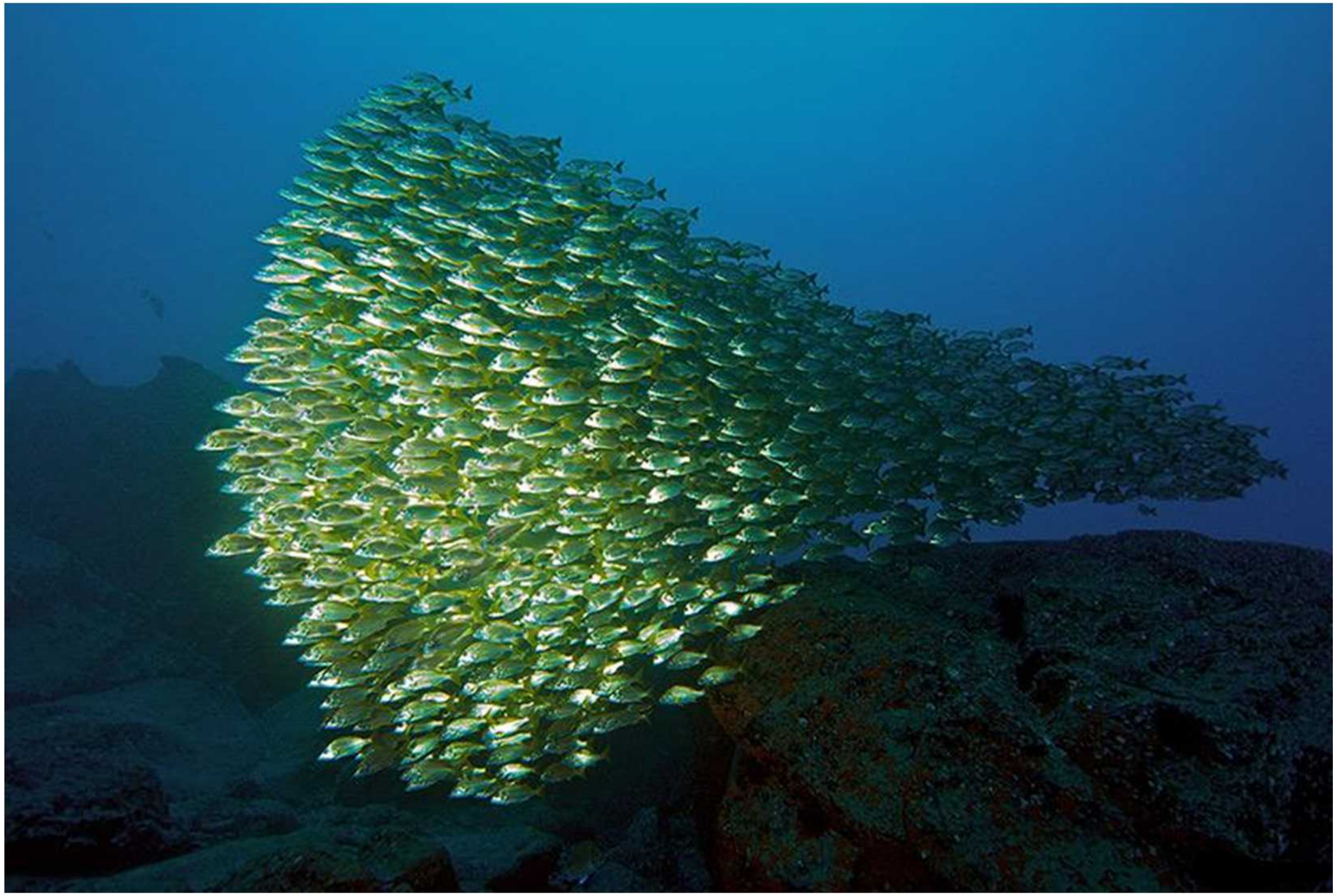
Swarm Intelligence



http://pixdaus.com/single.php?id=168307

Swarm Intelligence

- algorithms inspired by bird flocking, fish schooling, insect colonies etc.

- Particle Swarm Optimization
- Ant Colony Optimization
- Glow-worm Swarm Optimization
- …

- different behaviour in transfer of information

Swarm Intelligence

Advantages:

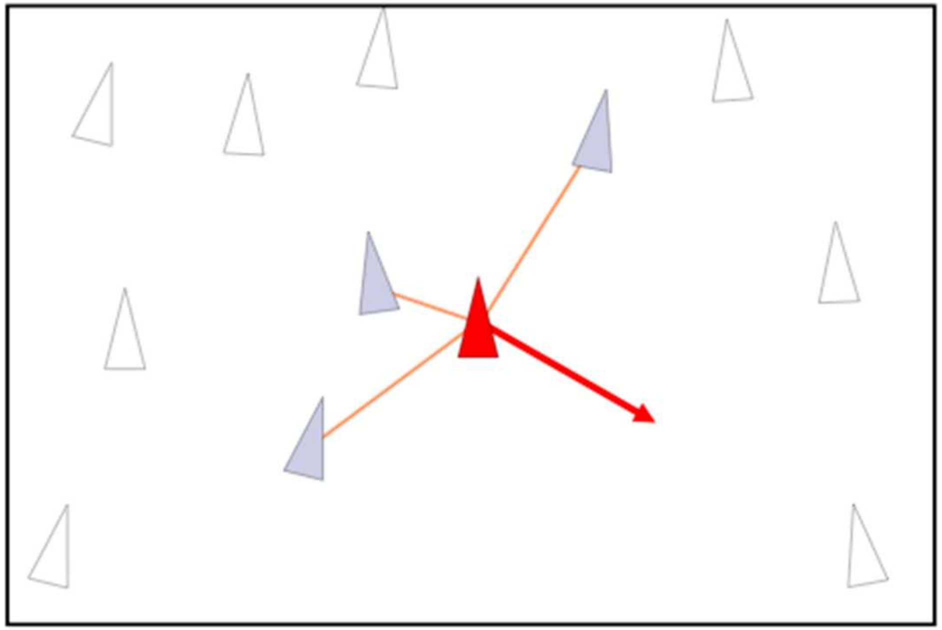
- easy to implement
- it is not necessary to calculate gradients or Hessian matrix

