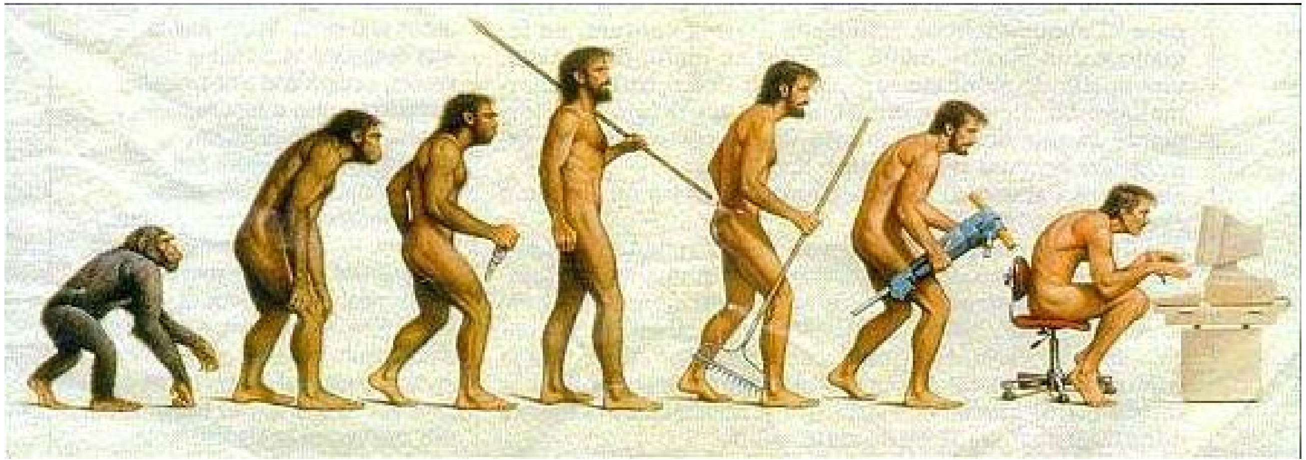
Evoluční algoritmy

- Kategorie vytvořená v 90. letech, aby se sjednotily jednotlivé metody, které využívaly „evoluční“ principy, tzn. Genetické algoritmy, Evoluční strategie a Evoluční programování (v těchto přednáškách nejmenovaná metoda taktéž vzniklá v 70. letech)

- Dále rozšiřována, zde uvedeme notaci a algoritmy vzniklé na katedře mechaniky, Fakulty stavební ČVUT

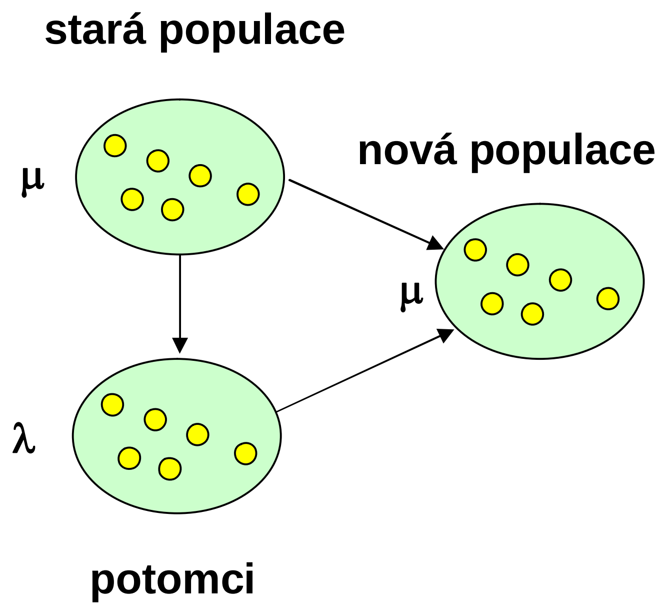
Vlastnosti EA



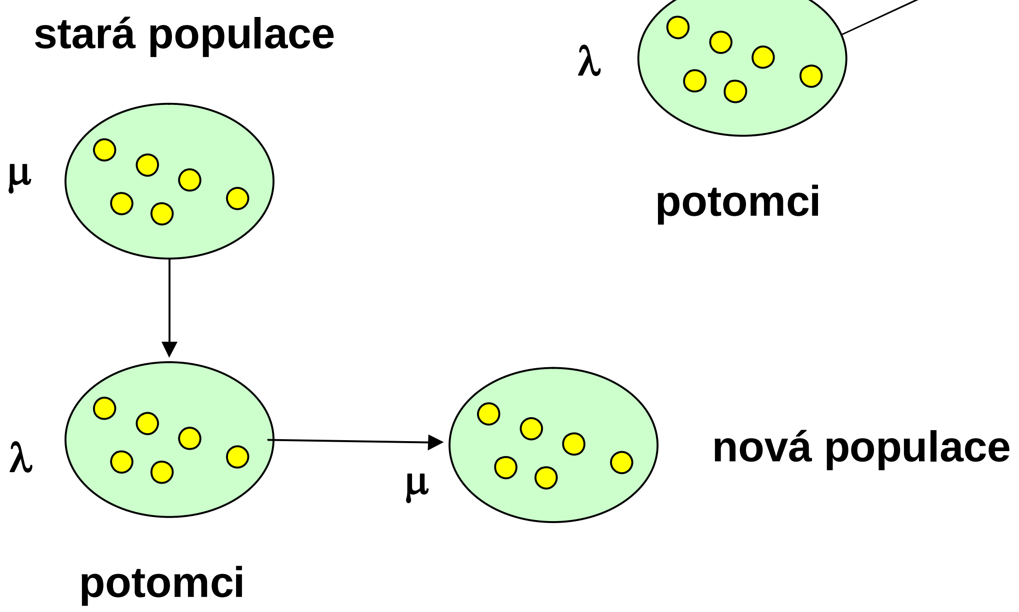
- Založeny na Darwinově myšlence „přežití nejsilnějších“
- Stochastické metody – vykazují náhodné chování
- Ne-gradientní metody- nevyžadují dokonce ani spojitost optimalizované funkce

