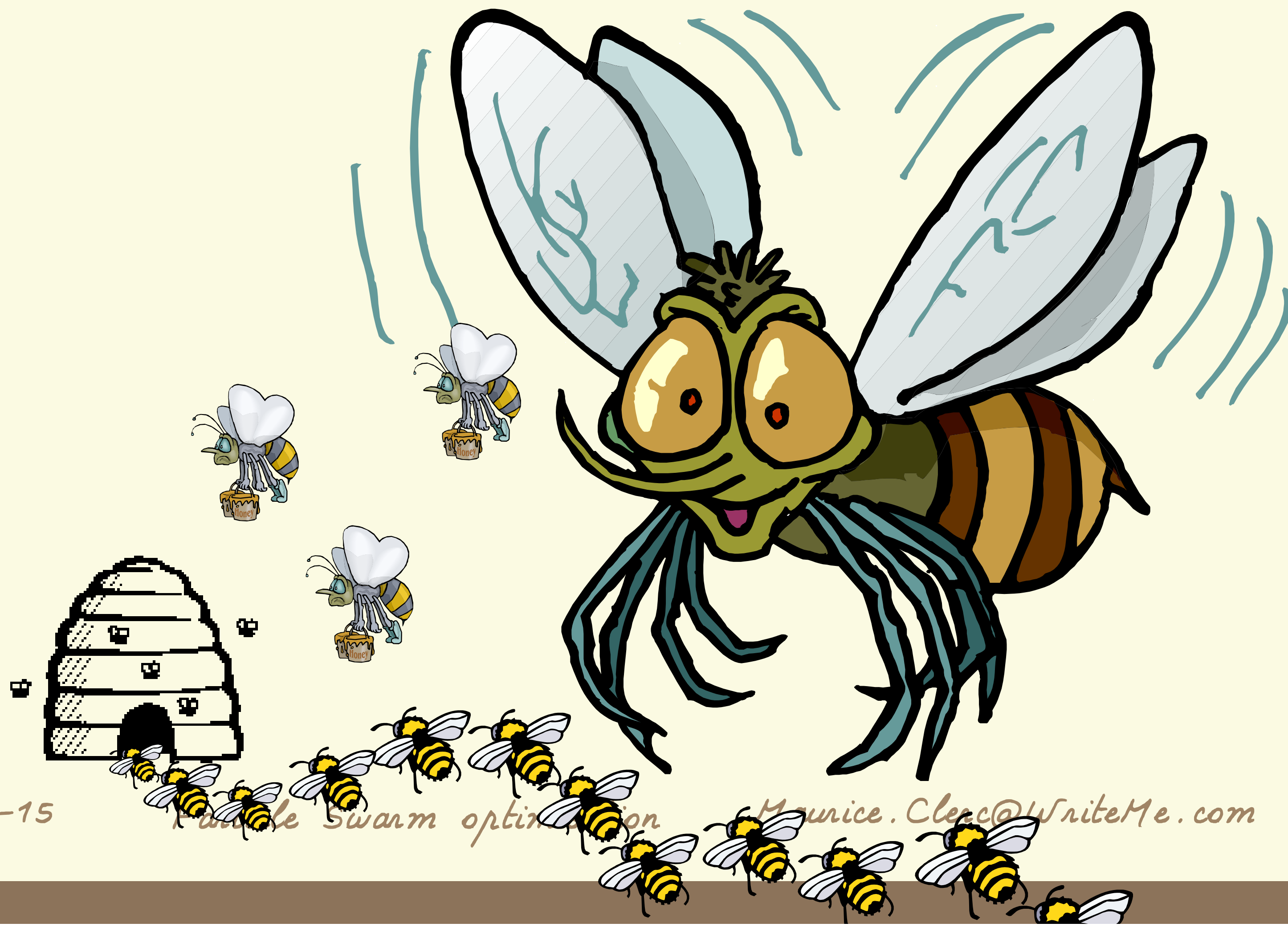
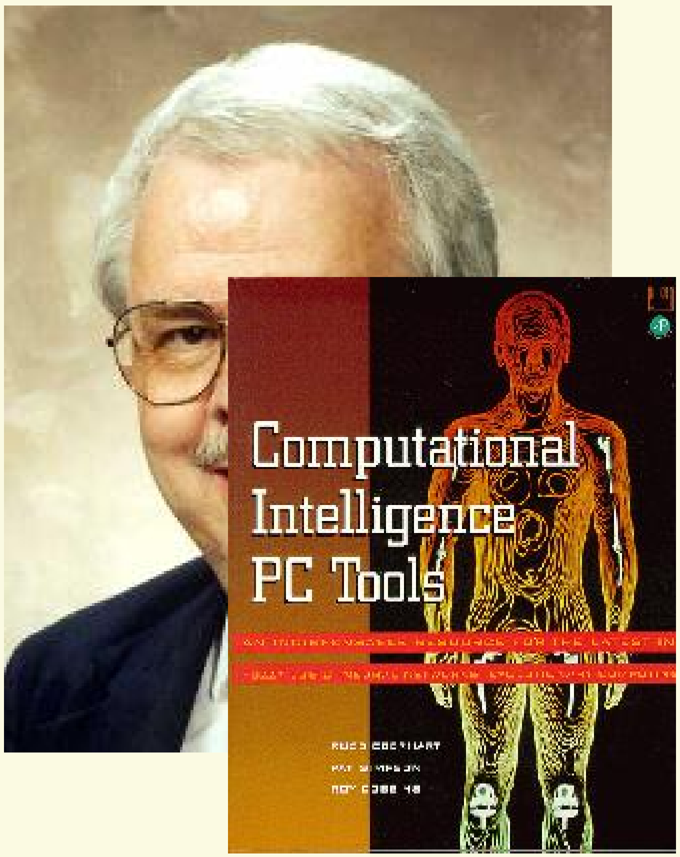
# Particle Swarm optimisation: A mini tutorial



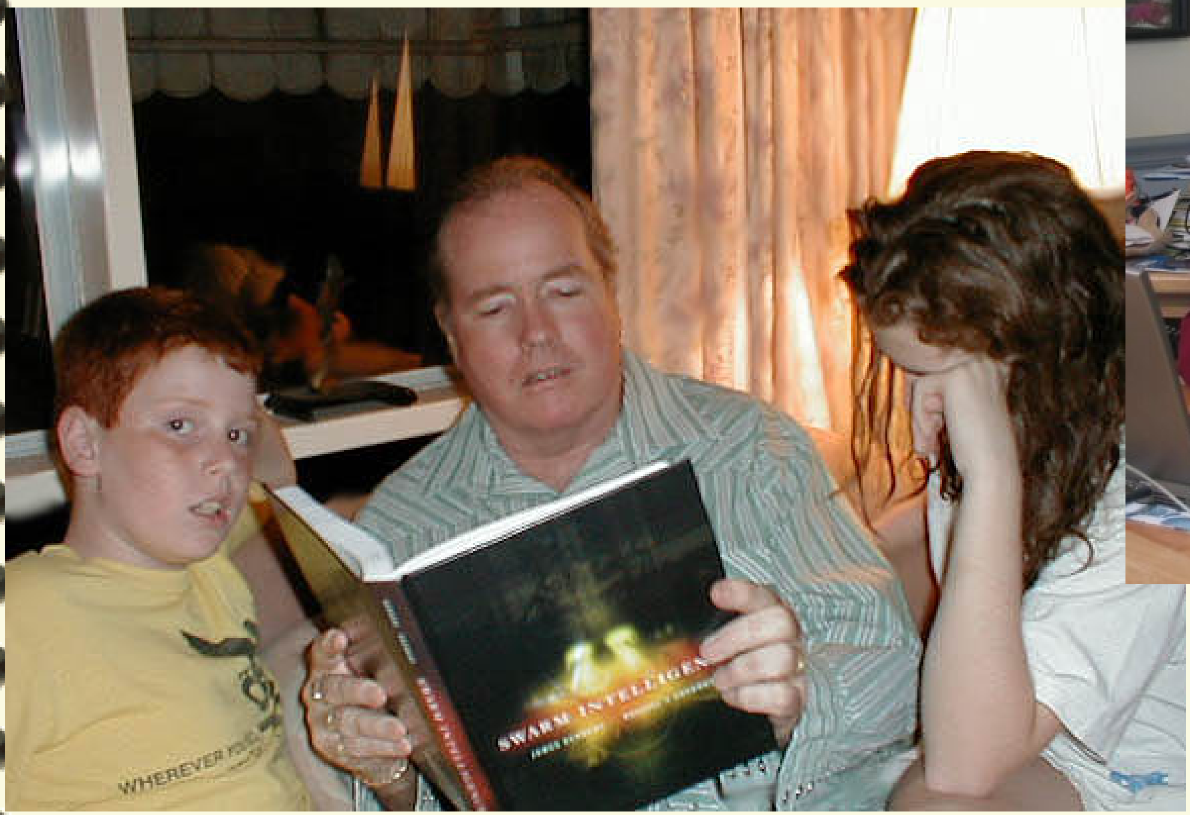
# The "inventors" (1)




