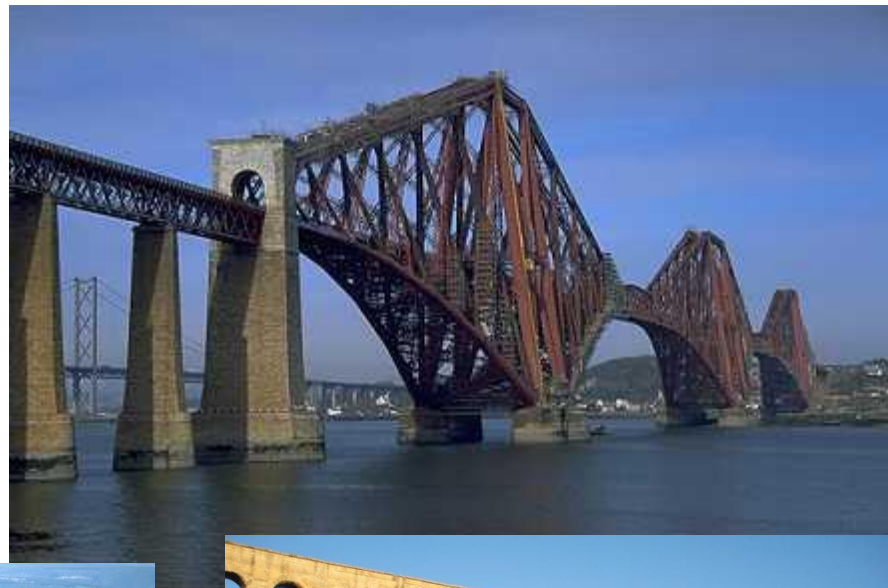
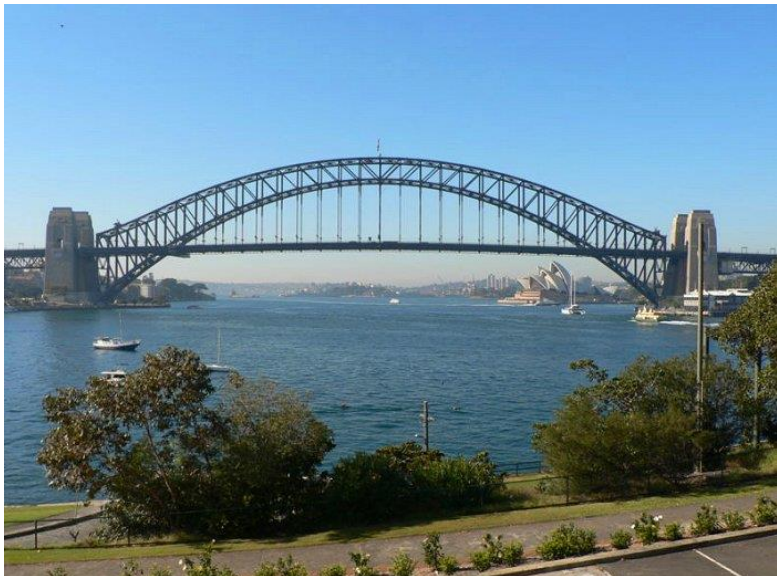
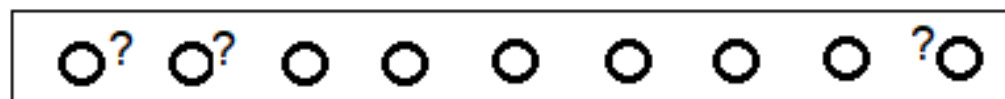


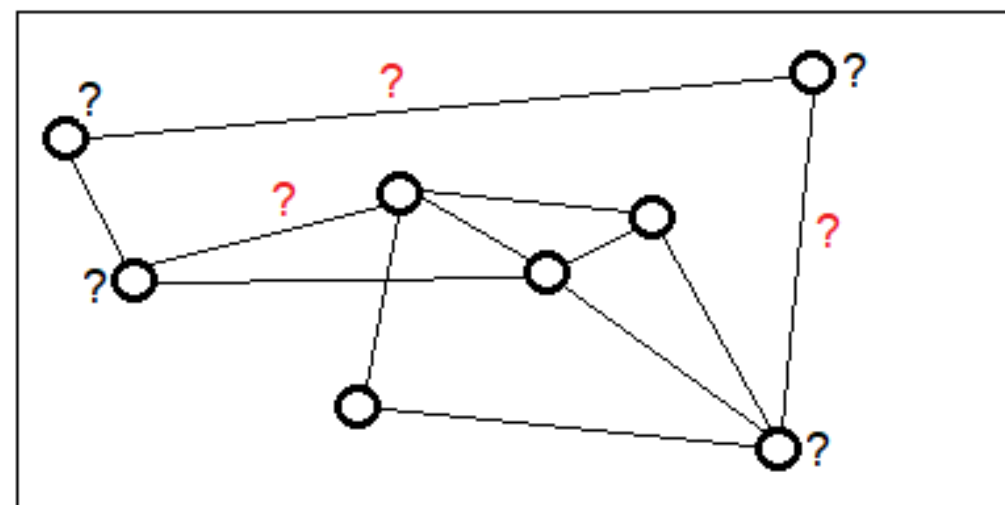
# **1.9 Genetické programovanie (GP)**

# Neznáma štruktúra aj neznáme parametre hľadaného objektu

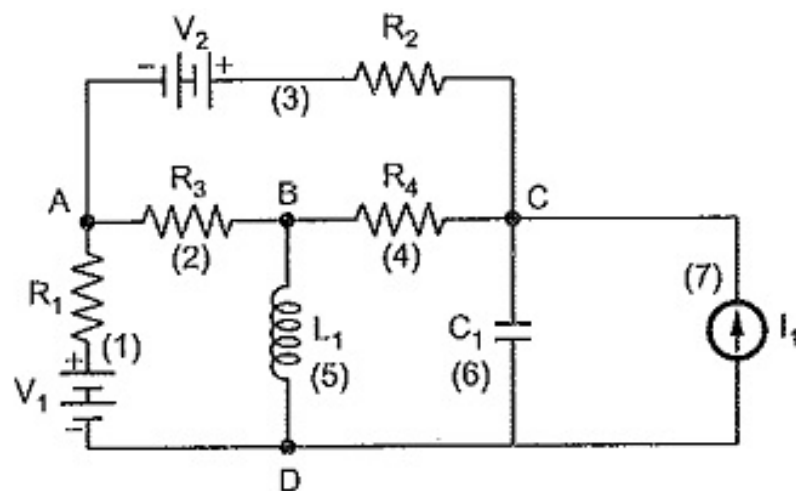




**GA** neznáme parametre



**GP** neznáme parametre  
neznáme väzby



# Porovnanie GP / GA

- **GA - optimalizácia dát (parametrov) v pevne definovanej štruktúre systému**
- **GP – optimalizácia štruktúry aj dát, optimalizácia vnútorných väzieb systému**
- **Pri GA je dĺžka chromozómu konštantná – lebo štruktúru objektu poznáme**
- **Pri GP je dĺžka chromozómu premenlivá – lebo štruktúru objektu nepoznáme**

# Cieľ optimalizácie v GP

- **funkcie (symbolická regresia)**
- **programy**
- **logické automaty**
- **grafy**
- **schémy**
- **elektrické obvody, evolúcia hardvéru**
- **regulačné obvody**
- **konštrukcie**

# Prvky reťazcov (gény)

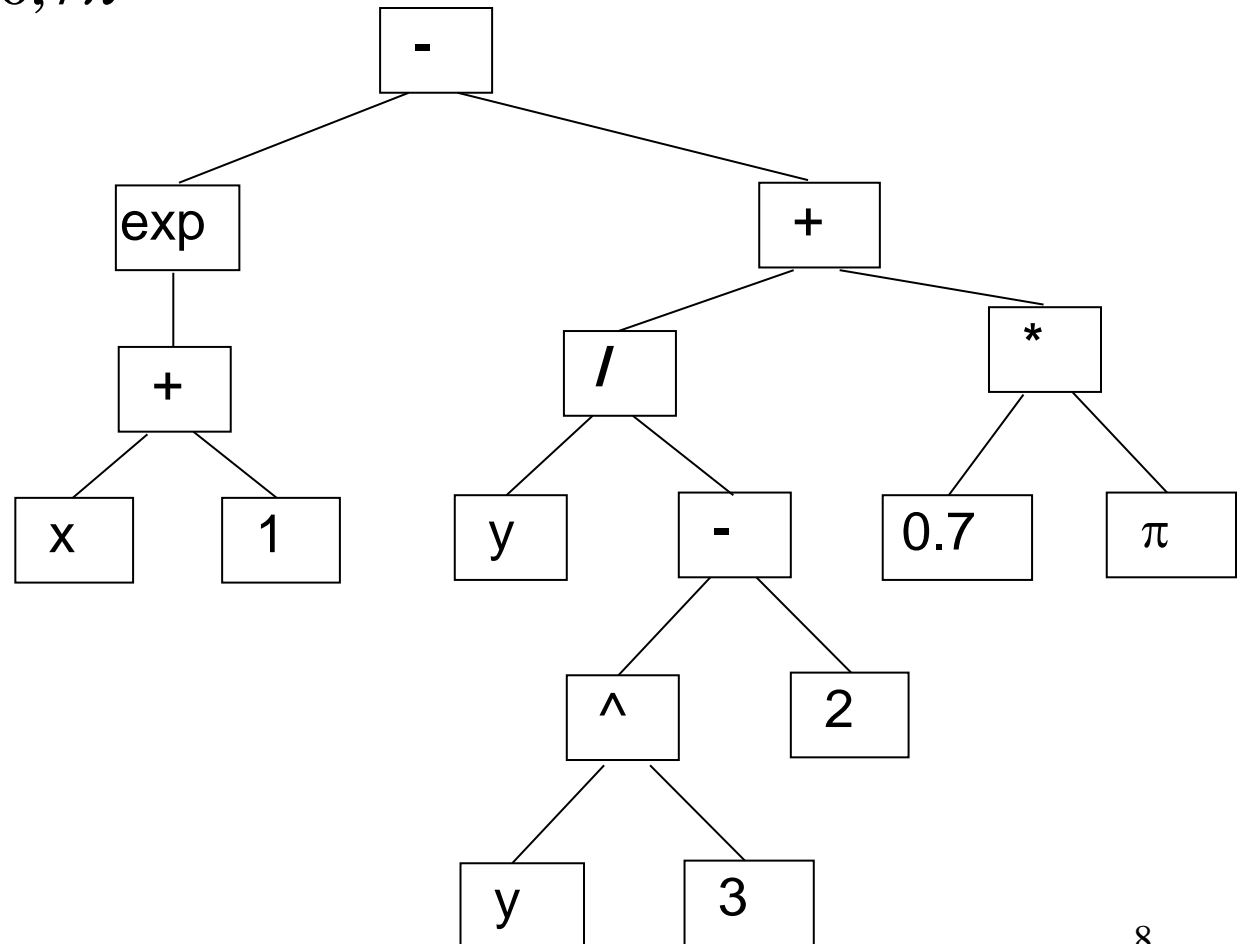
- aritmetické operácie:  $+$ ,  $-$ ,  $*$ ,  $/$
- funkcie: *sin, cos, exp, log, ln ...*
- logické funkcie: *and, or, not, xor*
- priradenie hodnoty premennej: napr.  $x=1$
- inštrukcie programu príslušného jazyka:  
*if, else, then, switch, goto, call, while-do, for-do, ...*
- podprogramy, makroinštrukcie alebo výkonné časti programu:  
načítanie informácie zo snímača,  
akčný zásah o určenej veľkosti, zatočenie doprava,  
zatočenie doľava ...
- prvky (orientovaných) grafov, prvky schém  
zapojenia, spojenia medzi prvkami schém ...

# Reprezentácia jedincov v GP

- **Stromová**
- **Lineárna**
- **Grafová**
- **Iné (tabuľky, dátové polia ...)**

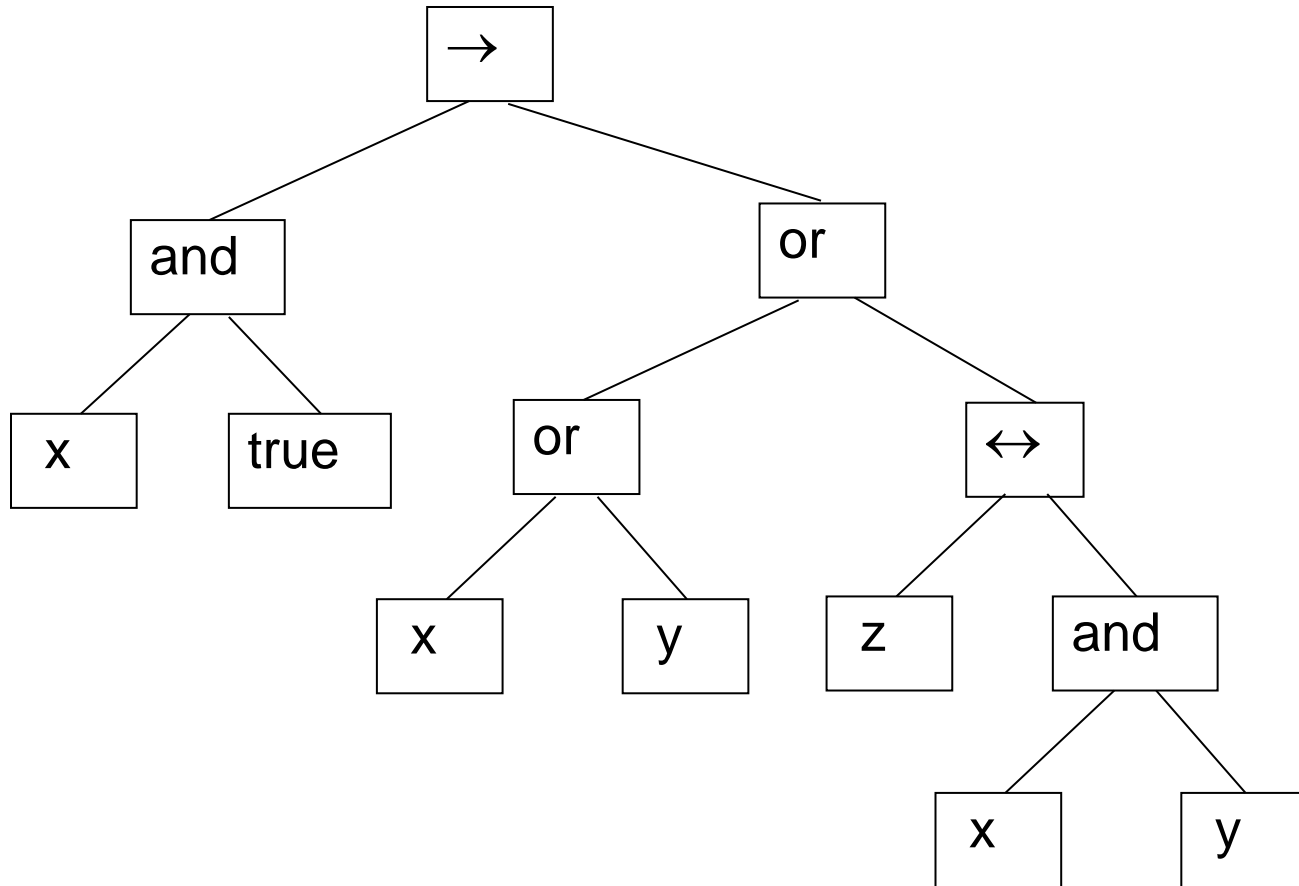
# Stromová reprezentácia

$$\exp(x+1) - \frac{y}{y^3-2} + 0,7\pi$$





$$(a \wedge \text{true}) \rightarrow ((a \vee b) \vee (z \leftrightarrow (a \wedge b)))$$



