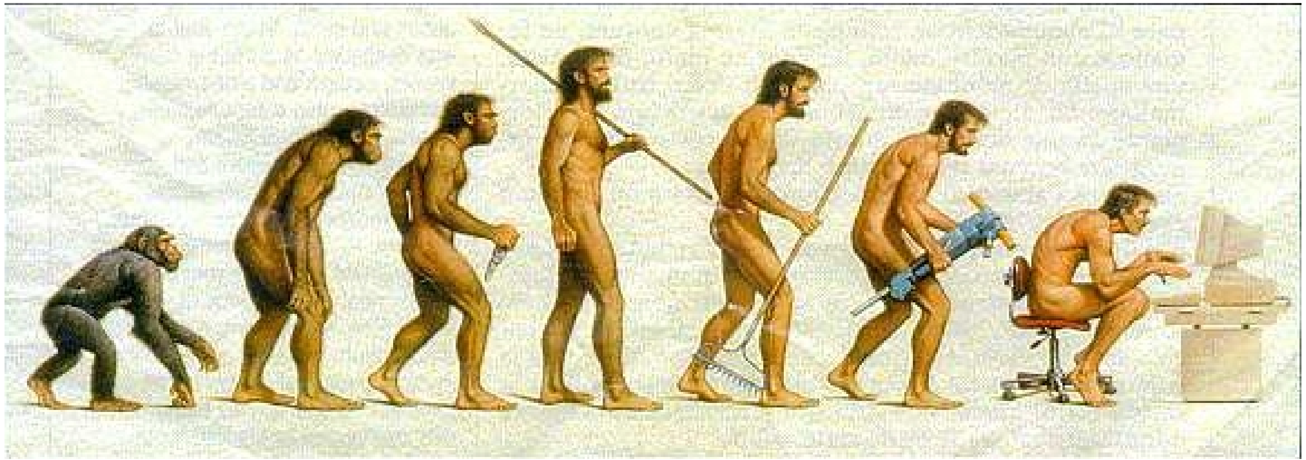
Evolutionary Algorithms

- Category created in 90's to unify individual "evolutionary" methods, i.e. Genetic Algorithms, Evolution Strategies and Evolutionary Programming (in these lectures not mentioned method developed also in 70's)

- Still developing area; only notation and methods developed at Department of Mechanics, Faculty of Civil Engineering at CTU Prague (FCE), will be presented