Disadvantages:

- it is necessary to determine correctly all constants and parameters for a given task

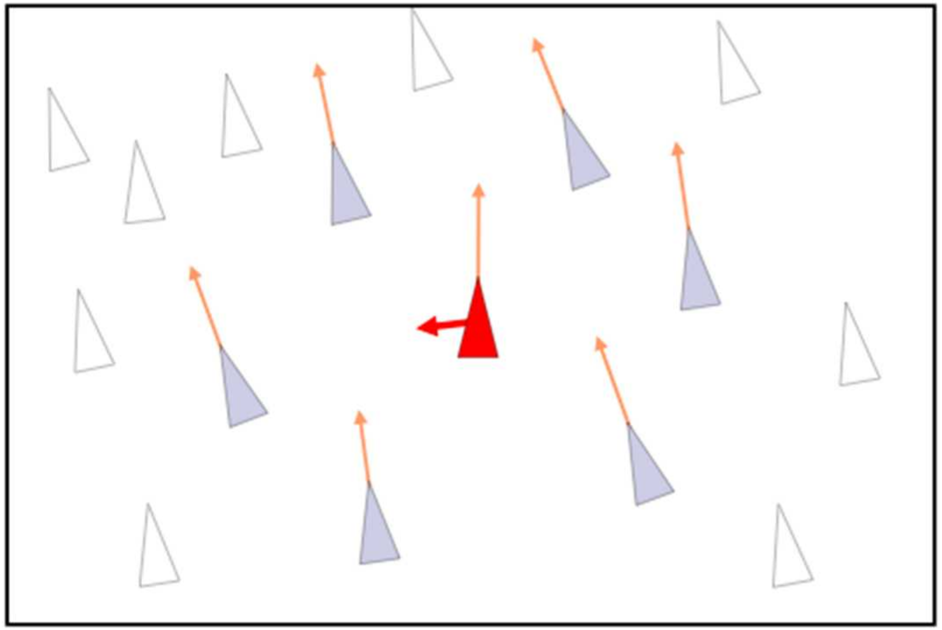
Rules of Swarm Intelligence

Rule no. 1: Avoid Collision with neighboring birds



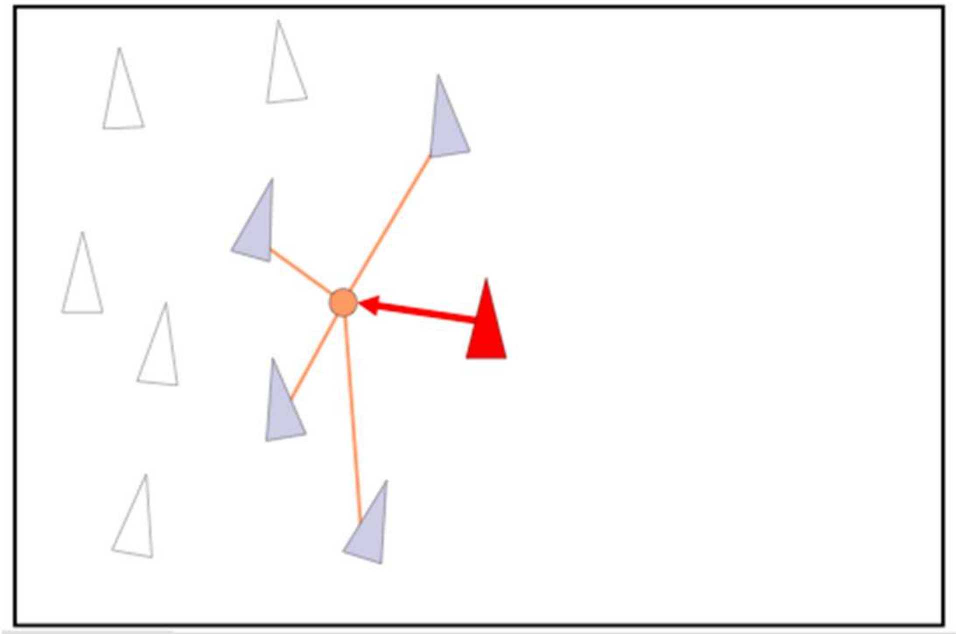
Rules of Swarm Intelligence

Rule no. 2: Match the velocity of neighboring birds



Rules of Swarm Intelligence

Rule no. 3: Stay near neighboring birds



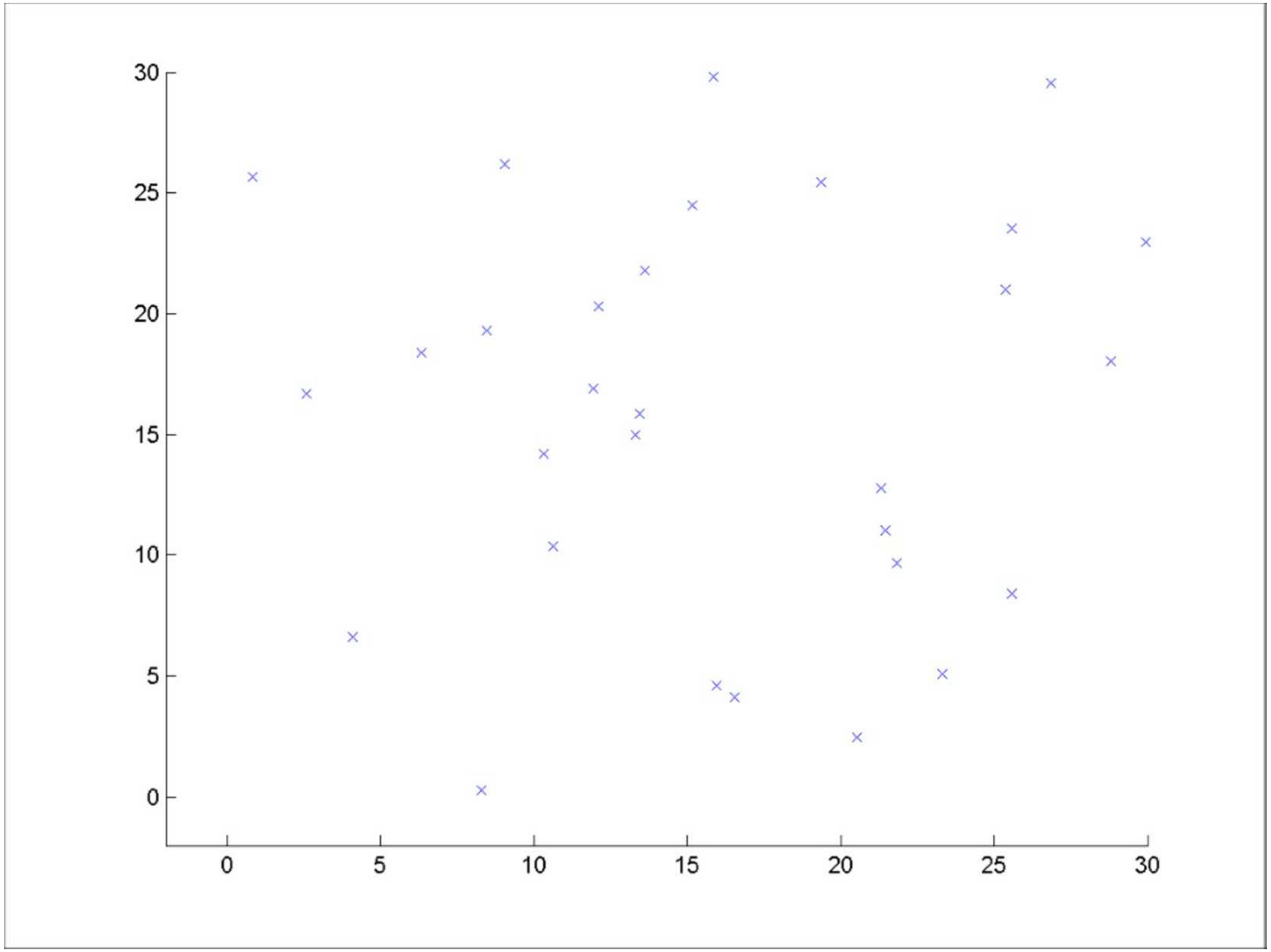
Particle Swarm Optimization

http://www.crackedcamera.com/flock-of-birds-san-diego-ca/