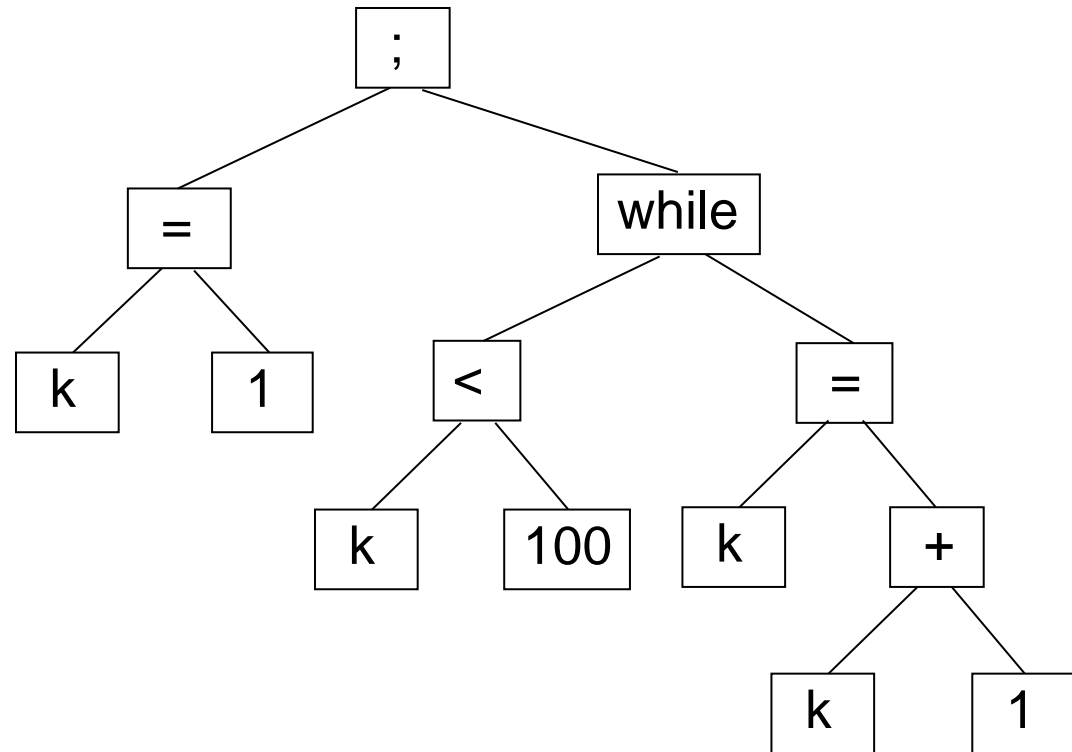
$k=1;$

*while* ( $k<100$ )

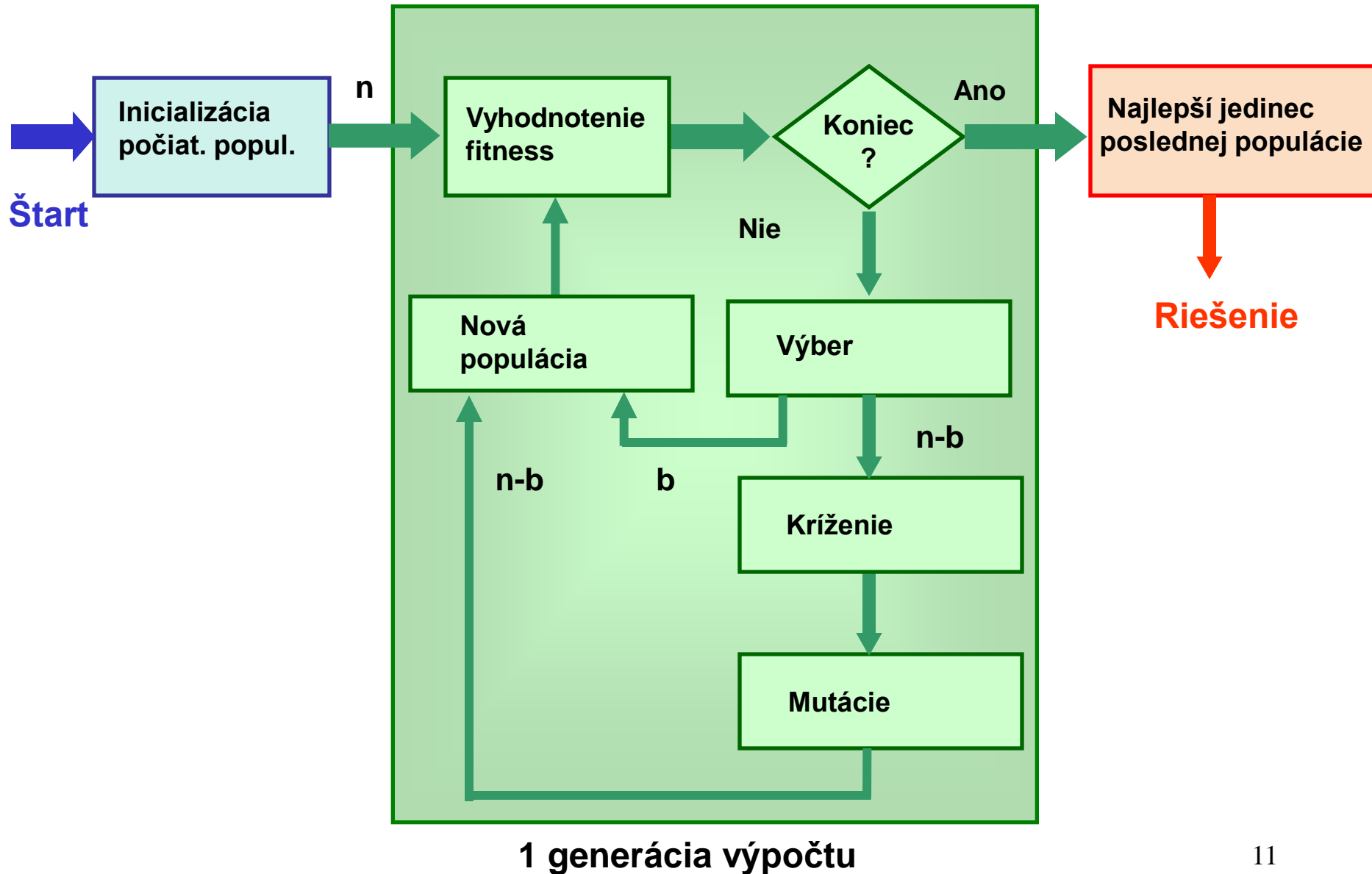
{

$k=k+1$

}

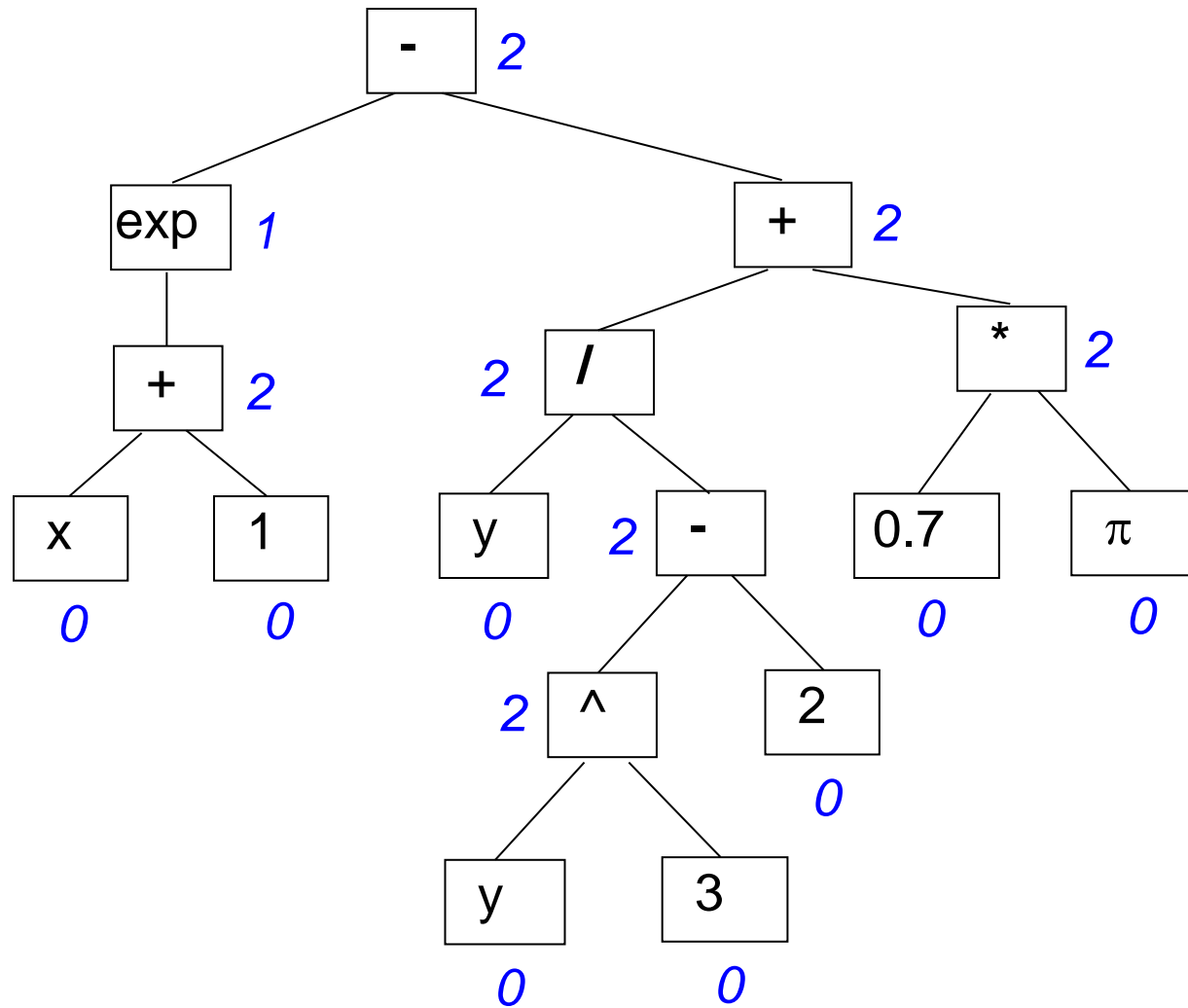


# Algoritmus GP

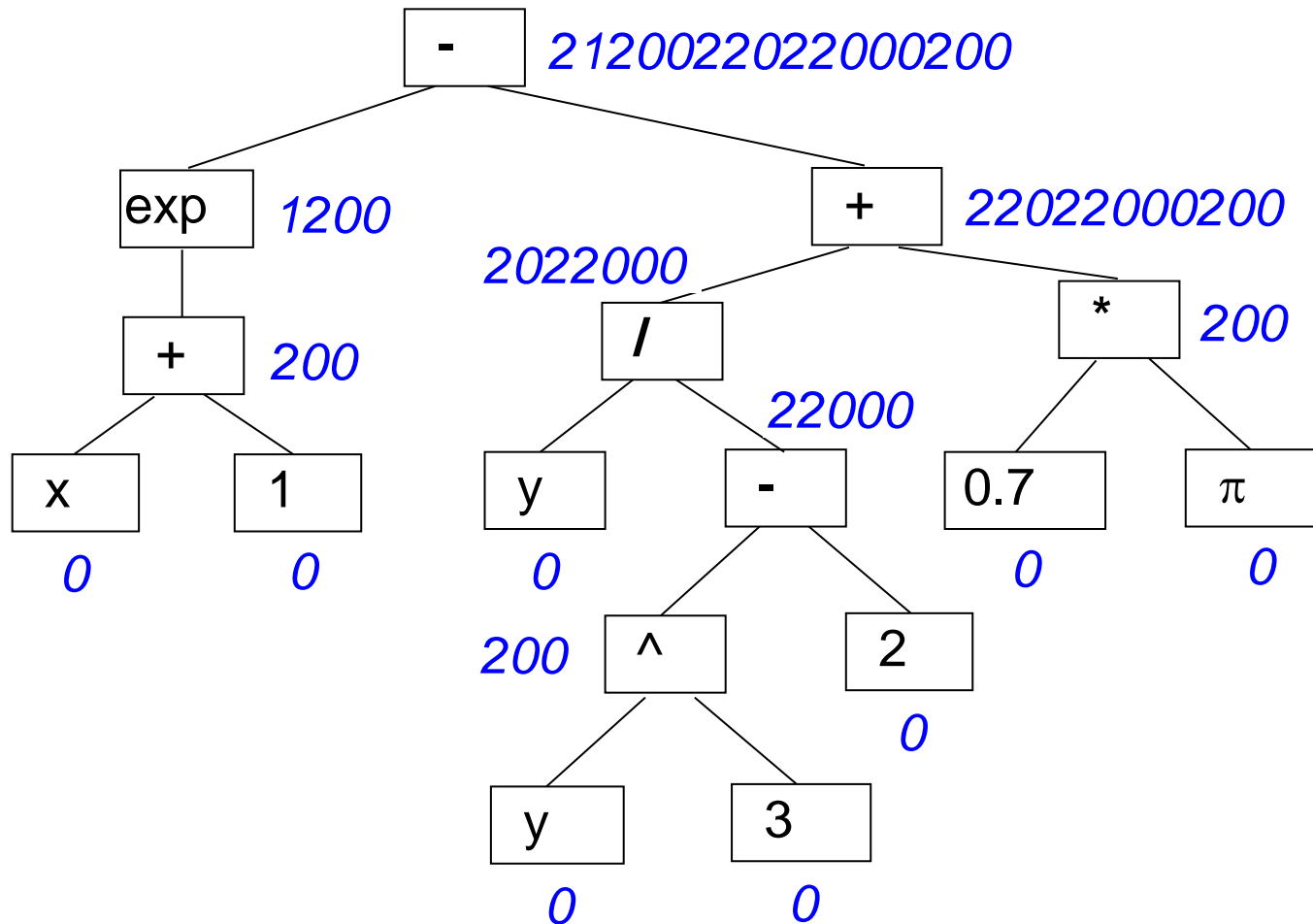


# **Readov lineárny kód - transformácia stromovej štruktúry do lineárneho reťazca znakov**

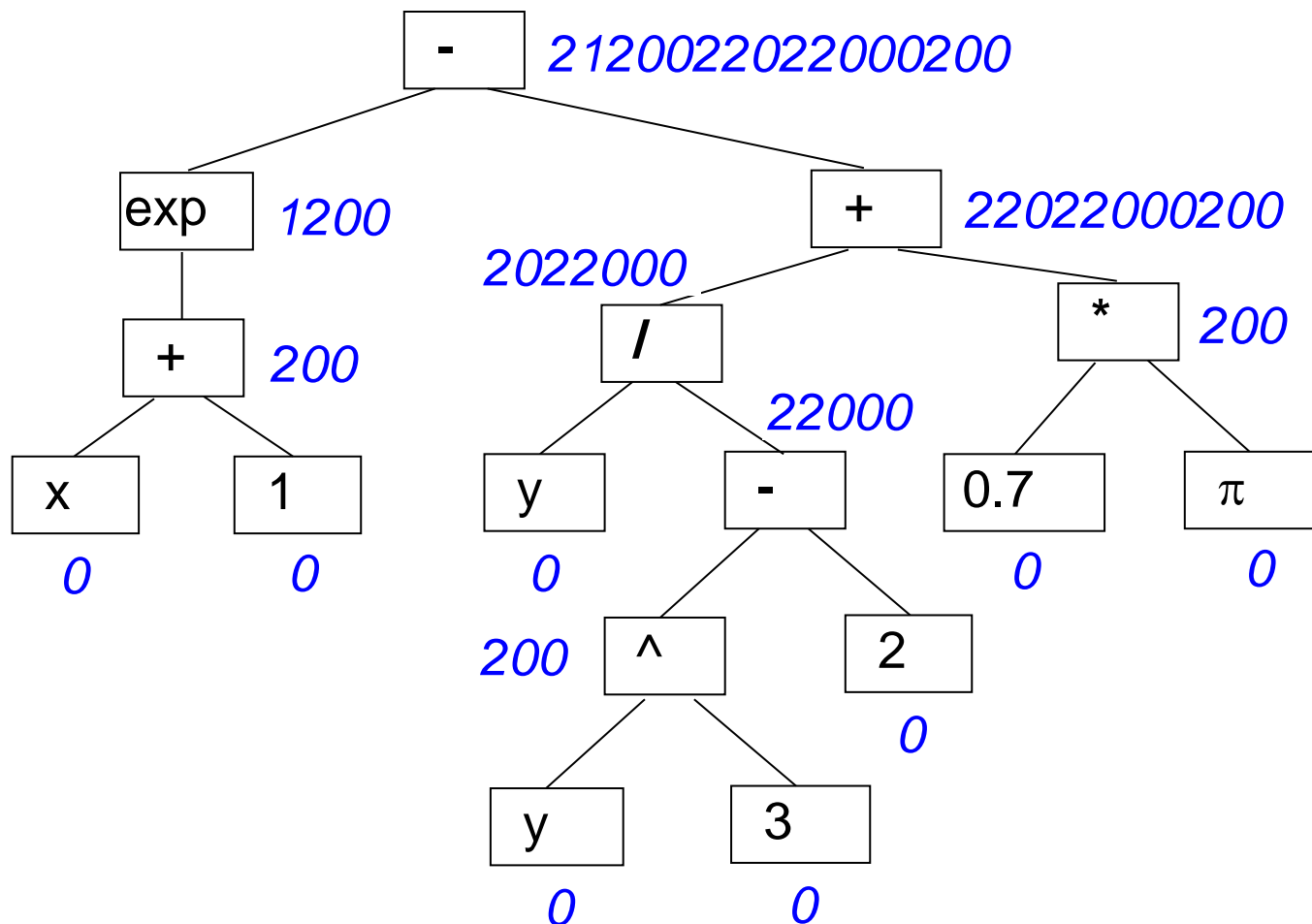
# Valencia uzlov stromu



# Readov kód stromu



# Kompletný readov kód stromu



**$[(2,-),(1,\text{exp}),(2,+),(0,x),(0,1),(2,+),(2,/),(0,y),(2,-),(2,\wedge),(0,y),(0,3),(0,2),(2,*),(0,0.7),(0,\pi)]$**

