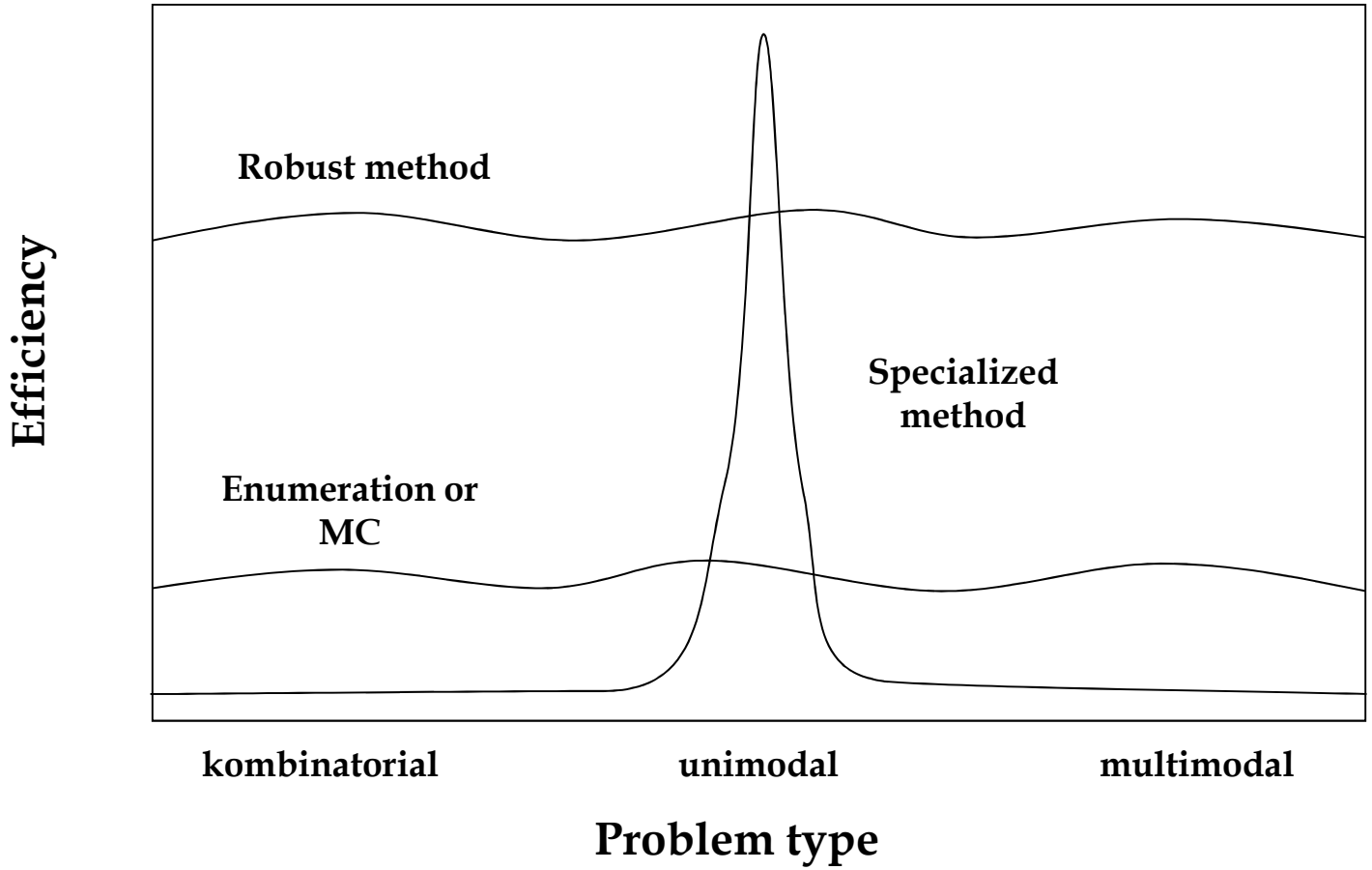
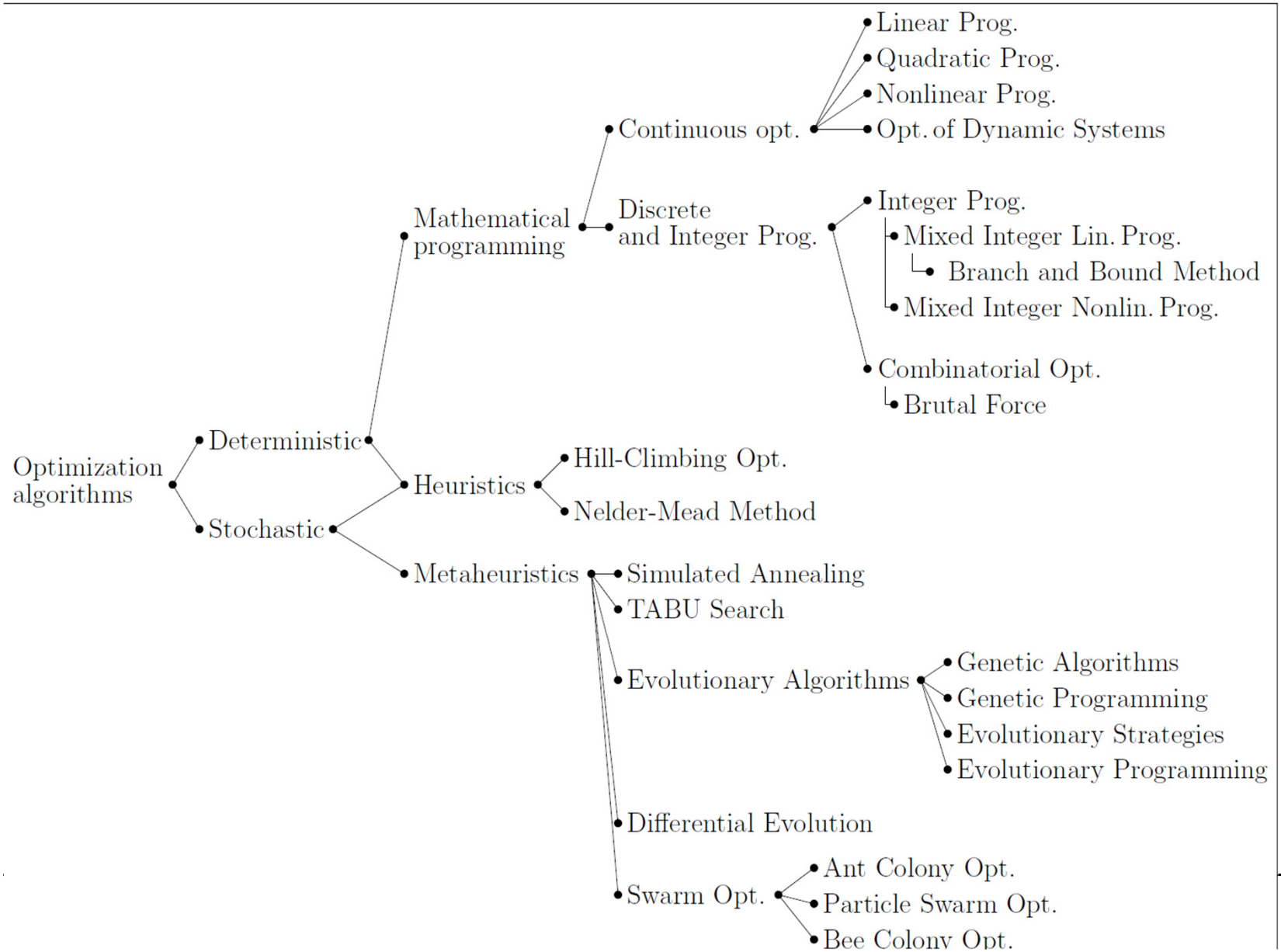
Introduction to Stochastic Optimization Methods (meta-heuristics)