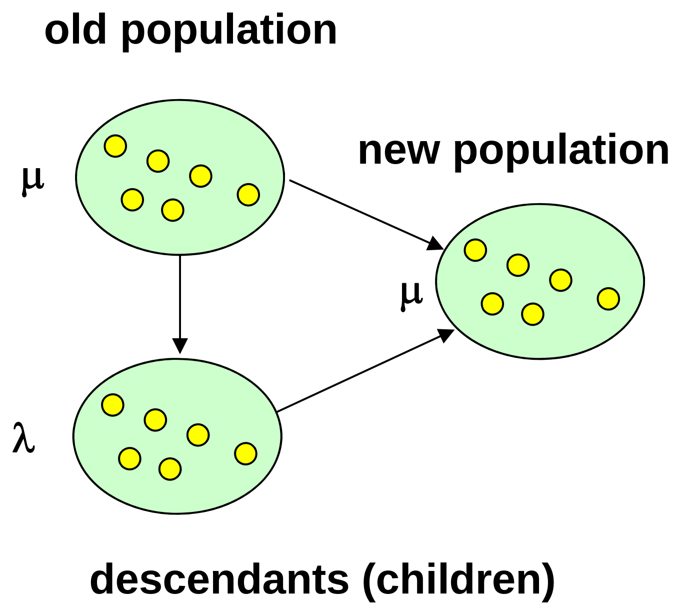
Properties of EAs



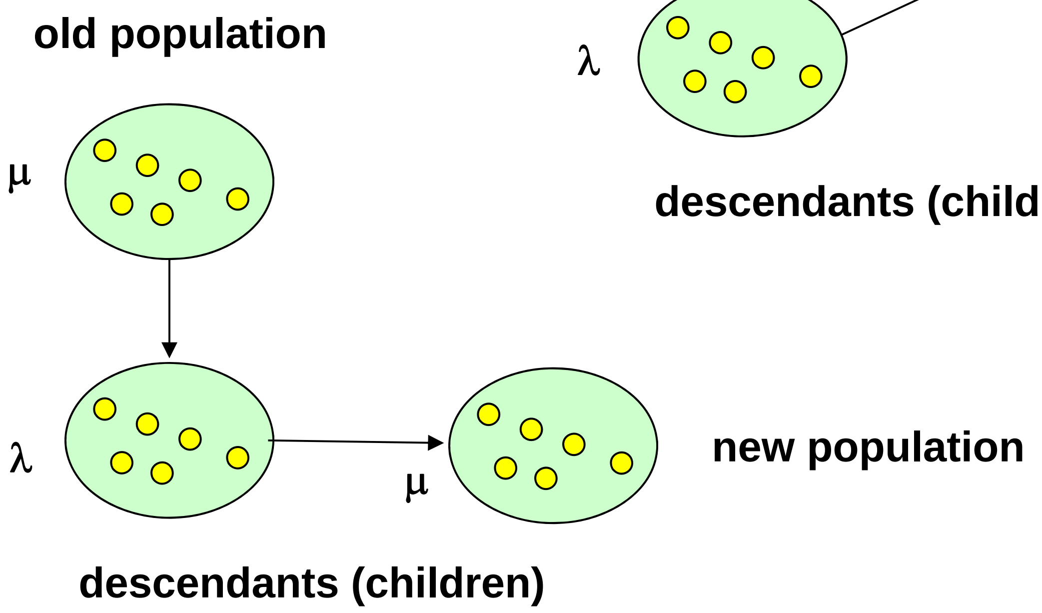
- Based on Darwinian idea „survival of the fittest“
- Stochastic optimization methods – produce random behavior
- Non-gradient methods – no need for function continuity

Notation taken from ES's

- **Two types of algorithms:**
  - **$(\mu+\lambda)$-ES**



  - **(μ,λ)-ES**



Operators' notation

- **Combination of three operators:**
  - **Recombination**

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

  - **Mutation**

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

  - **Selection**

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

Description of several algorithms

- **Simulated Annealing**

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

- **Genetic Algorithm**

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

- **Differential Evolution**

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

- **Evolution Strategies**

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

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

Algorithms developed at FCE

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

SADE algorithm

- **Simplified Atavistic Differential Evolution**
- **Based on Differential Evolution**
- **Combines GA and DE principles**
- **Mainly introduces mutation**
- **Author's webpage:**
  - **http://klobouk.fsv.cvut.cz/~anicka/**

SADE algorithm

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

SADE algorithm

- **New differential operator**

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

- **Mutation and local mutation added**

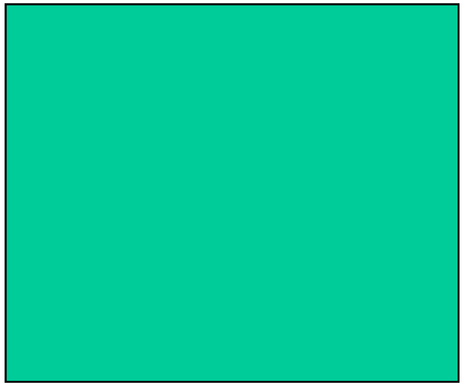
$$ch_{ij} = ch_{kj} + MR\,(rp_{qj} - ch_{kj})$$

- **Inverse tournament selection**

Parameters

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

Example of SADE with 20 dimensions

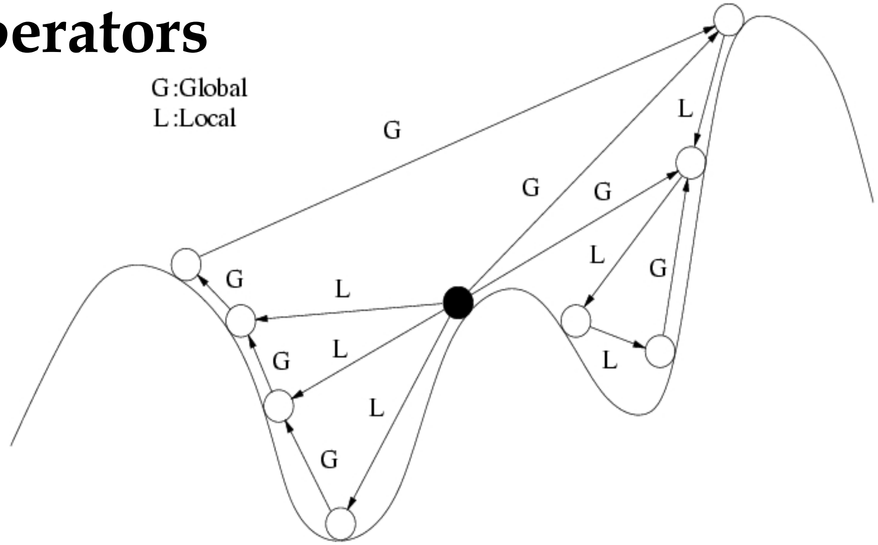


RASA

- Real-valued augmented simulated annealing
- Based on GA and SA principles

RASA

- **GA origins**
  - **Selection and genetic operators**
    4 mutations, 4 crossover



- **SA ideas**
  - **Cooling schedule,**
    re-annealing

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

Mutation

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.