Russell Eberhart

eberhart@engr.iupui.edu

# The "inventors" (2)

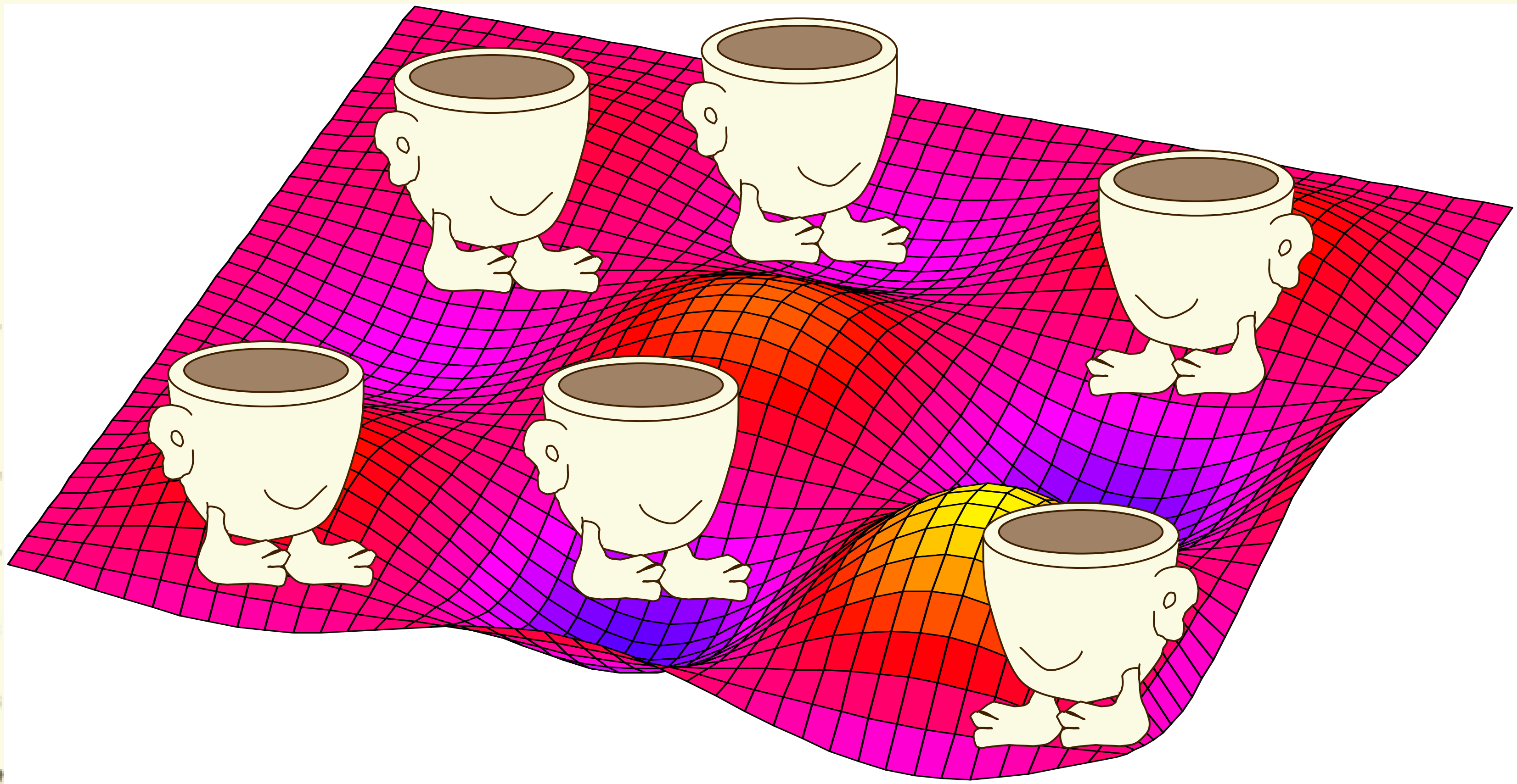


Jim at work

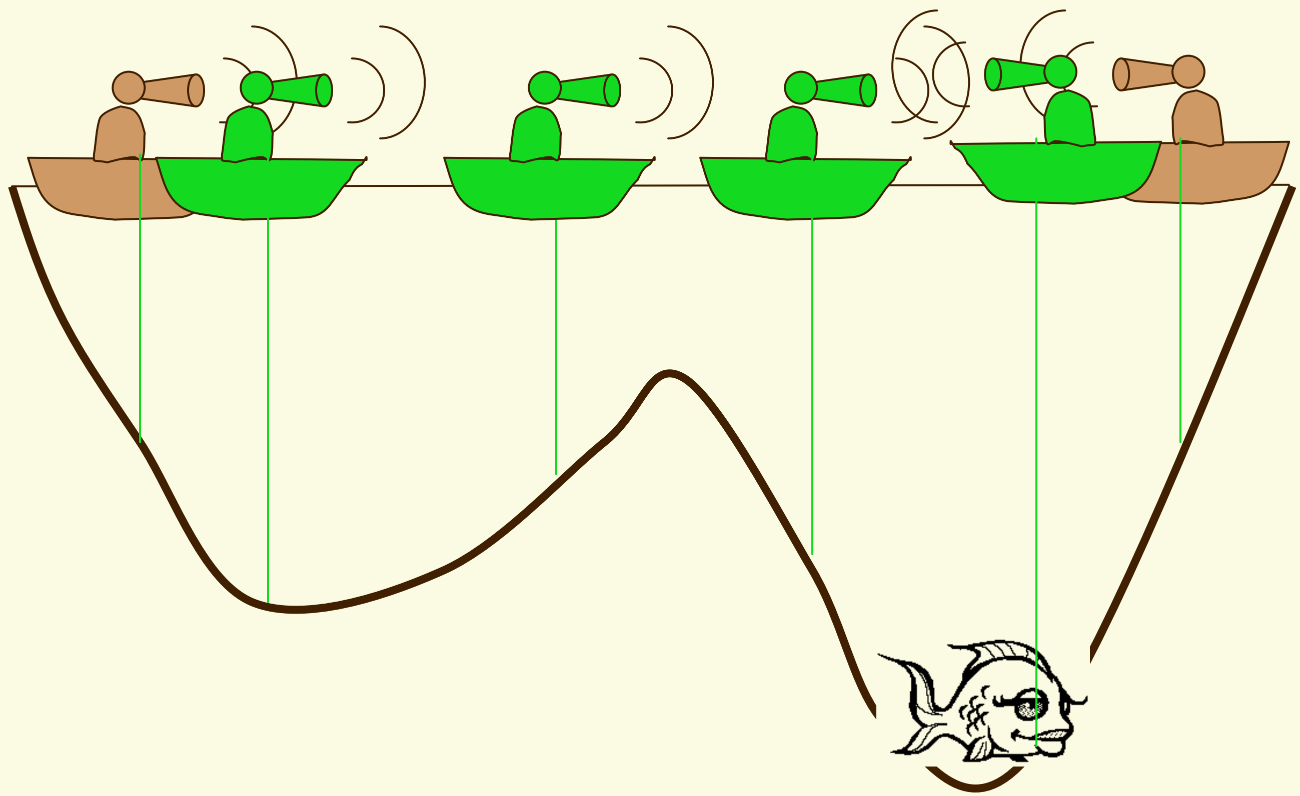
James Kennedy

Kennedy_Jim@bls.gov

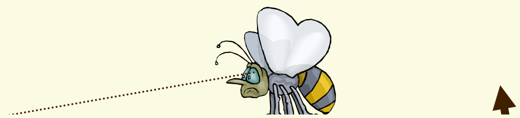
# Part 1: United we stand



# Cooperation example

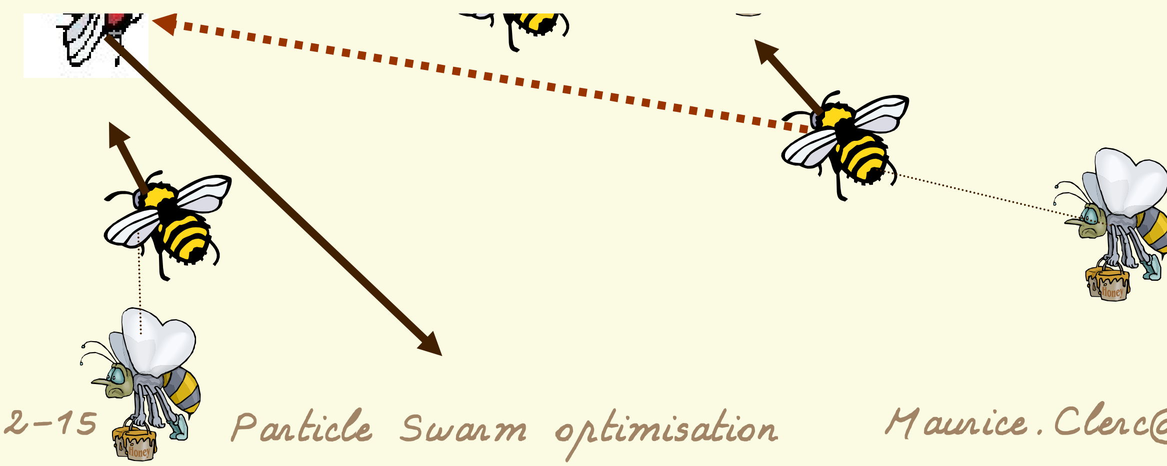


# Memory and informers



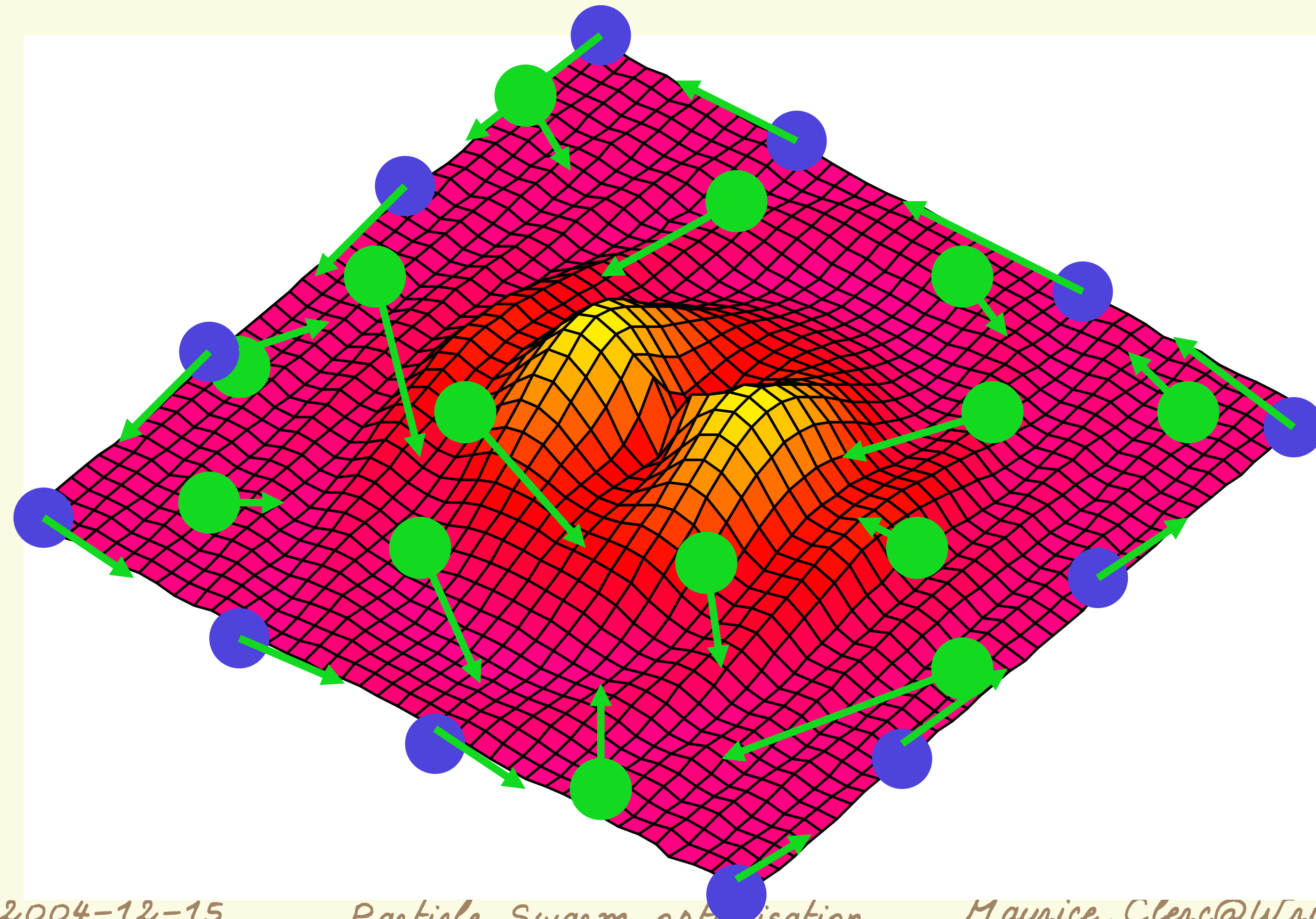
*It is thanks to these eccentrics, whose behaviour is not conform to the one of the other bees, that* all *fruits sources around the colony are so quickly found.*