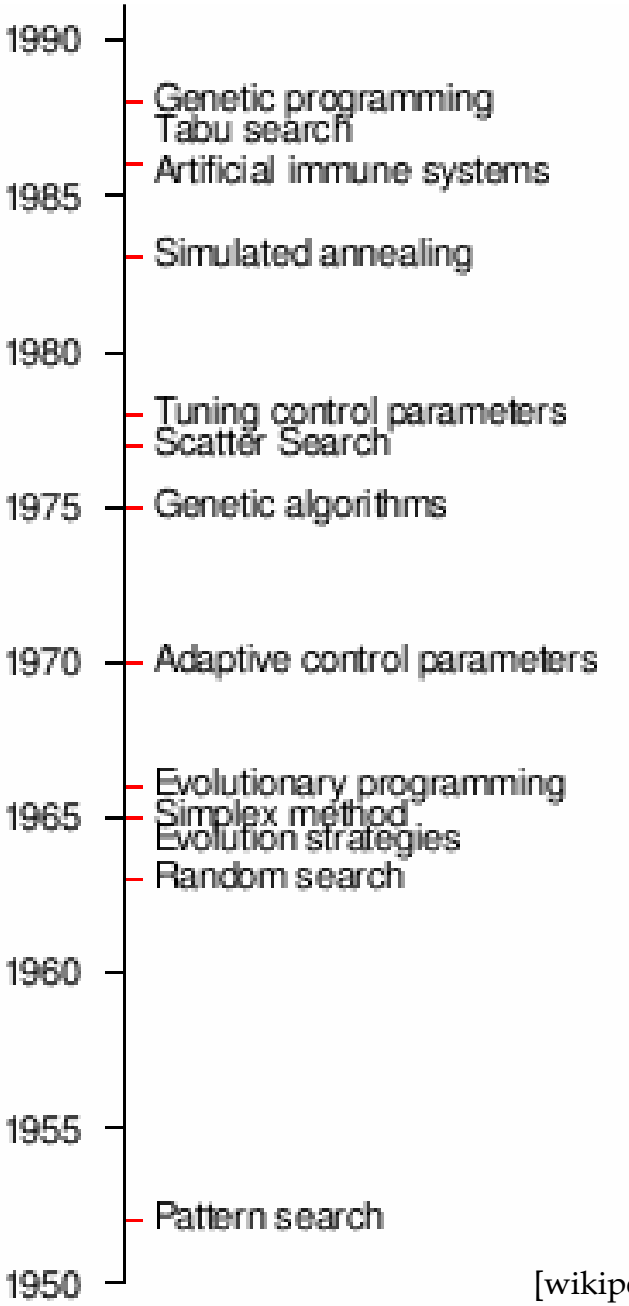
Efficiency of optimization methods



Classification of optimization methods



History of metaheuristics



[wikipedia]

Direct search methods

- Without knowledge of gradients (zero-order methods)
- **Heuristics:**
  - „... a heuristic is an algorithm that ignores whether the solution to the problem can be proven to be correct, but which usually produces a good solution or solves a simpler problem that contains or intersects with the solution of the more complex problem. Heuristics are typically used when there is no known way to find an optimal solution, or when it is desirable to give up finding the optimal solution for an improvement in run time.“ [Wikipedia]
  - For instance: Nelder-Mead algorithm
- **Metaheuristics:**
  - „designates a computational method that optimizes a problem by **iteratively** trying to improve a **candidate solution**.“ [Wikipedia]