Crossover

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:

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

Parameters

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 × pop_size	10 × pop_size
num_counter_max	50 × pop_size	50 × pop_size
num_heu_max	20	20
precision (step 4a)	See Section 4.3	10^{-4}

IASA

- **Integer augmented simulated annealing**
- **Combines GA, SA, ES and DE principles**

Integer coding

- Function given by $f(x) = f(x_1, x_2, \ldots, x_n)$

$$x_i \in D \subseteq \{N, R\} \quad \min_i \le x_i \le \max_i$$

- Then, positive, integer representation for precision p_i reads

$$y_i = \left\lfloor \frac{x_i - \min_i}{p_i} \right\rfloor$$

- With inverse relationship

$$x_i = y_i p_i + \min_i$$

Integer coding

- E.g.: $x = 314.151$, $min_x = 0.0$, $p_x = 0.001$
  $y = 314151$
- The same expression can be used also e.g. for rounding:
  - To integers: $p_i = 1$
  - To even numbers: $p_i = 2$ and min_i should be even
- Note.: Precision of real numbers is not infinite in computers (they are stored in binary code as well!)

IASA

- **Differential crossover**

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

- **Mutation**

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

- **Integer coding**

```
T = Tmax, t = 0
create P0, evaluate P0
while (not stopping_criterion) {
    count = succ = 0
    while( count < countmax & succ < succmax)
    {
        count = count + 1, t = t + 1
        select individuals It from Pt-1
        select operator O
        alter It with O, It' is result
        evaluate It'
        p = 1/(1+exp ((F(It') - F(It))/T))
        if ( random_number u[0, 1] <= p ) {
                succ = succ + 1
                insert It' into Pt
        }
    }
    decrease T
}
```

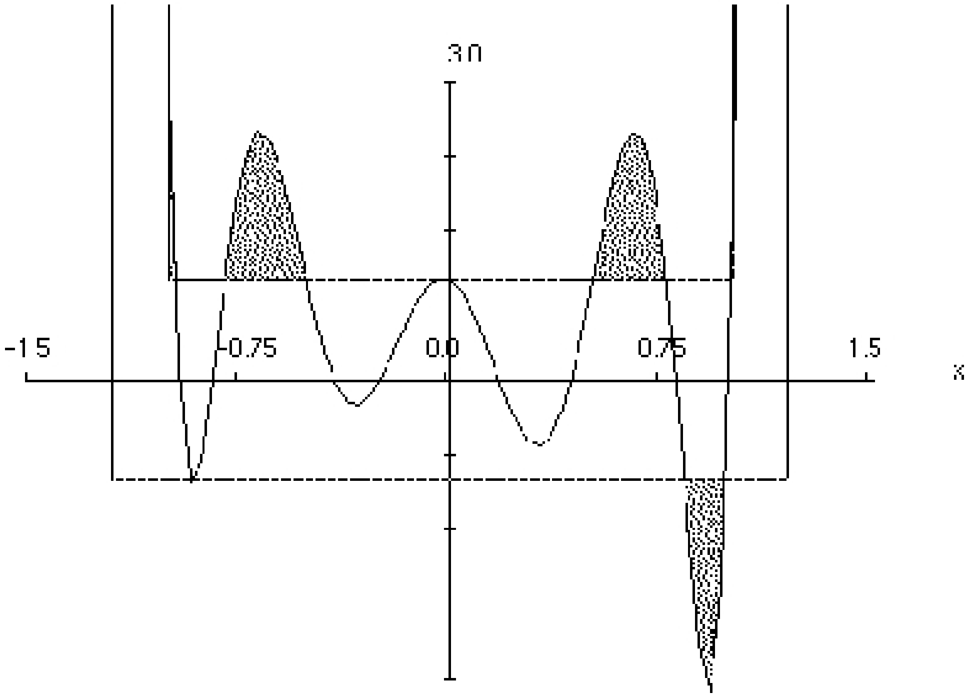
Parameters

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

Comparison of four methods

- Each algorithm was designed for one particular problem
  - DE $\Leftarrow$ Chebychev polynomial problem
  - IASA $\Leftarrow$ Design of RC beam
  - RASA $\Leftarrow$ Optimal periodic cell
  - SADE $\Leftarrow$ Type “0” function
- Combination of “artificial” as well as “practical” problems
- Validation of individual methods

