov:

$$A_{tr} = \sum_{j=1}^N \sum_{i=1}^n (\alpha_{b,i,j} - \alpha_{a,i,j})$$

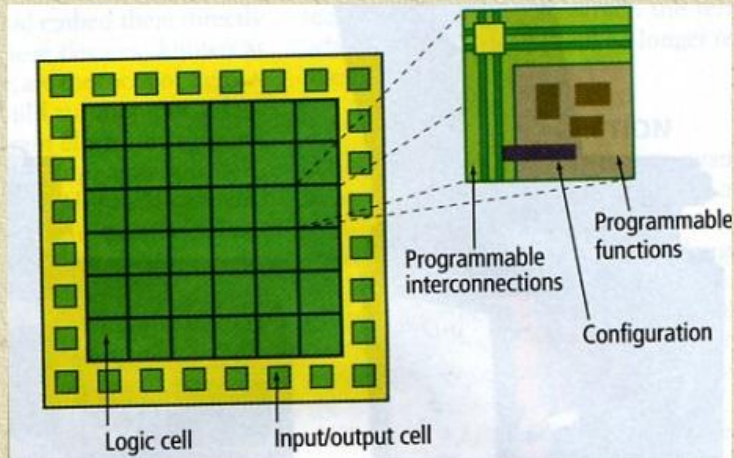
Evolučný hardvér (Evolvable Hardware)



Evolúcia elektrických obvodov



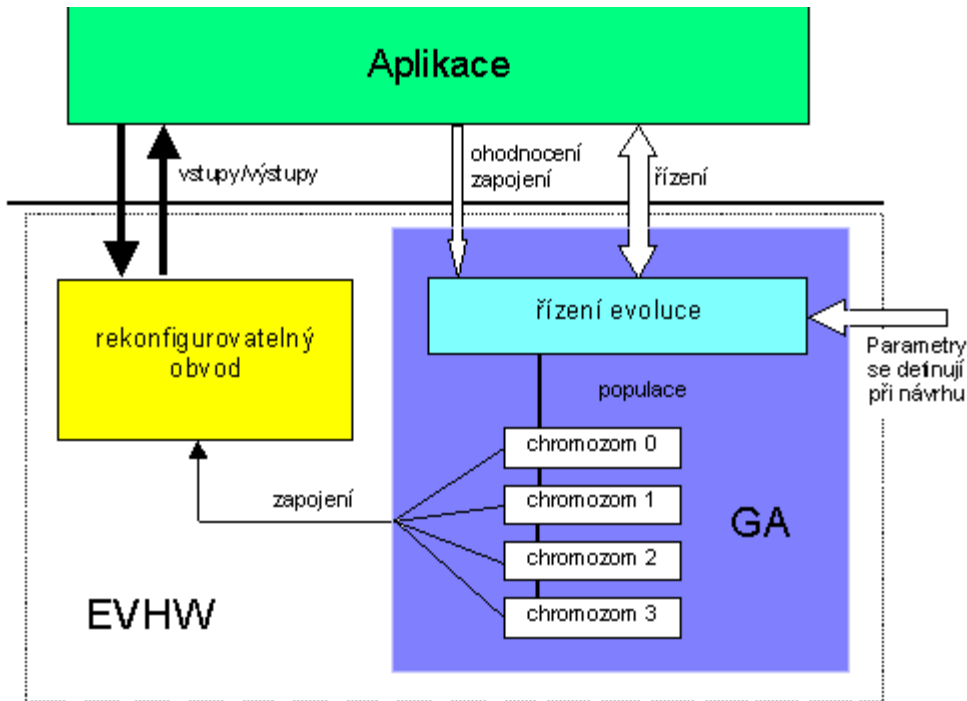
FPGA:



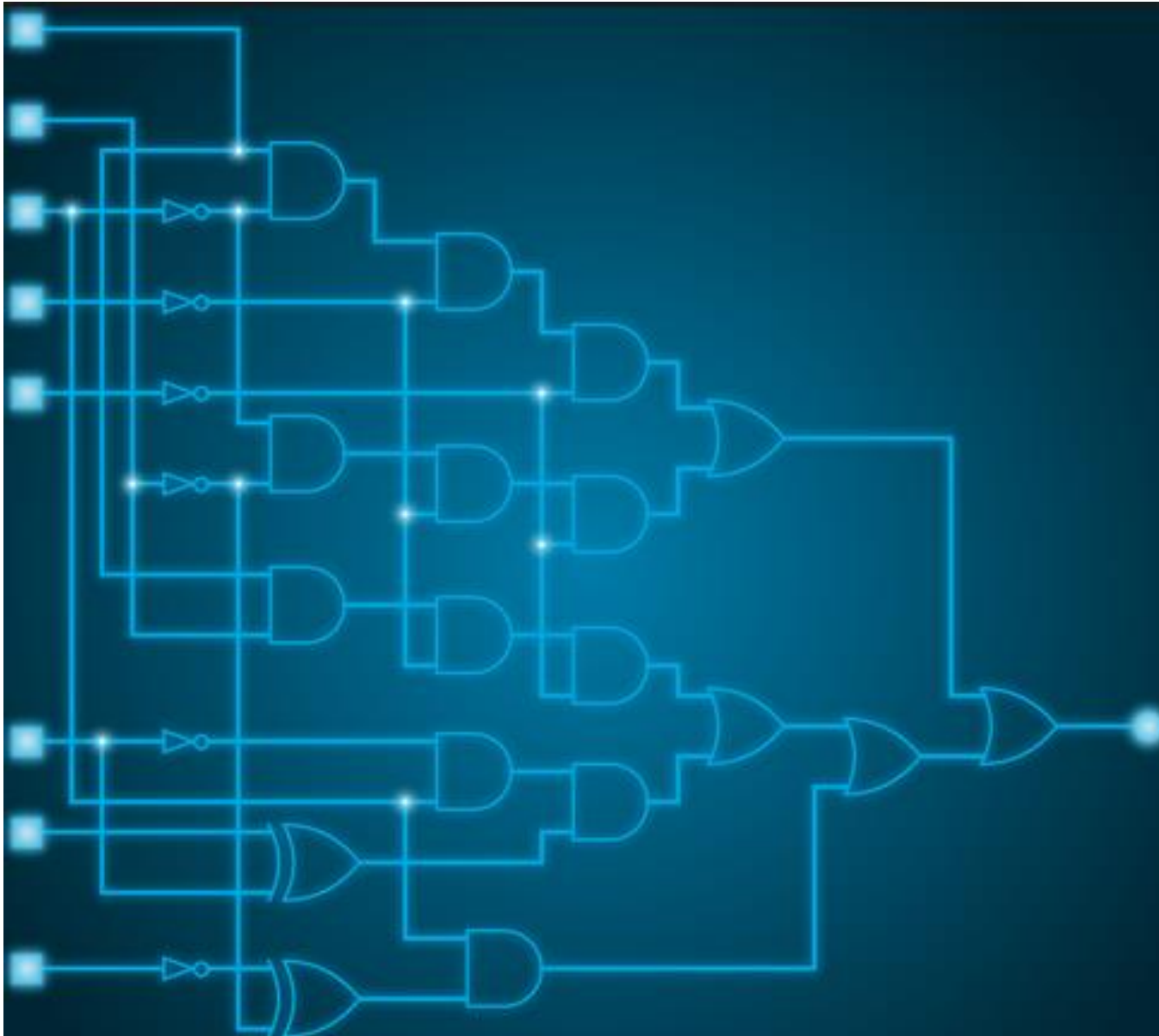
Vyhodnotenie fitness:

- simulačný softvér elt. obvodov
- rekonfigurovatelné elt. obvody

(číslicové/analógové, logické obvody, hradlové polia, polia operač.zosil. atď...)