Nelder-Mead method

- Heuristic method
- Known also as a *simplex method*
- Suitable for Dim<10

Nelder-Mead method

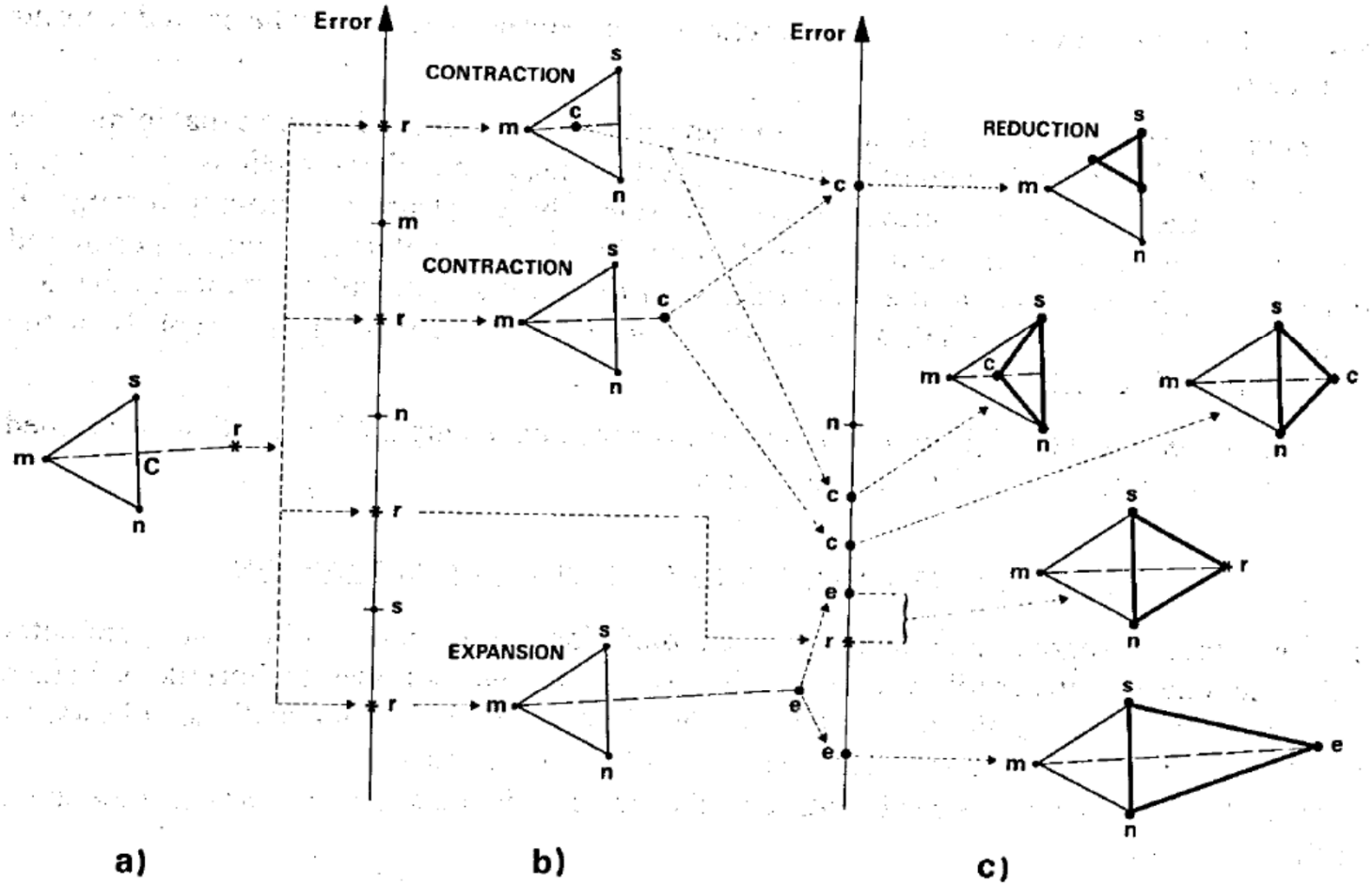
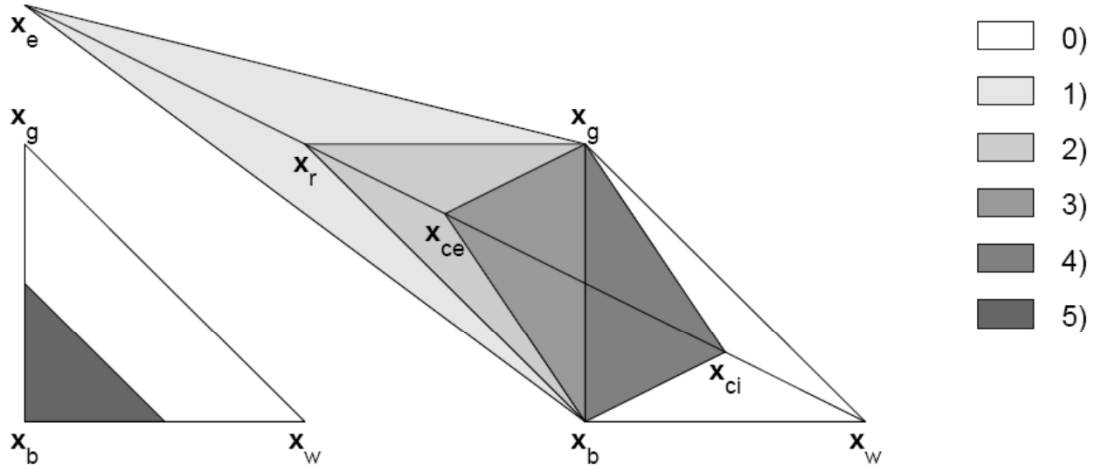
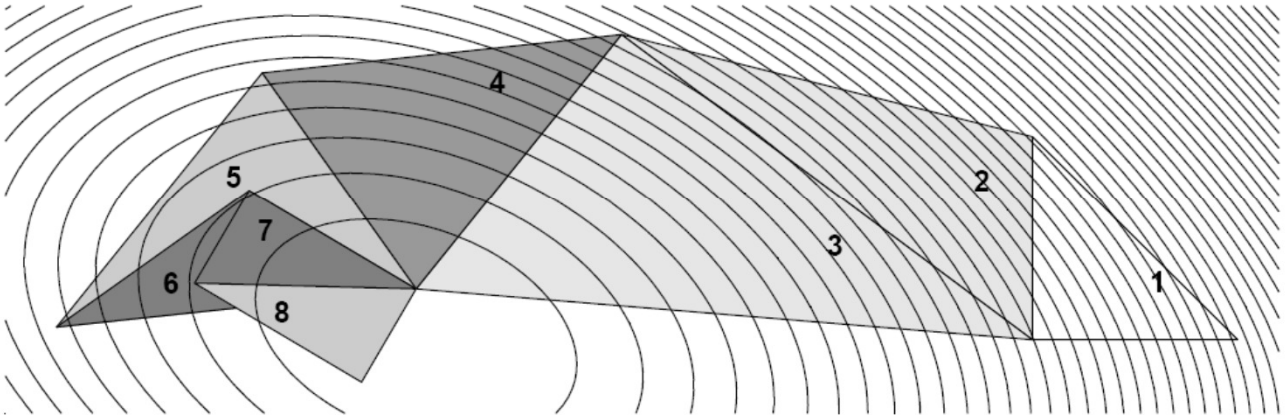


Figure 3. An illustration of the flexible polyhedron strategy for $N = 2$ parameters. (a) Current simplex and reflection (of m with respect to C into r): m = vertex with max error; n = vertex with next to the max error, s = vertex with smallest error; C = centroid of the vertices except m. (b) Comparisons of error levels and consequent moves. (c) Second error comparisons and final moves

[Gioda & Maier, 1980]