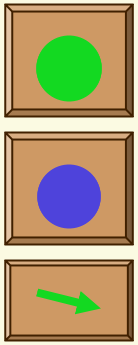
Karl von Frish *1927*



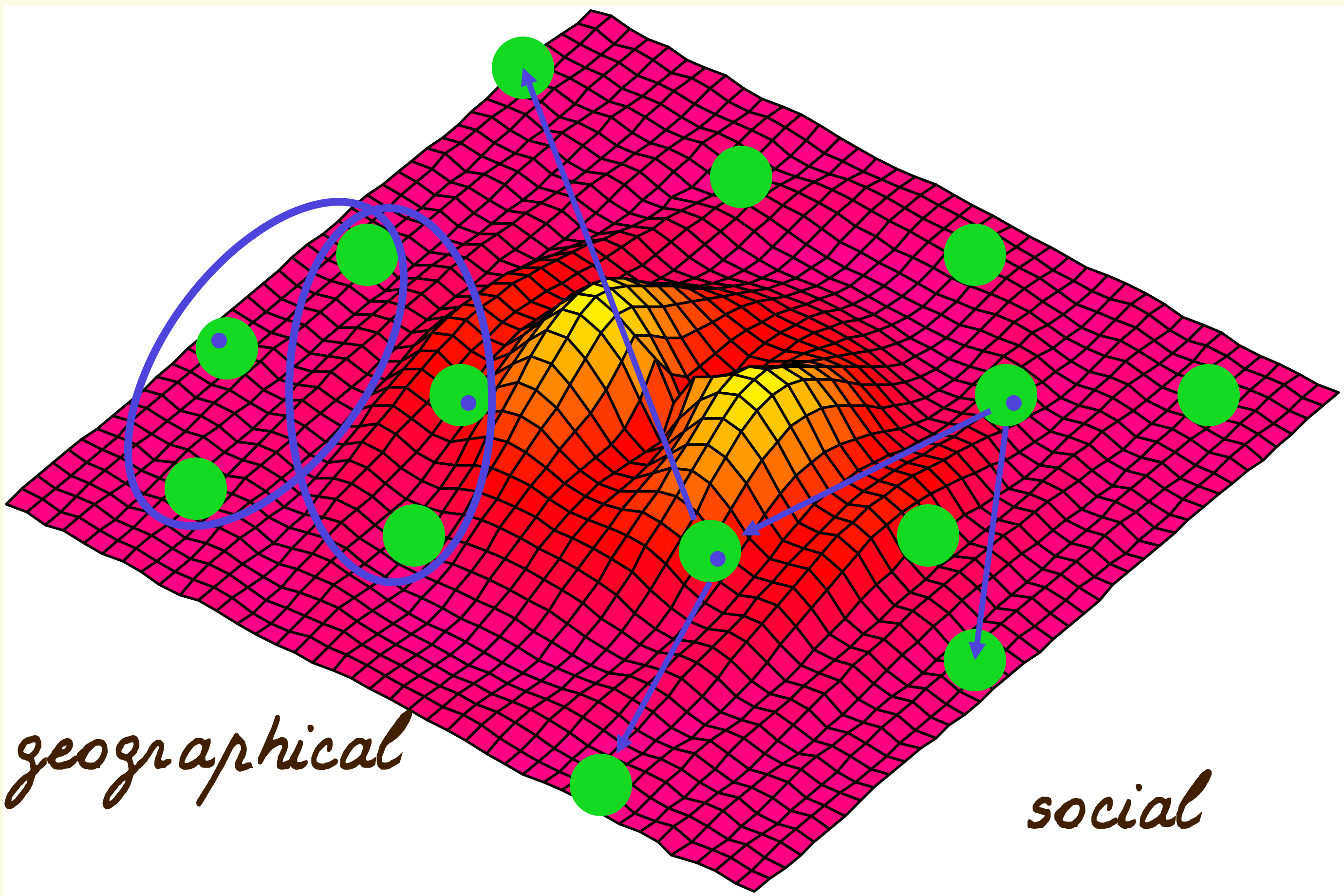
DPNP

# Initialisation. Positions and velocities

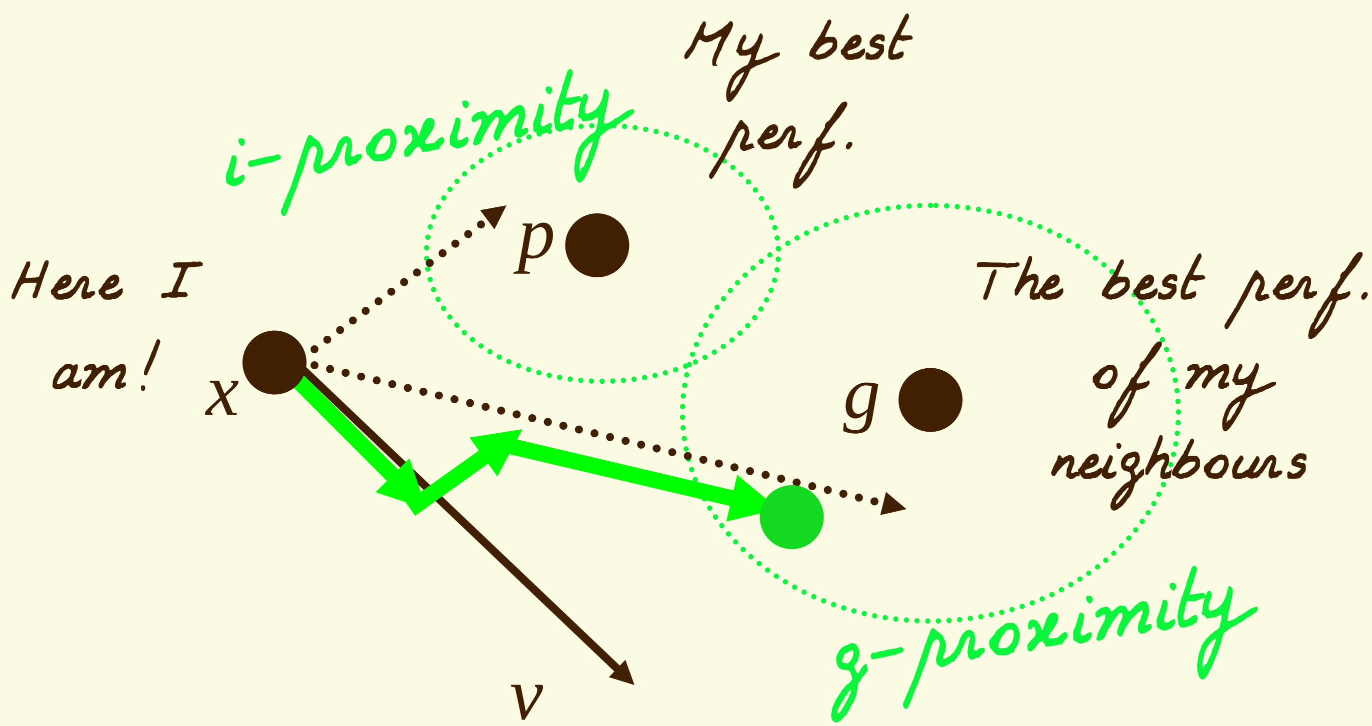




# Neighbourhoods



# Psychosocial compromise



# The historical algorithm

At each time step $t$

for each particle

for each component $d$

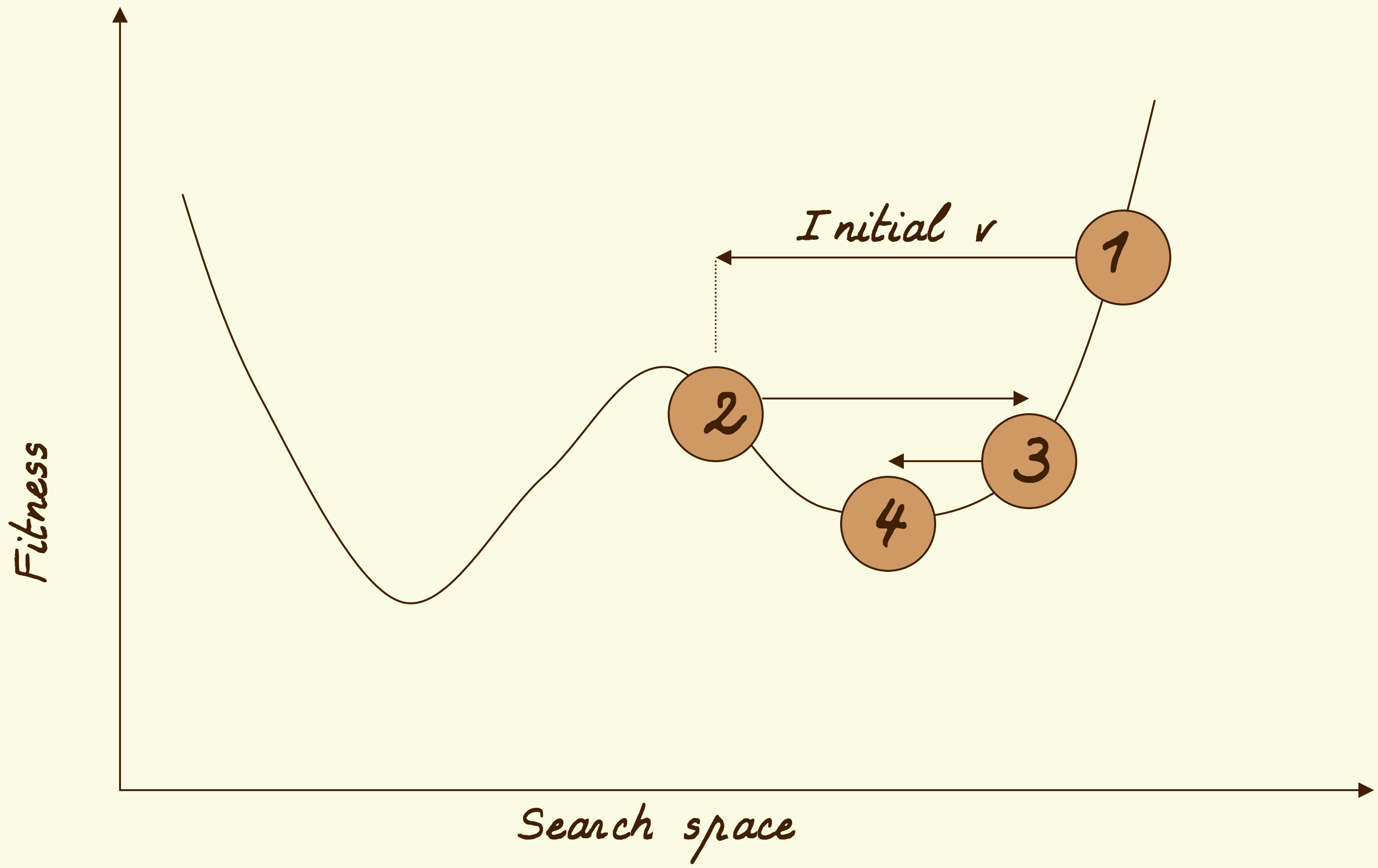
update the velocity

$$v_d(t+1) = \alpha v_d(t) + \beta rand(0,\varphi_1)\left(p_d - x_d(t)\right) + \beta rand(0,\varphi_2)\left(g_d - x_d(t)\right)$$