Particle Swarm Optimization

- firstly published in 1995 by authors Kennedy and Eberhart [1]
- stochastic method
- it is not necessary to calculate gradients or Hessian matrix
- for discrete, continuous and combined problems

Particle Swarm Optimization



Particle Swarm Optimization

Original approach:

$$v_{id}^{k+1} = w \cdot v_{id}^{k} + c_1 \cdot rand() \cdot (p_{id} - x_{id}) + c_2 \cdot Rand() \cdot (p_{gd} - x_{id})$$

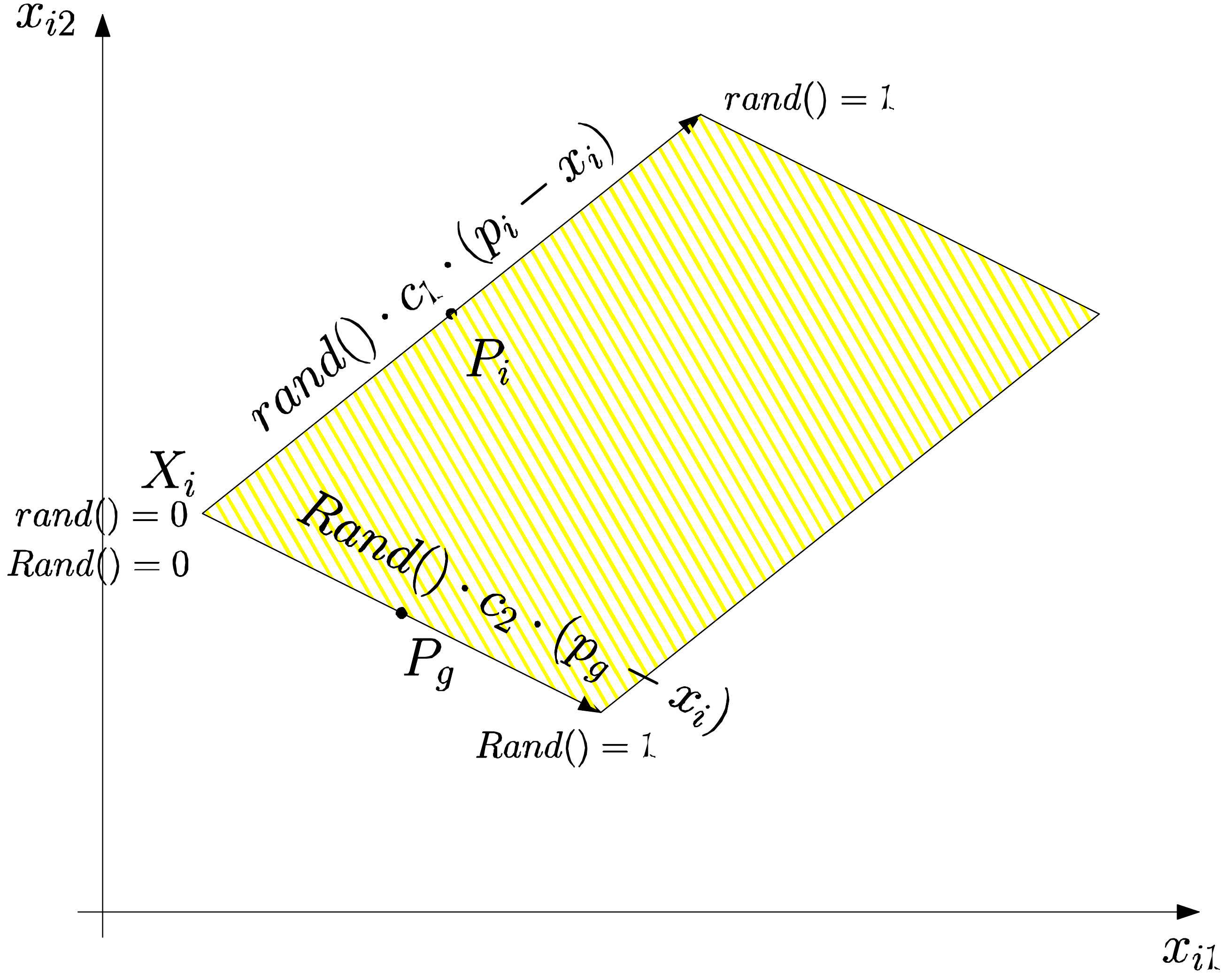
$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}$$