Nelder-Mead method

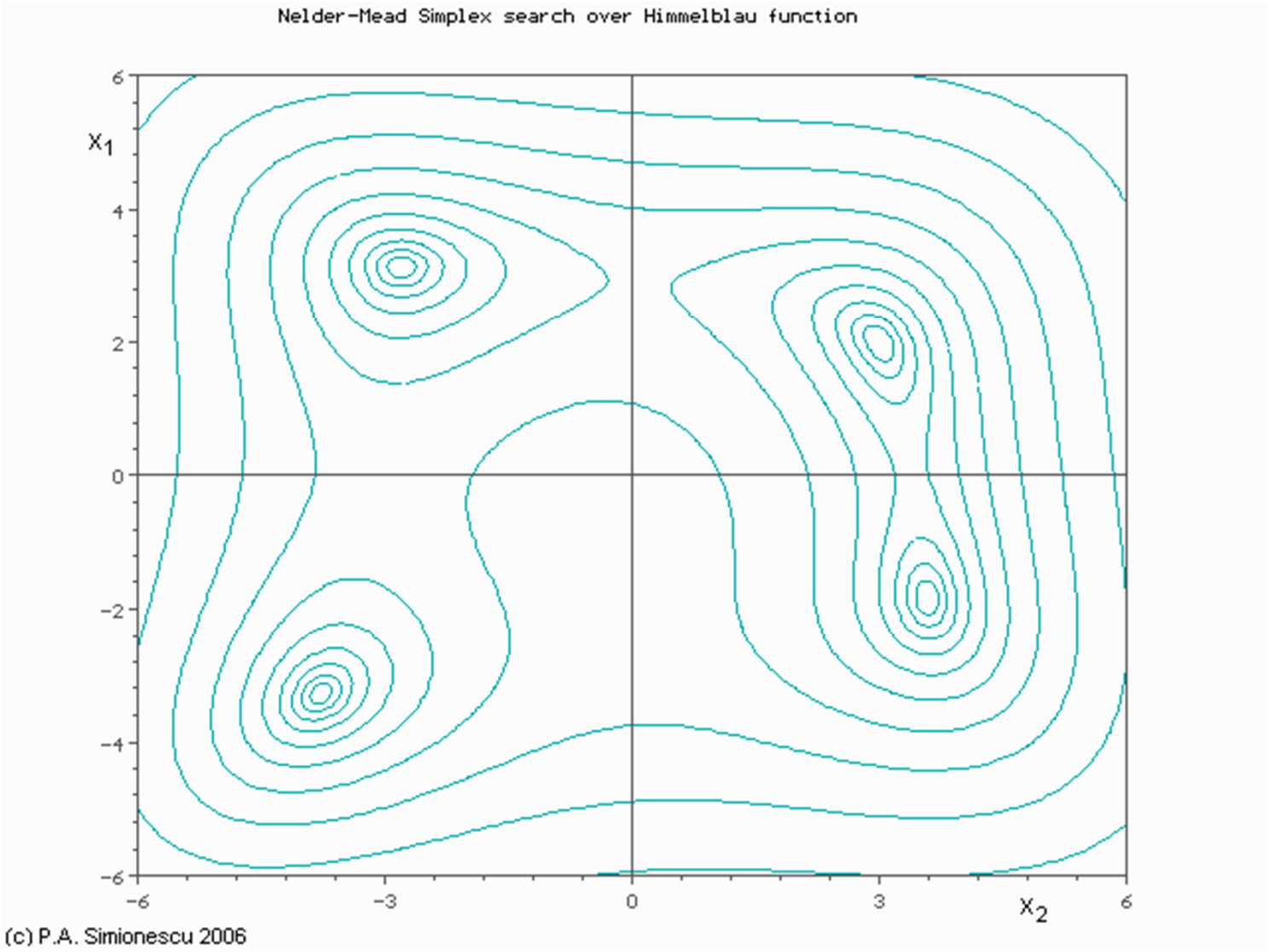




0) original triangle, 1) expansion, 2) reflexion, 3) outer contraction, 4) inner contraction, 5) reduction

[Čermák & Hlavička, VUT, 2006]

Nelder-Mead method



Meta-heuristic

- Also called **stochastic optimization methods**– usage of random numbers => random behavior
- Altering of existing solutions by local change
- Examples:
  - Monte Carlo method
  - Dynamic Hill-climbing algorithm
  - Simulated Annealing, Threshold Acceptance
  - Tabu Search

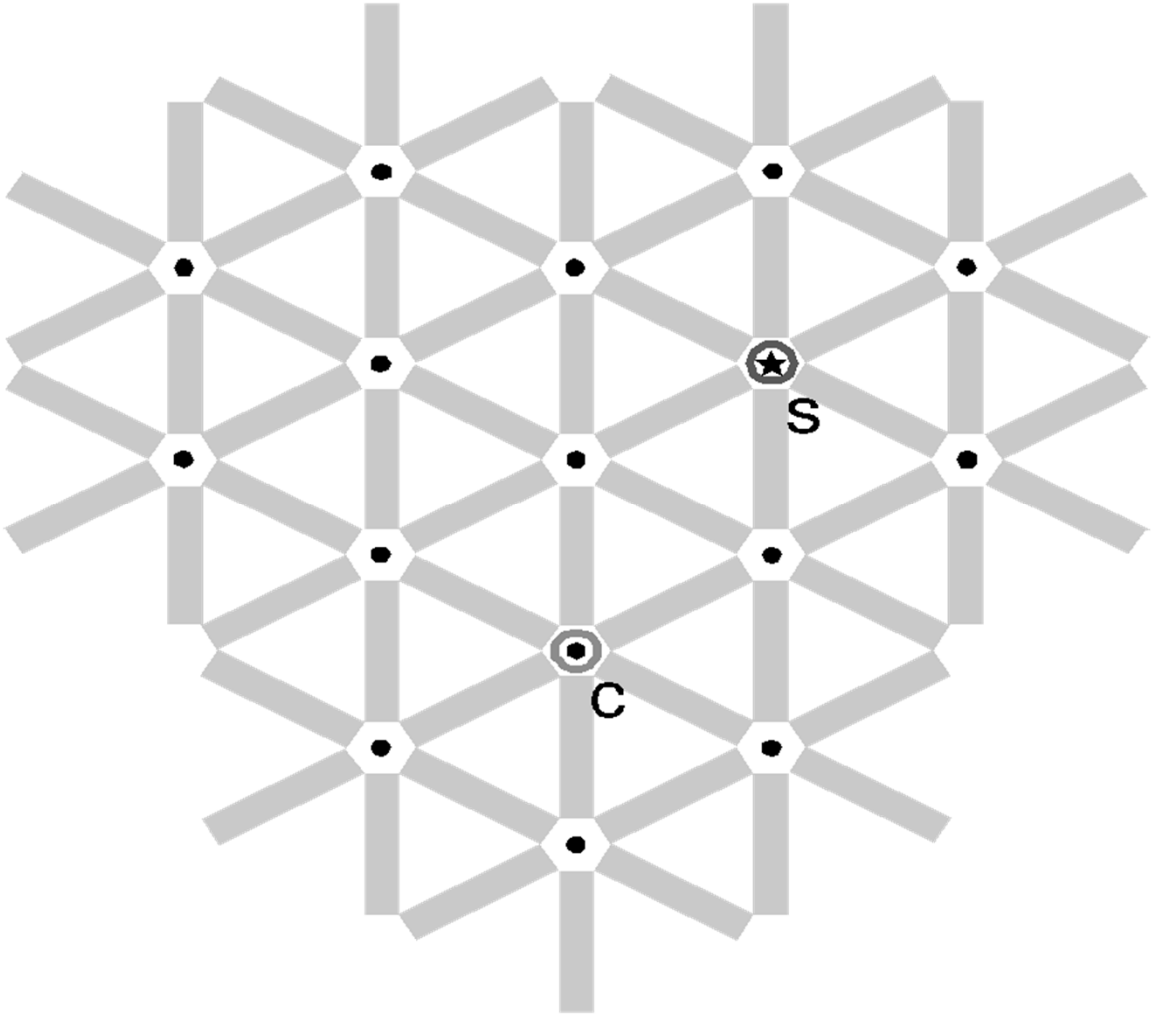
Monte Carlo method

also (brute-force search)