Notace použita z ES

- **Dva druhy algoritmů:**
  - **(μ+λ)-EA**



  - **(μ,λ)-EA**



Notace operátorů

- **Kombinace tří operátorů:**
  - **Re-kombinace**

$$\mathbf{rec}_i^j \qquad I^i \to I^j \quad i \le \mu\,, \quad j \le \lambda$$

  - **Mutace**

$$\mathbf{mut}_1^1 \qquad I^1 \to I^1$$

  - **Výběr/selekce**

$$\mathbf{sel}_i^j \qquad I^i \to I^j \quad i \le \mu \,\|\, \mu{+}\lambda\,, \quad j \le \mu$$

Zápis některých algoritmů

- **Simulované žíhání**

$$opt_{SA}({}^{t+1}I) = \mathbf{sel}_2^1(\mathbf{mut}_1^1({}^tI), {}^tI) \; .$$

- **Genetický Algoritmus**

$$opt_{SGA}({}^{t+1}I^{\mu}) = \mathbf{mut}_1^1(\mathbf{rec}_2^2({}^{t+1}\overline{I}^{\lambda}), {}^{t+1}\overline{I}^{\lambda}) \; , \quad {}^{t+1}\overline{I}^{\lambda} = \mathbf{sel}_{\mu}^{\lambda}({}^tI^{\mu}) \; ,$$

- **Diferenciální evoluce**

$$opt_{DE}({}^{t+1}I^{\mu}) = \mathbf{sel}_2^1(\mathbf{rec}_4^1({}^tI, {}^tI_{best}), {}^tI) \; ,$$

- **Evoluční strategie**

$$opt_{(\mu+\lambda)-ES}({}^{t+1}I^{\mu}) = \mathbf{sel}_{\mu+\lambda}^{\mu}(\mathbf{mut}(\mathbf{rec}({}^tI^{\mu})), {}^tI^{\mu}) \; ,$$

$$opt_{(\mu,\lambda)-ES}({}^{t+1}I^{\mu}) = \mathbf{sel}_{\lambda}^{\mu}(\mathbf{mut}(\mathbf{rec}({}^tI^{\mu}))) \; ,$$

Algoritmy vzniklé na FSv

- SADE - Simplified Atavistic Differential Evolution
- RASA : Real-valued augmented simulated annealing
- IASA : Integer augmented simulated annealing

SADE algoritmus

- Simplified Atavistic Differential Evolution
- Odvozená z Diferenciální evoluce
- Kombinuje vlastnosti DE s Genetickými algoritmy
- Zejména zavádí mutace
- Web stránky autorky:
  - http://cml.fsv.cvut.cz/~anicka/

SADE algoritmus

```
void SADE ( void )
{
   FIRST_GENERATION ();
   while ( to_continue )
   {
      MUTATE ();
      LOCAL_MUTATE ();
      CROSS ();
      EVALUATE_GENERATION ();
      SELECT();
   }
}
```

SADE algoritmus

- **Nový diferenciální operátor**

$$ch_{ij} = ch_{pj} + CR\,(ch_{qj} - ch_{rj})$$

- **Přidána mutace a lokální mutace**

$$ch_{ij} = ch_{kj} + MR\,(rp_{qj} - ch_{kj})$$

- **Je použit inverzní turnajový výběr**

Parametry

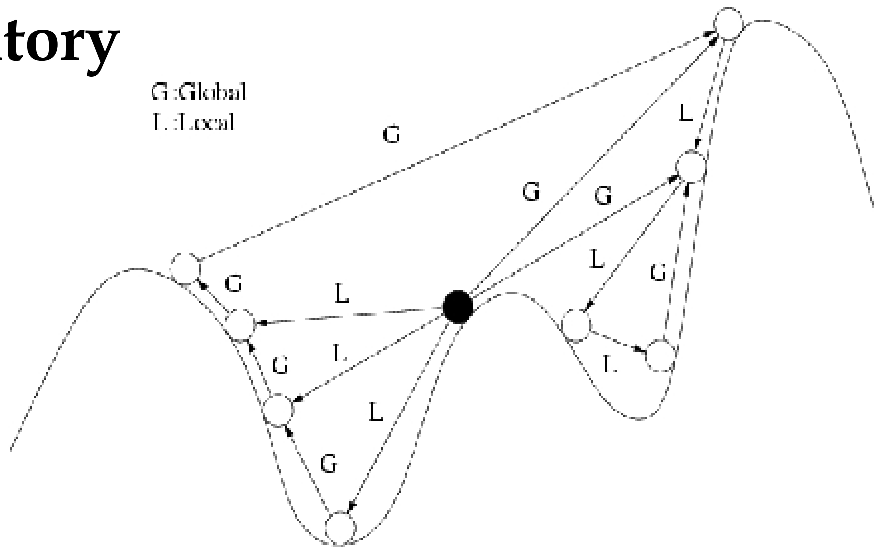
Parameter	Cheby-chev	Type 0	Beam	PUC
pop_size	$10 \times$ dim	$25 \times$ dim	$10 \times$ dim	$10 \times$ dim
CR	0.44	0.1	0.3	0.2
Radioactivity	0	0.05	0.05	0.3
MR	0.5	0.5	0.5	0.5