**Alebo:  $[2,1,2,0,0,2,2,0,2,2,0,0,0,2,0,0,-,\text{exp},+,x,1,+,/,y,-,\wedge,y,3,2,*,0.7,\pi]$**  <sup>15</sup>

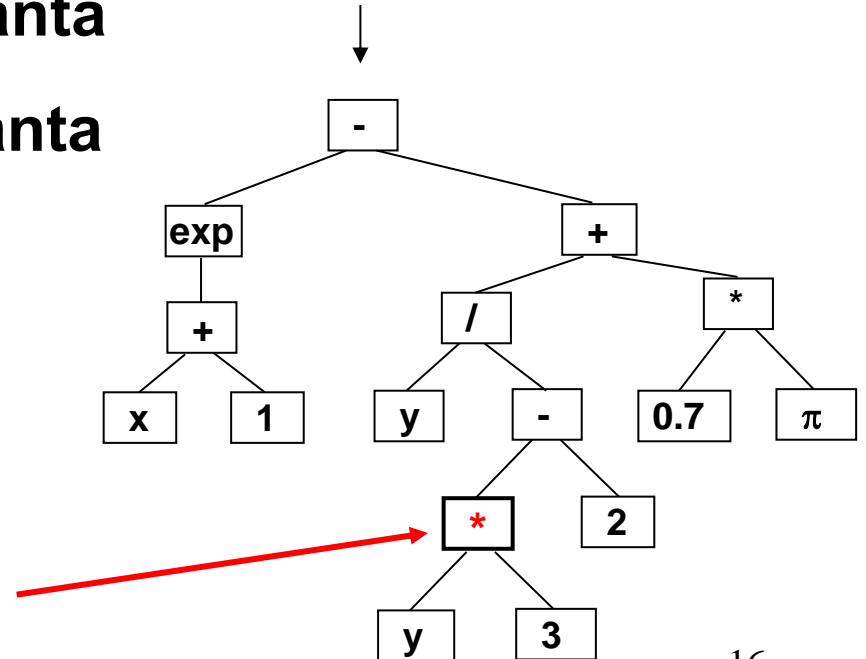
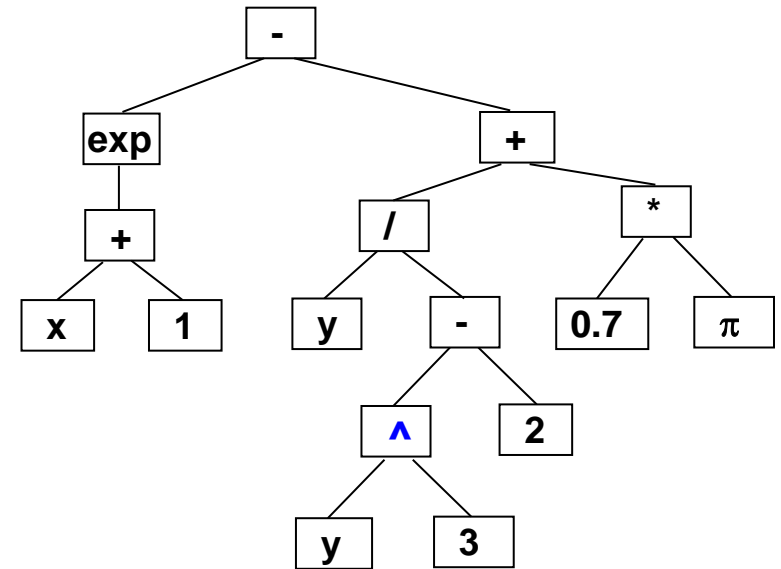
# Mutácie stromu

náhrada uzla iným uzlom

funkcia → funkcia

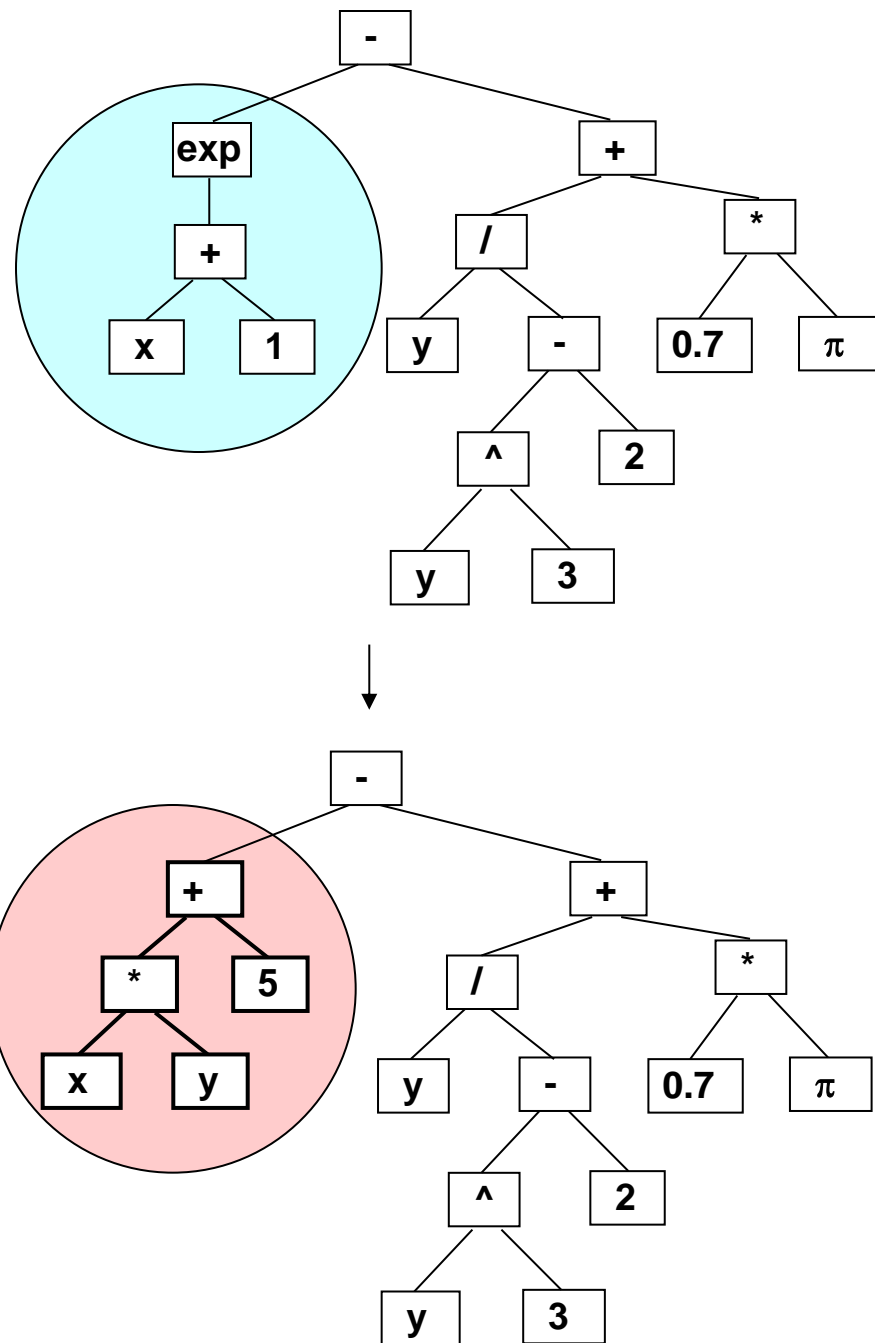
premenná → premenná/konštanta

konštanta → premenná/konštanta

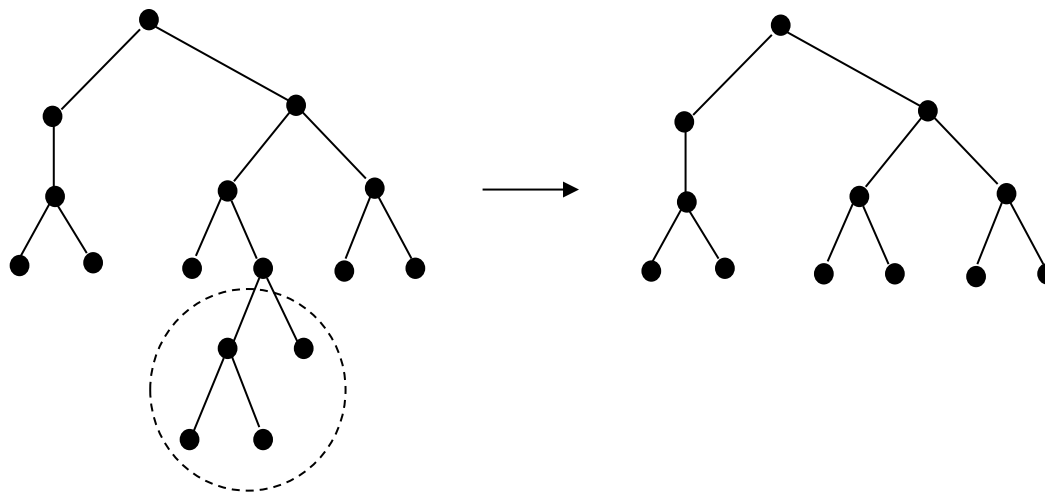




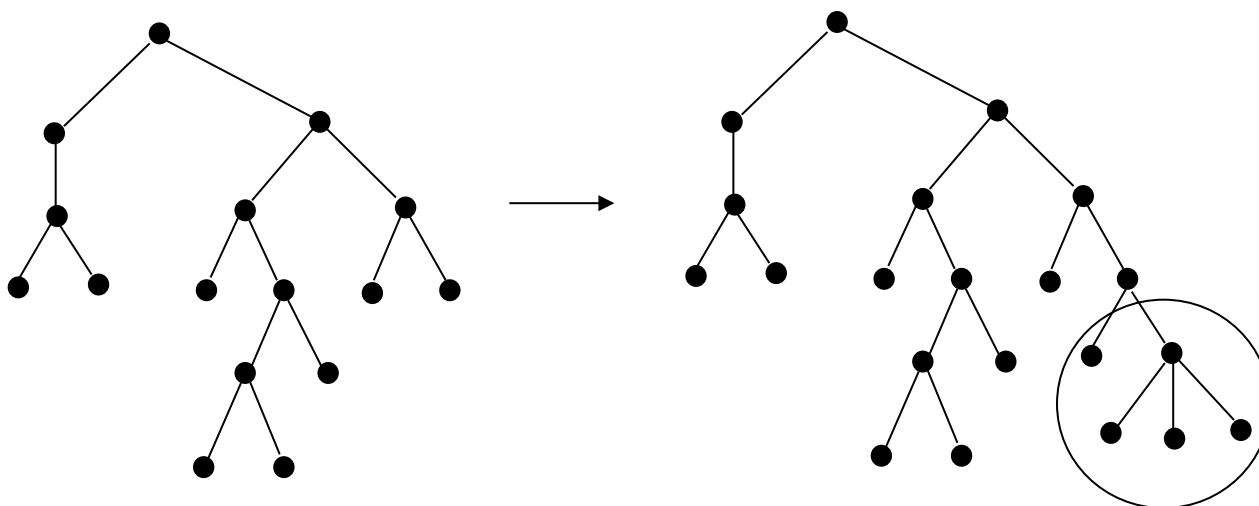
## Náhrada podstromu iným podstromom



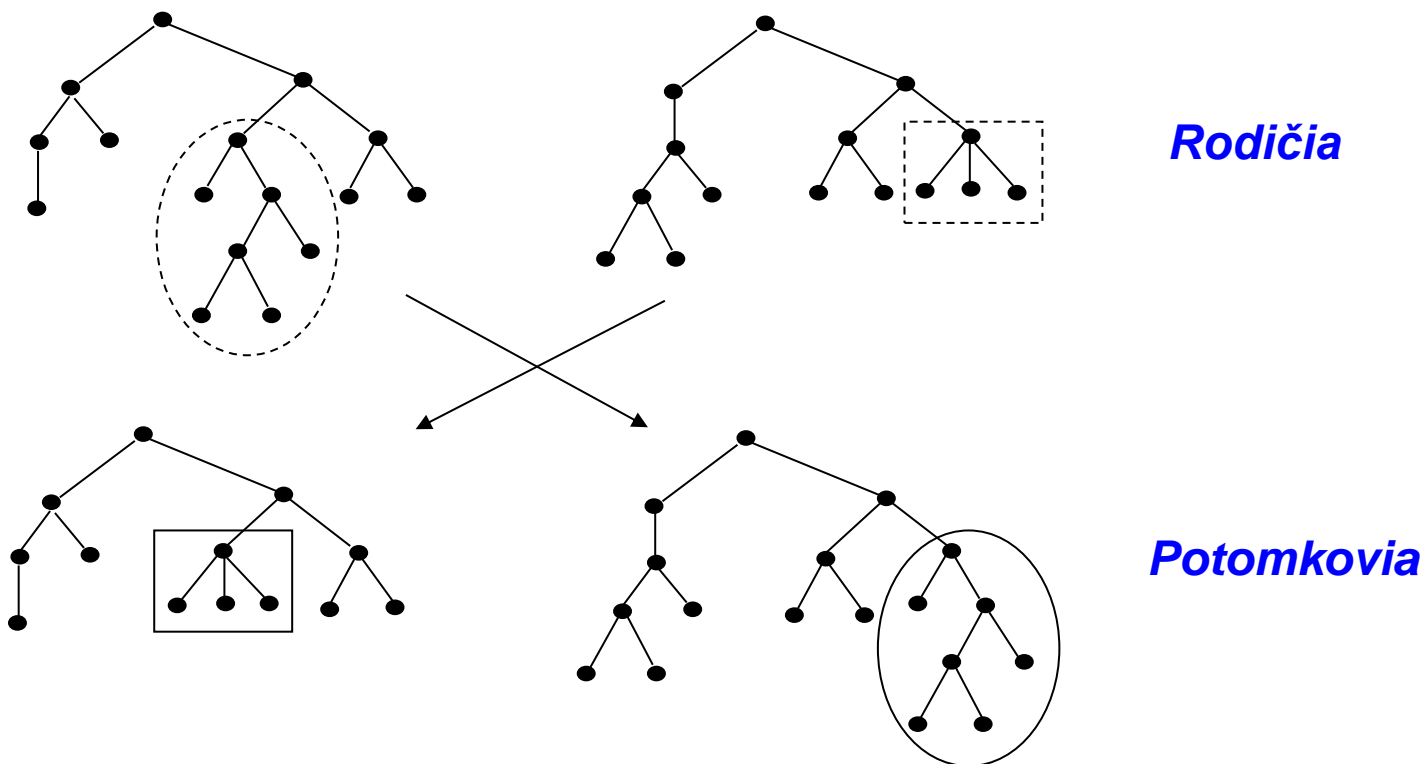
## Mutácia odstránením podstromu



## Mutácia pridaním podstromu



# Kríženie stromov

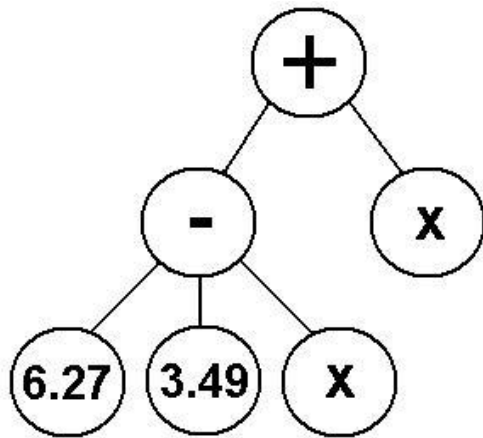


# **Príklady praktických aplikácií genetického programovania**

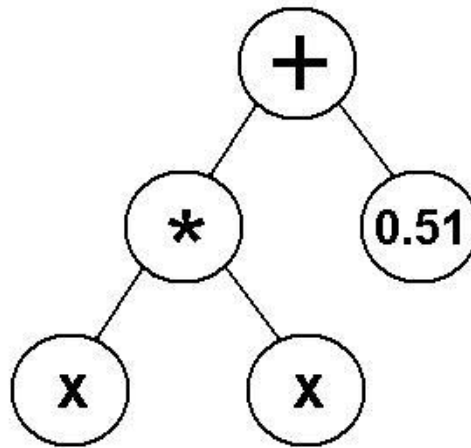
## Príklad 1: symbolová regresia

$$f(x)=x^2+1;$$

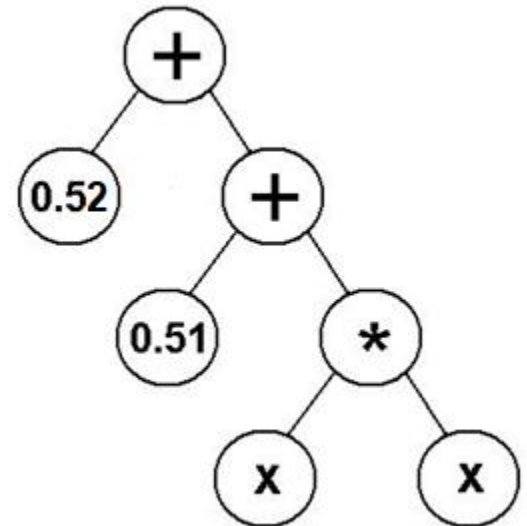
$$x=\{-2,-1,0,1,2\}; \quad f(x)=\{5,2,1,2,5\}.$$



1. gen.