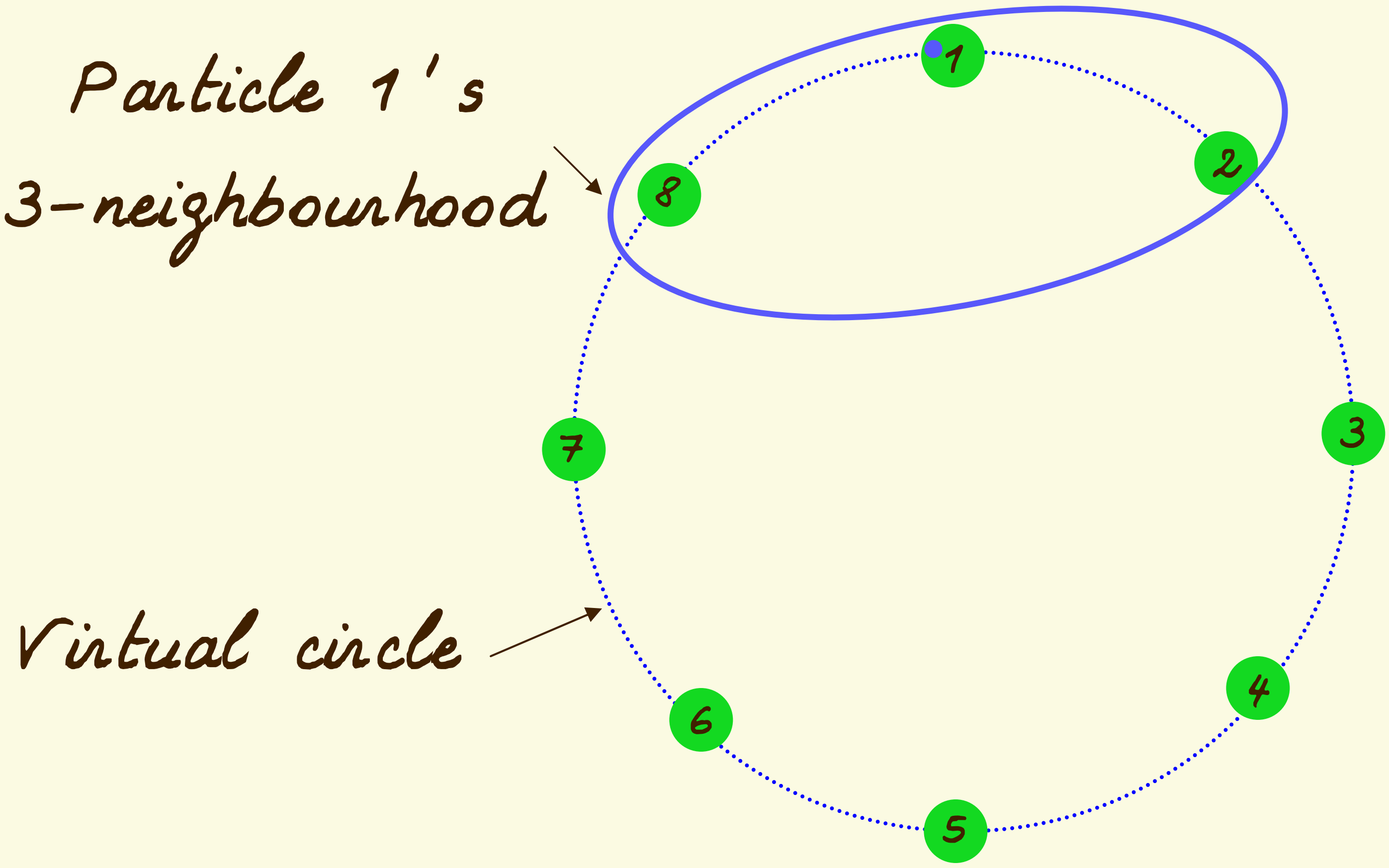
then move $x(t+1) = x(t) + v(t+1)$

Randomness inside the loop

# Oscillations

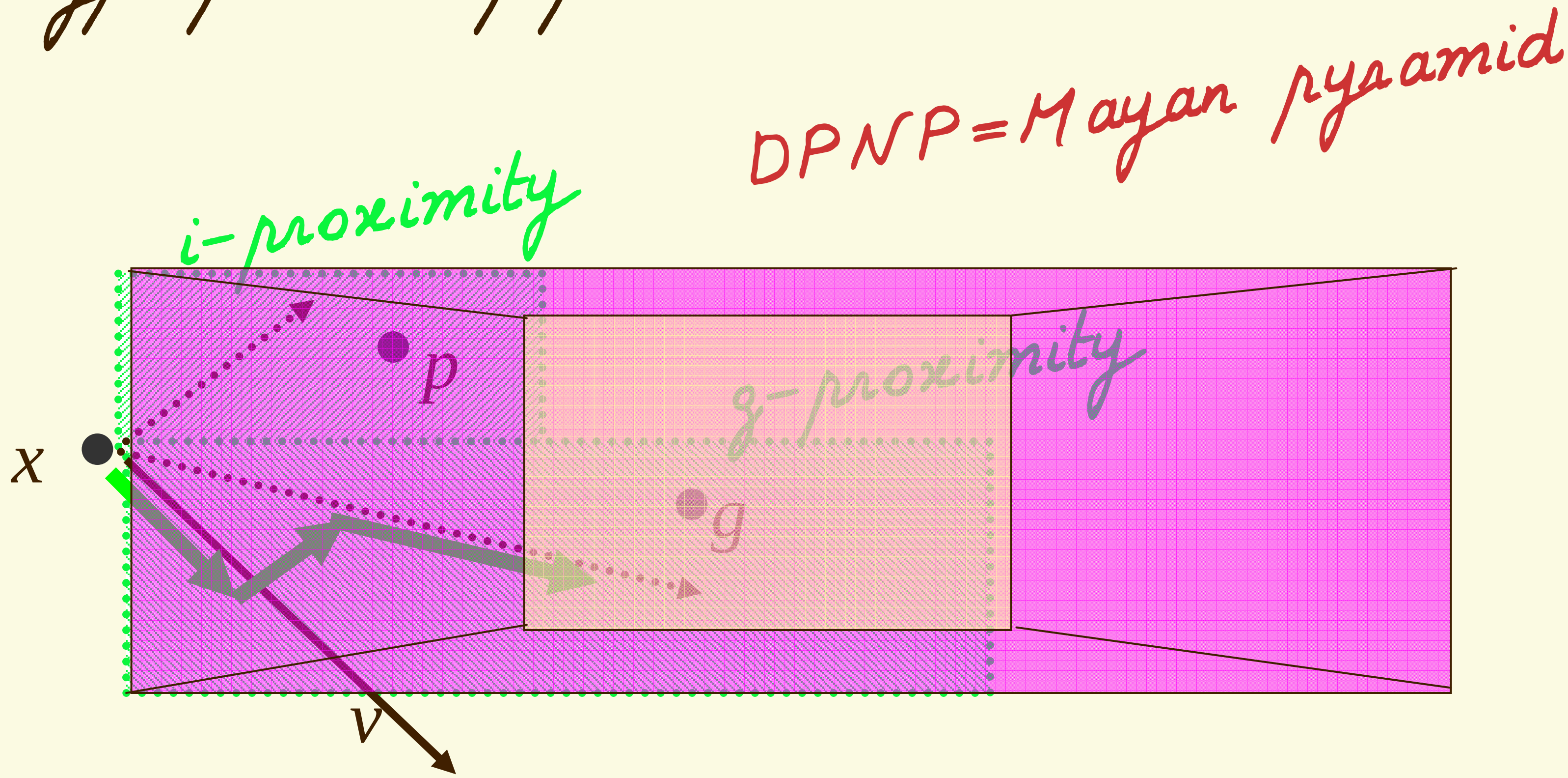


# The circular neighbourhood

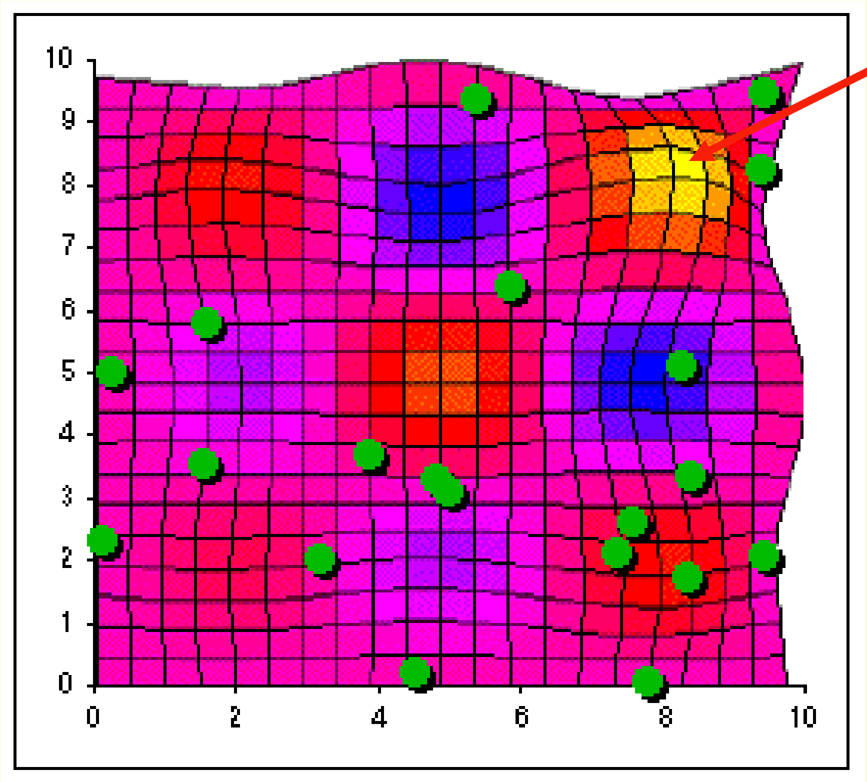


# Random proximity

Hyperparallelepiped => Biased