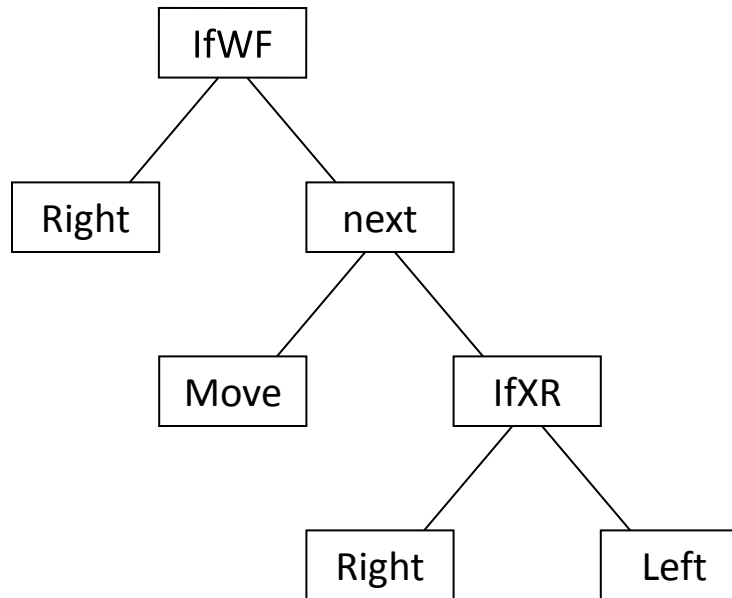
Evolúcia logických obvodov



Návrh (riadiaceho) programu – genetické programovanie

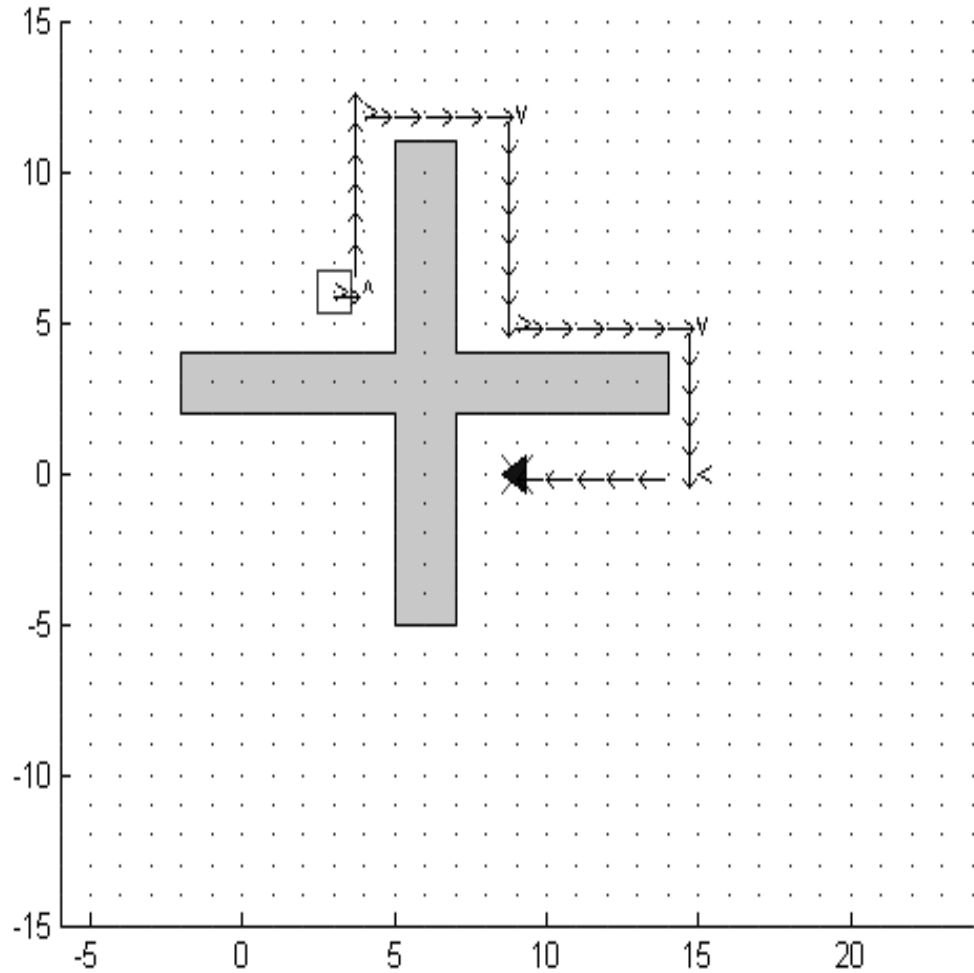
stromová reprezentácia jedinca

```
IfWF  
  Right  
else  
  Move  
  IfXR  
    Right  
  else  
    Left  
  end  
end  
end
```

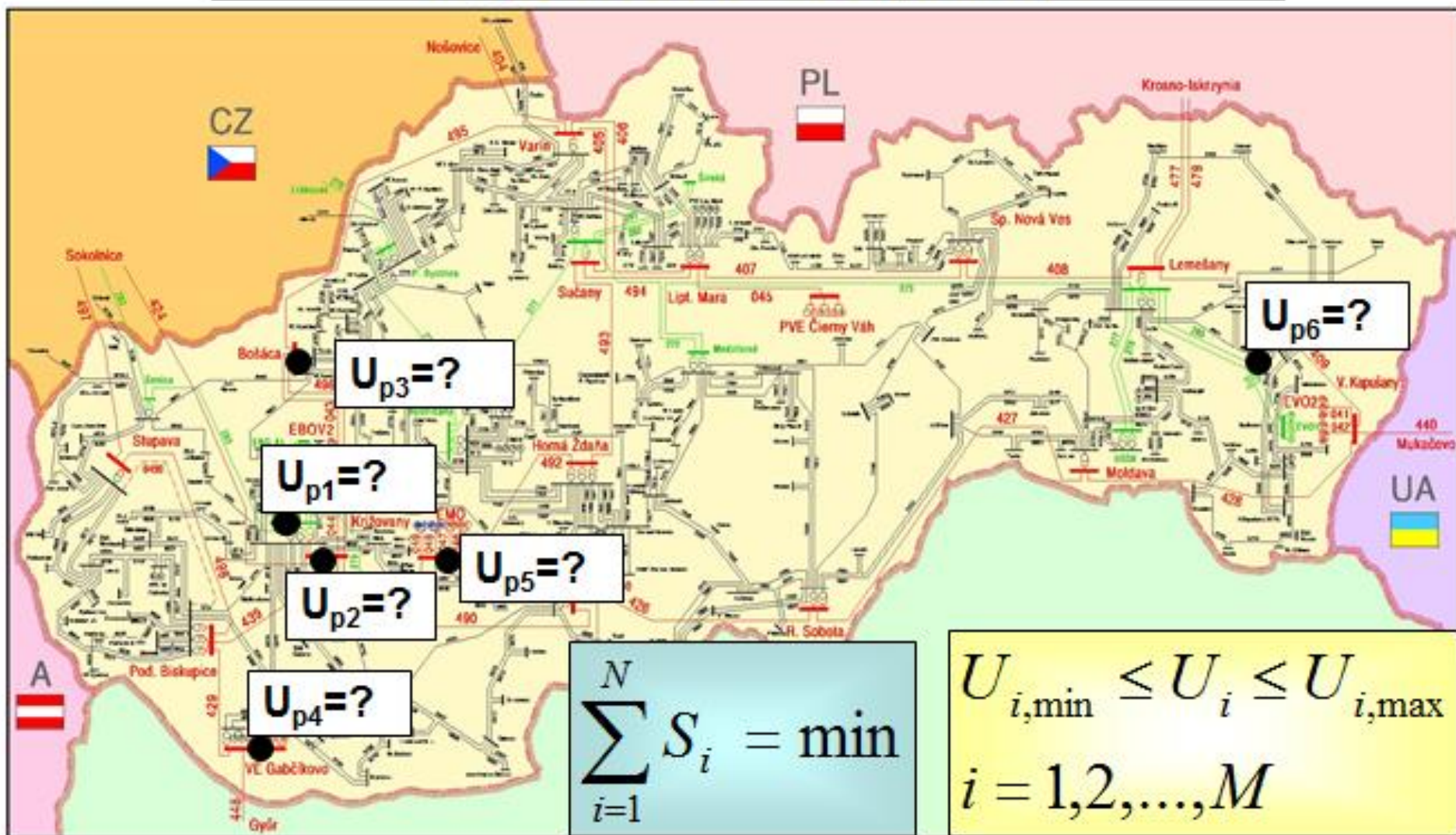


Výsledná trajektória robota

```
IfWF
  Right
    IfXF
      IfWL
        IfWF
          Left
        else
          IfWF
            Move
            Left
          else
            Move
          end
          Left
        end
      else
        Move
        Right
      end
    else
      Move
    ...
```



Optimalizácia ustáleného stavu napätí v ES SR (TRN)