Chebychev polynomial problem

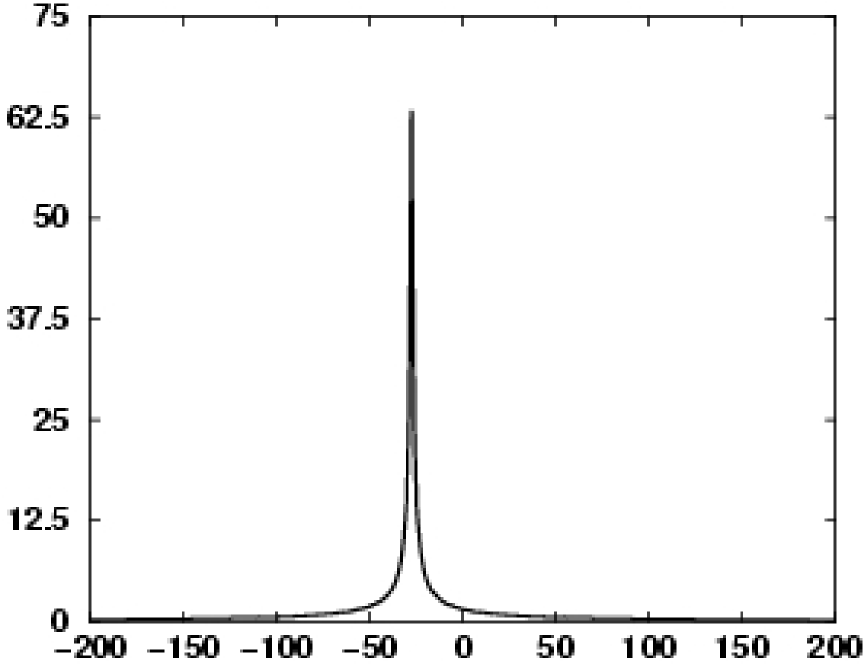


- **Fitting a polynomial into given bounds**
- **Polynomial given by**

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

- **Minimization of hashed areas**

Type 0 function



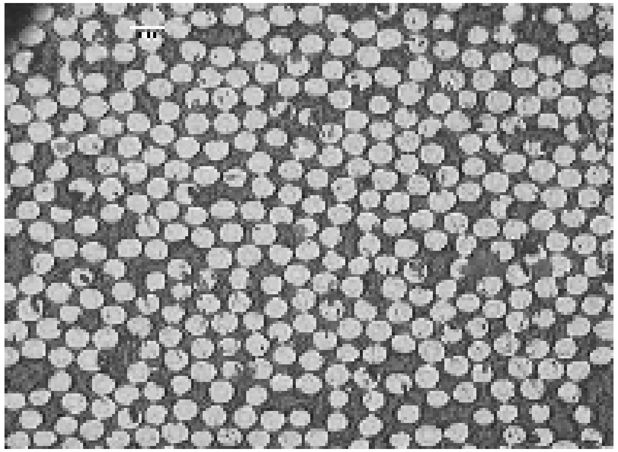
- **One global optimum**

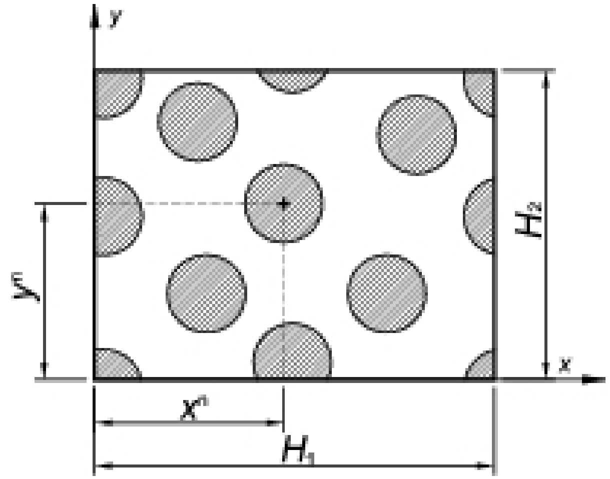
- **Function given by**