RASA

- Real-valued augmented simulated annealing
- Založena na principech GA a SA

RASA

- **Založena na GA**
  - **Výběr a genetické operátory**
    4 mutace, 4křížení
- **SA principy**
  - **Cooling schedule,**
    re-annealing



$$p = \frac{1}{1 + e^{\Delta C / T}} \qquad \Delta C = C_{NEW} - C_{OLD}$$

Mutace

Uniform mutation: Let $k = [1, n]$

$$\mathrm{ch}_{ij}(t+1) = \begin{cases} u(L_j, U_j), & \text{if } j = k \\ \mathrm{ch}_{ij}(t), & \text{otherwise,} \end{cases} \tag{17}$$

Boundary mutation: Let $k = u[1, n]$, $p = u(0, 1)$ and set:

$$\mathrm{ch}_{ij}(t+1) = \begin{cases} L_j, & \text{if } j = k, p < 0.5 \\ U_j, & \text{if } j = k, p \geqslant 0.5 \\ \mathrm{ch}_{ij}(t), & \text{otherwise,} \end{cases} \tag{18}$$

Non-uniform mutation: Let $k = [1, n]$, $p = u(0, 1)$ and set:

$$\mathrm{ch}_{ij}(t+1) = \begin{cases} \mathrm{ch}_{ij}(t) + (L_j - \mathrm{ch}_{ij}(t))f, & \text{if } j = k, p < 0.5 \\ \mathrm{ch}_{ij}(t) + (U_j - \mathrm{ch}_{ij}(t))f, & \text{if } j = k, p \geqslant 0.5 \\ \mathrm{ch}_{ij}(t), & \text{otherwise,} \end{cases} \tag{19}$$

where $f = u(0, 1)(T_t/T_0)^{\mathtt{b}}$ and $\mathtt{b}$ is the shape parameter.

Multi-non-uniform mutation: Apply non-uniform mutation to all variables of CH_i.

Křížení

Simple cross-over: Let $k = [1, n]$ and set:

$$\mathrm{ch}_{il}(t+1) = \begin{cases} \mathrm{ch}_{il}(t), & \text{if } l < k \\ \mathrm{ch}_{jl}(t), & \text{otherwise,} \end{cases}$$

$$\mathrm{ch}_{jl}(t+1) = \begin{cases} \mathrm{ch}_{jl}(t), & \text{if } l < k \\ \mathrm{ch}_{il}(t), & \text{otherwise.} \end{cases}$$

Simple arithmetic cross-over: Let $k = u[1, n]$, $p = u(0, 1)$ and set:

$$\mathrm{ch}_{il}(t+1) = \begin{cases} p\,\mathrm{ch}_{il}(t) + (1-p)\,\mathrm{ch}_{jl}(t), & \text{if } l = k \\ \mathrm{ch}_{il}(t), & \text{otherwise,} \end{cases} \tag{20}$$

$$\mathrm{ch}_{jl}(t+1) = \begin{cases} p\,\mathrm{ch}_{jl}(t) + (1-p)\,\mathrm{ch}_{il}(t), & \text{if } l = k \\ \mathrm{ch}_{jl}(t), & \text{otherwise.} \end{cases} \tag{21}$$

Whole arithmetic cross-over: Simple arithmetic cross-over applied to all variables of CH_i and CH_j.

Heuristic cross-over: Let $p = u(0, 1)$, $j = [1, n]$ and $k = [1, n]$ such that $j \neq k$ and set:

$$\mathrm{CH}_i(t+1) = \mathrm{CH}_i(t) + p(\mathrm{CH}_j(t) - \mathrm{CH}_k(t)). \tag{22}$$

Parametry

Parameter	Beam	Others
pop_size	64	32
q	0.04	0.04
p_uni_mut	0.525	0.05
p_bnd_mut	0.125	0.05
p_nun_mut	0.125	0.05
p_mnu_mut	0.125	0.05
p_smp_crs	0.025	0.15
p_sar_crs	0.025	0.15
p_war_crs	0.025	0.15
p_heu_crs	0.025	0.35
b	0.25	2.0
T_frac	10^{-2}	10^{-10}
T_frac_min	10^{-4}	10^{-14}
T_mult	0.9	0.9
num_success_max	$10 \times$ pop_size	$10 \times$ pop_size
num_counter_max	$50 \times$ pop_size	$50 \times$ pop_size
num_heu_max	20	20
precision (step 4a)	See Section 4.3	10^{-4}

IASA

- **Integer augmented simulated annealing**
- **Kombinuje GA, SA, ES a DE principy**