w – inertia weight

c_1 – cognitive factor

c_2 – social factor

Stochastic contribution to velocity



Particle Swarm Optimization

Constriction method (Clerc, 1999) [2, 3]

$$v_{id}^{k+1} = K \cdot \left[v_{id}^{k} + c_1 \cdot rand() \cdot (p_{id} - x_{id}) + c_2 \cdot Rand() \cdot (p_{gd} - x_{id}) \right]$$

$$K = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}, \; where: \; \varphi = c_1 + c_2, \; \varphi > 4$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}$$

K – constriction factor

Orientation values of parameters

c_1, c_2	2
rand(), Rand()	(0, 1)
w	0,4 - 0,9 for const.
K	0,729 (for ϕ = 4,1)

Notes: w - analogy to SA
- it is possible to use constant values, linearly changing values etc.

Algorithm

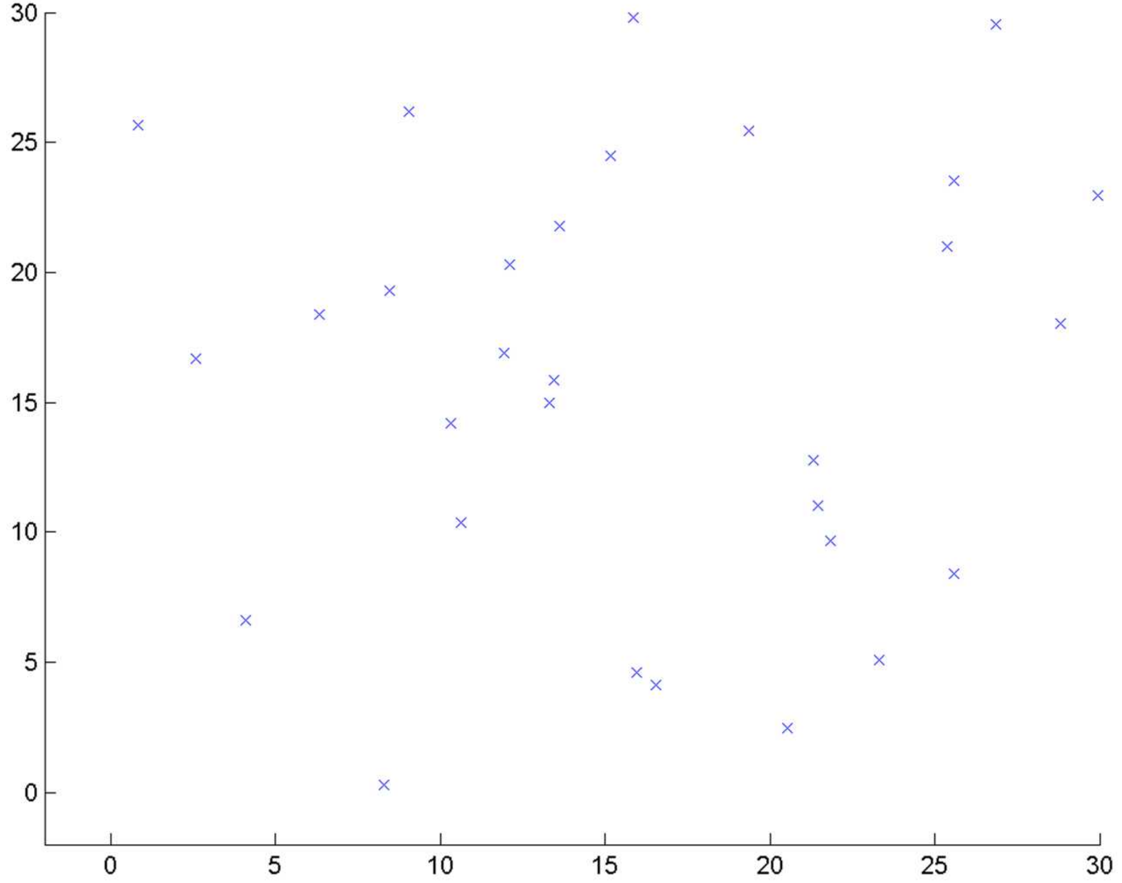
setting of constants, factors, …

initial placement of particles in space

setting of initial velocity

```
while (termination condition != true )
    k++                         % cycle iteration
    for i = 1 : number of particles
        enumeration of an objective function f_i^k
        if (f_i^k < p_i)  then p_i = f_i^k
        if (f_i^k < p_g)  then p_g = f_i^k
        updating particle velocity V_i
        updating particle location X_i
    end
end
```