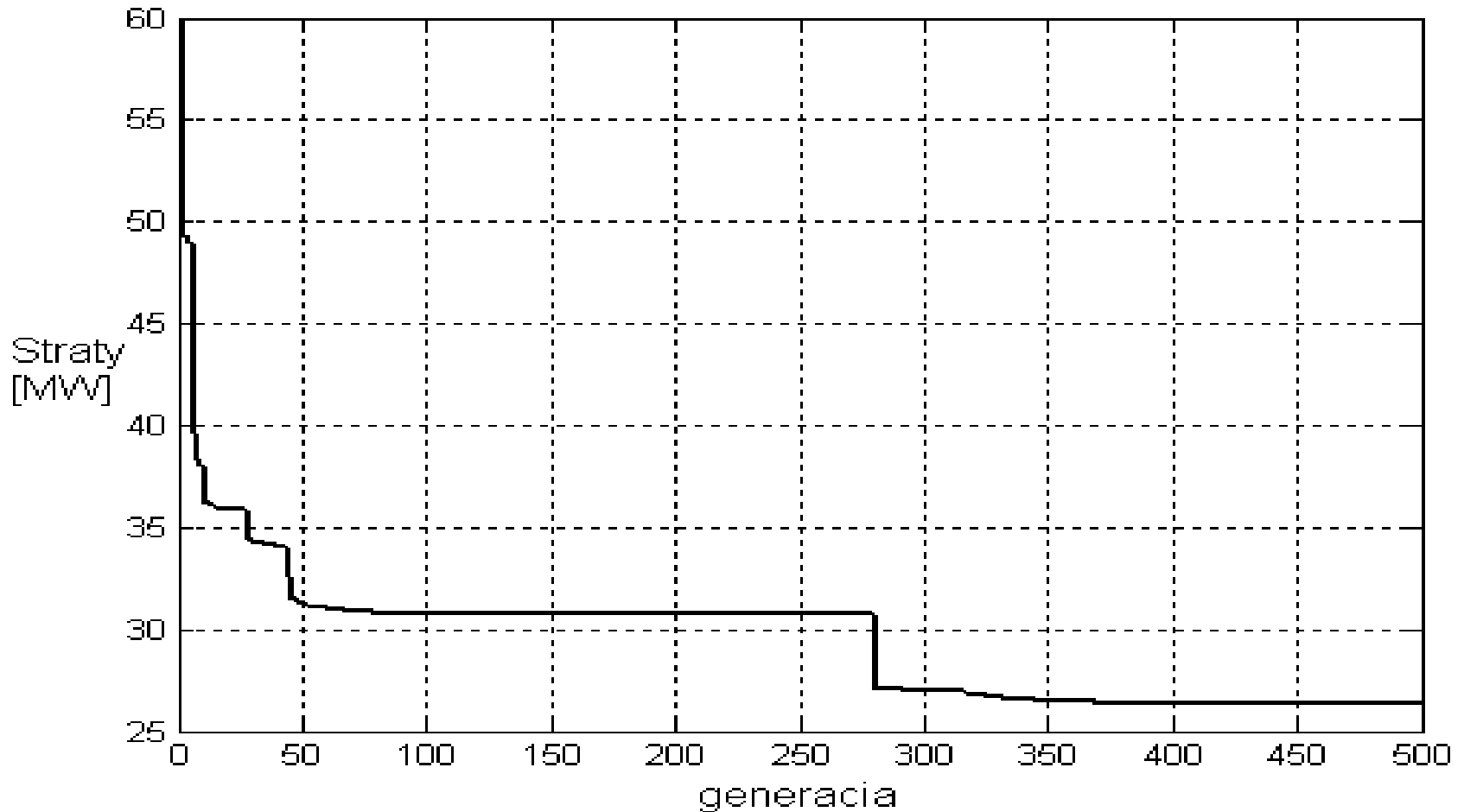


$$\sum_{i=1}^N S_i = \min$$

$$U_{i,\min} \leq U_i \leq U_{i,\max}$$

$$i = 1, 2, \dots, M$$

S – činné straty, N – počet vedení, M – počet uzlov siete

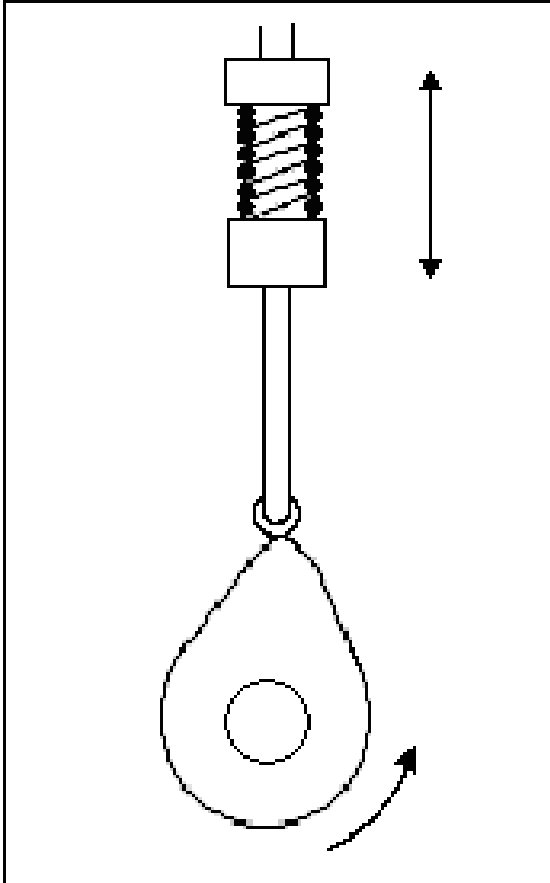


Počes výpočtov ustáleného stavu ES SR (fitnes)
počas riešenia GA: $500 \cdot 30 = 15\,000$
(*počet generácií x veľkosť populácie*)

Zníženie strát oproti skutočnému stavu o 2,42 %

Konštrukčné úlohy

Návrh tvaru vačky otáčavého stroja (GA)

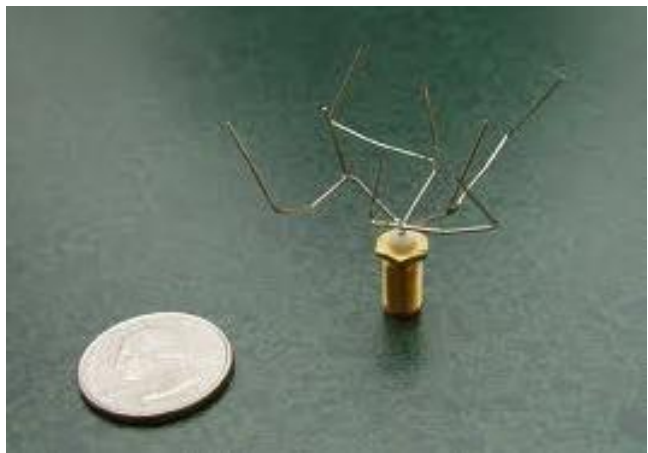


Reťazec: tvar a rozmery vačky zakódované pomocou parametrov B-spline funkcie

Účelová funkcia: simulácia a vyhodnotenie dynamických a statických mechanických a tepelných vlastností časti stroja