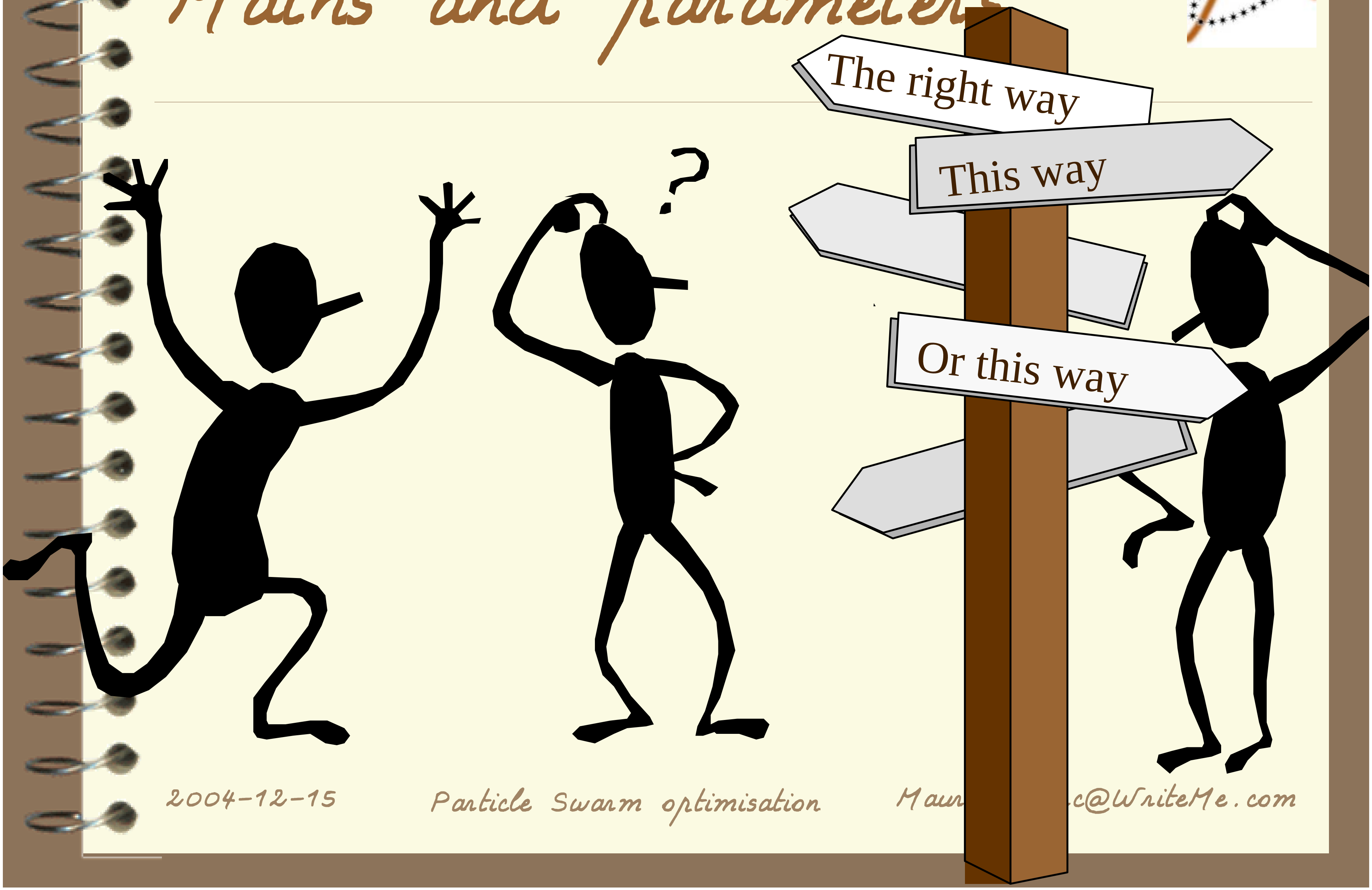


# Animated illustration

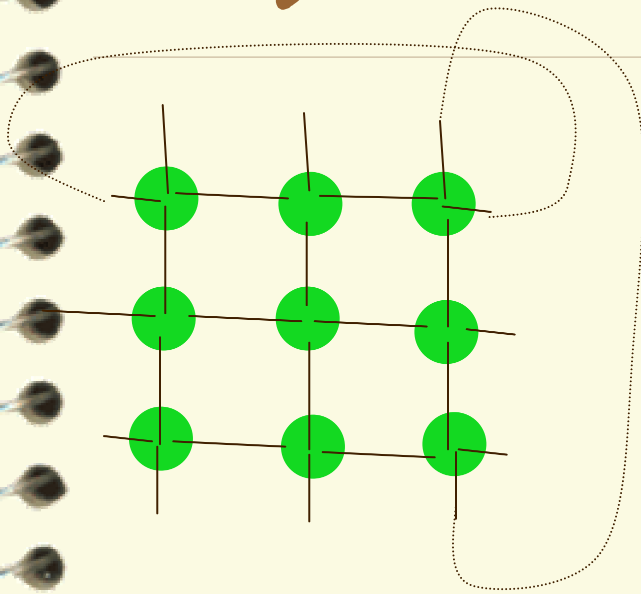


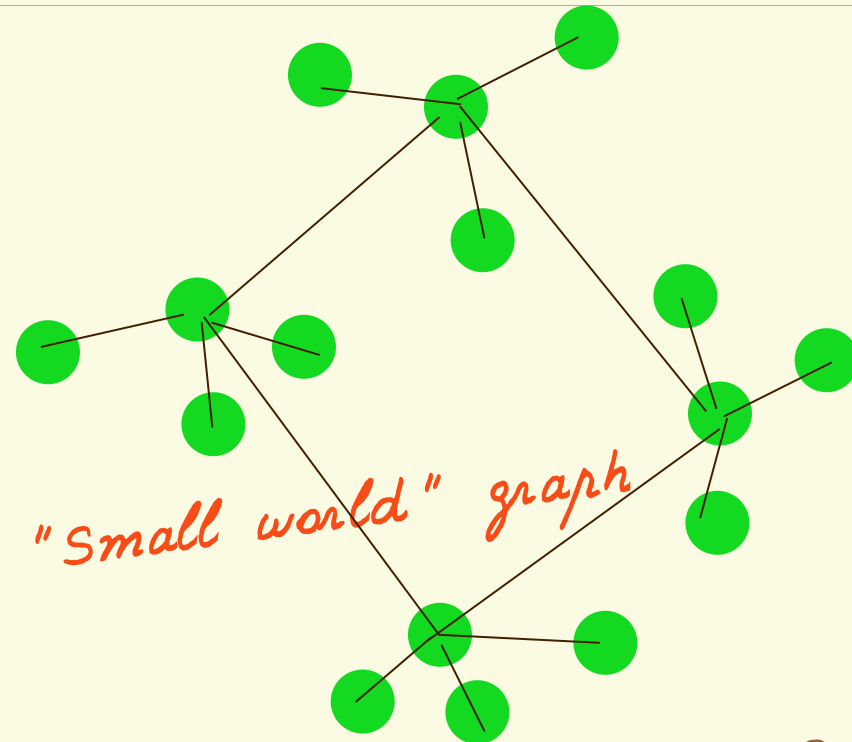

Global optimum

# Maths and parameters



# Neighbourhoods (topologies)






↔ or → ?