50. Gen



100.gen

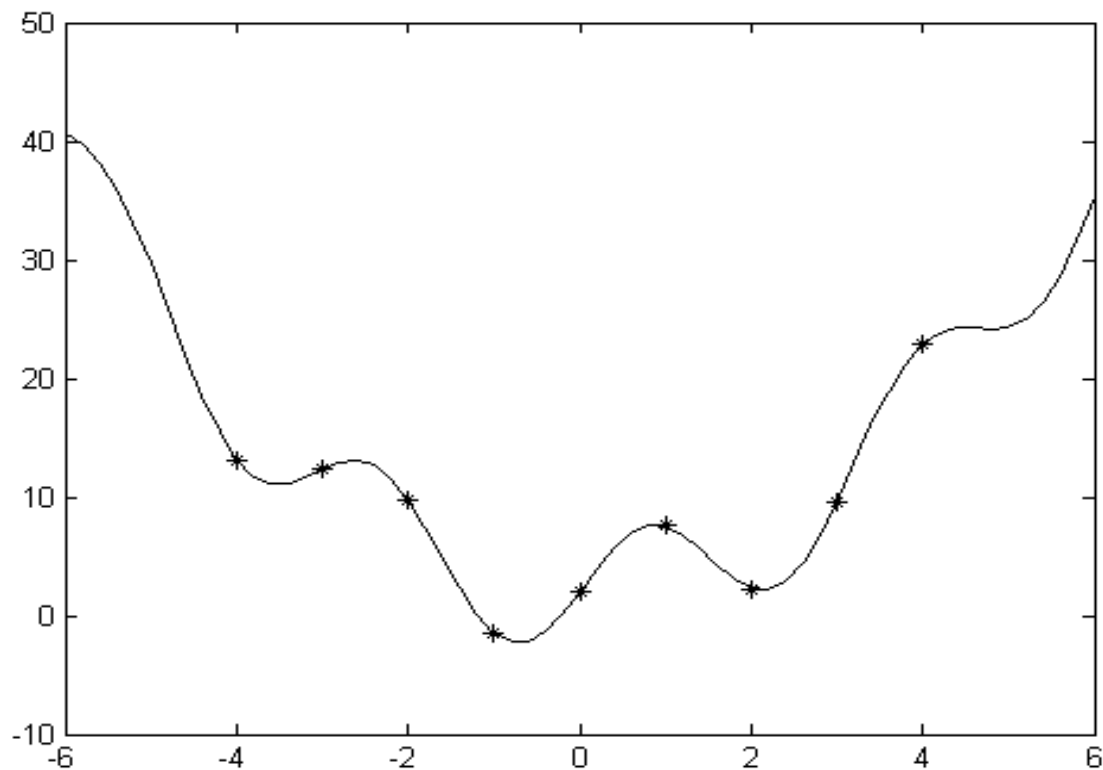
## Príklad 2

$$f(x) = x^2 + 5 \sin(2x) + 2$$

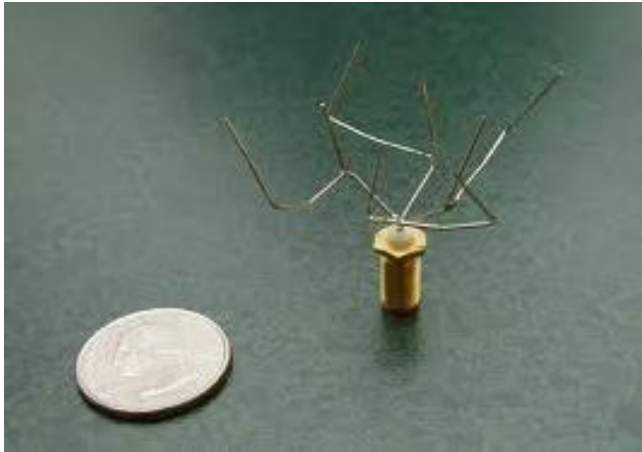


$$F(x) = 4.179 * \sin(x + x) + (0.8165 * 0.7806 * 3.1551) + x * x$$

$$F(x) = x^2 + 4.9179 * \sin(2x) + (2.0109)$$

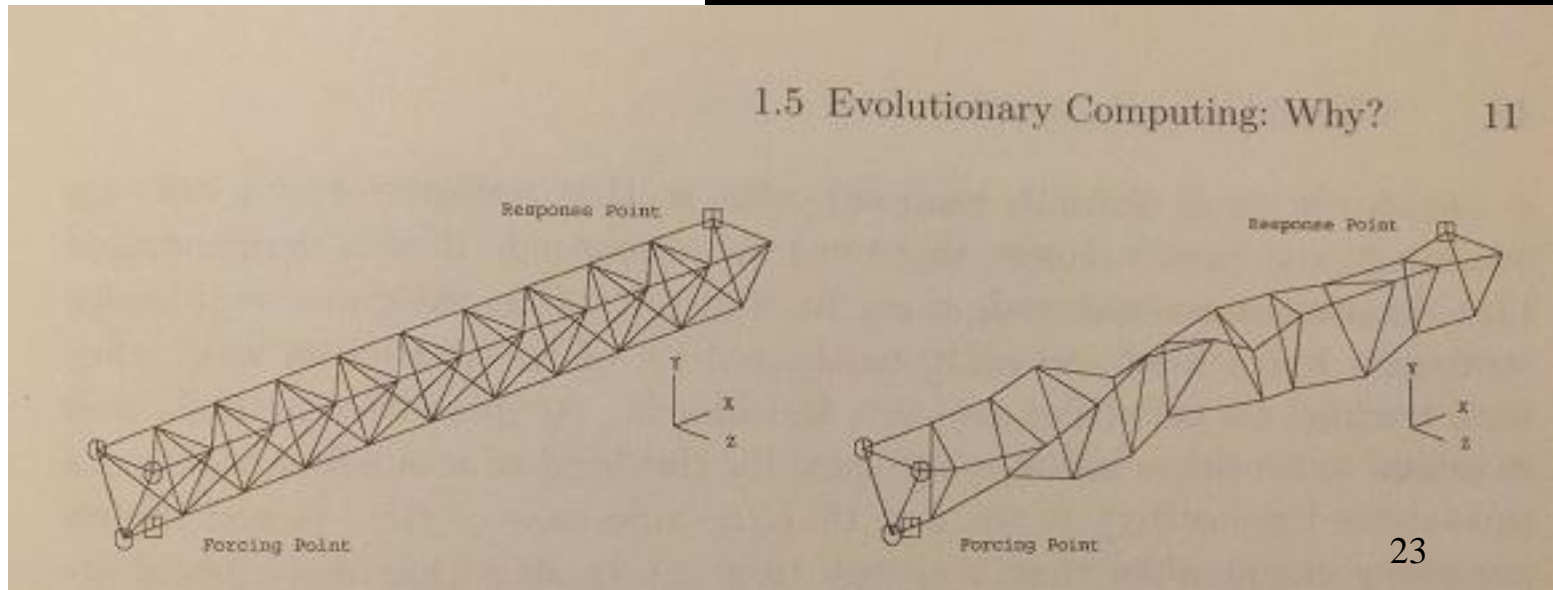
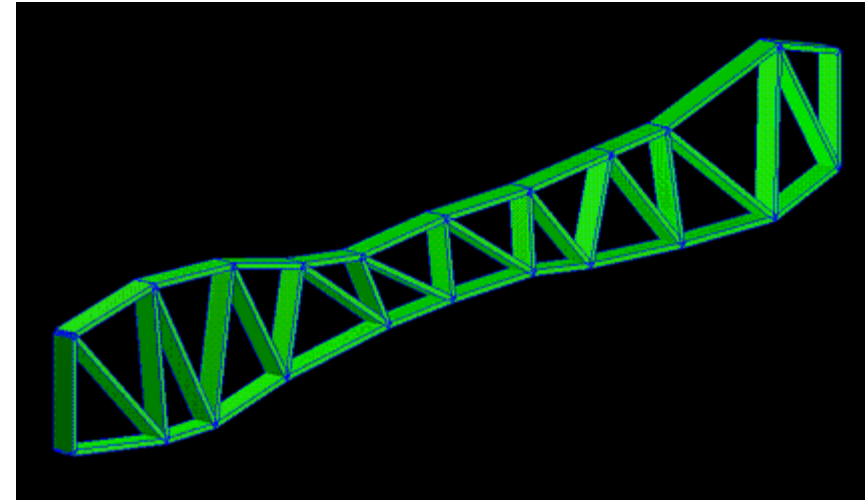