PSO.m

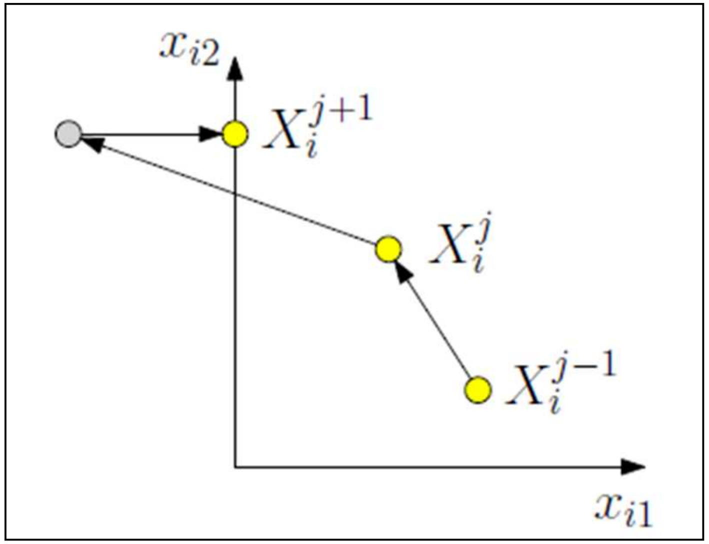


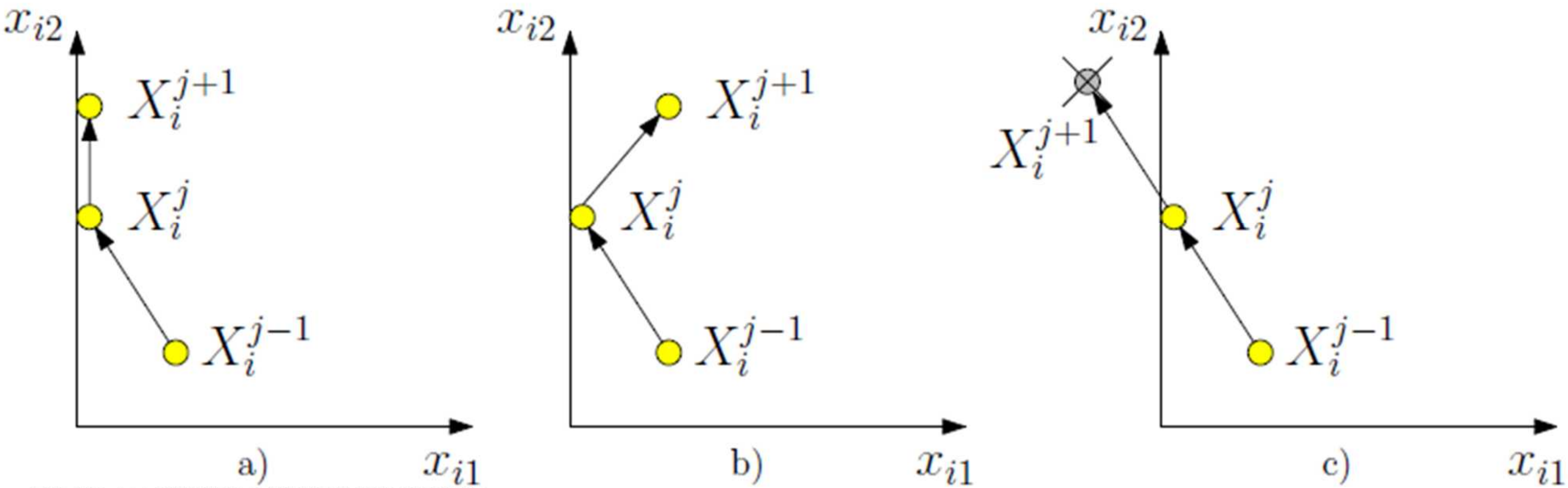
PSO variants [4]

- random reinitialization of particle velocities
- maximum velocity restriction
  - restriction of vector length
  - restriction of each component of the vector
- minimum velocity restriction
- craziness
- inertia weight (constant value, linearly changing value, …)
- constriction factor
- and many others