(informers)

# Type 1" form

Global constriction coefficient

$$\begin{cases} v(t+1) = \chi\left(v(t) + \varphi(q - x(t))\right) \\ x(t+1) = v(t+1) + x(t) \end{cases}$$

with

$$\varphi = rand(0,\varphi_1) + rand(0,\varphi_2) = \varphi'_1 + \varphi'_2$$

$$q = \frac{\varphi'_1 p + \varphi'_2 g}{\varphi'_1 + \varphi'_2}$$

$$\chi = \begin{cases} \dfrac{2\kappa}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}} & \text{for } \varphi > 4 \\ else\ \sqrt{\kappa} \end{cases}$$

Non divergence criterion

Usual values:

$\kappa=1$

$\varphi=4.1$

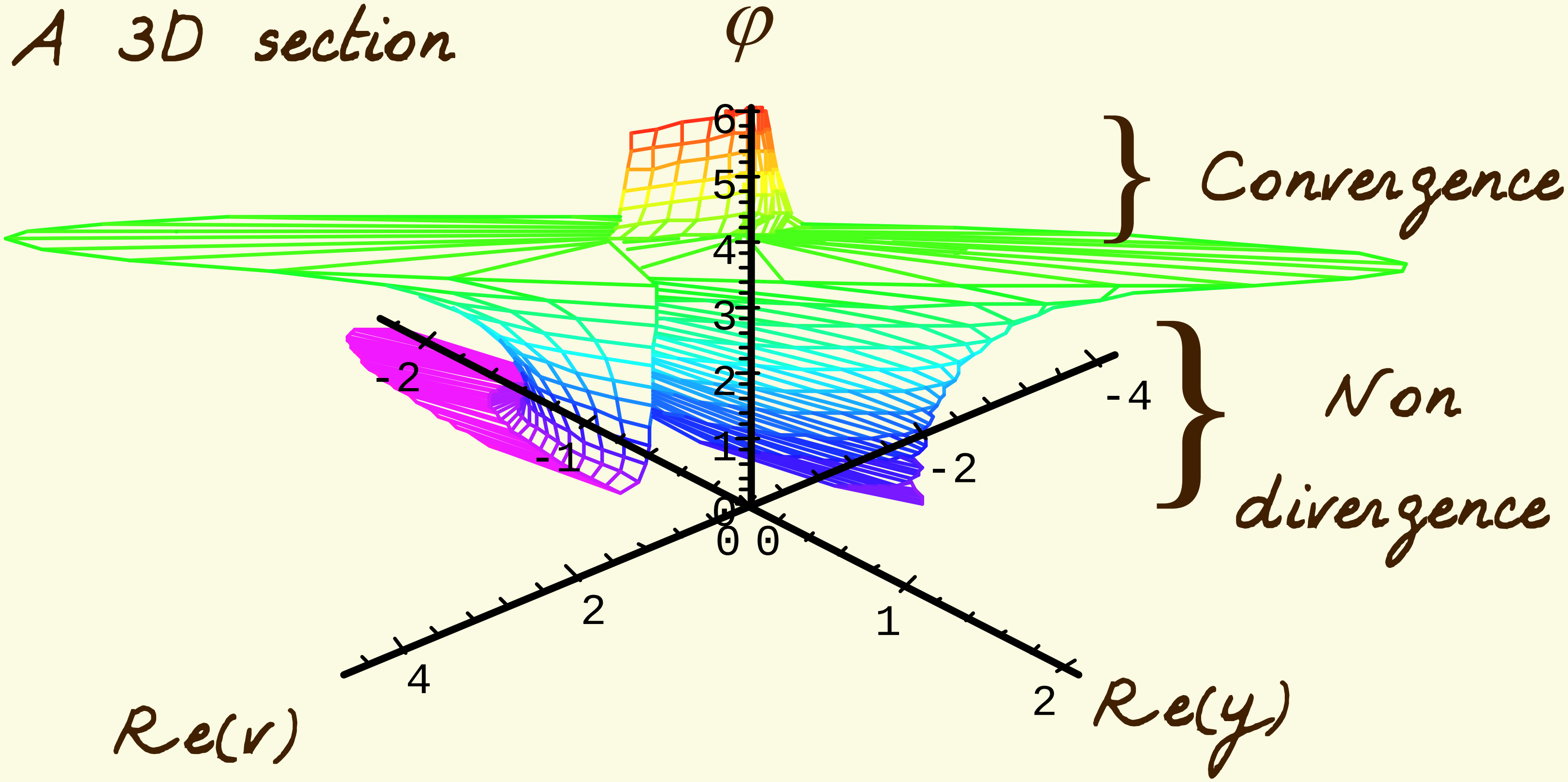
$\Rightarrow \chi=0.73$

swarm size=20

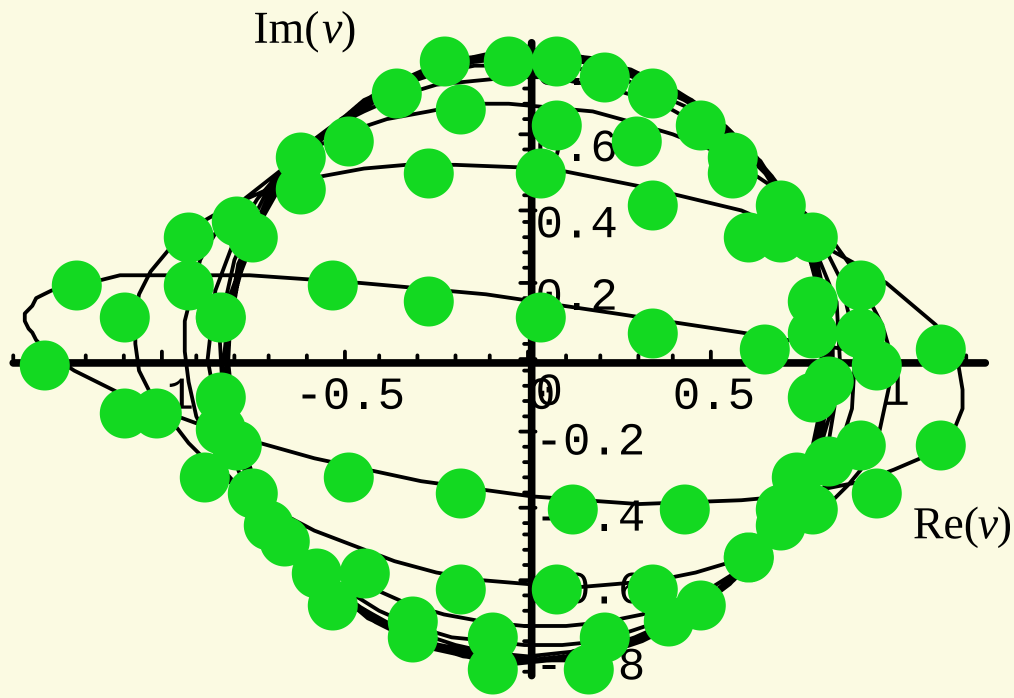
hood size=3

# 5D complex space

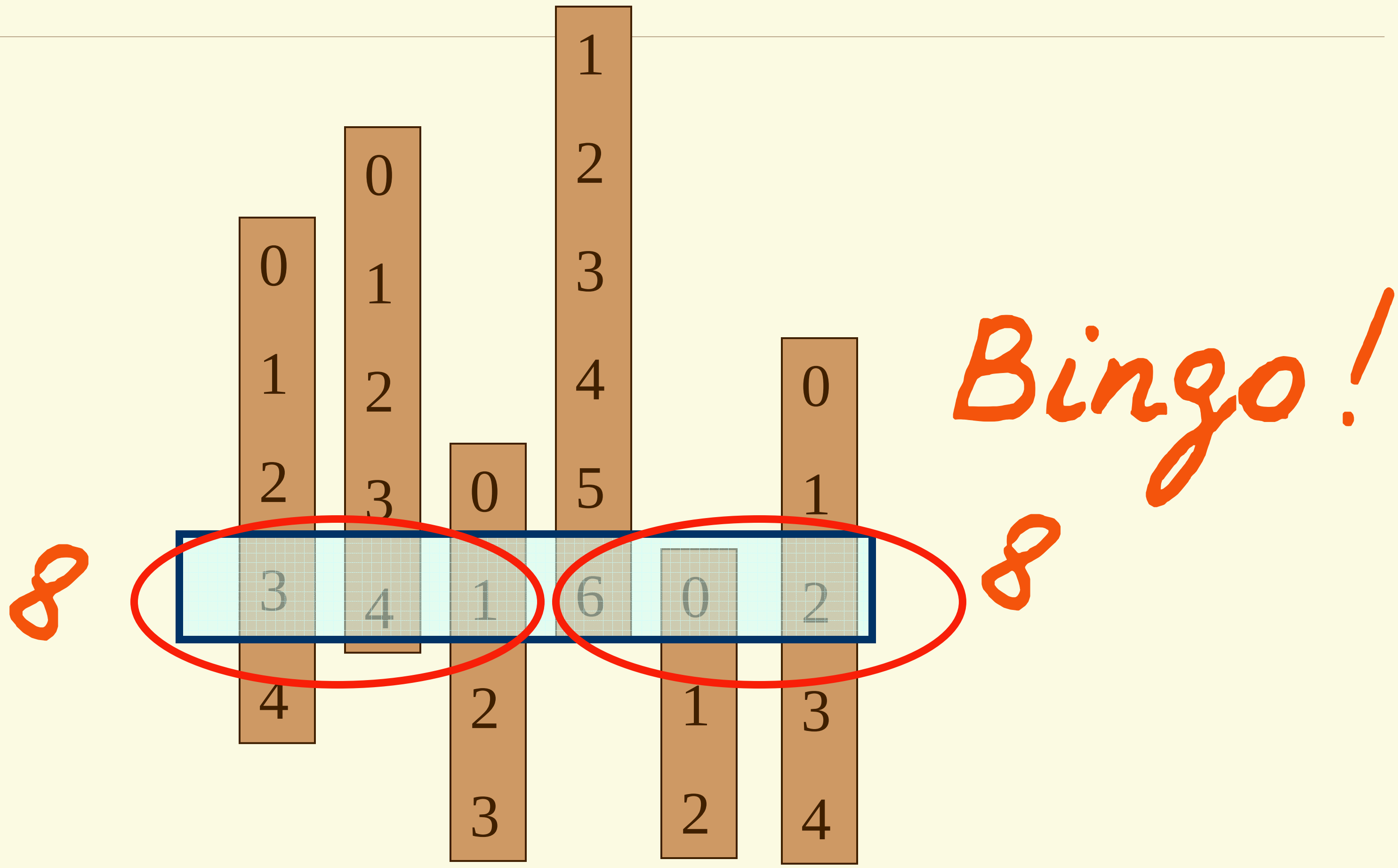
## A 3D section



# Move in a 2D section (attractor)



# Beyond real numbers



# Minimun requirements

## Comparing positions in the search space H

$$\forall (x,x') \in H \times H, \left(f(x) < f(x')\right) \vee \left(f(x) \geq f(x')\right)$$