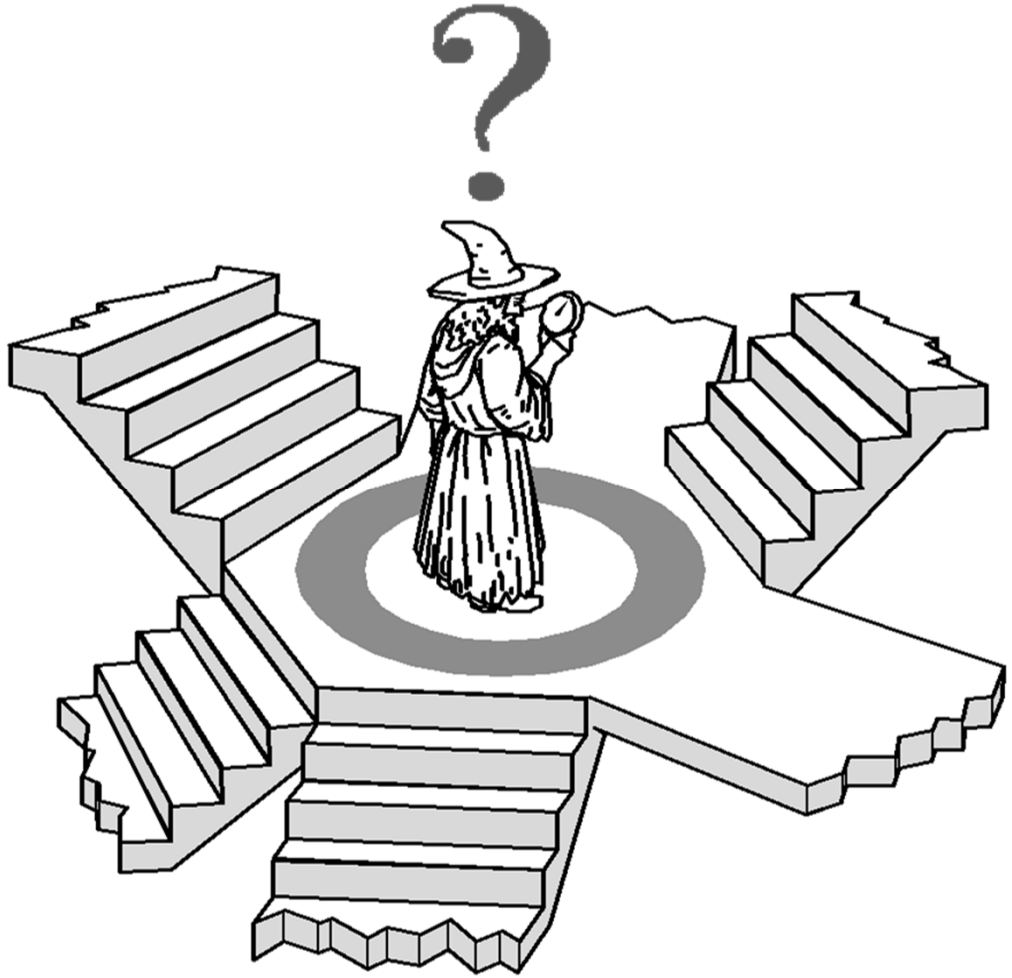
- Or „blind algorithm“
- Serve as a benchmark

```
1    t = 0
2    Create P, evaluate P
3    while (not stopping_criterion) {
4    t = t+1
5    Randomly create N, evaluate N
6    If N is better than P, then P=N
7    }
```

Global view on optimization

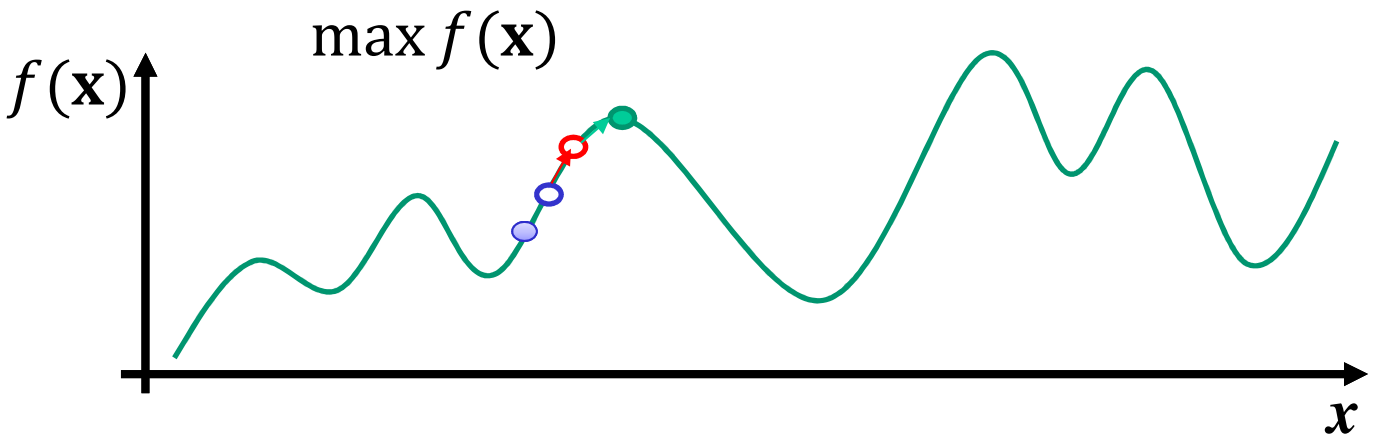


Local view on optimization



Hill-climbing algorithm

- Search in the vicinity of the current solution



Hill-climbing algorithm

- Search in the vicinity of the current solution

1 $t = 0$

2 Create P, evaluate P

3 **while** (**not** stopping_criterion) {

4 $t = t+1$

5 Randomly or systematically create N_i in the vicinity of P, evaluate N_i

6 The best N_i exchanges P

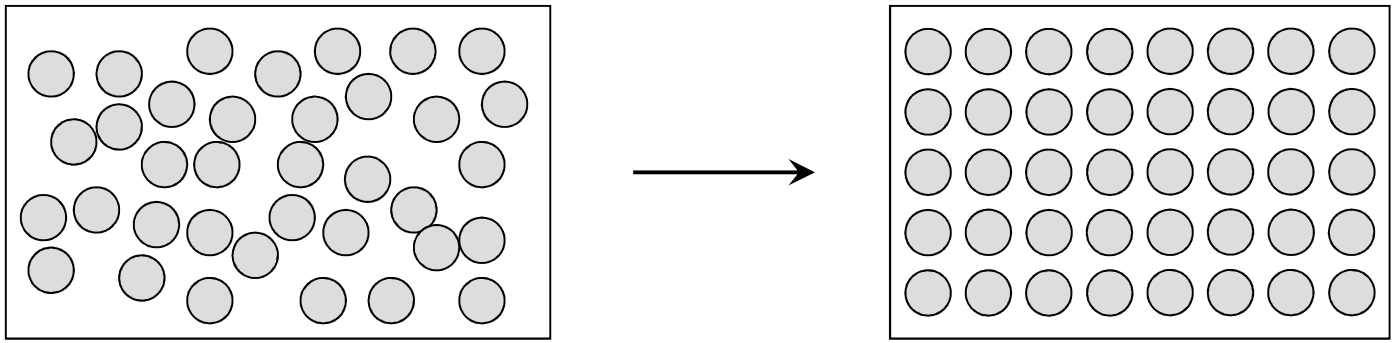
7 }

Simulated annealing **(SA)**

- Discovered in 80's independently by two authors [Kirkpatrick et al., 1983] and [Černý, 1985].
- Based on physical meaning, on an idea of annealing of metals, where the slow cooling leads material to the state of minimum energy. This is equivalent to the global minimization.
- There is a mathematical proof of convergence.