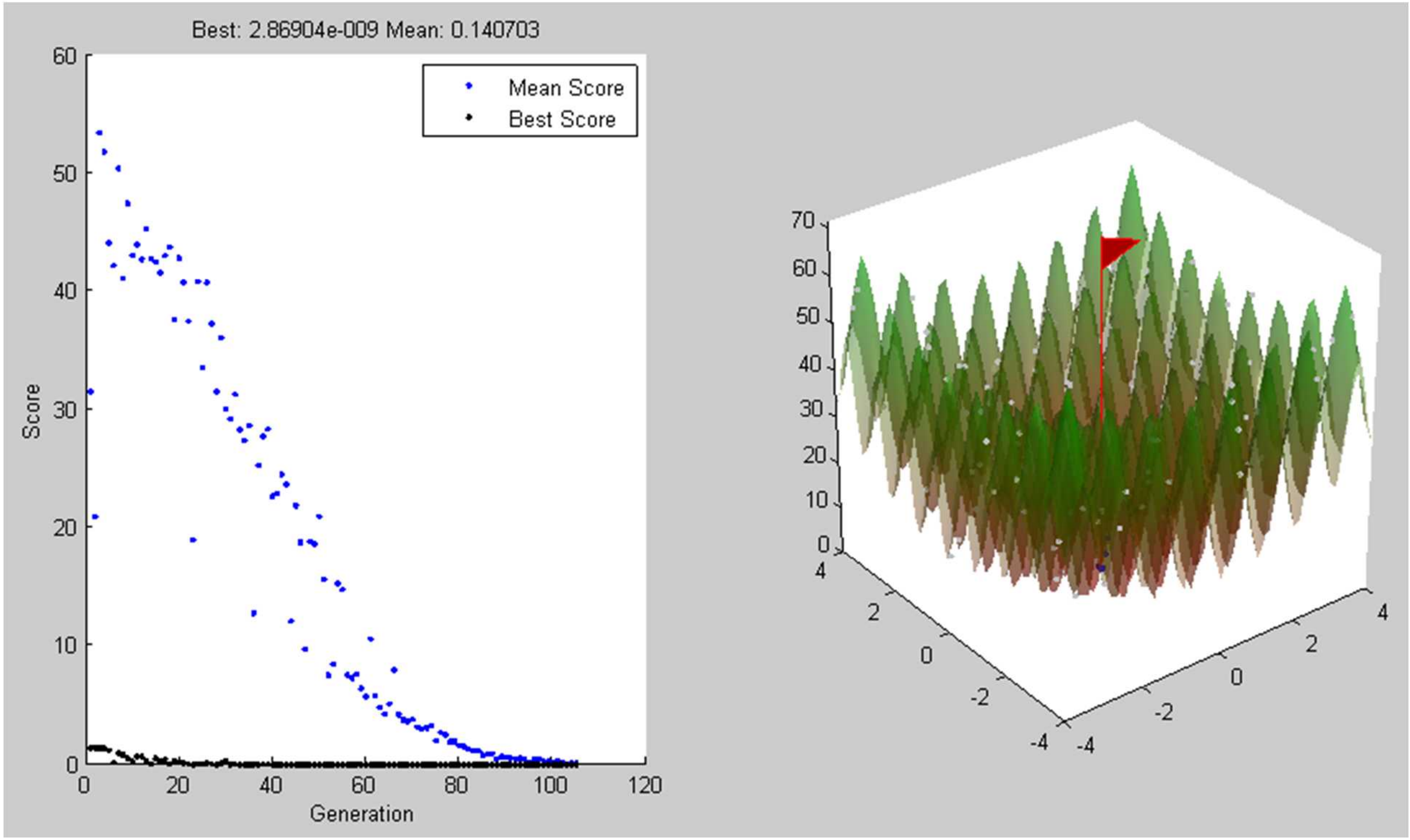
Boundary conditions

- Absorbing wall
- Reflecting wall
- Invisible wall





Another Particle Swarm Toolbox



References

[1] Kennedy, J.; Eberhart, R. C. (1995). Particle Swarm Optimization. IEEE International Conference of Neural Networks, 4: 1942 – 1948.

[2] Clerc, M. (1999). The Swarm and The Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. IEEE Congress on Evolutionary Computation CEC'99, 3: 1951 – 1957.

[3] Eberhart, R. C.; Shi, Y. (2000). Comparing Intertia Weights and Constriction Factors in Particle Swarm Optimization. IEEE Congress on Evolutionary Computation, 1: 84 – 88.

[4] Wilke, D. N.; Kok, D.; Groenwold, A. A. (2006). Comparison of Linear and Classical Velocity Update Rules in Particle Swarm Optimization : Notes on Diversity. International Journal for Numerical Methods in Engineering, 70: 962 – 984.

References

[5] Shi, Y.; Eberhart, R. C. (1999). Empirical Study of Particle Swarm Optimization. IEEE Congress on Evolutionary Computation CEC'99, 3: 1945 – 1950.

[6] Clerc, M. (2006). Particle Swarm Optimization. 1. ed. Wiley-ISTE. ISBN-13: 978-1-905209-04-0