IASA

- **Diferenciální křížení**

$$ch_{ij} = ch_{pj} + u(0.0, CR)(ch_{qj} - ch_{rj})$$

- **Mutace**

$$ch_{ij} = ch_{kj} + N\left(0.0, 0.5\left|ch_{kj} - ch_{pj}\right| + 1\right)$$

- **Celočíselné kódování**

```
T = T_max, t = 0
vytvoř P_0, ohodnoť P_0
while (not zastavovací_podmínka) {
    count = succ = 0
    while( count < countmax & succ < succmax)
    {
        count = count + 1, t = t + 1
        vyber jedince I_t z P_t-1
        vyber operátor O
        změň I_t operátorem O, I_t' je výsledek
        p = 1/(1+exp ((F(I_t') - F(I_t))/T))
        if ( náhodné_číslo u[0, 1] <= p ) {
                succ = succ + 1
                vlož I_t' do P_t
                ohodnoť P_t
        }
    }
    sniž T
}
```

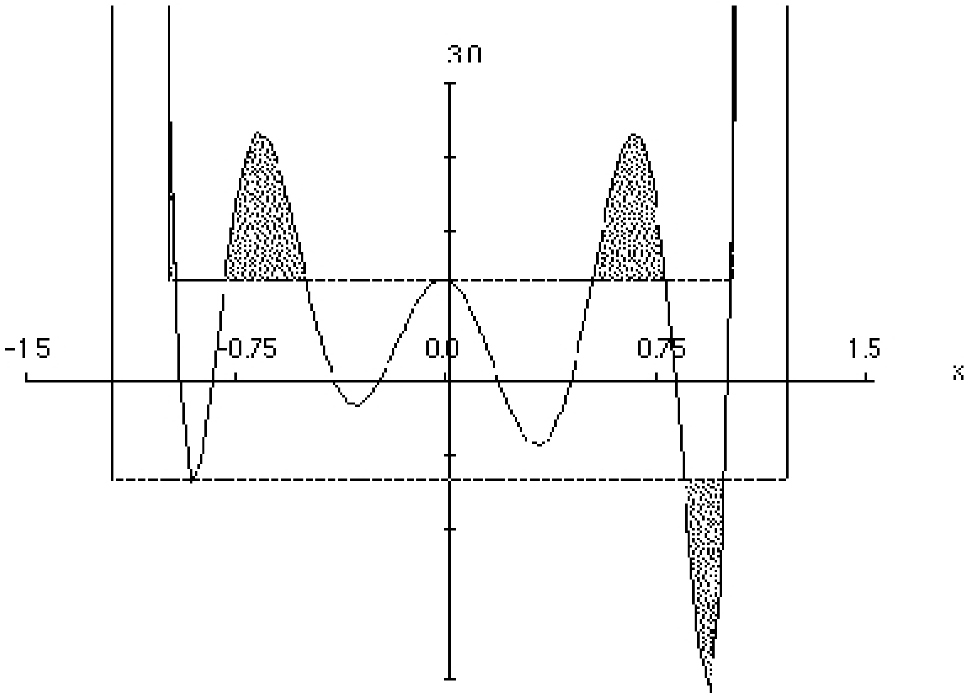
Parametry

Parameter	Cheby-chev	Type 0	Beam	PUC
OldSize	80	900	180	200
NewSize	5	600	250	100
T_max	10^{-5}	10^{-5}	10^{-4}	10^{-1}
T_min	10^{-7}	10^{-10}	10^{-5}	10^{-5}
SuccessMax	1000	1000	1000	1000
CounterMax	5000	5000	5000	5000
TminAtCallsRate	19%	100%	25%	20%
CrossoverProb	97%	92%	60%	90%
CR	0.5	0.6	1.3	1.0

Porovnání čtyř metod

- Každý algoritmus byl navržen pro jiný problém
  - DE $\Leftarrow$ Chebychevův problém
  - IASA $\Leftarrow$ Návrh ŽB nosníku
  - RASA $\Leftarrow$ Optimální periodická buňka
  - SADE $\Leftarrow$ Type "0" funkce
- Kombinace „umělých“ i „praktických“ příkladů
- Validace jednotlivých metod

Chebychevův problém



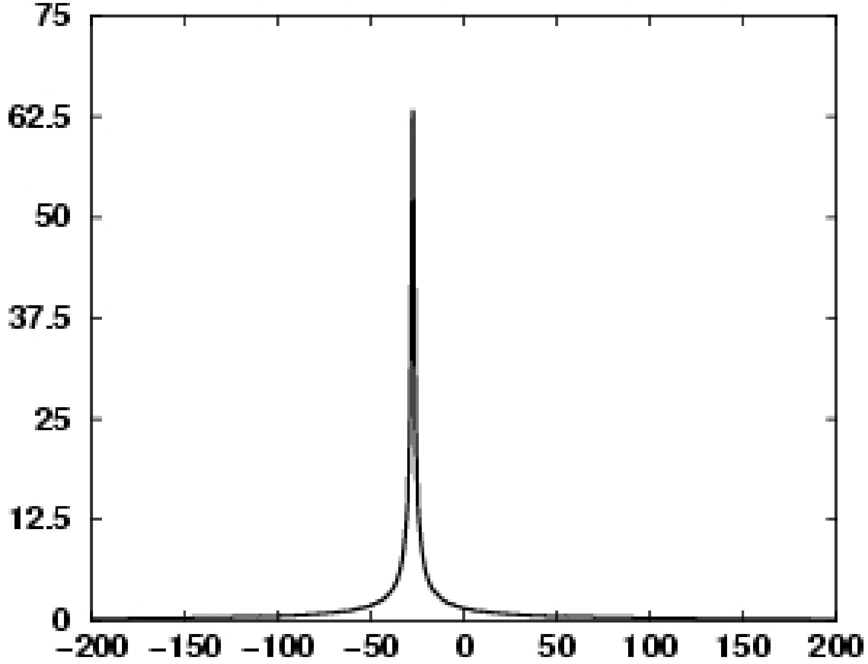
- **Vměstnání polynomu do daných hranic**