Physical background

- Theory based on monitoring of metallic crystals at different temperatures



High temperature

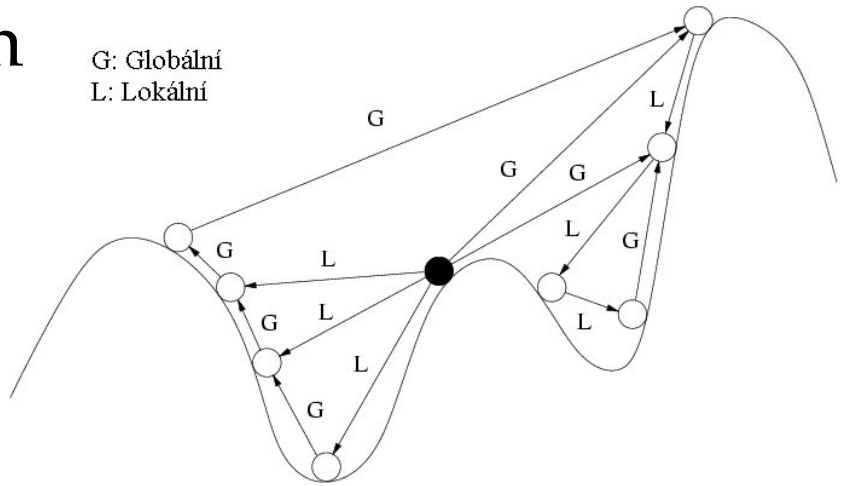
Low temperature

Principles of method

- stochastic optimization method
- Probability to escape from local minima

$$\Pr(E) = \exp\left(\frac{-\Delta E}{\kappa_B T}\right)$$

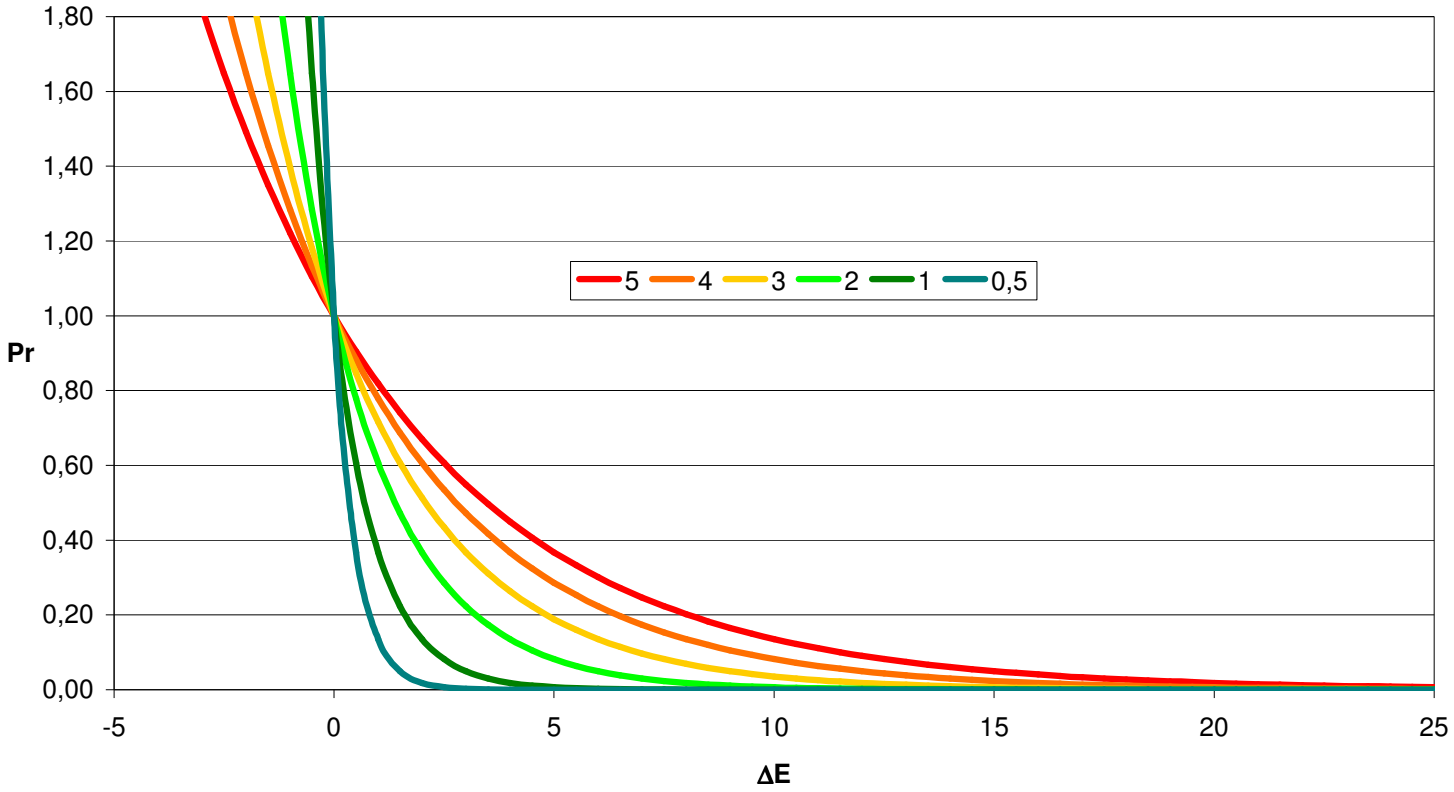
- consecutive decreasing of temperature, so-called *cooling schedule*



Temperature influence

- Probability

$$\Pr(E) = \exp\left(\frac{-\Delta E}{\kappa_B T}\right)$$



Algorithm

1 $T = T_{max,}$ Create P, evaluate P

2 **while** (**not** stopping_criterion) {

3 $count = succ = 0$

4 **while** ($count < count_{max}$ & $succ < succ_{max}$) {

5 $count = count + 1$

6 alter P by operator O, result is N

7 $p = \exp((F(N) - F(P))/T)$ (application of probability)

8 **if** (random number u[0, 1] < p) {

9 $succ = succ + 1$

10 P=N

11 $\}_{\text{if}}\}_{\text{while}}$

12 Decrease T (cooling schedule)

13 $\}_{\text{while}}$

Algorithm

- Initial temperature T_{max} is recommended at the level of e.g. 50% probability of acceptance of random solution. This leads to experimental investigation employing usually the "trial and error method"

- Parameter $count_{max}$ is for maximal number of all iteration and $succ_{max}$ is the number of successful iterations at the fixed temperature level (so-called Markov chain). Recommended ration is $count_{max} = 10\ succ_{max}$

Algorithm

- Stopping criterion is usually the maximum number of function calls

- As an „mutation“ operator, the addition of Gaussian random number with zero mean is usually used, i.e.

$$N = P + G(0,\sigma) ,$$

  where the standard deviation σ is usually obtained by the trial and error method

Cooling algorithms

- Classical: $T_{i+1} = T_{mult} T_i$
- Where T_{mult} =0,99 or 0,999
- Other possibilities exist. e.g.:

  $$T \approx \frac{T_0}{\ln t}$$

  see e.g. [Ingber]
- These settings usually do not provide flexibility to control the algorithm

Cooling algorithms

- Proposed methodology [Lepš, MSc. Thesis] [Lepš, Šejnoha]

$$T_{mult} = \left(\frac{T_{min}}{T_{max}} \right)^{\frac{succ_{max}}{iter_{max}}}$$

- This relationship enables at maximum iterations $iter_{max}$ reach the minimal temperature T_{min} given by