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

- **N-dimensional problem**

Periodic Unit Cell construction

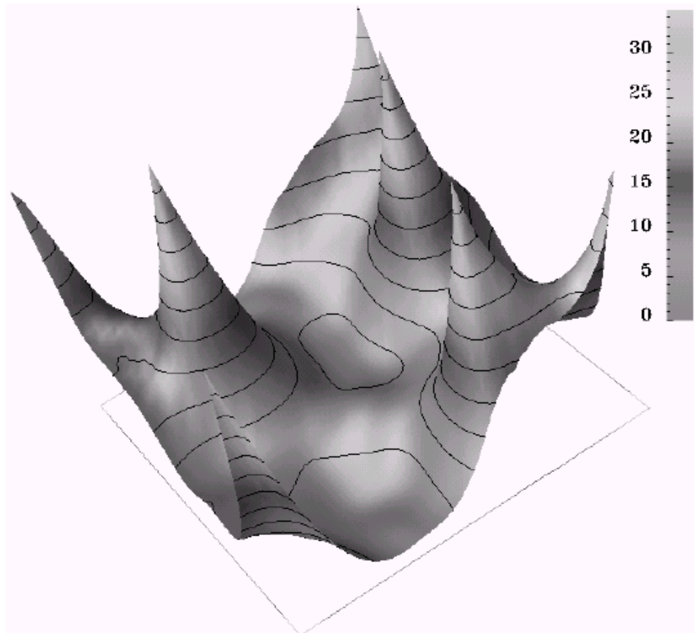




• **Search for PUC with same statistical descriptions as real material**

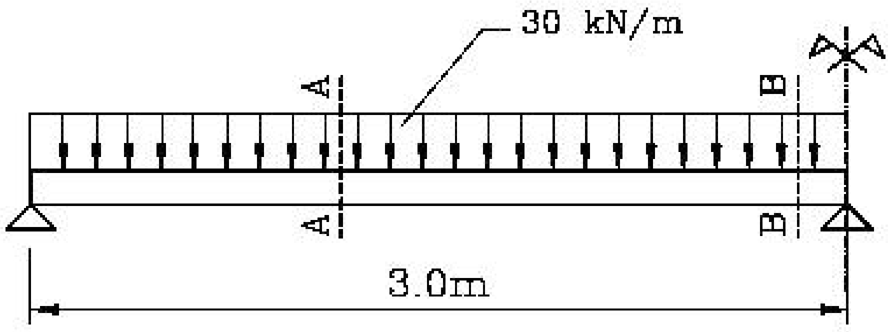
Periodic Unit Cell construction

- Problem's difficulty



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