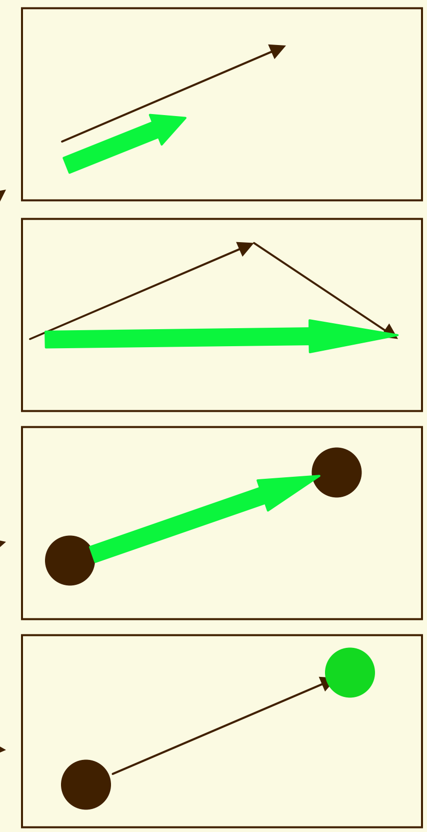
## Algebraic operators

$$(\text{coefficient}, \text{velocity}) \xrightarrow{\otimes} \text{velocity}$$

$$(\text{velocity}, \text{velocity}) \xrightarrow{\circ} \text{velocity}$$

$$(\text{position}, \text{position}) \xrightarrow{\Theta} \text{velocity}$$

$$(\text{position}, \text{velocity}) \xrightarrow{\oplus} \text{position}$$



# Pseudo code form

velocity = pos_minus_pos(position$_1$, position$_2$)

velocity = linear_combin($\alpha$,velocity$_1$,$\beta$,velocity$_2$)

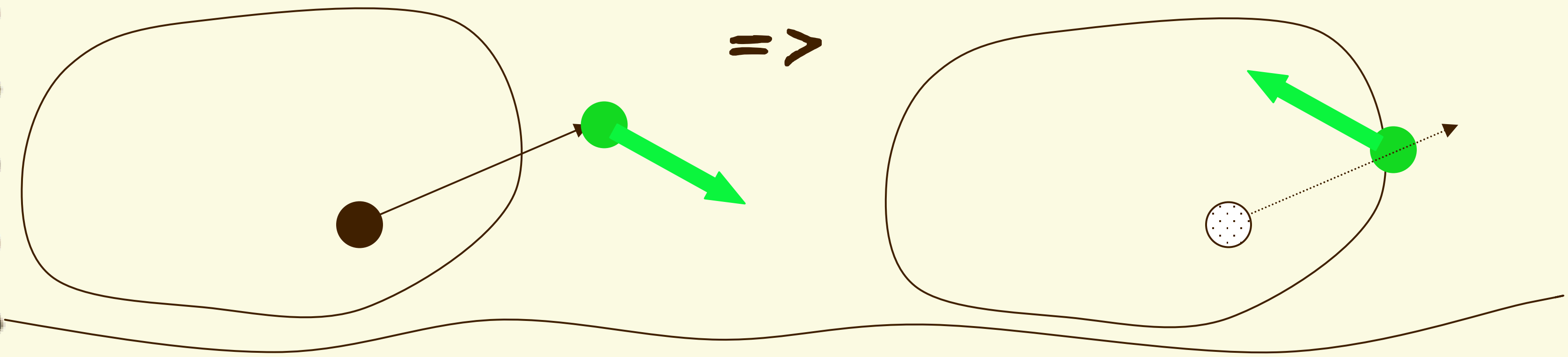
position = pos_plus_vel(position, velocity)

*algebraic operators*

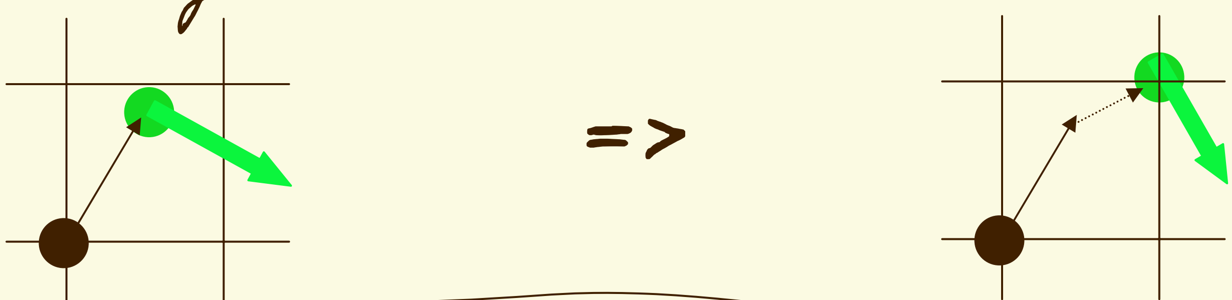
(position,velocity) = confinement(position$_{t+1}$,position$_t$)

# Confinements

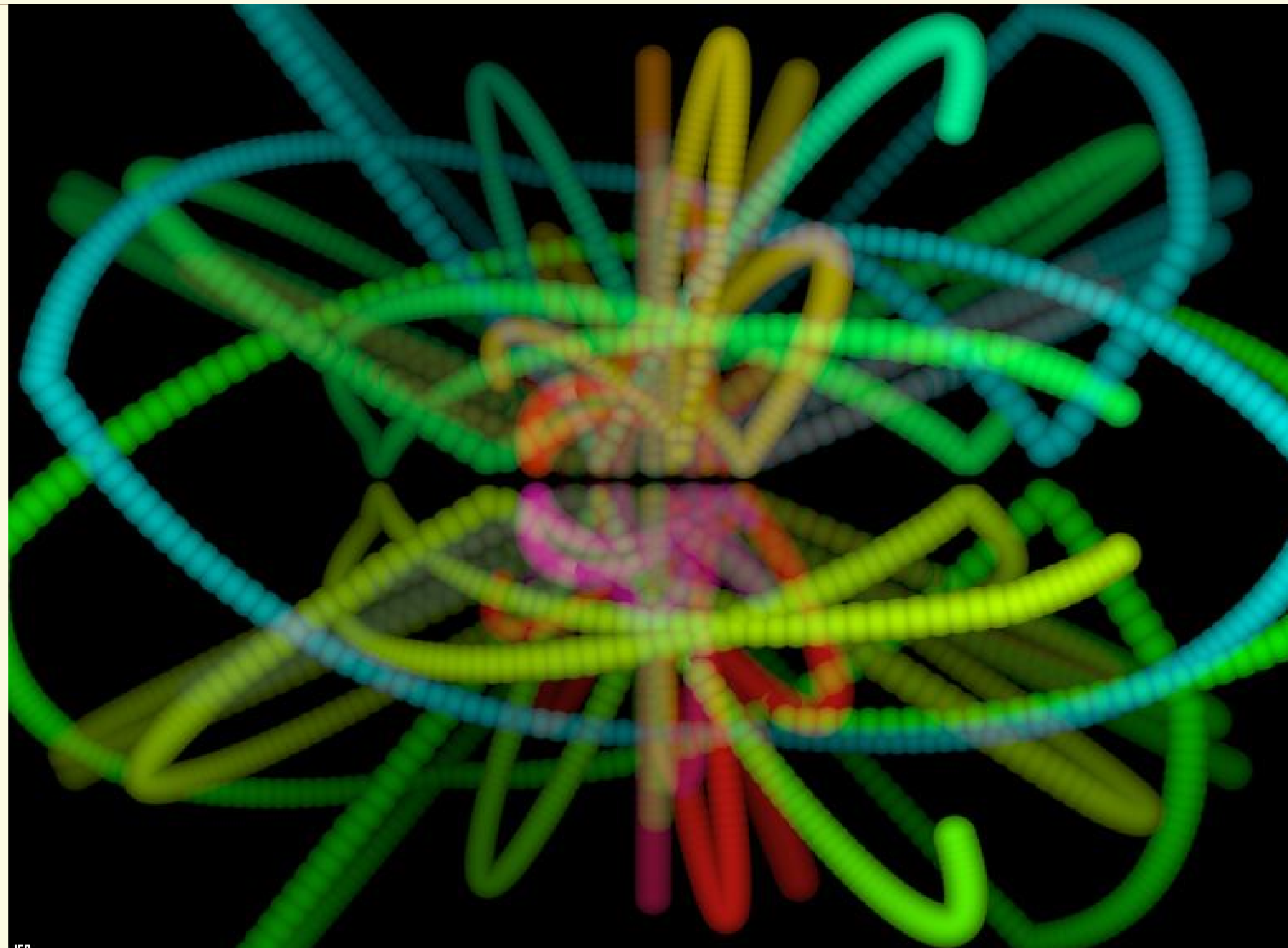
Frontiers (ex. : interval)