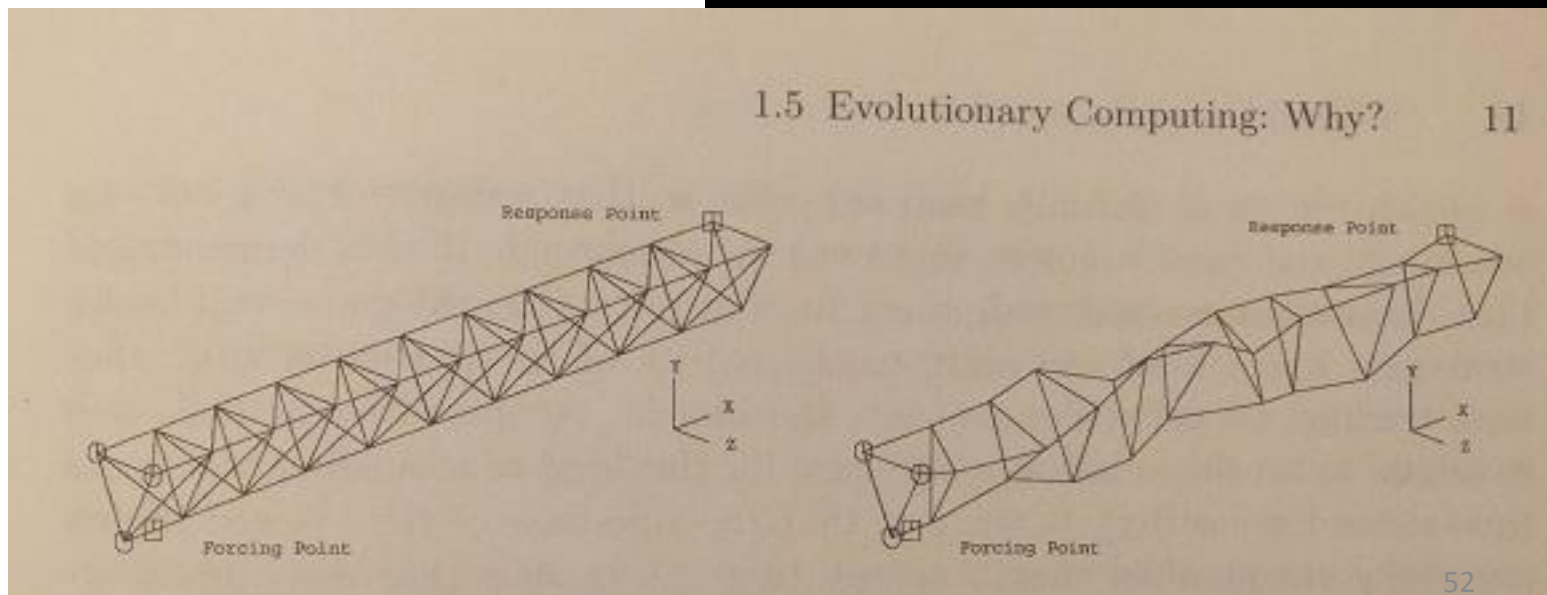
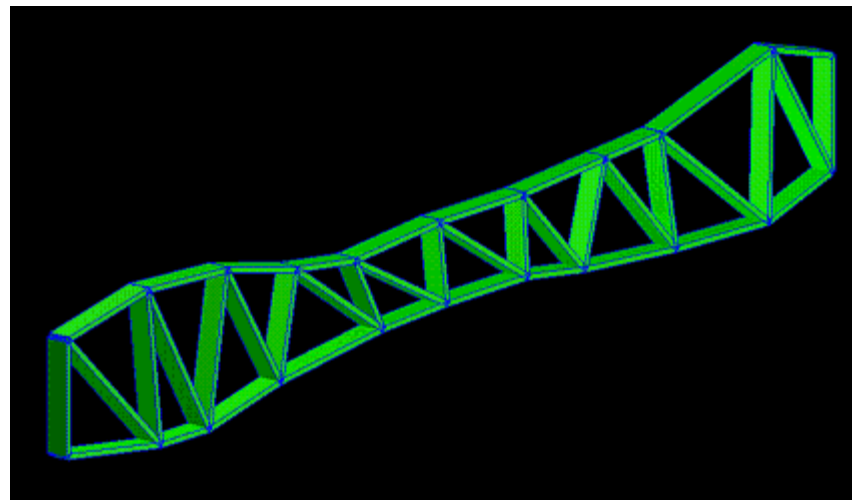


Rôzne konštrukčné aplikácie



Návrh antény satelitu

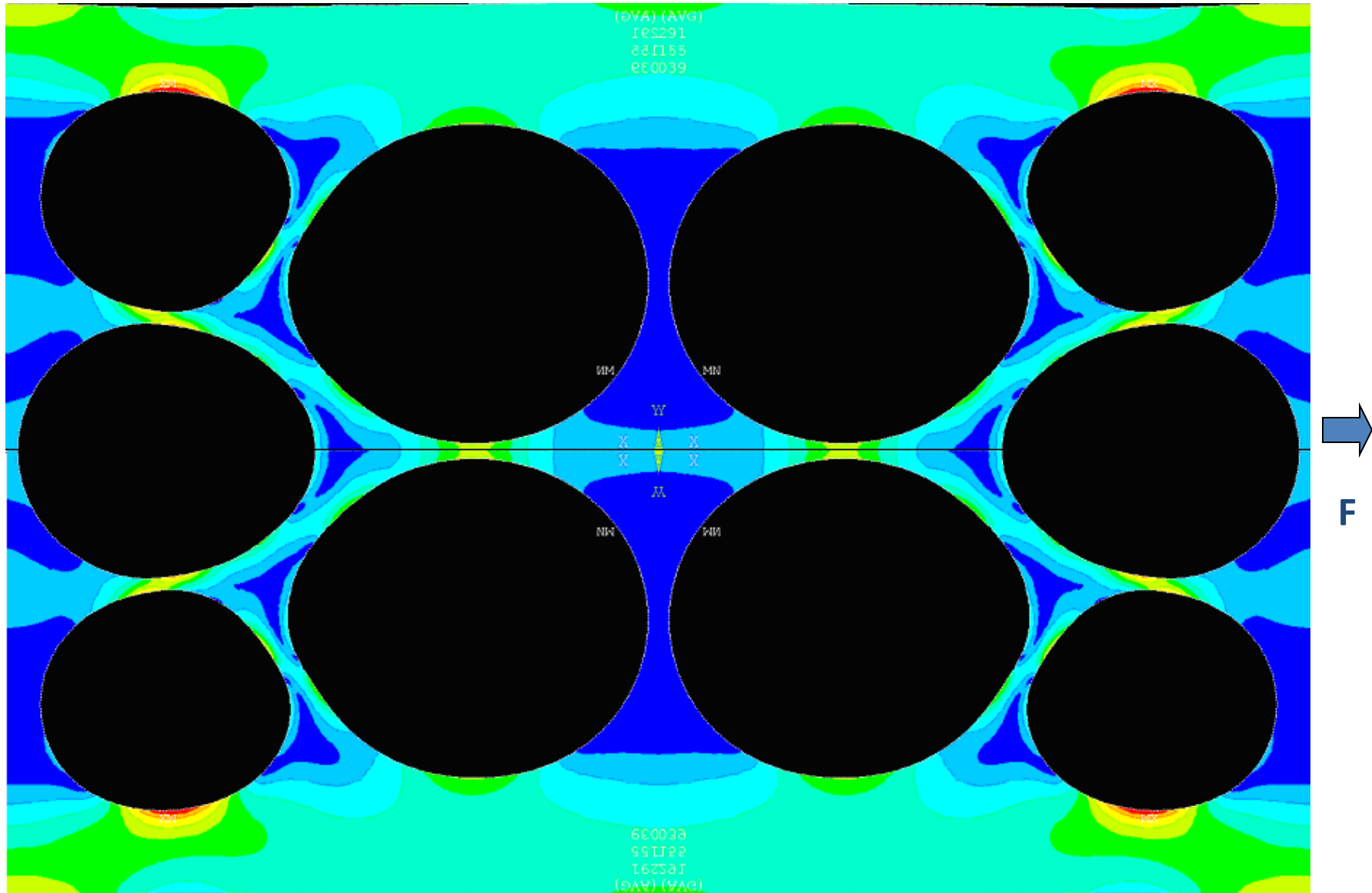
Mechanické konštrukcie



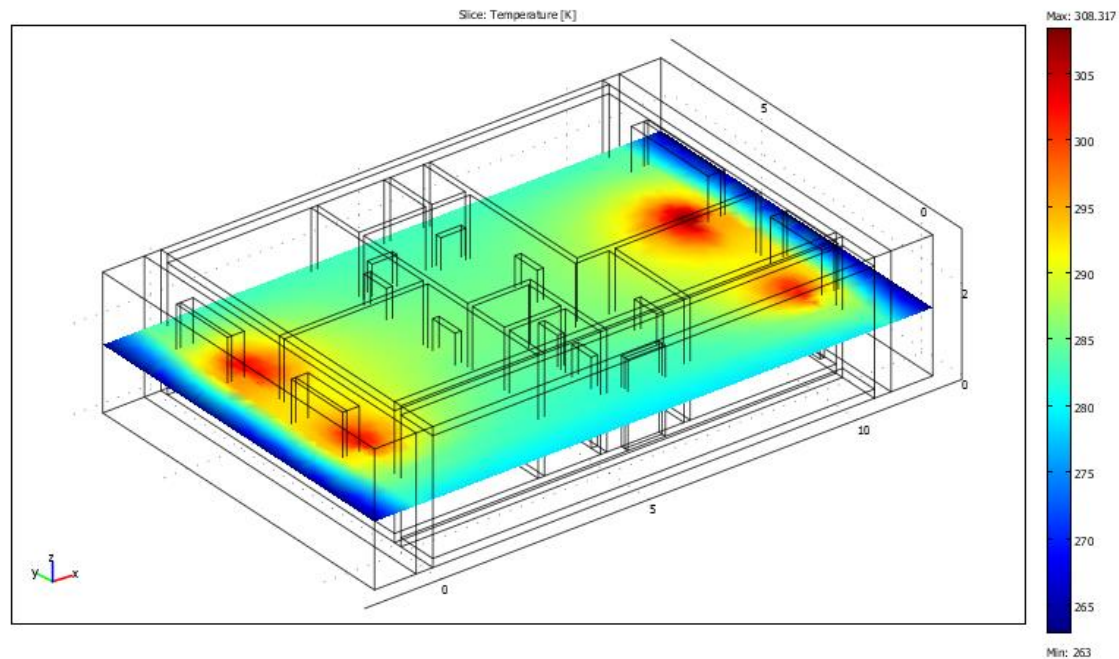
Spojenie EA s modelom na báze metódy konečných prvkov (MKP)

(SW: Comsol, Ansys ...)