Design of RC beam



- **Sear for optimal design**

- **Function as a price of the structure**

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

V_C Volume of concrete

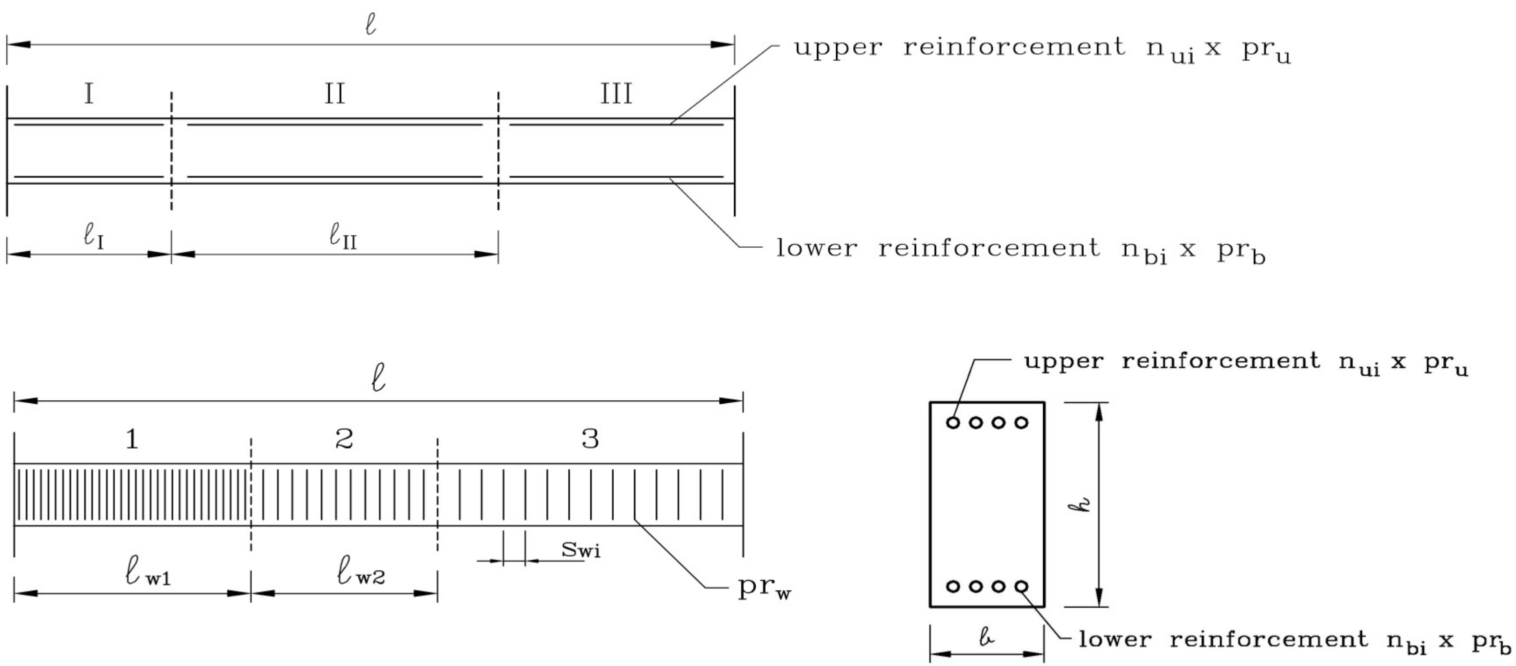
P_C Price of concrete

W_S Weight of steel

P_S Price of steel

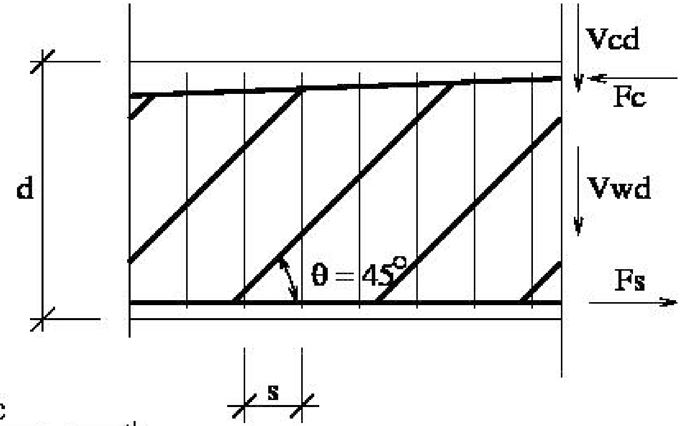
Design of RC beam: variables

- Overall 18 variables