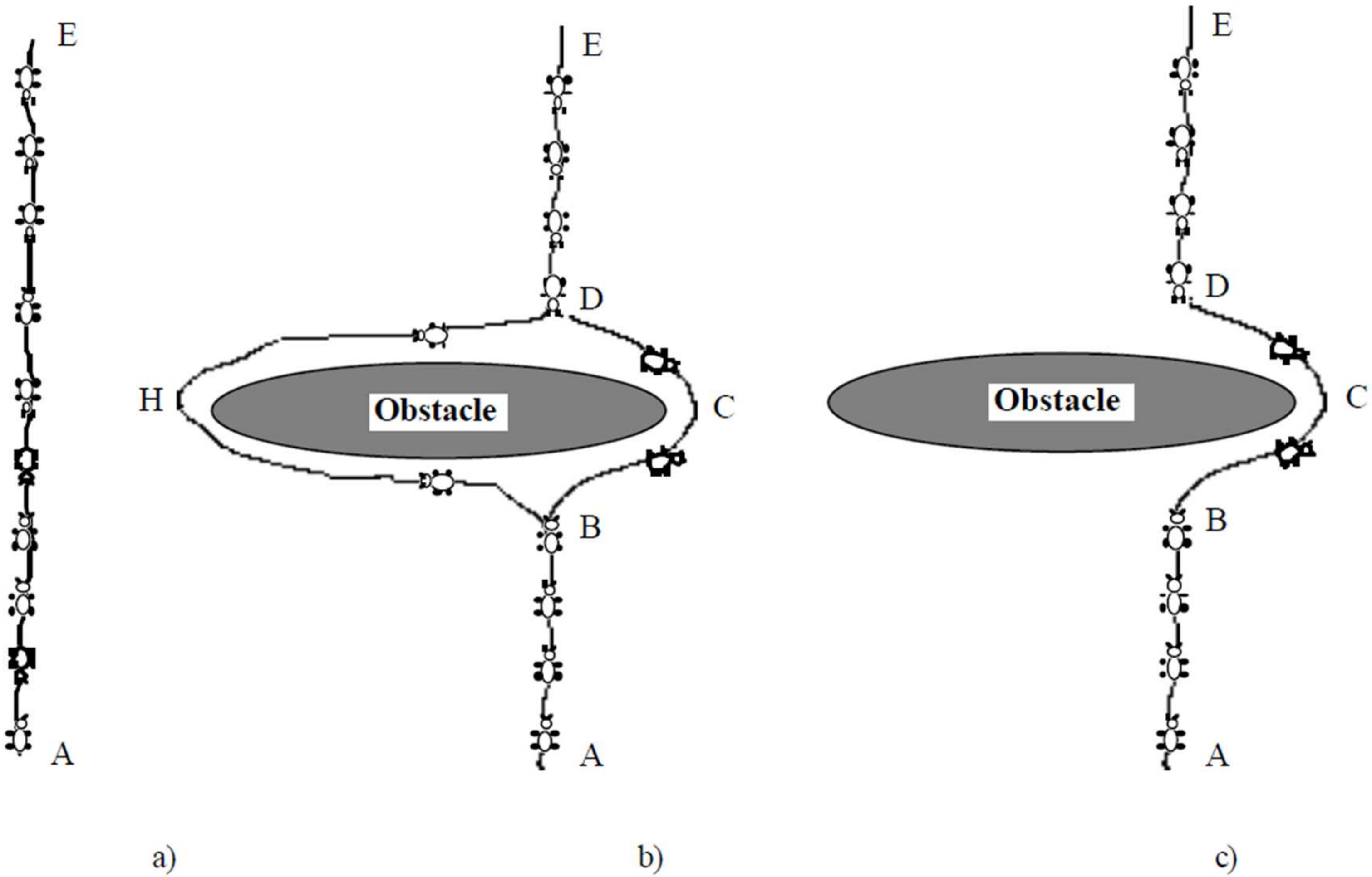
Ant Colony Optimization

http://ananthu-howtonameit.blogspot.com/2010/11/secret-of-success-get-mind-set-of-ant.html

Ant Colony Optimization

- first published in 90's by M. Dorigo et al.
- stochastic method
- it is not necessary to calculate gradients or Hessian matrix
- algorithm was originally tested on TSP (Travelling Salesman Problem)
- pheromone

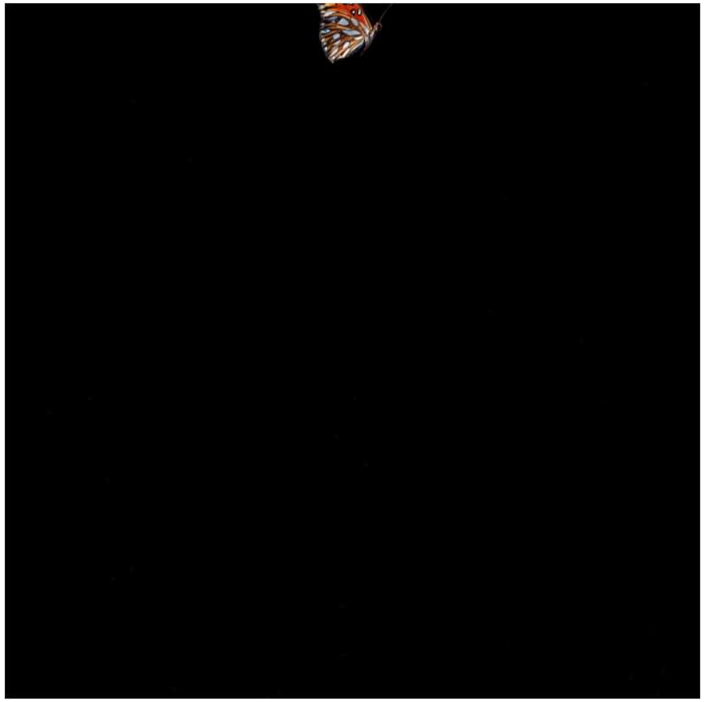
Ant Colony Optimization



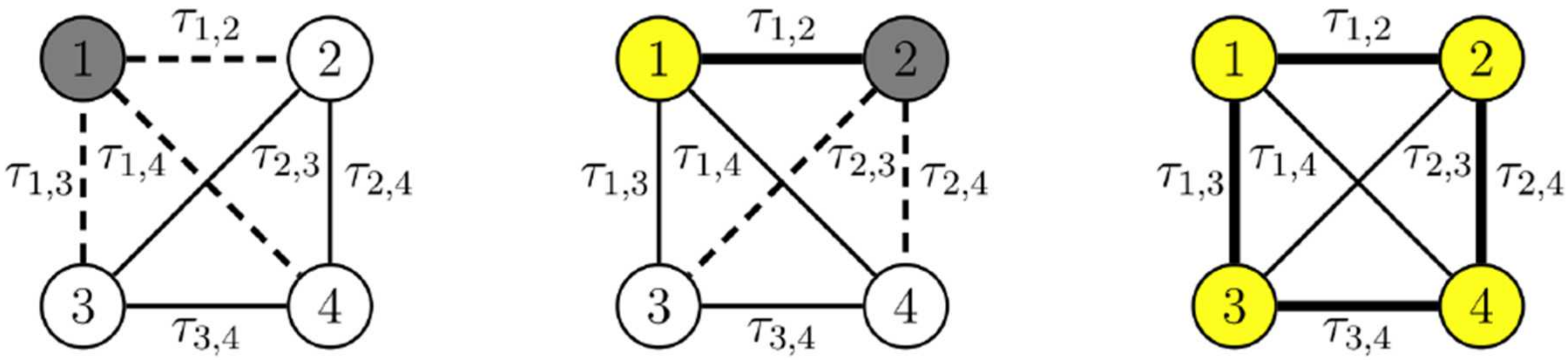
published in [7]