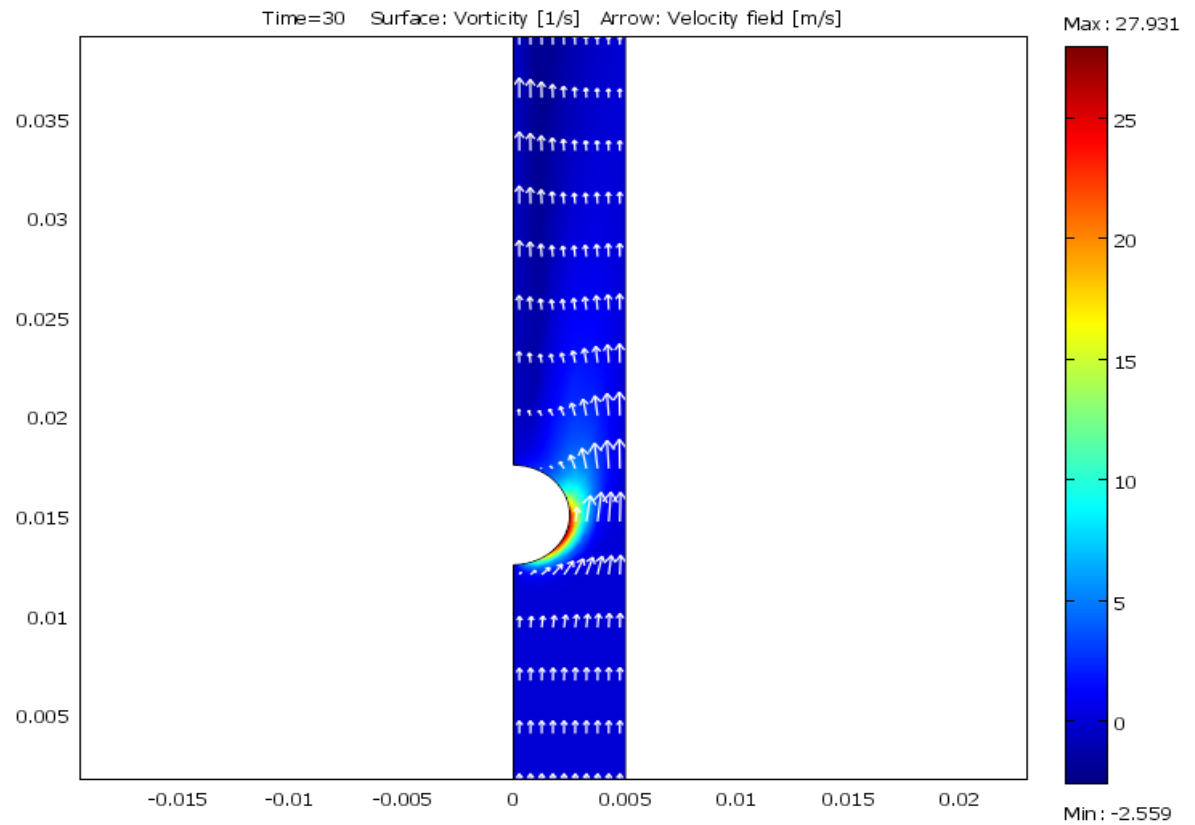
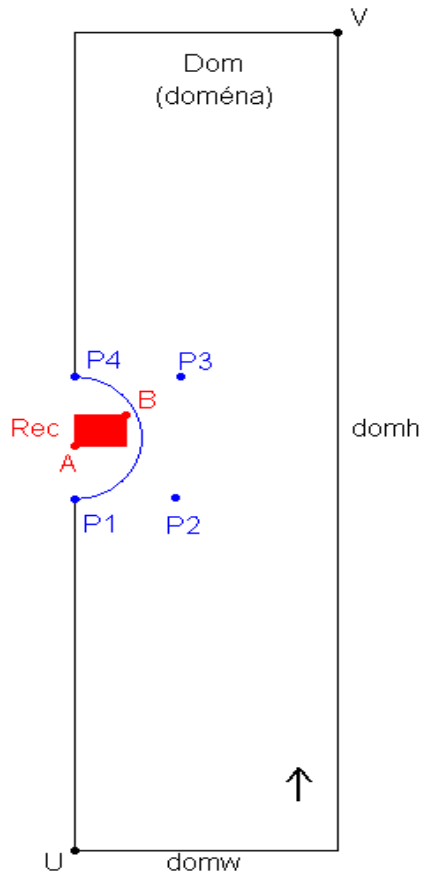
Optimalizácia platne namáhanej silou (GA) (SW: ANSYS)



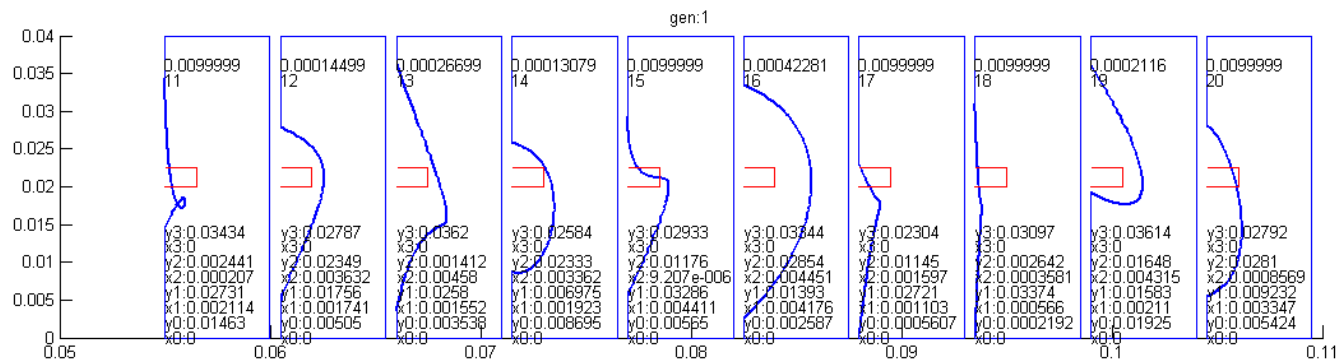
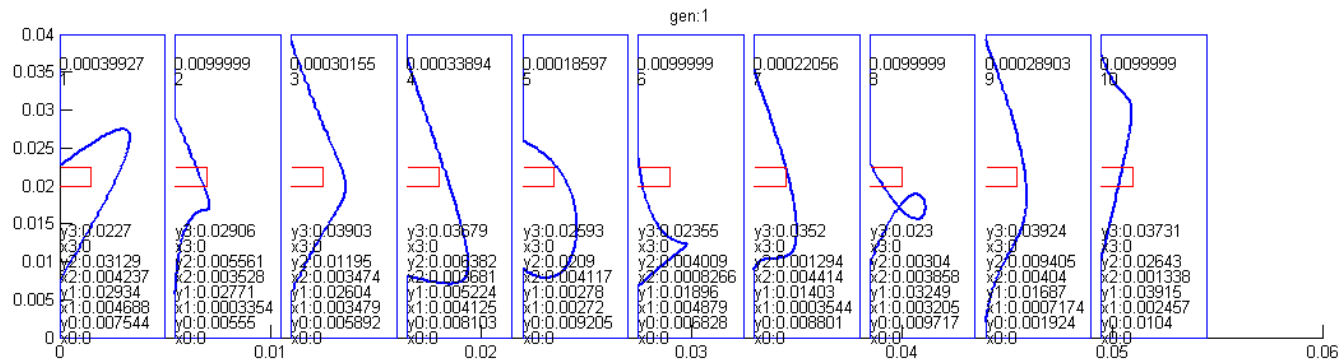
Optimalizácia distribúcie teploty v byte (SW: Comsol)