# Rôzne konštrukčné aplikácie

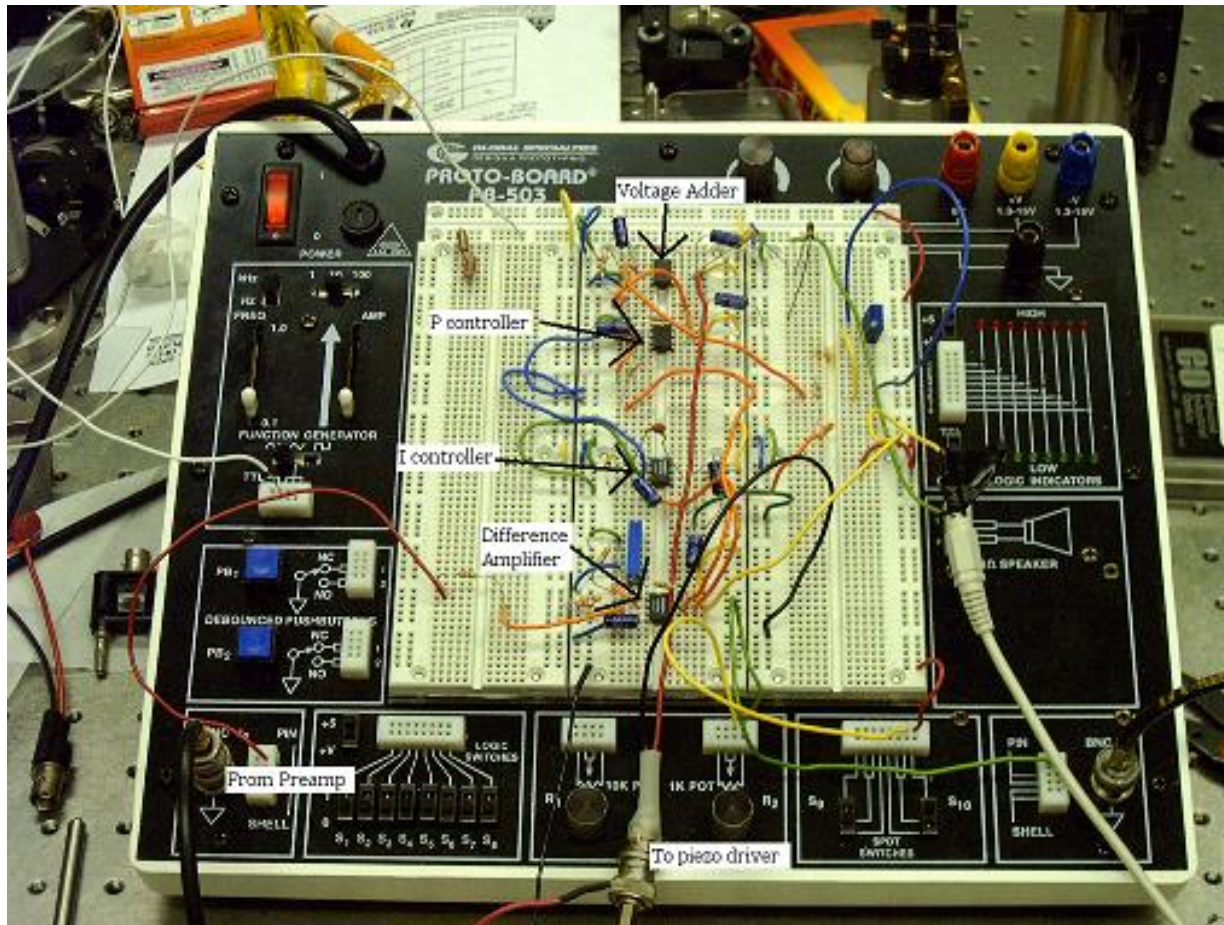


Návrh antény satelitu

## Mechanické konštrukcie

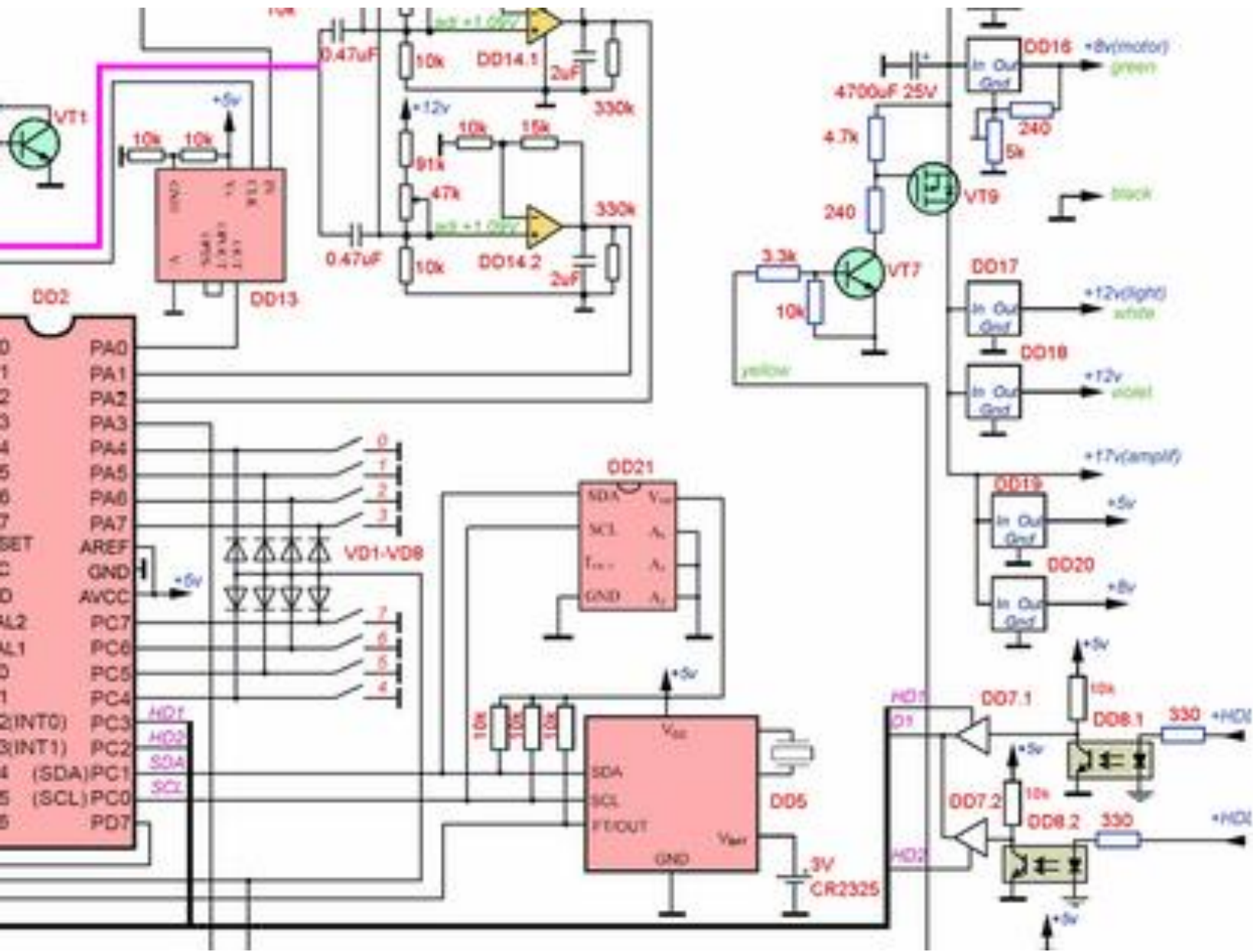


# Evolučný hardvér (Evolvable Hardware)

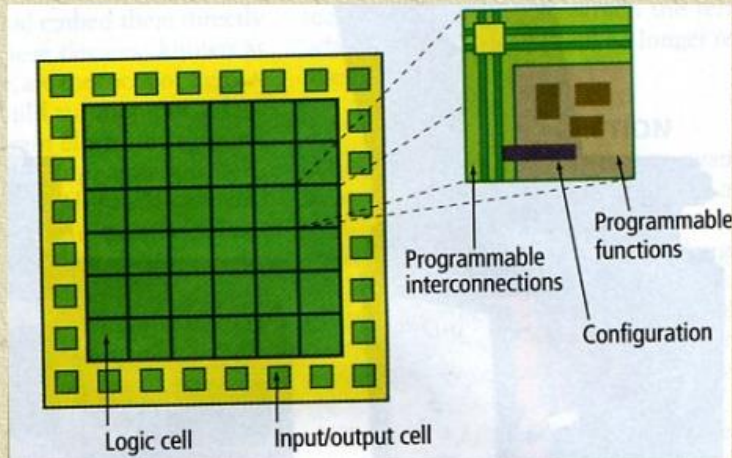




# Evolúcia elektrických obvodov



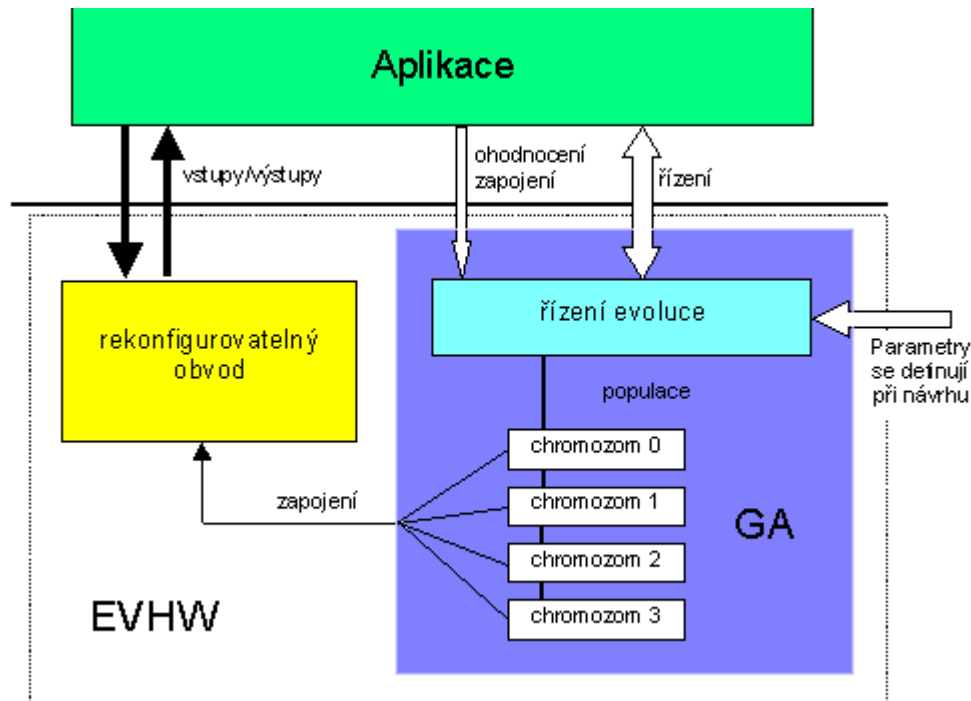
## FPGA:



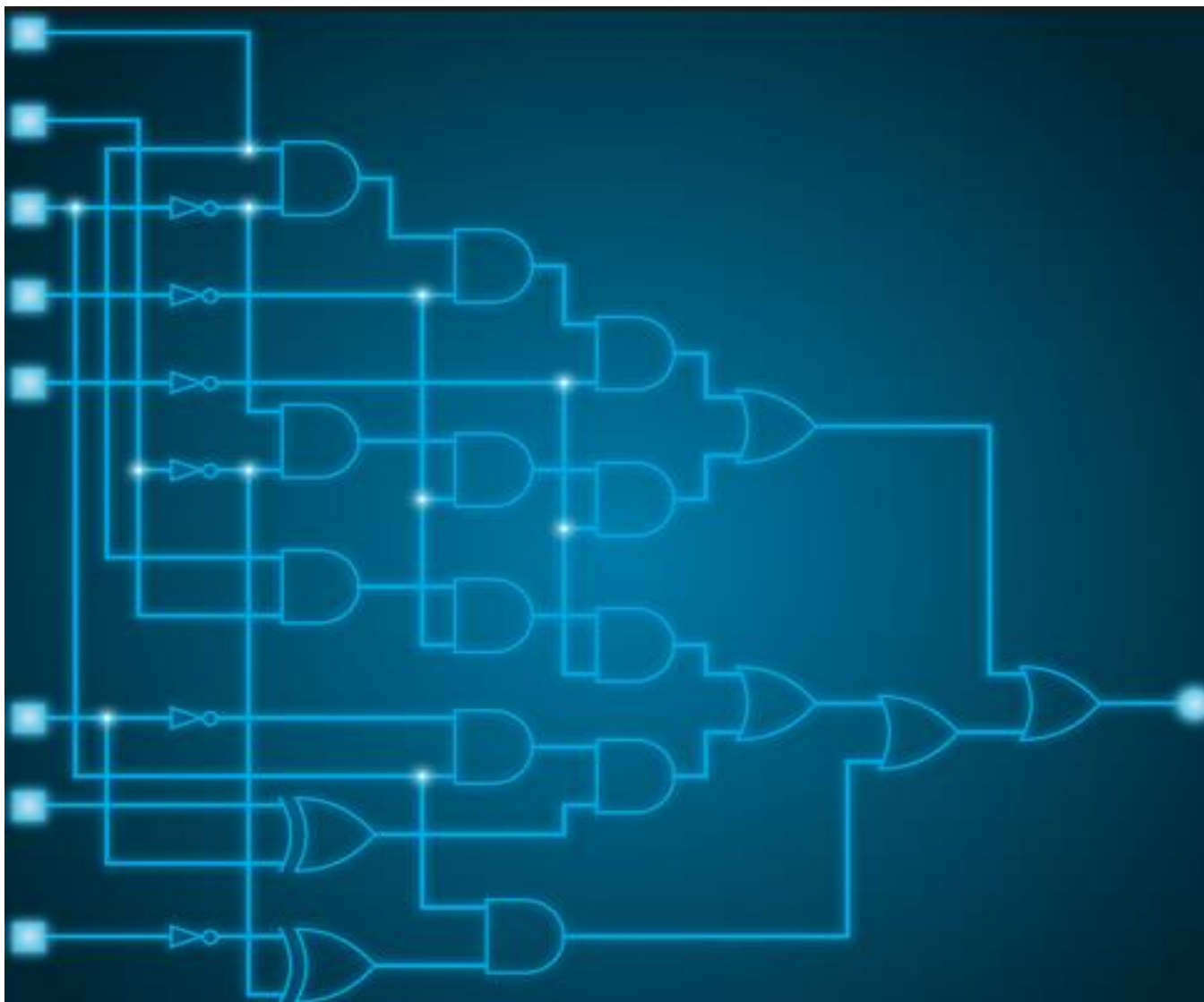
## Vyhodnotenie fitness:

- simulačný softvér elt. obvodov
- rekonfigurovateľné elt. obvody

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



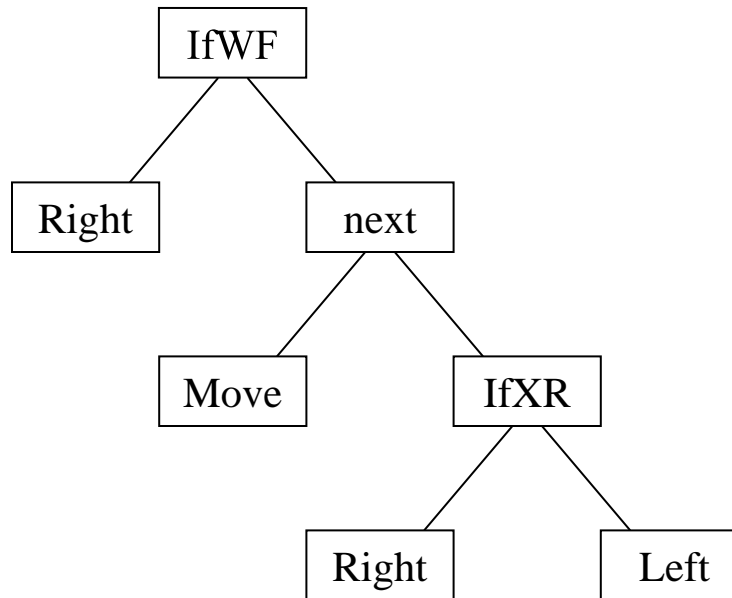
# Evolúcia logických obvodov



# Návrh (riadiaceho) programu – genetické programovanie

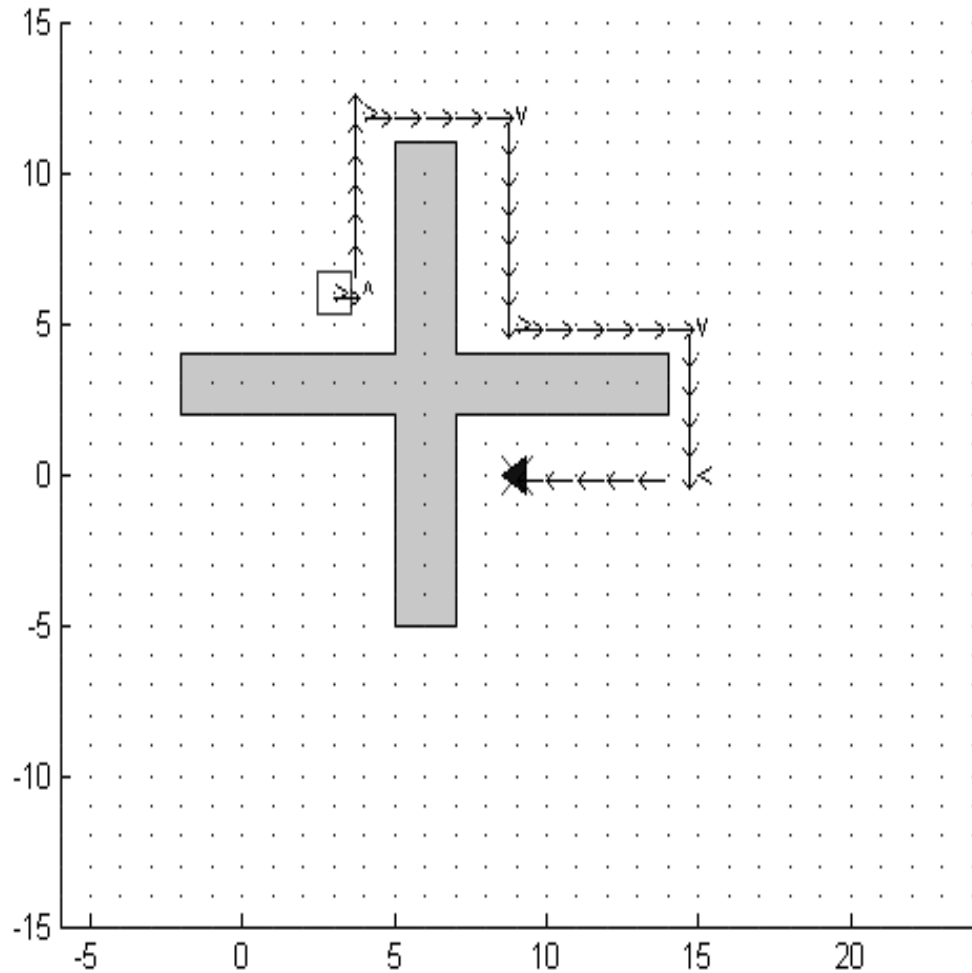
## stromová reprezentácia jedinca

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



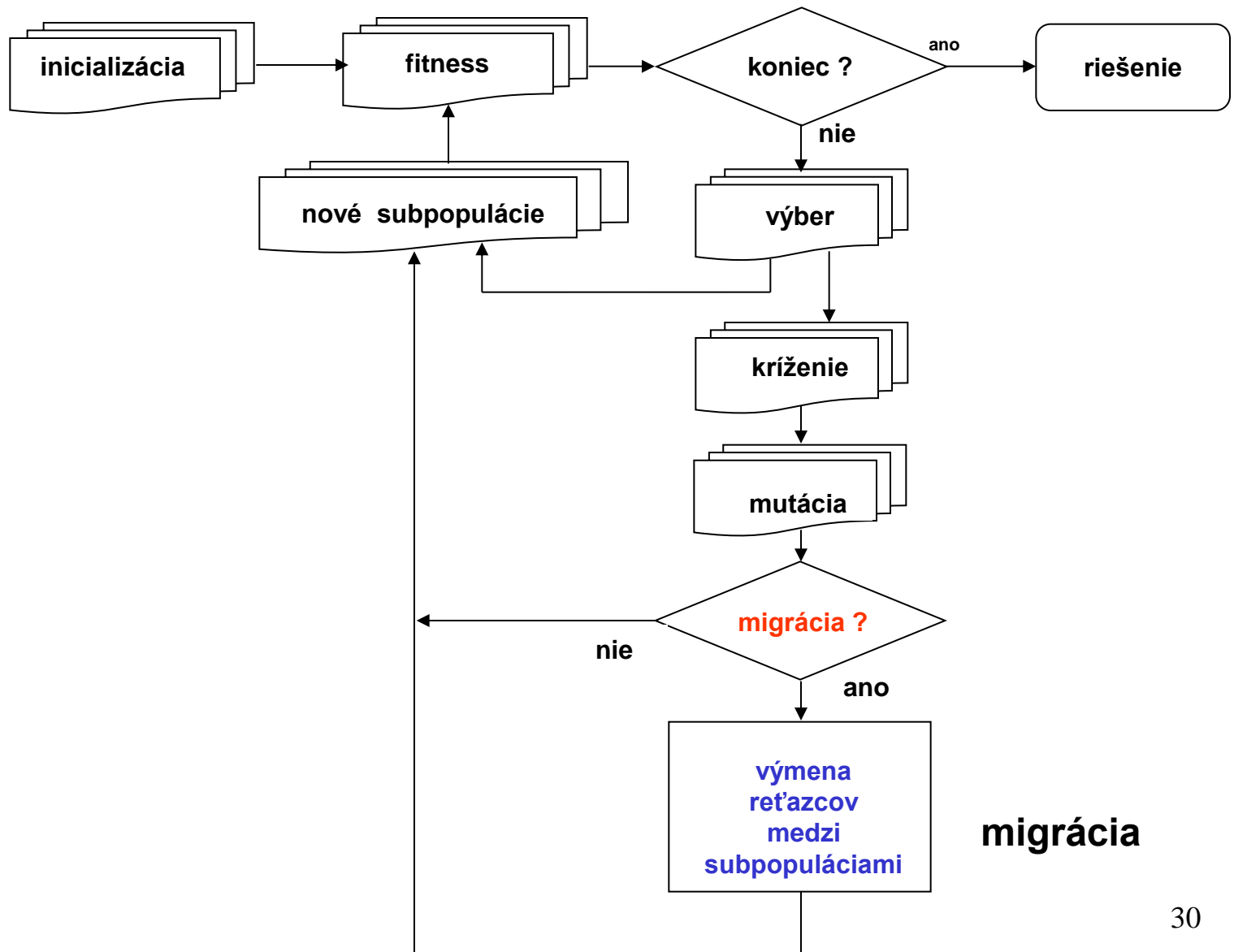
# Výsledná trajektória robota

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



# 1.10 Paralelné GA (PGA)

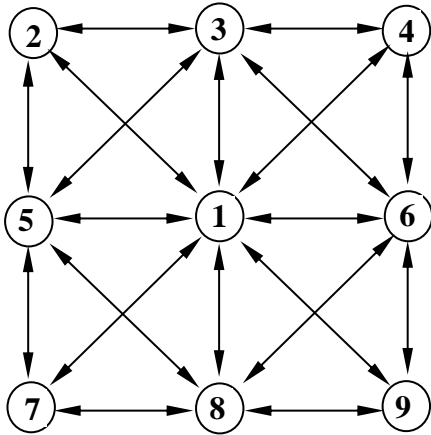
## GA s viacerými subpopuláciami



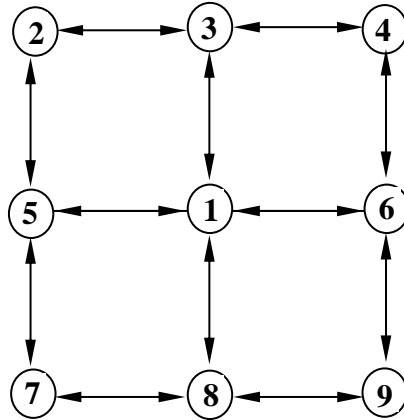
# Ostrovné Paralelné GA

**Subpopulácia (ostrov)** – menší počet väčších ostrovov (hrubozrnné PGA)  
viac jedincov

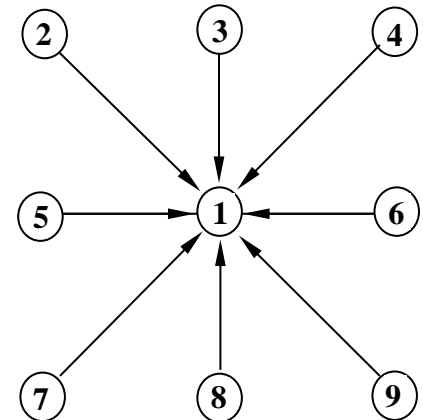
A



B



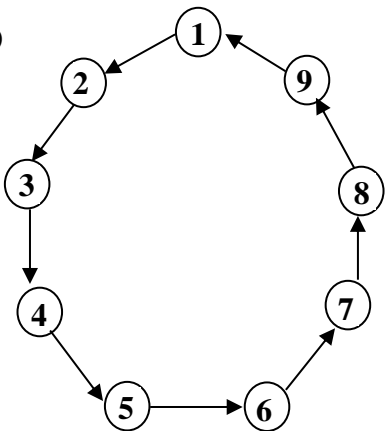
C



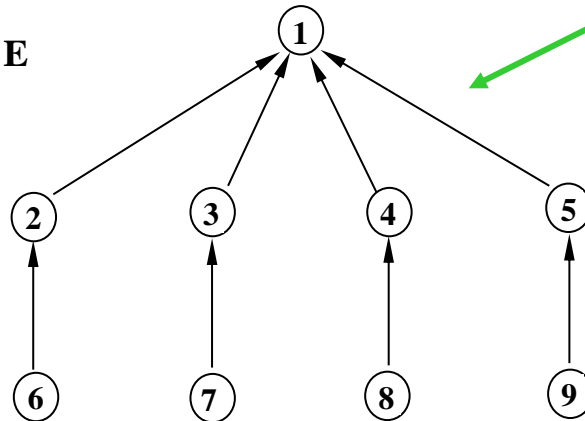
**Migrácia** – vybrané jedince (úspešné) z jedného ostrova  
nahradia vybrané jedince v inom ostrove (menej úspešné)

**Migračné väzby**

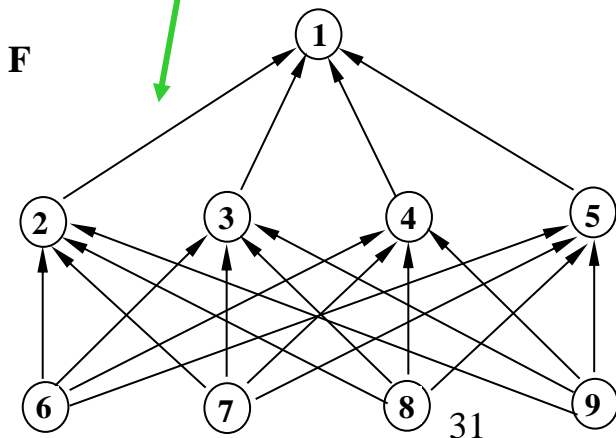
D



E



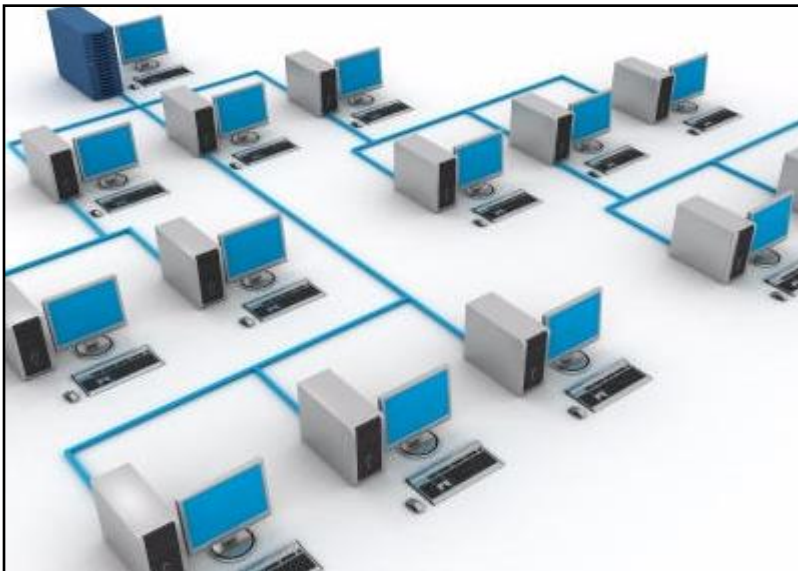
F





# Výpočtová platforma pre PGA

- Môžu sa realizovať na 1 alebo mnohých výpočtových jednotkách (jadrách, procesoroch, počítačoch).
- $N$  jadier zvyšuje výpočtový výkon (takmer)  $N$ -krát, (skracaie výpočtový čas  $N$ -krát – čas sa distribuuje).
- Môže sa uvažovať ľubovoľná architektúra PGA (jednopolulačná / mnohopolulačná).
- Vhodnou topológiou a migráciou v PGA sa môže (výrazne) redukovať aj sumárny počet potrebných vyhodnotení fitness funkcie.