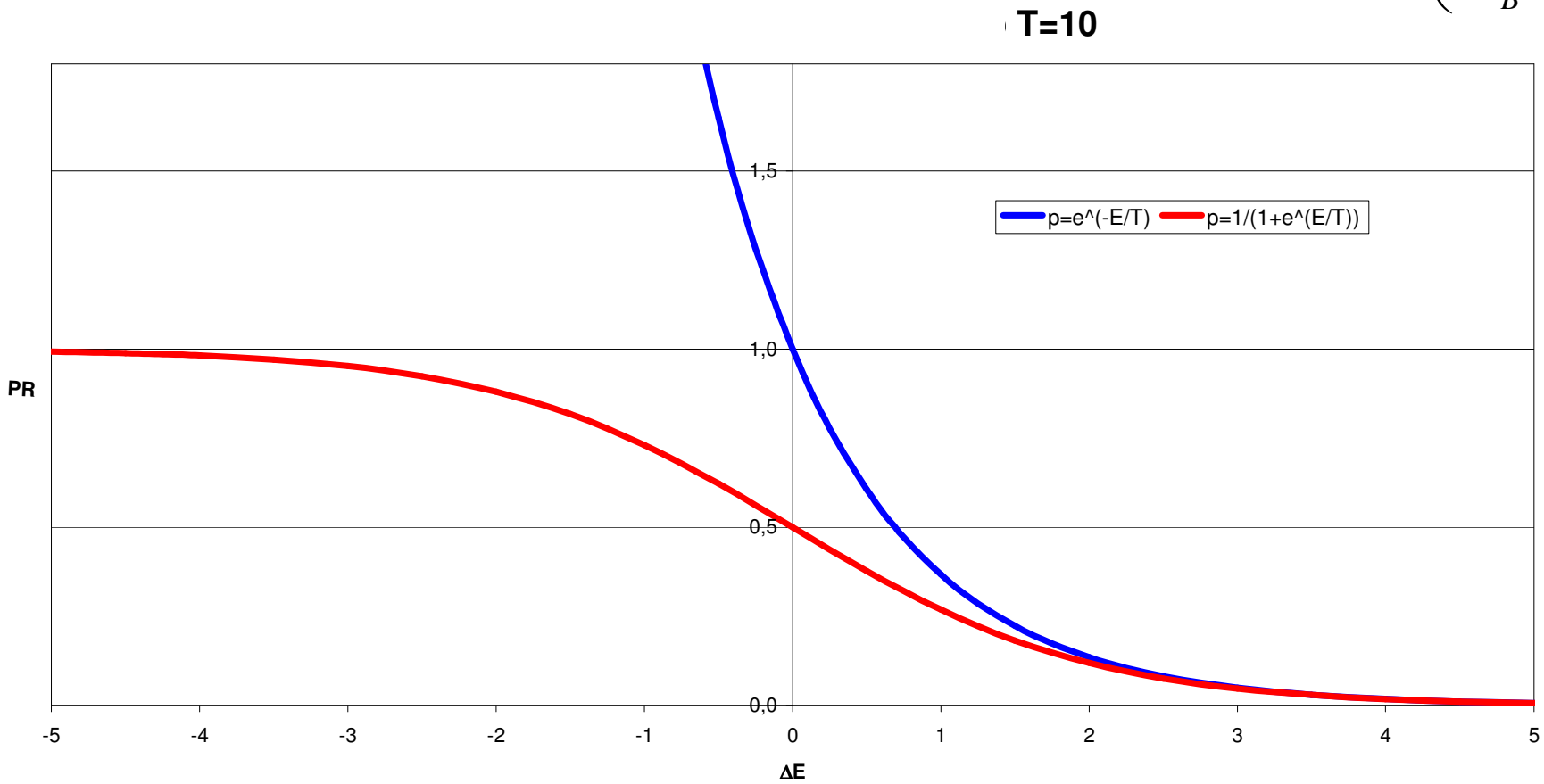
$$T_{min} = 0{,}01 T_{max}$$

- For instance: for $succ_{max}/iter_{max} = 0{,}001$ is $T_{mult}=0{,}995$

Note on probability

- For population based algorithms:

$$\Pr(E) = \frac{1}{1 + \exp\left(\frac{\Delta E}{\kappa_B T}\right)}$$



SA convergence

'Convergence' result for SA:

Under certain conditions (extremely slow cooling),
any sufficiently long trajectory of SA is guaranteed to end in
an optimal solution [Geman and Geman, 1984; Hajek, 1998].

Note:

- Practical relevance for combinatorial problem solving
  is very limited (impractical nature of necessary conditions)
- In combinatorial problem solving, *ending* in optimal solution
  is typically unimportant, but *finding* optimal solution
  during the search is (even if it is encountered only once)!

Threshold acceptance

- Simpler than SA
- Instead of probability, there is a threshold, or range, within which the worst solution can replace the old one
- Threshold is decreasing similarly as temperature in SA
- There is a proof that this method does not ensure finding the global minima.

References

[1] Kirkpatrick, S., Gelatt, Jr., C., and Vecchi, M. P. (1983). Optimization by Simulated Annealing. Science, 220:671–680.

[2] Černý, J. (1985). Thermodynamical approach to the traveling salesmanproblem: An efficient simulation algorithm. J. Opt. Theory Appl., 45:41–51.

[3] Ingber, L. (1993). Simulated annealing: Practice versus theory. Mathematical and Computer Modelling, 18(11):29–57.

[4] Ingber, L. (1995). Adaptive simulated annealing (ASA): Lessons learned. Control and Cybernetics, 25(1):33–54.

[5] Vidal, R. V. V. (1993). Applied Simulated Annealing, volume 396 of Notes in Economics and Mathematical Systems. Springer-Verlag.

References

[6] J. Dréo, A. Pétrowski, P. Siarry, E. Taillard, A. Chatterjee (2005). Metaheuristics for Hard Optimization: Methods and Case Studies. Springer.

[7] M. Lepš and M. Šejnoha: New approach to optimization of reinforced concrete beams. Computers & Structures, 81 (18-19), 1957-1966, (2003).

[8] Burke, Edmund K., and Graham Kendall. *Search methodologies*. Springer Science+ Business Media, Incorporated, 2005.

[9] Weise, Thomas, et al. "Why is optimization difficult?" *Nature-Inspired Algorithms for Optimisation*. Springer Berlin Heidelberg, 2009. 1-50.

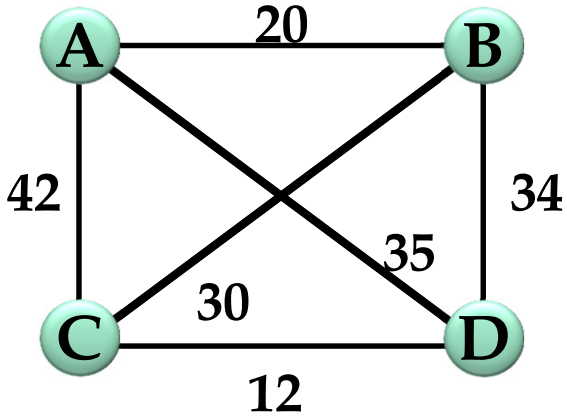
[10] S. Venkataraman and R.T. Haftka: Structural optimization complexity: What has Moore's law done for us? Struct Multidisc Optim 28, 375–387 (2004).