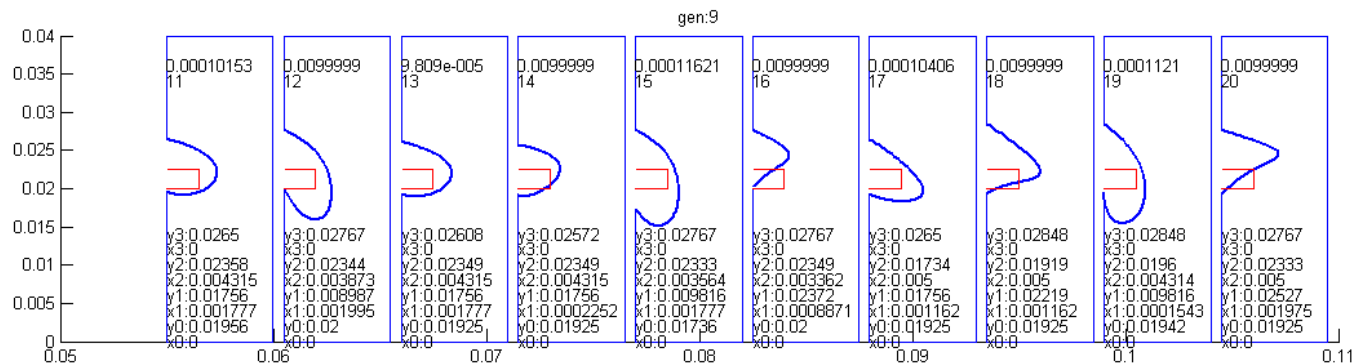
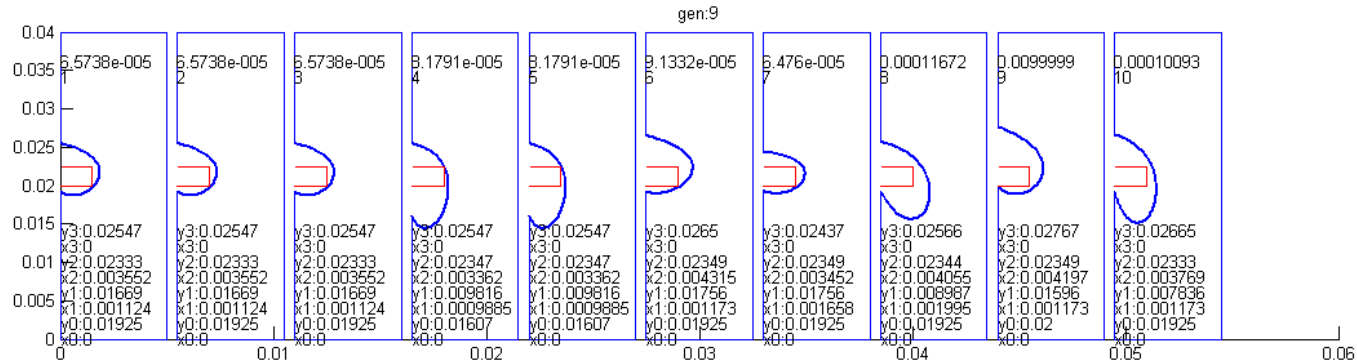
Optimalizácia tvaru prekážky v potrubí pretekanom tekutinou