# Matlab - parallel computing

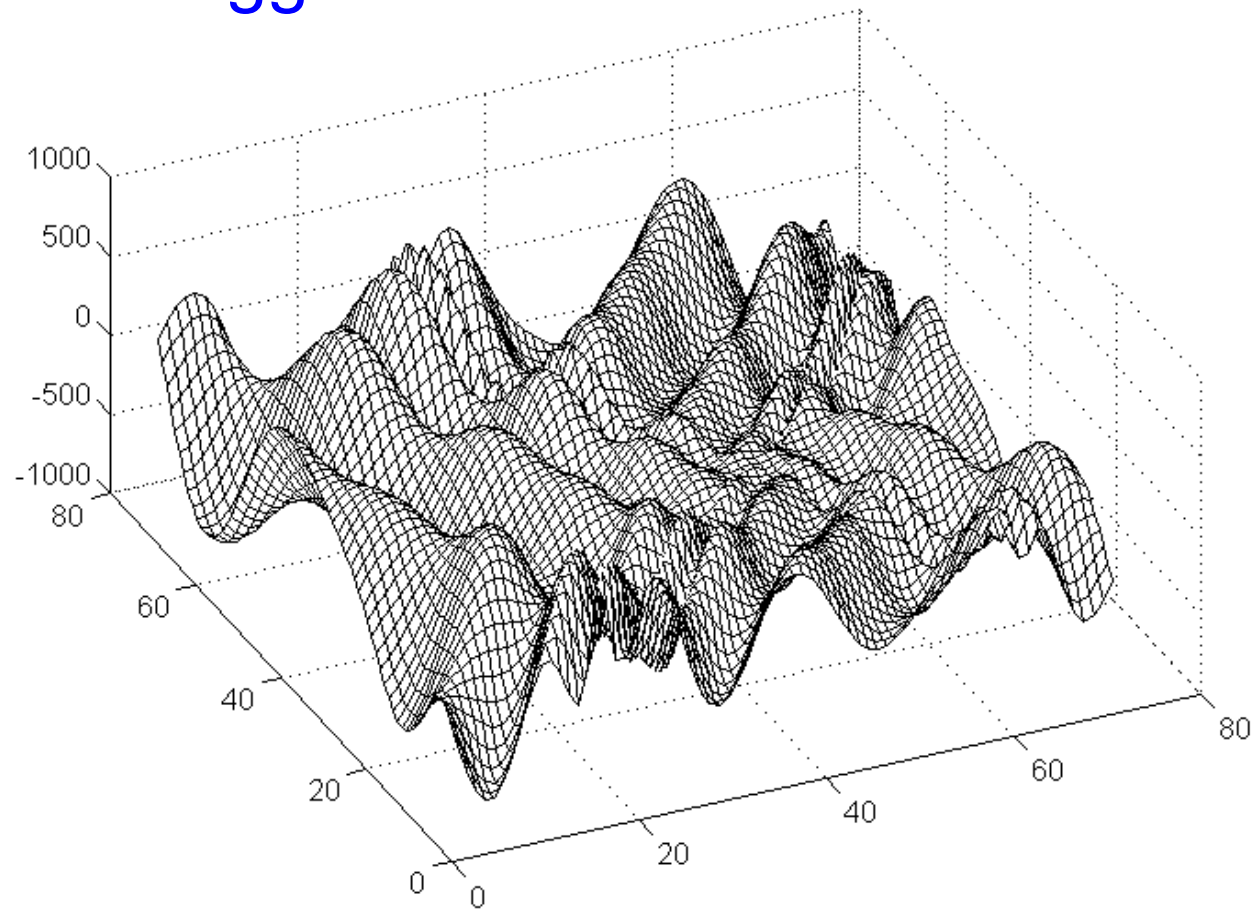
## Parallel Computing Toolbox

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

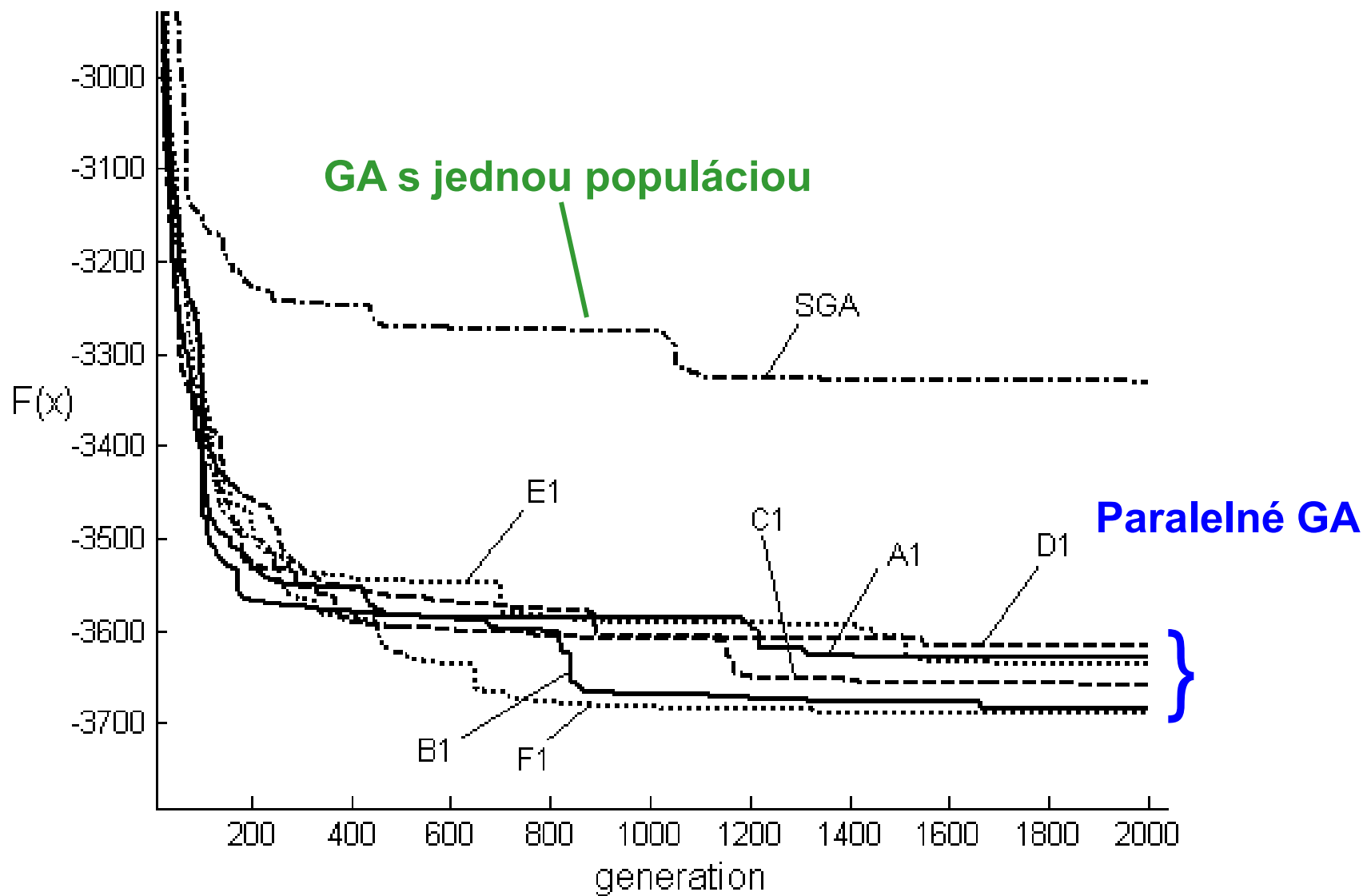
## Matlab Distributed Computing Server

- 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

## Príklad : Eggholder fn.



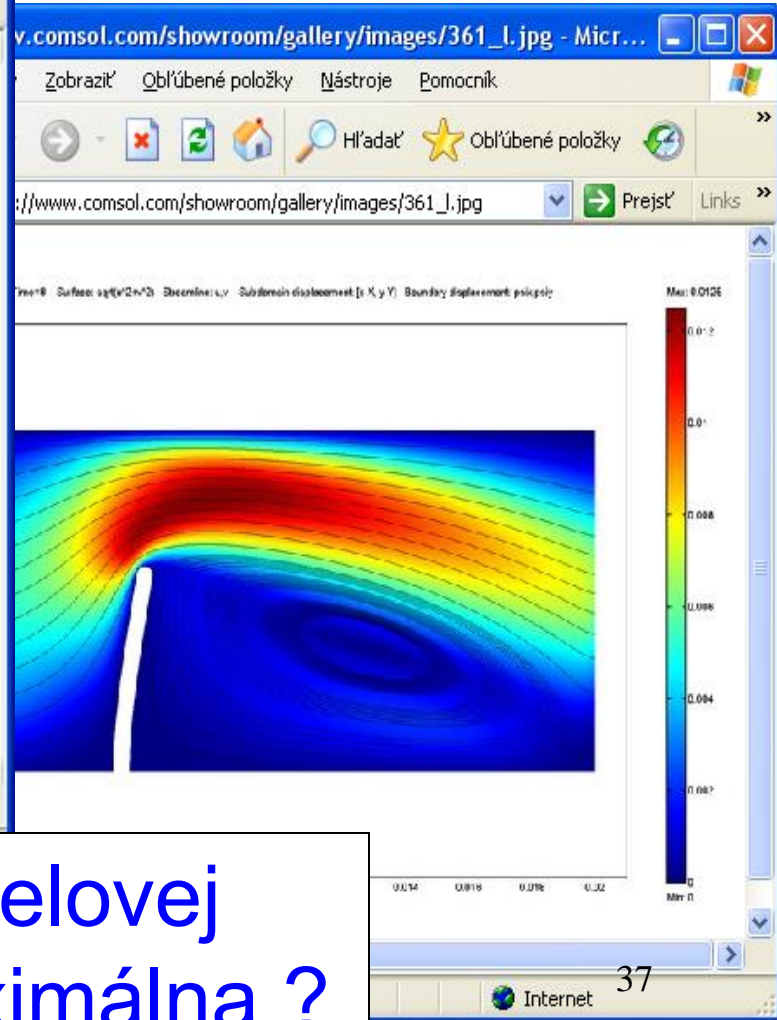
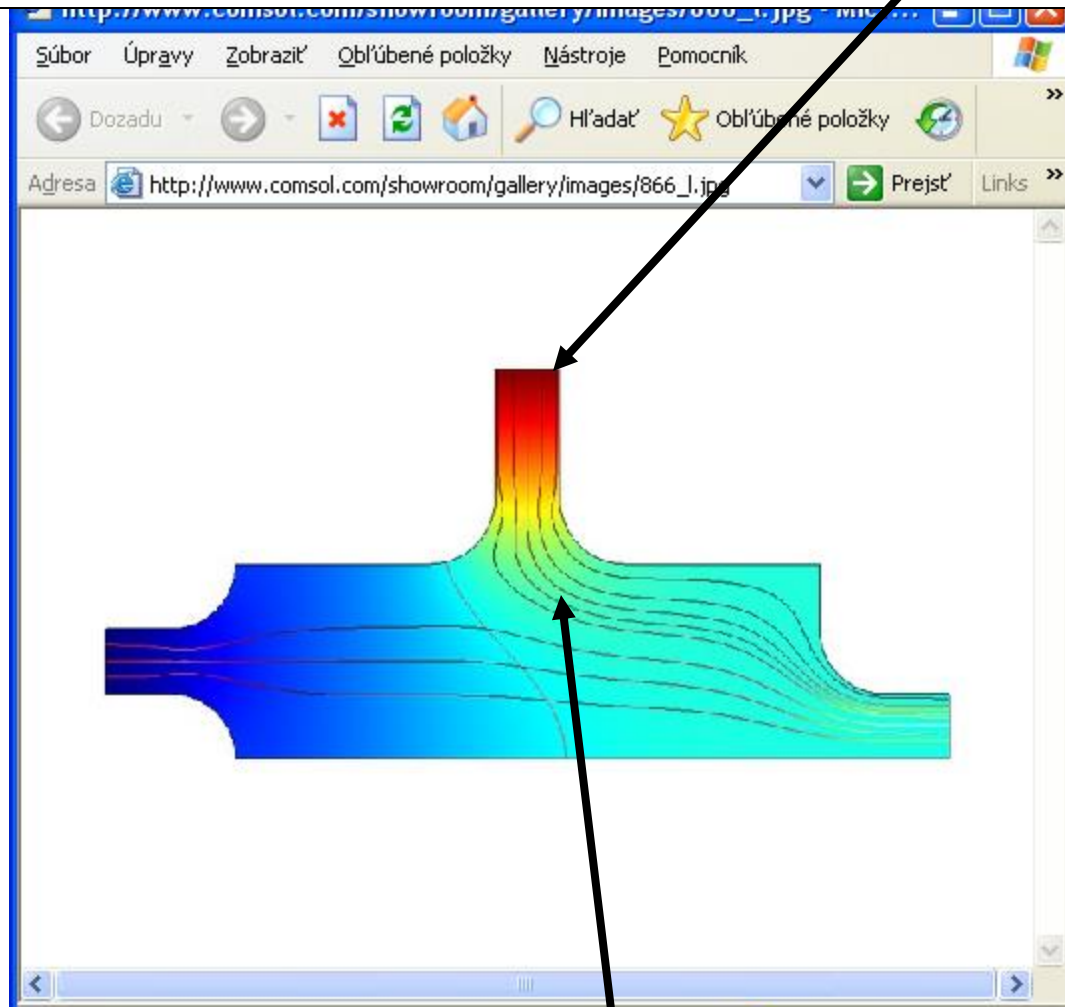
$$f_1(X) = \sum_{i=1}^{n-1} \left( -x_i \sin\left(\sqrt{|x_i - (x_{i+1} + 47)|}\right) - (x_{i+1} + 47) \sin\left(\sqrt{\left|x_{i+1} + 47 + \frac{x_i}{2}\right|}\right) \right)$$



**Paralelný GA potrebuje menej vyhodnotení účelovej funkcie (fitness) v porovnaní s GA s jednou populáciou (s rovnakým počtom jedincov).**

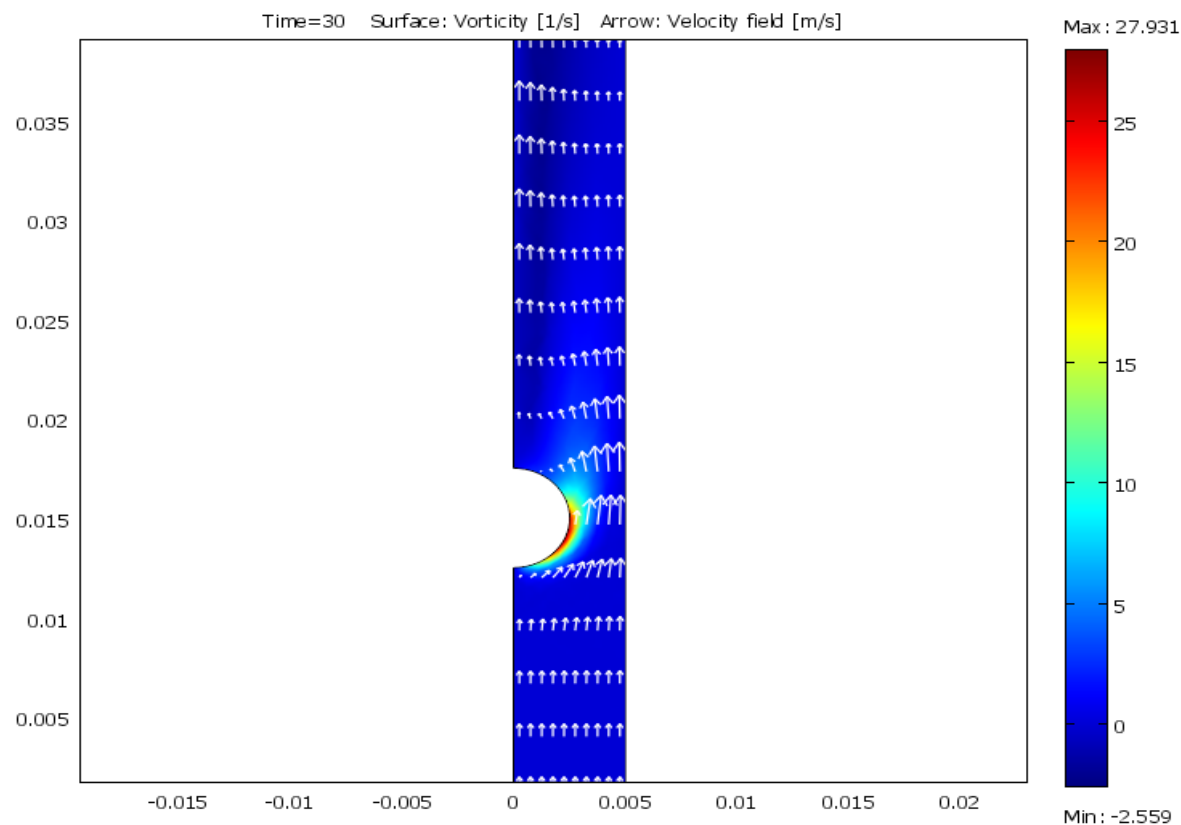
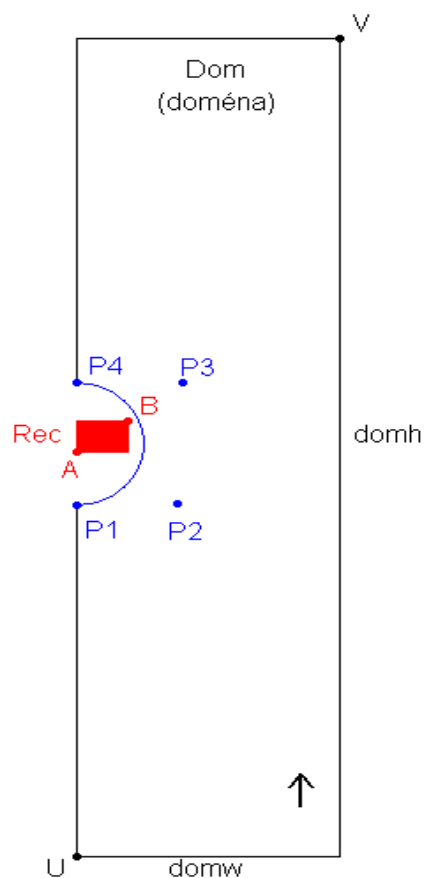
## **1.11 Praktické aplikácie evolučných optimalizačných metód v konštrukčných úlohách**

Aké majú byť parametre objektu (zariadenia),

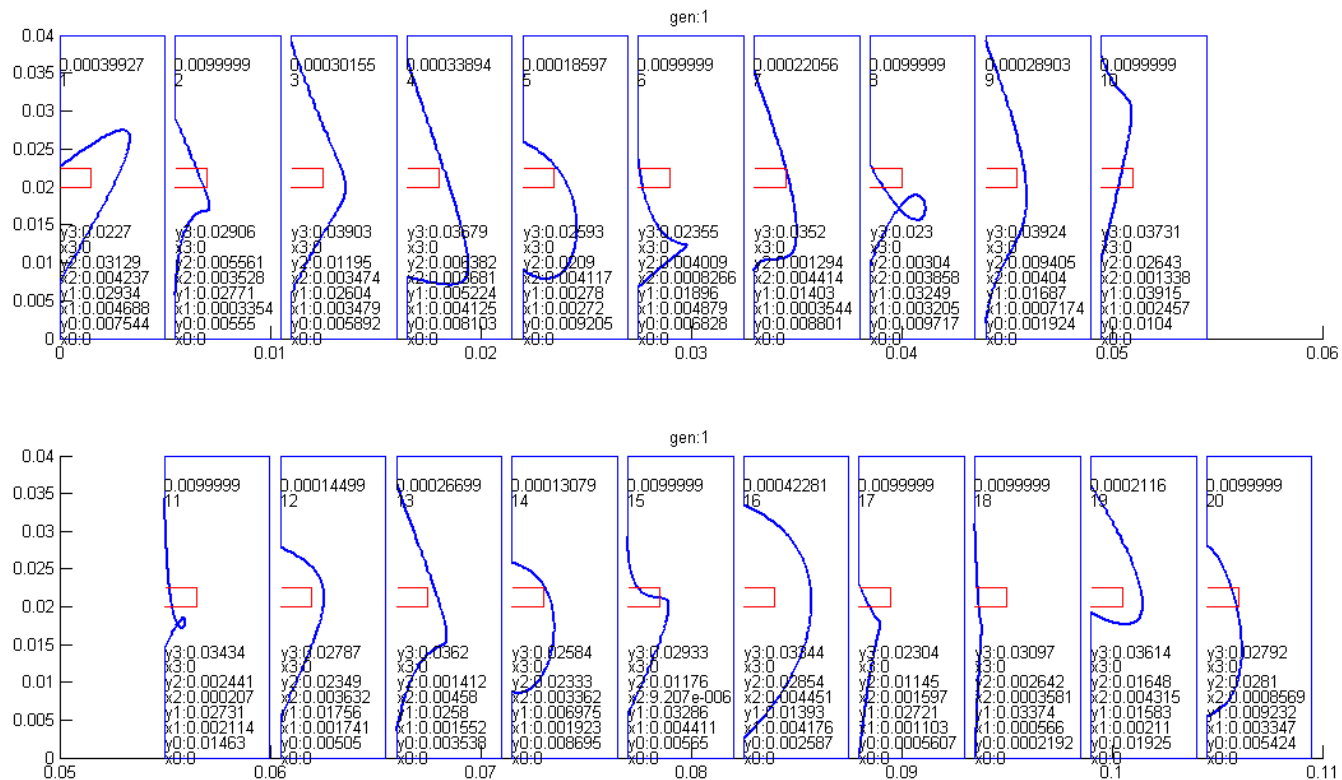


aby hodnota definovanej účelovej funkcie bola minimálna/maximálna ?

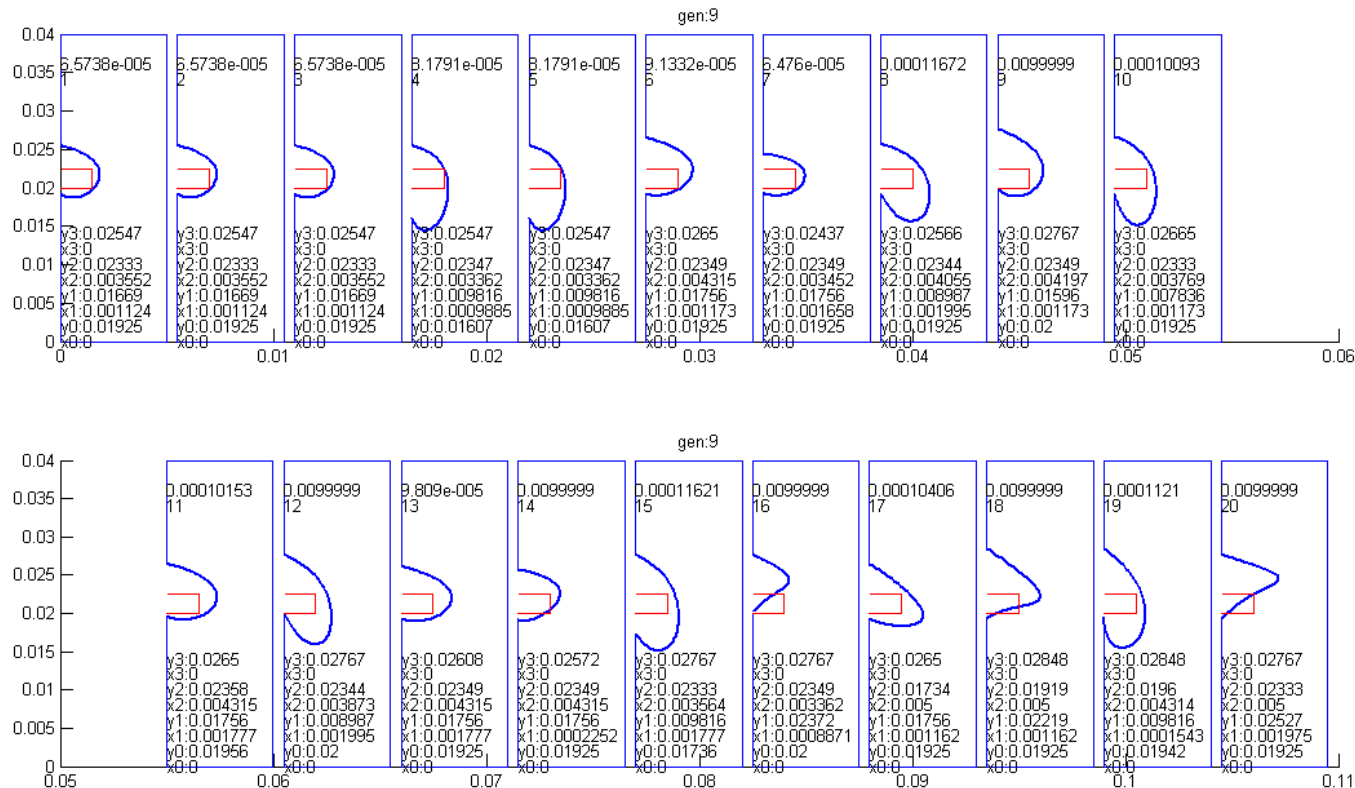
# Optimalizácia tvaru prekážky v potrubí pretekanom tekutinou