Ant Colony Optimization

http://users.sussex.ac.uk/~tn41/antStage2/war/AntAppletFull.html



Travelling Salesman Problem



$$\mathbf{p}(e_{1,j}) = \frac{\tau_{1,j}}{\tau_{1,2} + \tau_{1,3} + \tau_{1,4}}$$

$$\mathbf{p}(e_{2,j}) = \frac{\tau_{2,j}}{\tau_{2,3} + \tau_{2,4}}$$

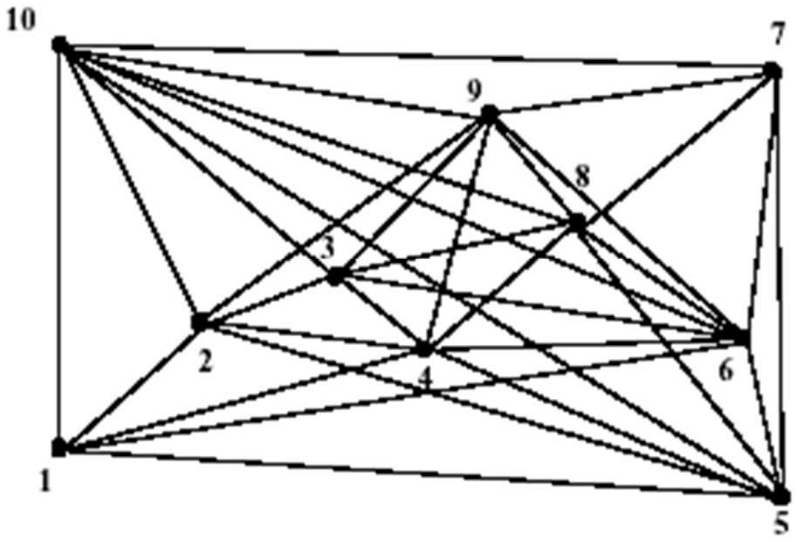
(a) First step of the solution construction.

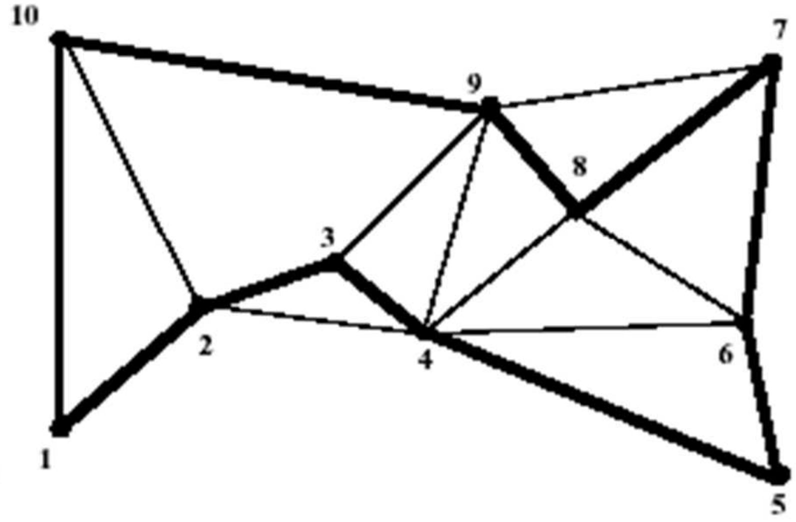
(b) Second step of the solution construction.

(c) The complete solution after the final construction step.

published in [8]

Travelling Salesman Problem





Algorithm

```
setting ant number and their random locations
setting initial pheromone values

while (termination condition != true )
     t++                          % cycle iteration (time)
     constructing ant solutions
     (local search)
     updating pheromone values
     relocation of ants
end
```

Ant Colony Optimization

- Ant System (AS)
- *MAX-MIN* Ant System
- Ant Colony System
- Hyper-cube AS

Methods of pheromone updating

- Ant System
  - pheromone values are updated by all ants

$$\tau_{ij}^{t+1} = (1-\rho)\cdot\tau_{ij}^{t} + \sum_{k=1}^{m}\Delta\tau_{ij}^{t},\ \ \Delta\tau_{ij}^{t} = \begin{cases} Q/L_k & ^{*} \\ 0^{**} & \end{cases}$$

* if k-ant used edge (i,j) on its tour

** otherwise

Methods of pheromone updating

- *MAX-MIN* Ant System
  - pheromone values are updated by individual ants with the best solution found (best in each iteration, best of all)

$$\tau_{ij}^{t+1} = \left[(1-\rho)\cdot\tau_{ij}^{t} + \Delta\tau_{ij}^{best}\right]_{\tau_{\min}}^{\tau_{\max}},$$

$$\Delta\tau_{ij}^{best} = \begin{cases} 1/L_{best}^{\;*} \\ 0^{**} \end{cases}$$

* if (i,j) belongs to the best tour

** otherwise

Methods of pheromone updating

- Ant Colony System
  - local pheromone update after each iteration by all ants

$$\tau_{ij} = (1-\varphi)\cdot\tau_{ij} + \varphi\cdot\tau_0$$

  - global pheromone update at the end of iteration as in *MAX-MIN* AS

$$\tau_{ij} = \begin{cases} (1-\rho)\cdot\tau_{ij} + \rho\cdot\Delta\tau_{ij}^{\;*} \\ \tau_{ij}^{\;**} \end{cases}$$

* if (i,j) belongs to the best tour, $\Delta\tau_{ij} = 1/L_{best}$

** otherwise

References

[7] Colorni, A.; Dorigo, M.; Maniezzo, V. (1991). Distributed Optimization by Ant Colonies. In European Conference on Artificial Life, 134-142.

[8] Blum, Ch. (2005). Ant Colony Optimization: Introduction and Recent Trends. Physics of Life Reviews, 2(4): 353 – 373.

[9] Dorigo, M. (2004). Ant Colony Optimization. 1. vyd. The MIT Press. ISBN: 978-0262042192.

[10] Dorigo, M.; Birattari, M.; Stutzle, T. (2006). Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique. Technical Report No. RT/IRIDIA/2006-023, Université Libre de Bruxelles, IRIDIA.

Swarm Intelligence - summary

- stochastic methods – do not guarantee the global optima retrieval
- difference in „communication“
  - PSO – on swarm level
  - ACO – locally by pheromones

References

http://www.mathworks.com/matlabcentral/fileexchange/11559-particle-swarm-optimization-simulation

http://www.mathworks.com/matlabcentral/fileexchange/25986-another-particle-swarm-toolbox

http://www.liacs.nl/~baeck/NC/slides/Applet/ants.html

http://www.mathworks.com/matlabcentral/fileexchange/14543

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: 25.11.2011
Version: 001