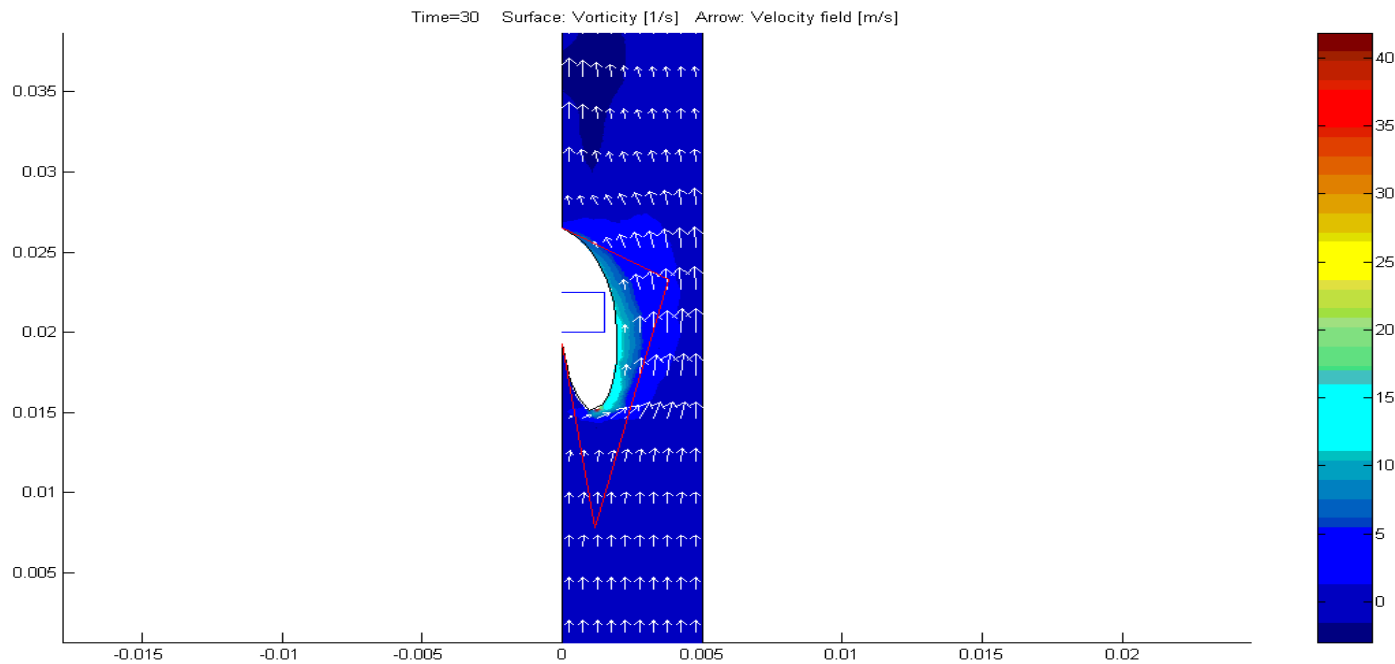
Populácia v 1. generácii výpočtu GA



Populácia v 10. generácii výpočtu GA



výsledok

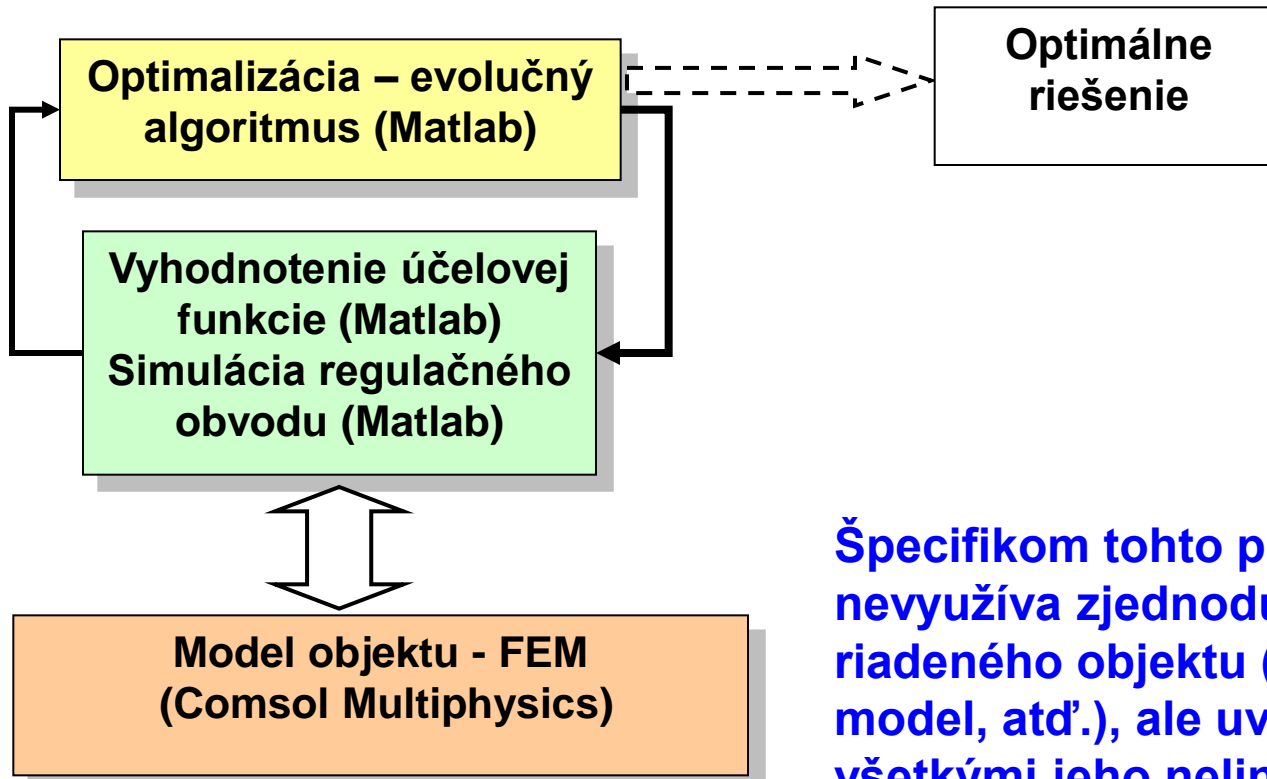


Účelová funkcia reprezentuje mieru turbulencií v potrubí, ktoré sa minimalizujú

Aplikačné oblasti MKP

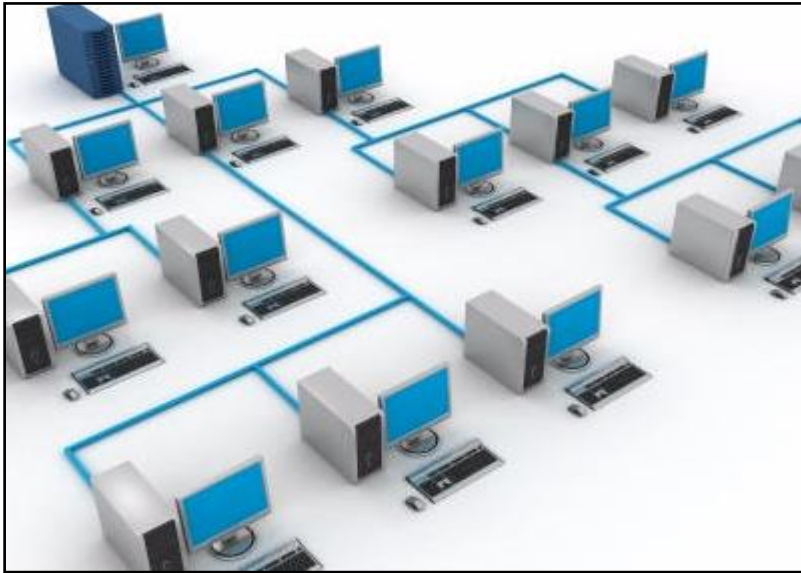
- **elektromagnetické polia, elektrické, magnetické polia**
- **сила, pružnosť, pevnosť, mechanické aplikácie**
- **šírenie tepla**
- **aerodynamika, hydrodynamika**
- **chémia ...**
- **riadenie v hore uvedených doménach**
- **a iné ...**

Princíp riešenia v Matlabe/Comsole



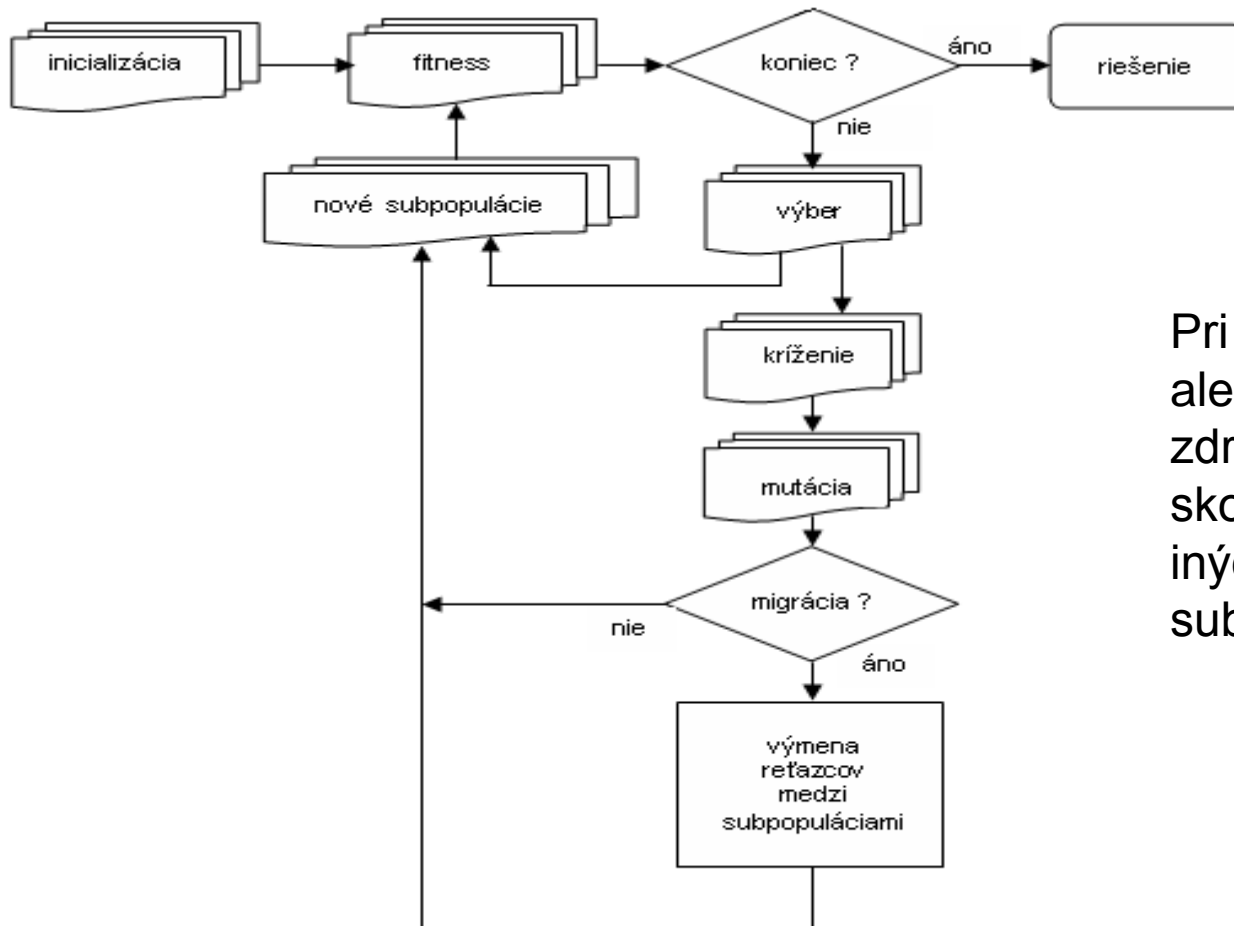
Špecifikom tohto prístupu je, že nevyužíva zjednodušenú náhradu riadeného objektu (napr. linearizovaný model, atď.), ale uvažuje model so všetkými jeho nelinearitami, okrajovými podmienkami a ohraničeniami.

Paralelizácia GA / BIOM



Paralelný GA

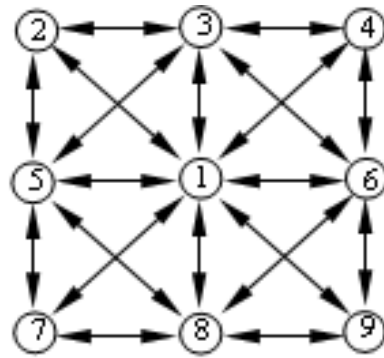
- Riešenie prebieha v jednotlivých ostrovoch (subpopuláciách) istý čas izolovane.
- Vo vhodných okamihoch nastáva výmena informácia medzi ostrovmi – migrácia.



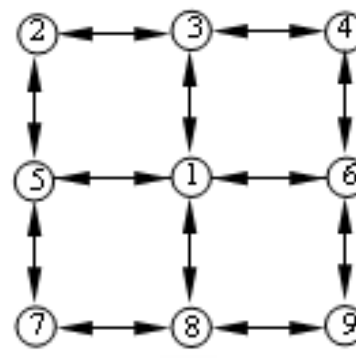
Pri migrácii sa jeden alebo viac jedincov zo zdrojovej subpopulácie skopíruje a nahradí iných jedincov v cieľovej subpopulácii.