# Populácia v 1. generácii výpočtu GA

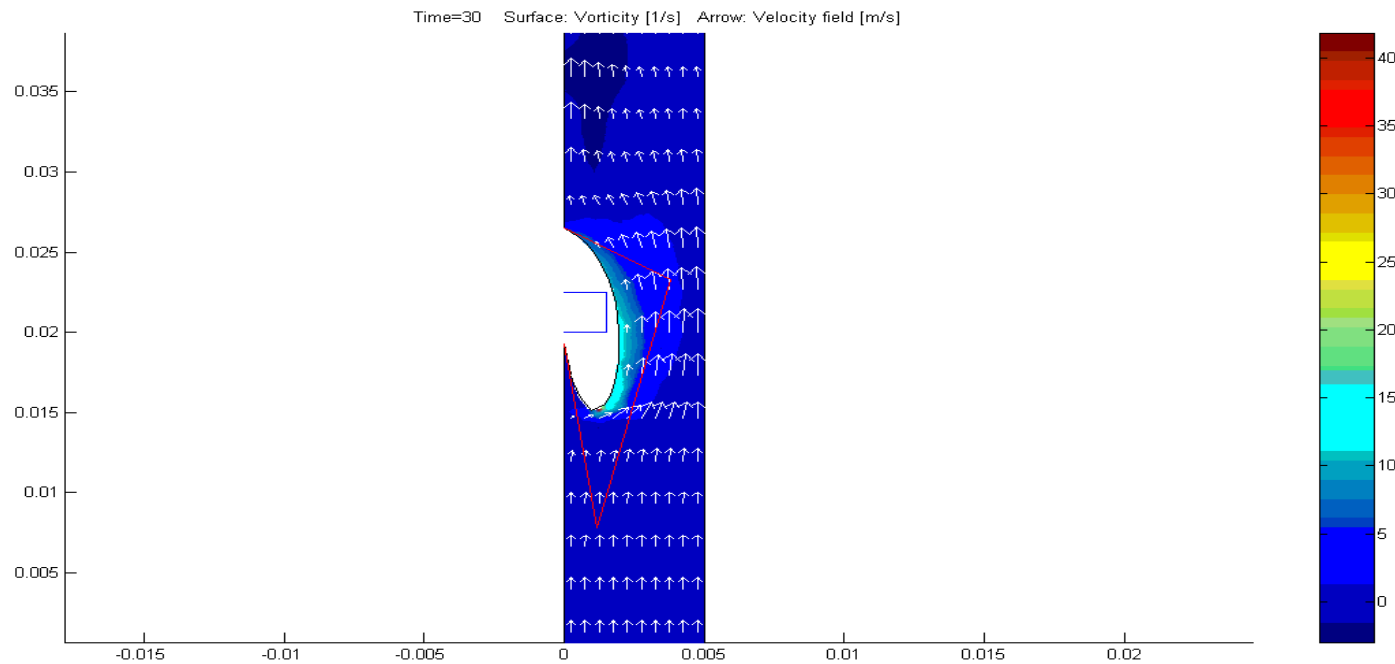


# Populácia v 10. generácii výpočtu GA



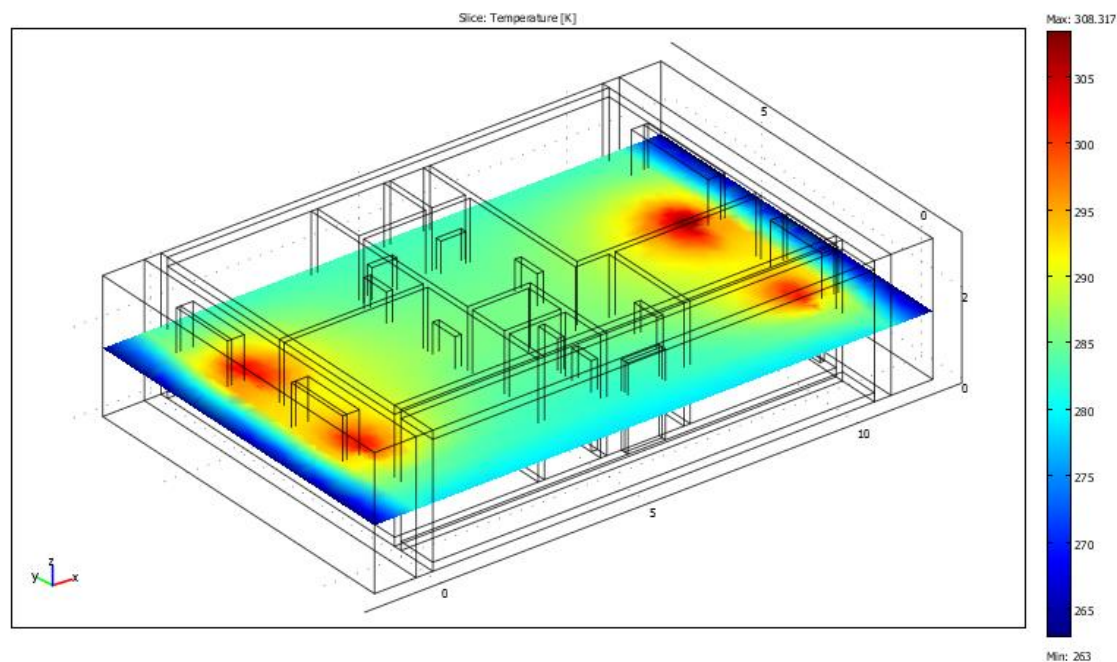


# výsledok



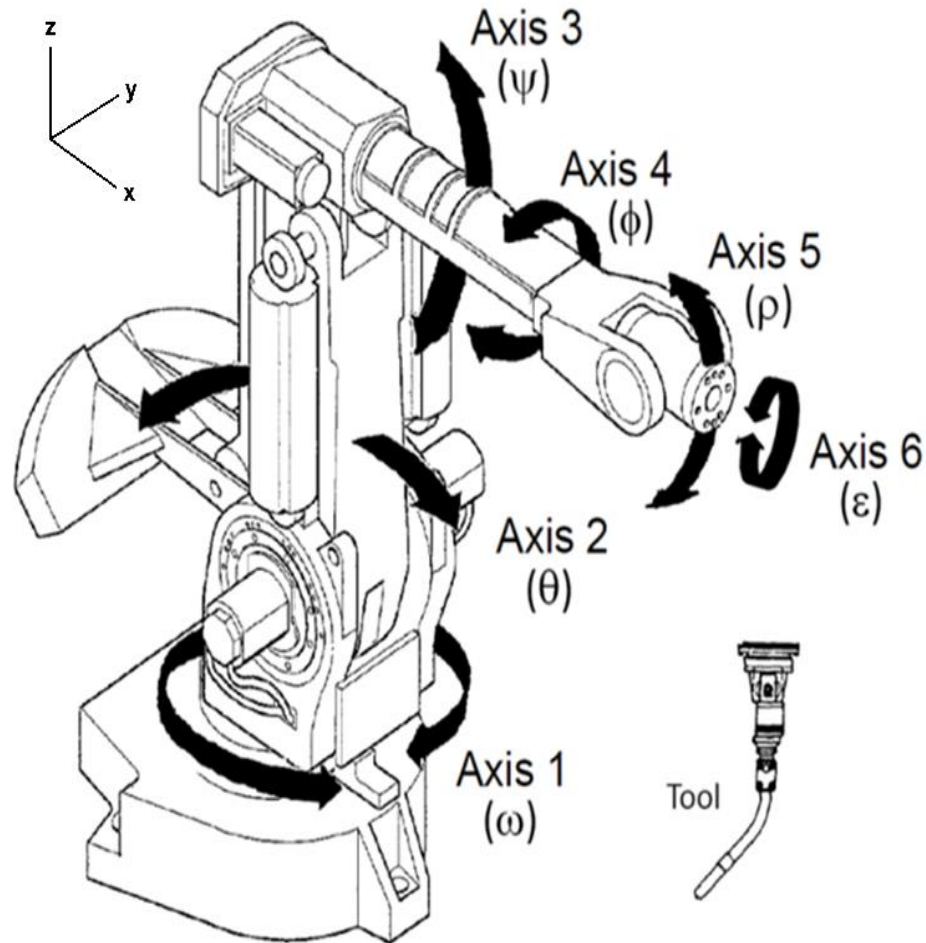
Účelová funkcia reprezentuje mieru turbulencií v potrubí, ktoré sa minimalizujú

# Optimalizácia distribúcie teploty v byte (SW: Comsol)

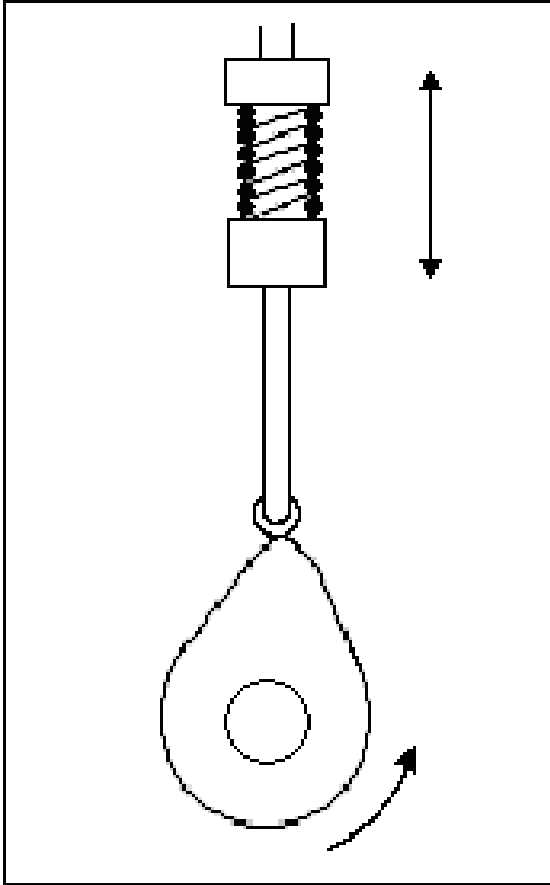


# Optimalizácia trajektórie robotického ramena – príklad ( Robot ABB IRB 6400 )

[http://www.youtube.com/watch?v=p2wCSyl\\_f6M](http://www.youtube.com/watch?v=p2wCSyl_f6M)



# Návrh tvaru vačky otáčavého stroja (GA)

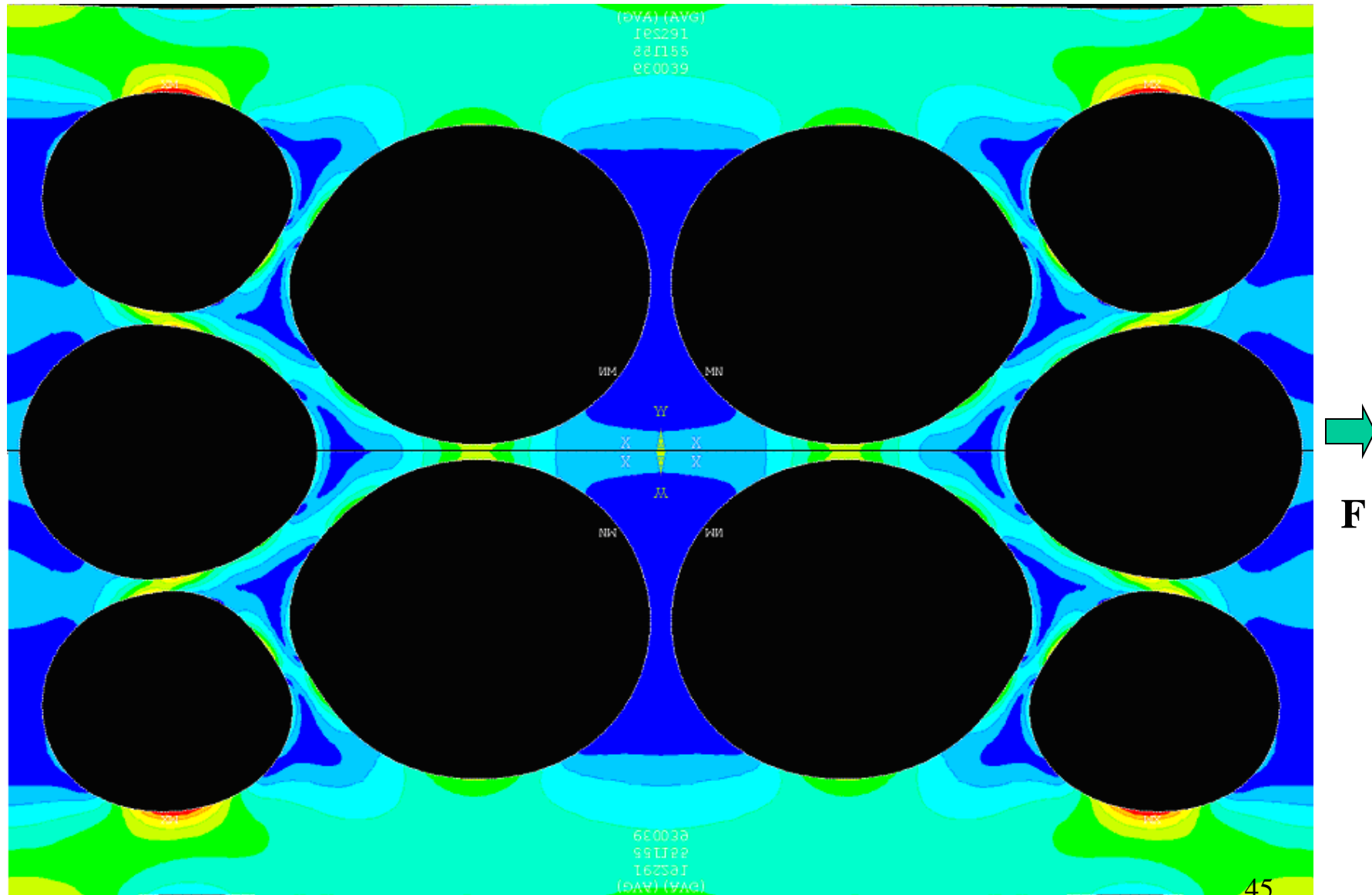


Reťazec: tvar a rozmery  
vačky zakódované pomocou  
parametrov B-spline funkcie

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

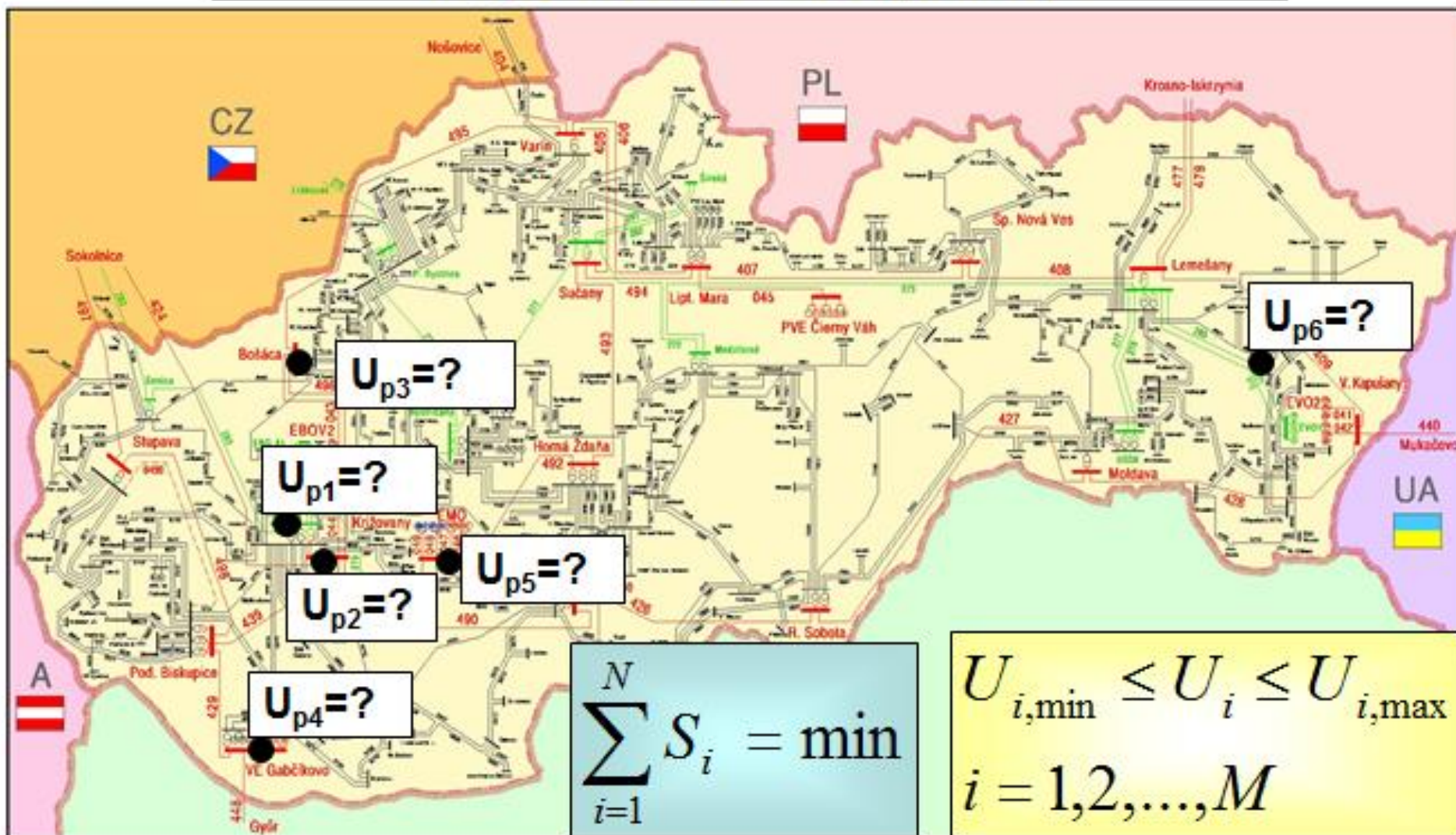


# Optimalizácia platne namáhanej silou (GA) (SW: ANSYS)





# Optimalizácia ustáleného stavu napätí v ES SR (TRN)



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

# **Zhrnutie:**

## **Bio-inšpirované optimalizačné metódy**

- **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**
- **Vysoká výpočtová/časová náročnosť**