- **Polynom dán výrazem**

$$f(x) = \sum_{i=0}^{n} a_i x^i$$

- **Minimalizace vyšrafované oblasti**

Type 0 funkce



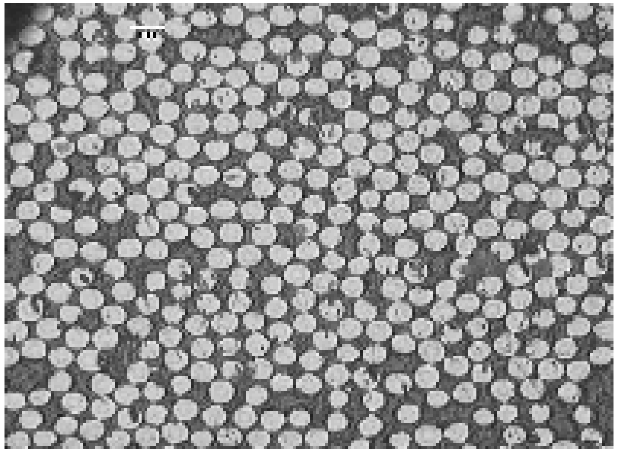
- **Nalezení jedinného optima**

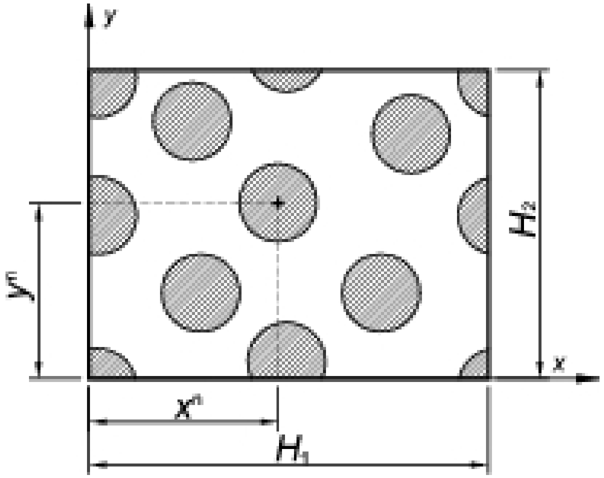
- **Předpis funkce**

$$f(x) = y_0 \left(\frac{\pi}{2} - \arctan \frac{\| x - x_0 \|}{r_0} \right)$$

- **Nalezení vrcholu v N-dimensionálním prostoru**

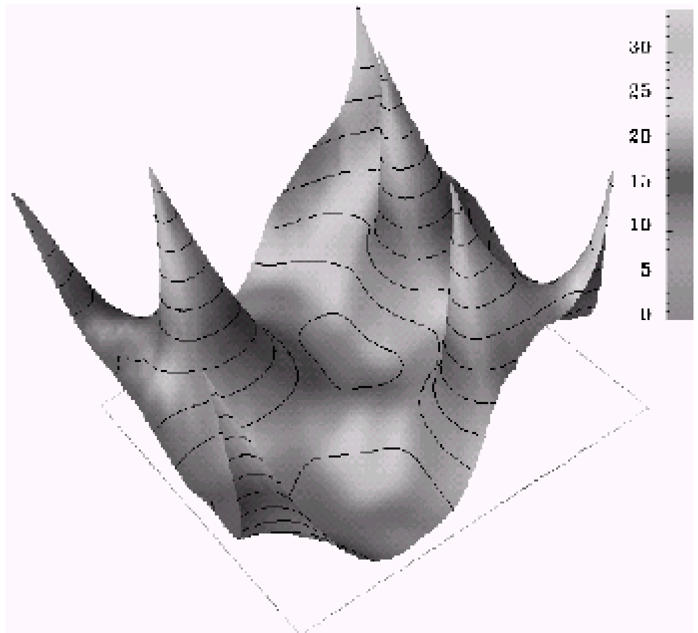
Tvorba jednotkové periodické buňky





- **Hledání buňky se stejnými statistickými parametry jako má reálný materiál**

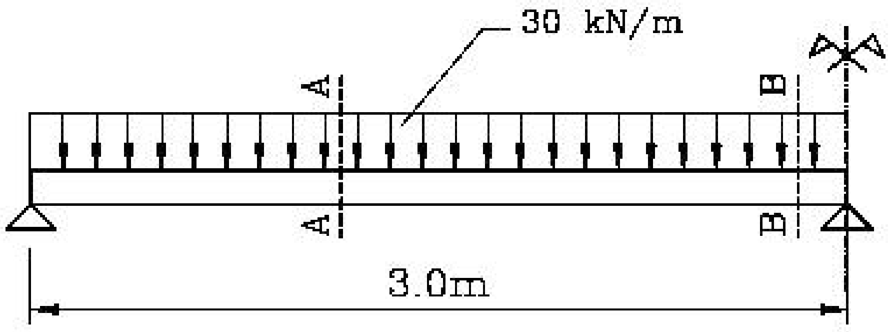