- All of them discrete or from the discrete list

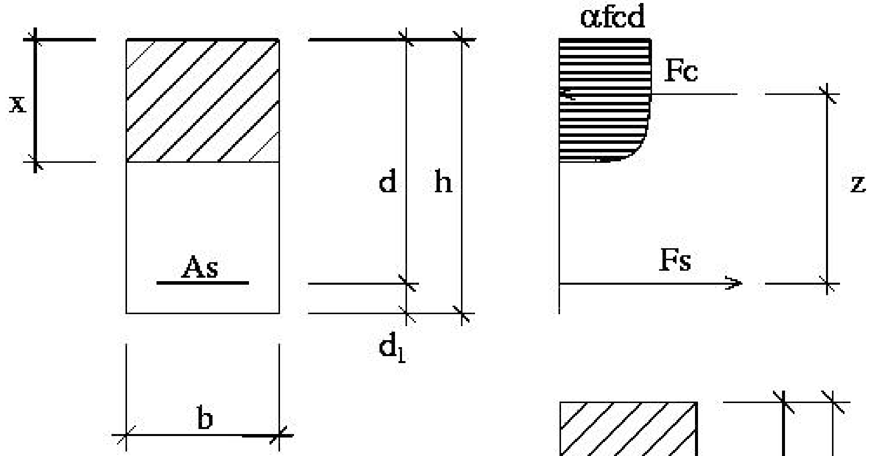


Check by EC 2 standard

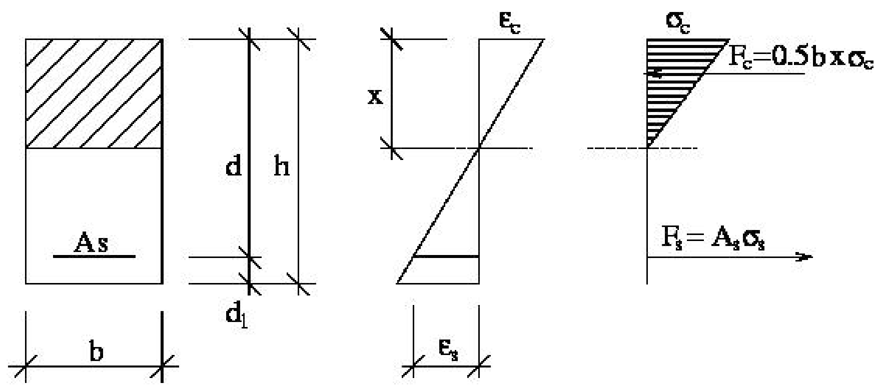
- Shear



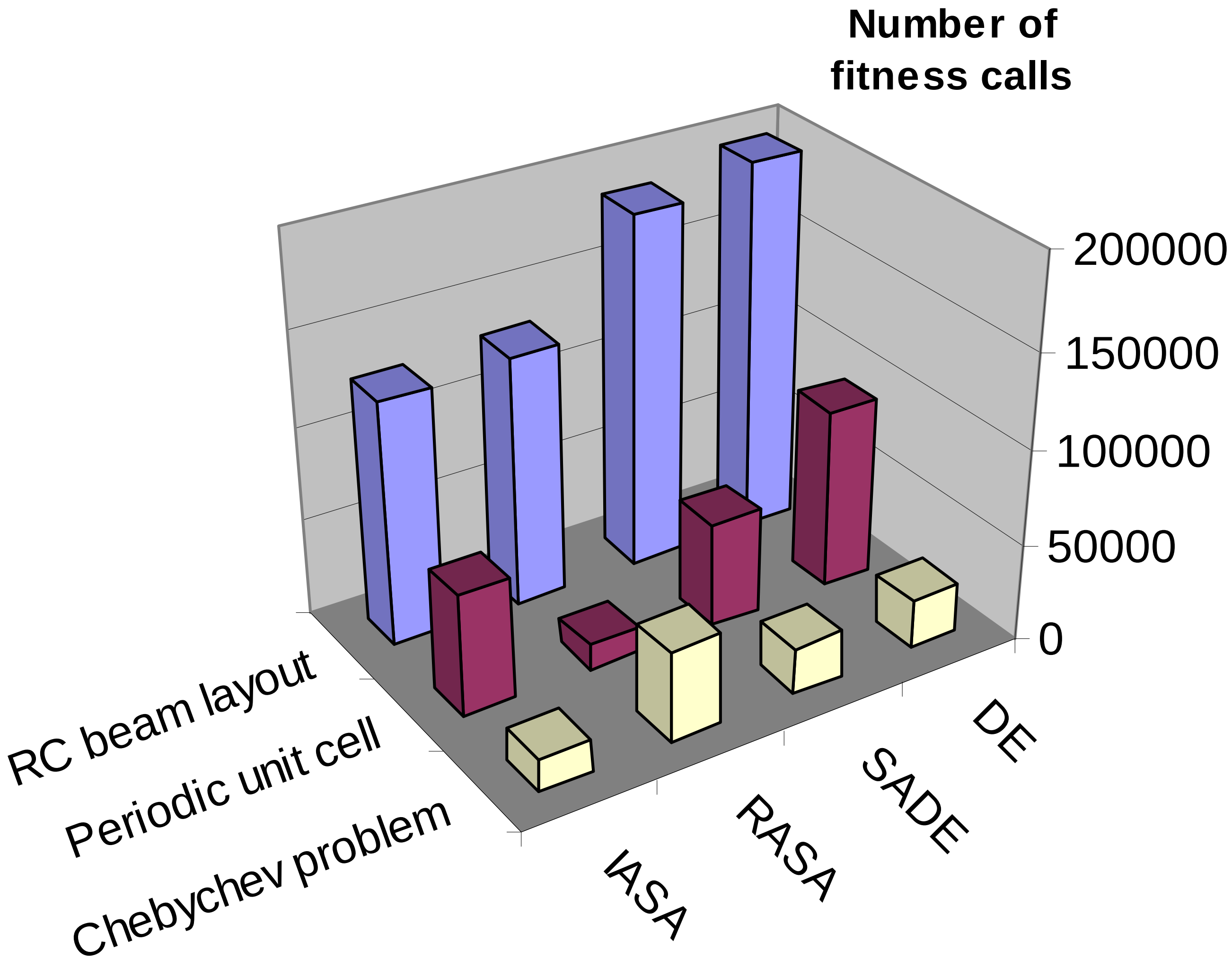
- Bending



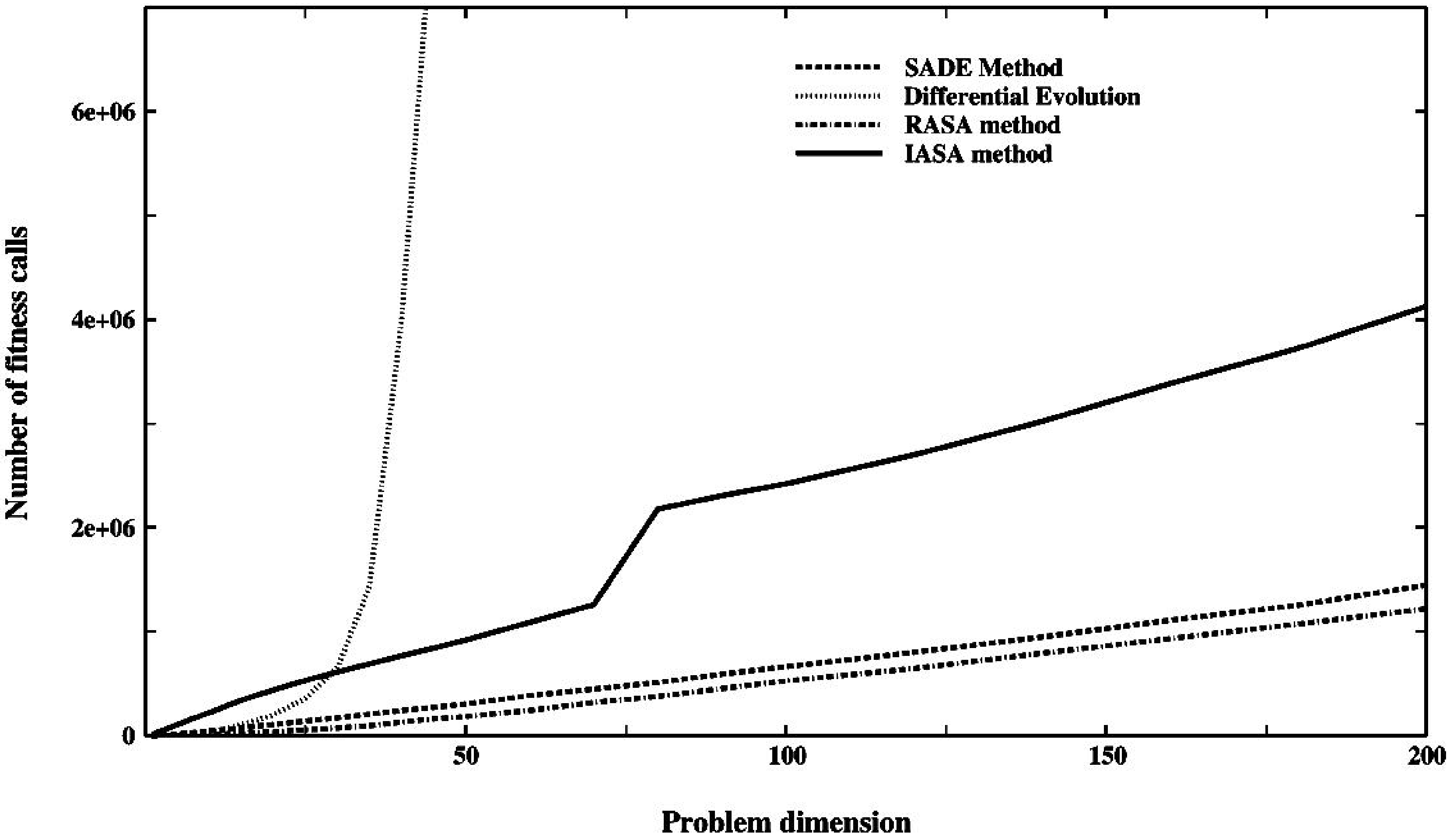
- Serviceability limit state



Results of comparison



Type 0 function



Results of comparison

	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

Results of comparison

- Number of tuning parameters

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

References

[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.

A humble plea. Please feel free to e-mail any suggestions, errors and typos to **`matej.leps@fsv.cvut.cz`**.

Date of the last version: 2.4.2008
Version: 001