Travelling Salesman problem (TSP)

- One of the most famous discrete optimization problems
- The goal is to find closed shortest path in edge-oriented weighted graph
- For n cities, there are (n–1) ! /2 possible solutions
- NP-complete task

Travelling Salesman problem (TSP)

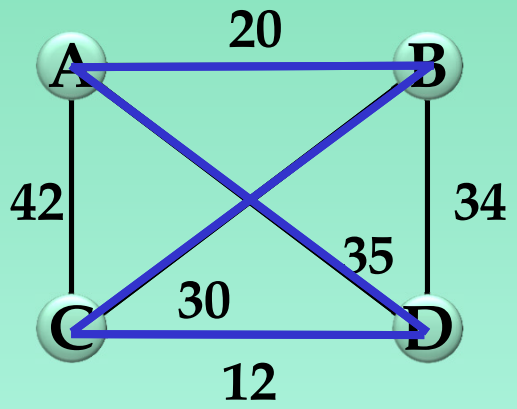
- Example



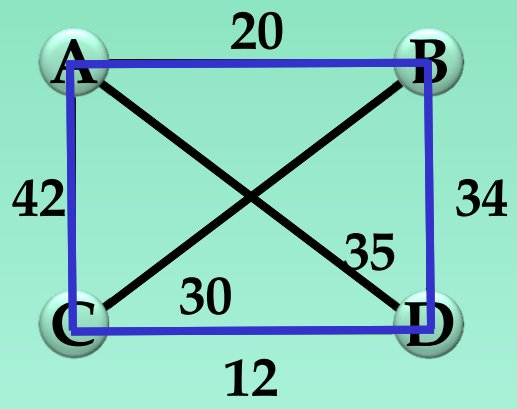
	A	B	C	D
A	0	20	42	35
B	20	0	30	34
C	42	30	0	12
D	35	34	12	0

Travelling Salesman problem (TSP)

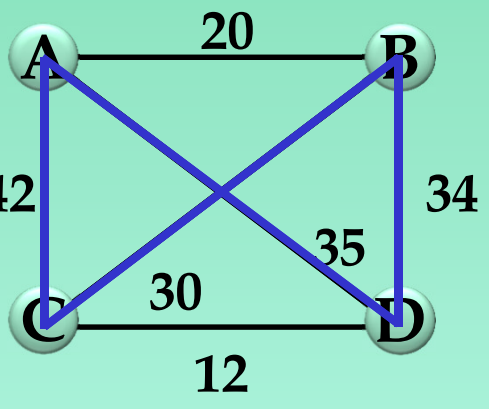
Solution I.



Solution II.



Solution III.



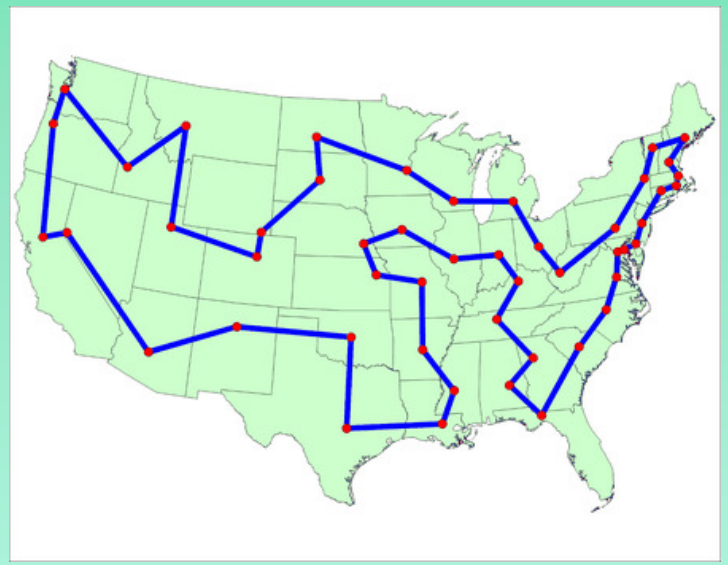
No.	Order of the city				distance
	1	2	3	4	
I.	A	B	C	D	97
II.	A	B	D	C	108
III.	A	C	B	D	141

Travelling Salesman problem (TSP)

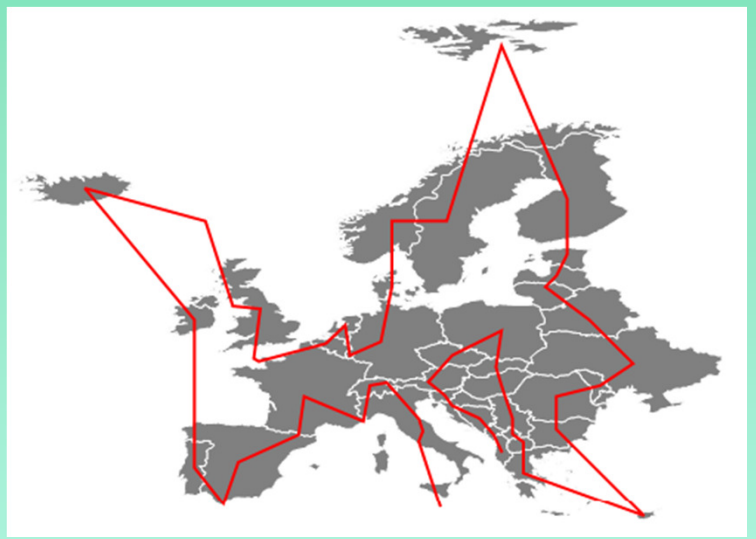
No. cities	Possible paths	Computational demands
4	3	$3{,}0 \cdot 10^{-09}$ seconds
5	12	$1{,}2 \cdot 10^{-08}$ seconds
6	60	$6{,}0 \cdot 10^{-08}$ seconds
7	360	$3{,}6 \cdot 10^{-07}$ seconds
8	2520	$2{,}5 \cdot 10^{-06}$ seconds
9	20160	$2{,}0 \cdot 10^{-05}$ seconds
10	181440	$1{,}8 \cdot 10^{-04}$ seconds
20	$6{,}1 \cdot 10^{16}$	19,3 years
50	$3{,}0 \cdot 10^{62}$	$9{,}6 \cdot 10^{46}$ years
100	$4{,}7 \cdot 10^{157}$	$1{,}5 \cdot 10^{140}$ years

Based on 10^9 operations per 1 second computer speed

Travelling Salesman problem (TSP)



The OPTNET Procedure [1]



Mathematica software[2]

http://gebweb.net/optimap/

[1] http://support.sas.com/documentation/cdl/en/ornoaug/65289/HTML/default/viewer.htm#ornoaug_optnet_examples07.htm
[2] http://mathematica.stackexchange.com/questions/15985/solving-the-travelling-salesman-problem

Example

- tsp.m

Some parts of this presentation have been kindly prepared by Ing. Adéla Pospíšilová with Faculty of Civil Engineering, CTU in Prague.

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

News 21.10.2013: Added Nelder-Mead method and the introduction redone.
News 5.11.2014: Added TSP after the end.
News 10.11.2015: Enhancement of TSP.

Date of the last version: 10.11.2015
Version: 003