Tvorba jednotkové periodické buňky



- Náročnost problému

Methoda	
BFGS/CG	0/20
Simplex	2/20
Randomized local search (Yuret,1994)	5/20

Návrh ŽB nosníku



- **Nalezení optimálního návrhu nosníku**

- **Funkce vyjádřená v ceně konstrukce**

V_C Objem betonu

P_C Cena betonu

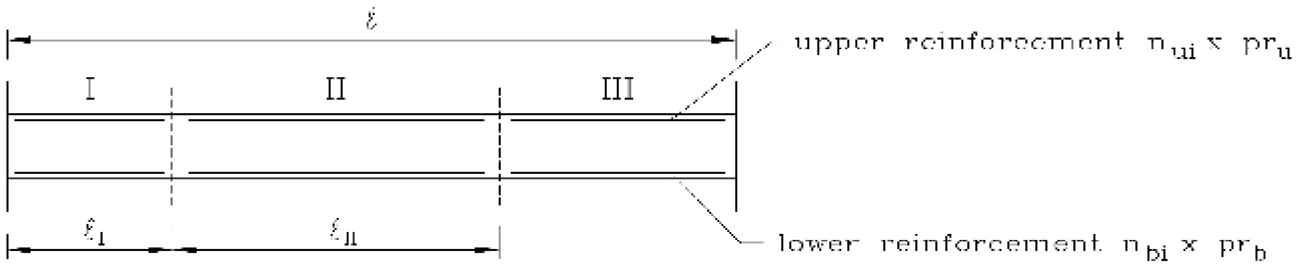
W_S Hmotnost oceli

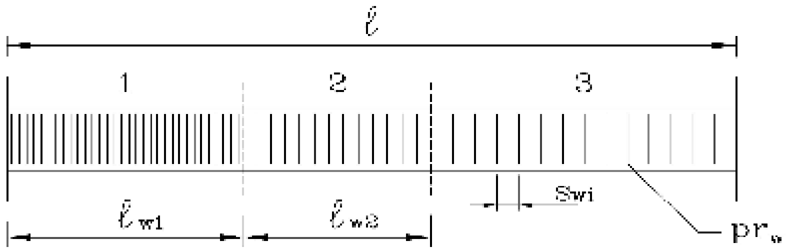
P_S Cena oceli

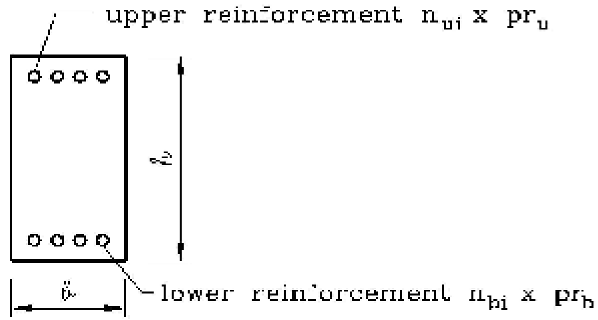
$$f(x) = V_C P_C + W_S P_S + \sum pf_i$$

Návrh ŽB nosníku - proměnné

- Celkem 18 proměnných
- Všechny celočíselné nebo vybírané ze seznamu diskrétních hodnot