Rôzne topológie paralel-GA

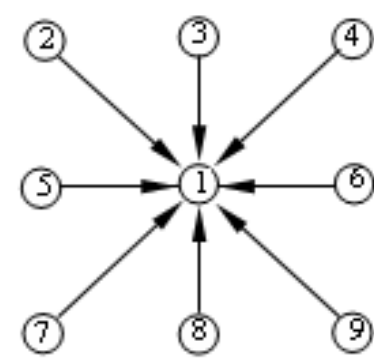
- Migrácie prebiehajú v smere šípok v definovaných okamihoch
- Okamihy migrácie a voľba a počet migrujúcich jedincov sú rôzne
- V jednotlivých ostrovoch môžu byť rovnaké/rôzne algoritmy a ich parametre (typ, selektívny tlak, diverzita)



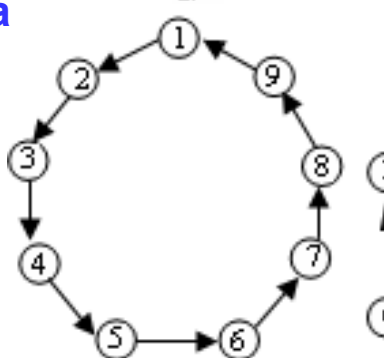
a



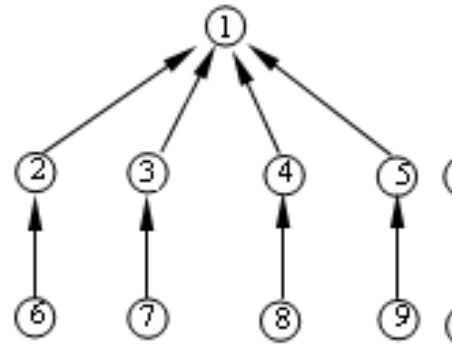
b



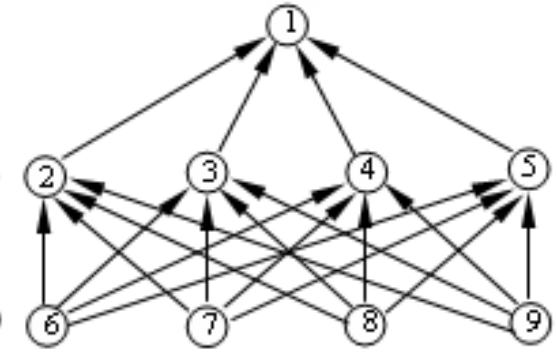
c



d



e



f

a podobné ...

1 uzel = 1 subpopulácia

Matlab - parallel computing

Parallel Computing Toolbox

- Umožňuje paralelizovať výpočty v Matlabe
- využívame 2 módy:
 - 1) parfor - paralelný for cyklus
 - 2) spmd – Single Program Multiple Data

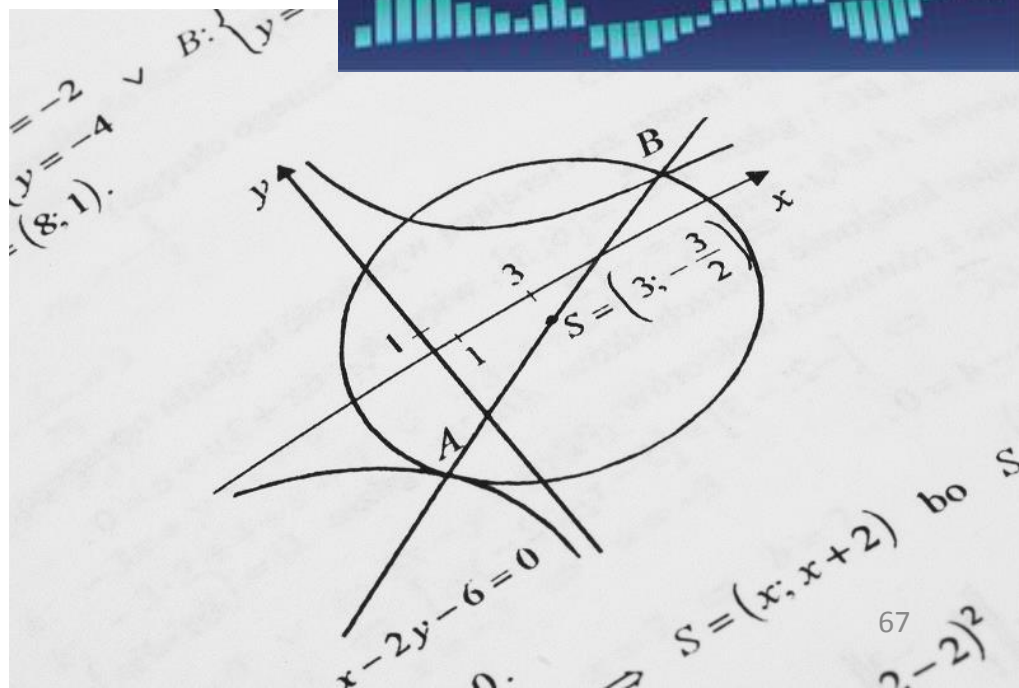
Matlab Distributed Computing Server

- Lokálny režim paralelizácie: do 12 výpočtových uzlov (max. 12 jadier na 1 procesore – 1 PC). Dokáže rozdeliť úlohu v Matlabe na viac jadier na 1 PC bez inštalácie MDCS.
- Pokiaľ chceme využiť viac PC, je potrebné inštalovať MDCS

Iné aplikačné domény BIOM

Matematika

- extremalizácia funkcií
- kombinatorika
- grafové úlohy
- regresná analýza
- teória hier, optimálne stratégie
- a veľa iného ...



Ekonomía a finančníctvo

- optimalizácia výrobného sortimentu
- plánovanie výroby
- dopravné úlohy
- alokácia investícií
- optimalizácia finančných operácií
- a veľa iného ...



Technika

- **konštrukčné úlohy, stavby**
- **strojové učenie**
- **optim. chemických reakcií**
- **evolučný hardvér**
- **a veľa iného ...**

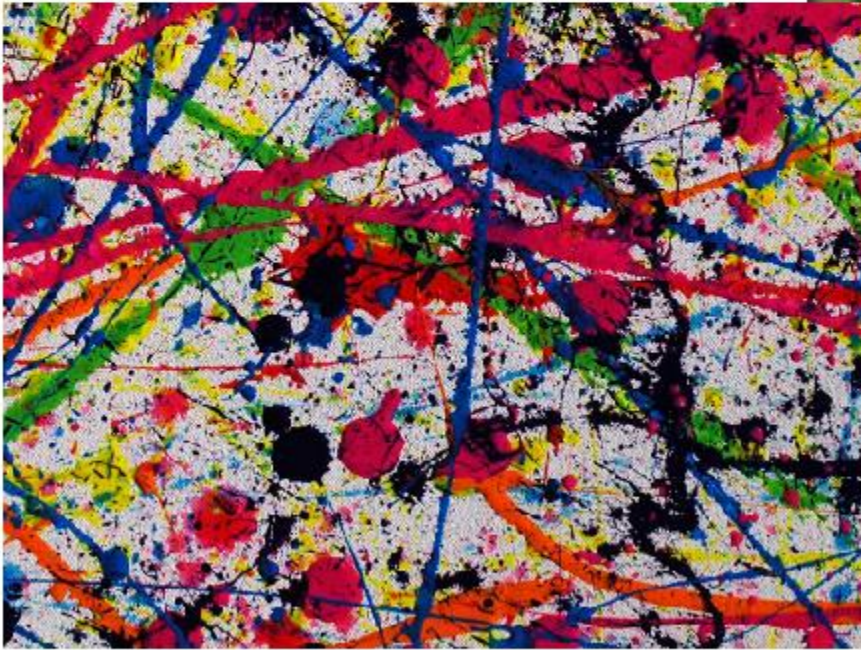
Informatika a komunikácia

- optim. spojenia v poč./telekom. sieti
- de / šifrovanie
- automatizované programovanie
- strojové učenie
- ...
- ?



Umenie (Design)

- „evolúcia“ výtvarných diel
- evolúcia hudby
- ... ?



Evolutionary Art ...

Zhrnutie - BIOM

- **veľmi efektívne/výkonné metódy v porovnaní s bežnými optimalizačnými prístupmi**
- **širokospektrálne, univerzálne použiteľné**
- **jednoduché použitie z hľadiska používateľa – prenášajú ťažisko riešenia problému z človeka na počítač**

Koniec

Ivan Sekaj

***Ústav robotiky a kybernetiky
FEI STU Bratislava***

E-mail: ivan.sekaj@stuba.sk