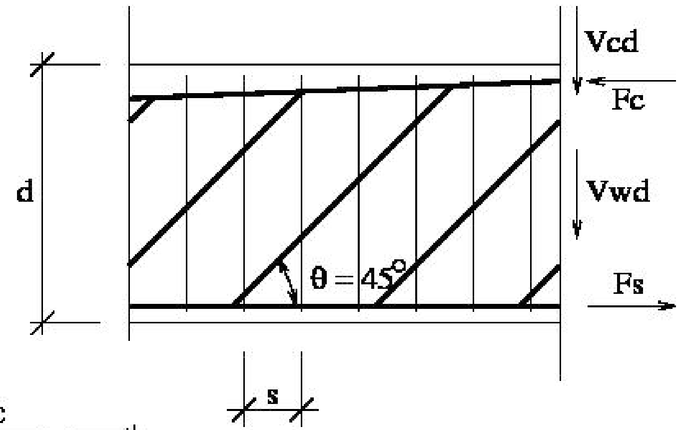




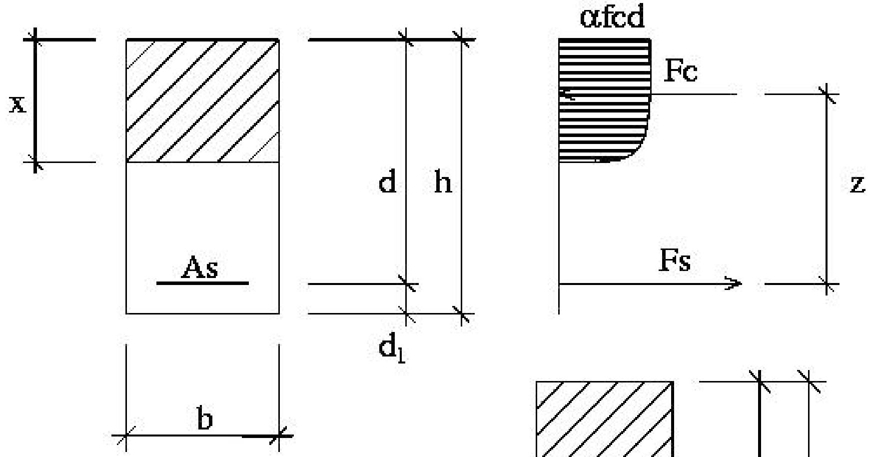


Posudek: EC 2

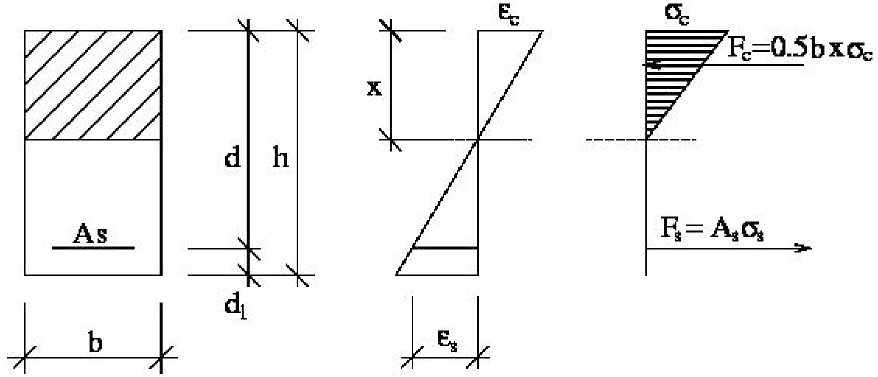
- Smyk



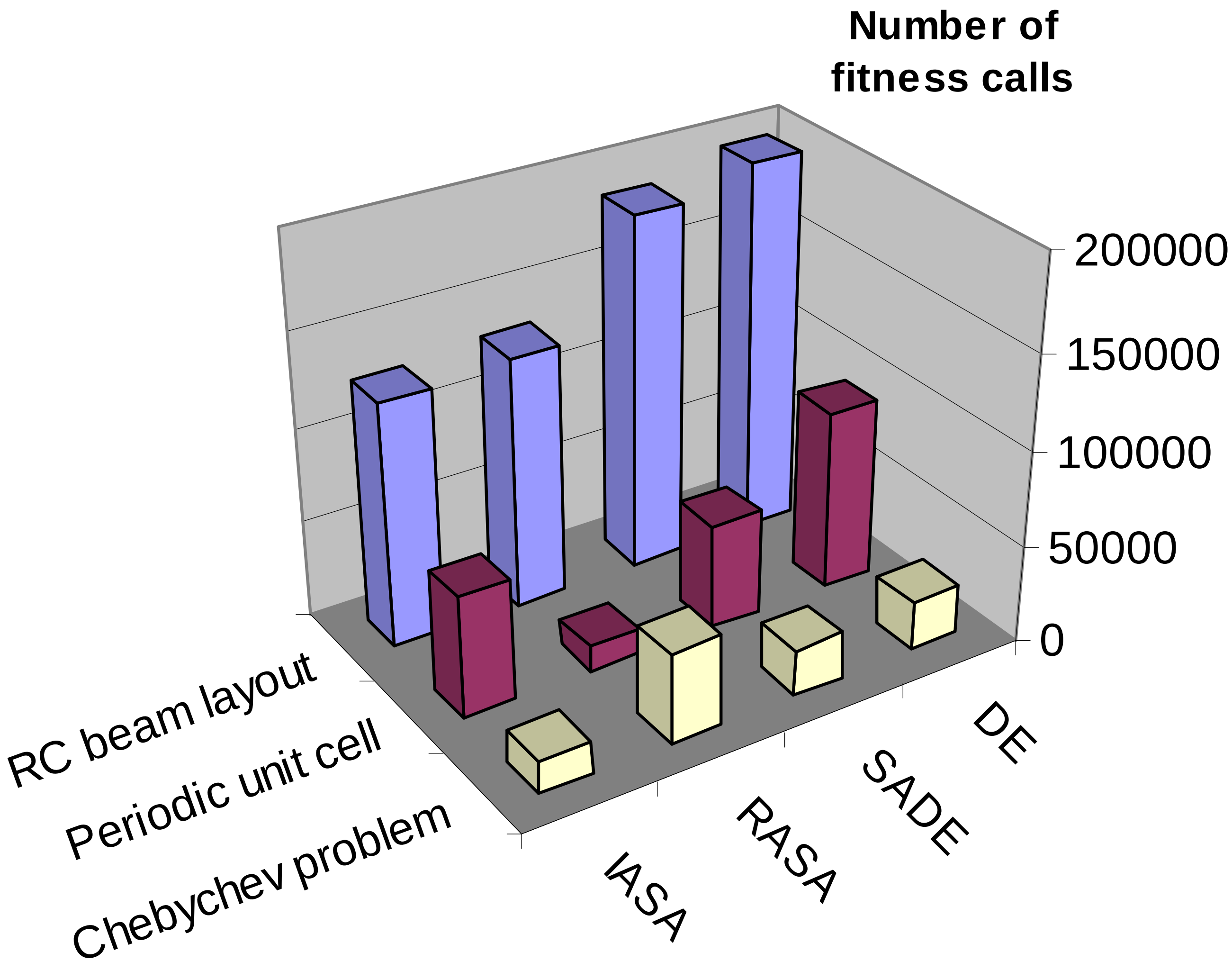
- Ohyb



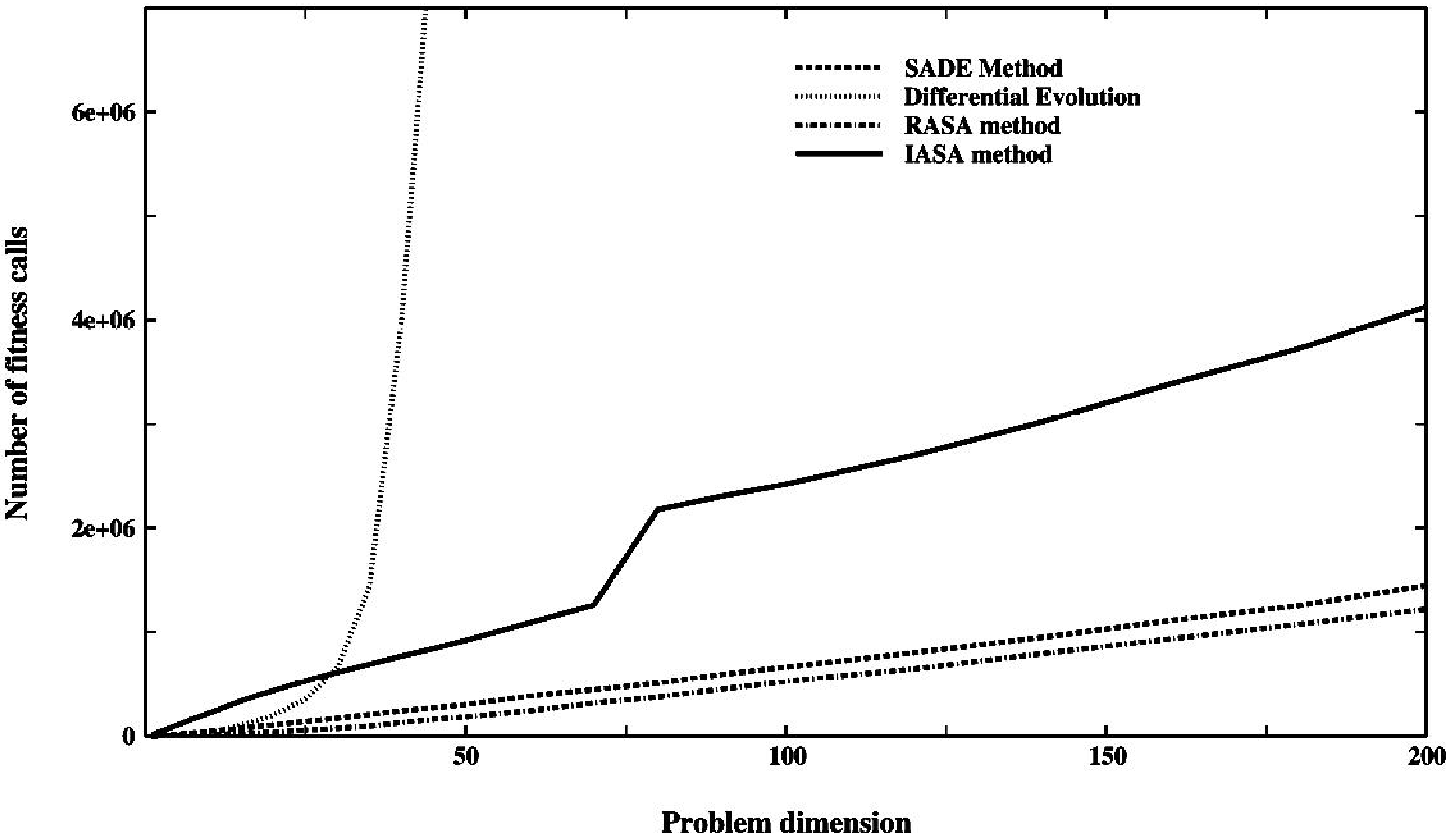
- MSP



Výsledky porovnání



Type 0 funkce



Výsledky porovnání

	DE	SADE	RASA	IASA
Chebychev polynomial problem	3	2	4	1
Type 0 test function	4*	2	1	3
Reinforced concrete beam layout	4	3	2	1
Periodic unit cell construction	4	2	1	3
Σ	15	9	8	8

Výsledky porovnání

- Počty parametrů algoritmu

DE	**4**
SADE	**5**
RASA	**17**
IASA	**9**

Reference

[1] Goldberg, D. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

[2] Michalewicz, Z. (1992). Genetic Algorithms + Data Structures = Evolution Programs. AI Series. Springer-Verlag, New York.

[3] Hrstka, O., Kučerová, A., Lepš, M., and Zeman, J. (2003). A competitive comparison of different types of evolutionary algorithms. Computers & Structures, 81(18–19):1979–1990.

[4] Lepš, M. (2005). Single and Multi-Objective Optimization in Civil Engineering with Applications, PhD thesis, CTU in Prague.

Prosba. V případě, že v textu objevíte nějakou chybu nebo budete mít námět na jeho vylepšení, ozvěte se prosím na **`matej.leps@fsv.cvut.cz`**.

Datum poslední revize: 19.11.2007
Verze: 001