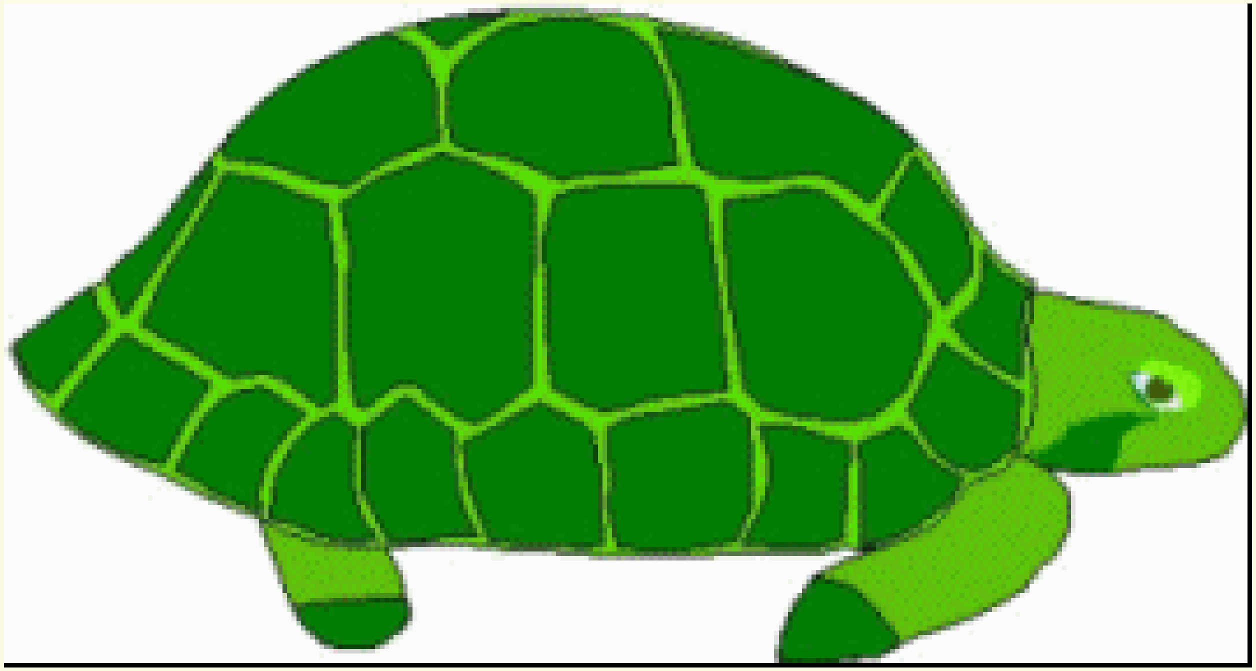
Granularity



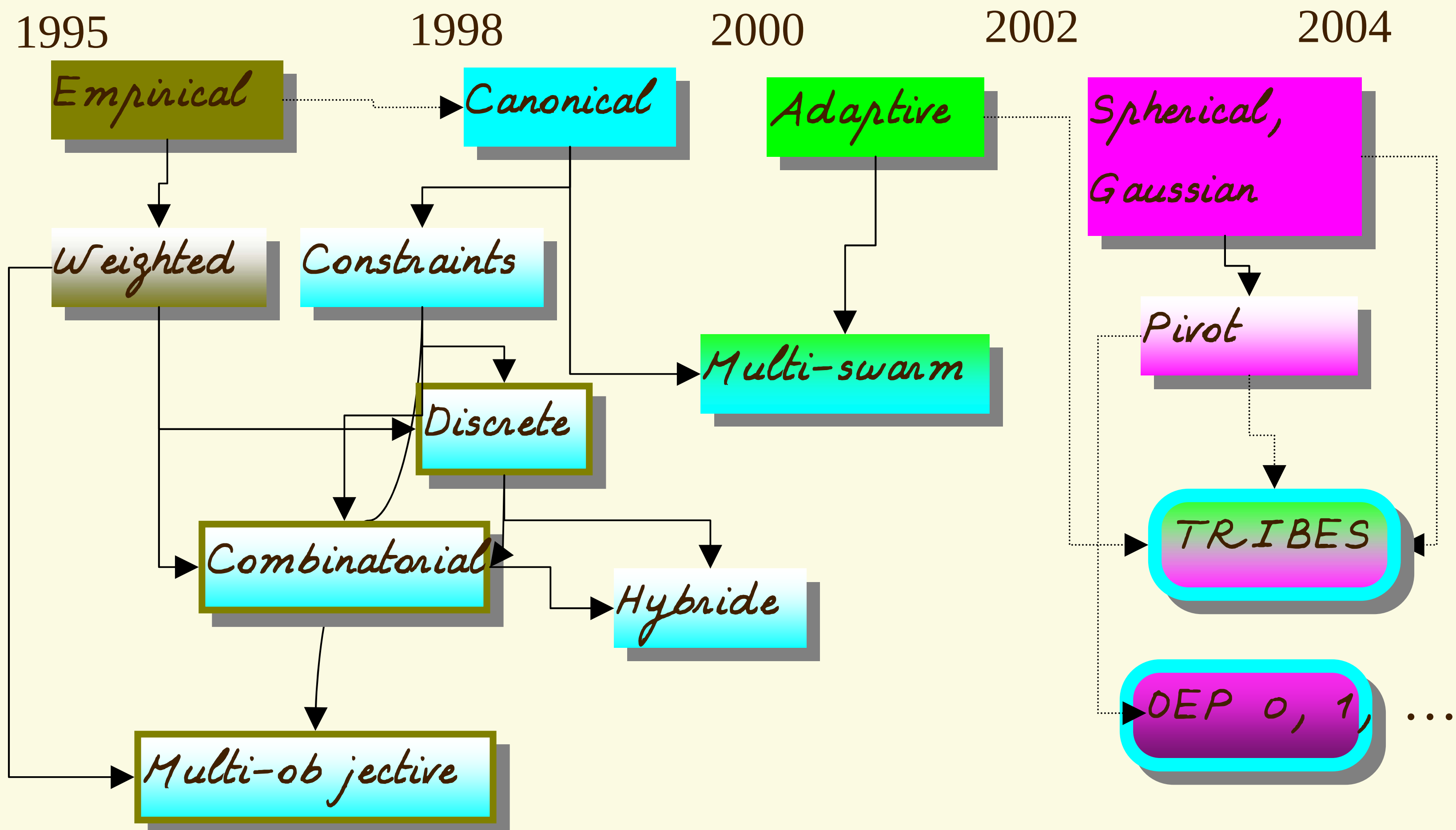
# End of Part 1



# Part 2: When the algo mutates

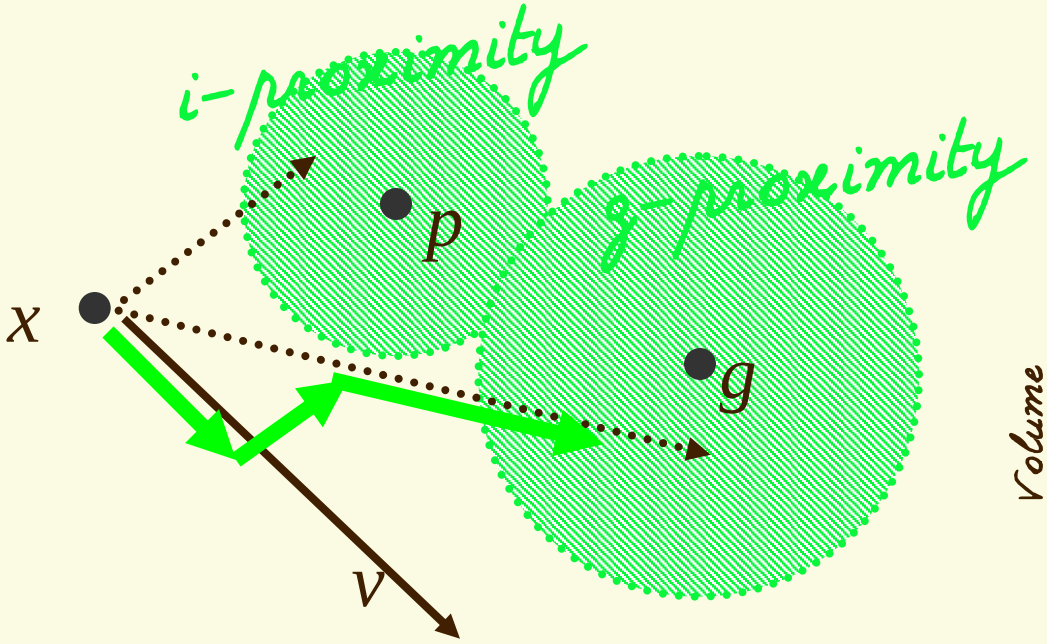


# The PSO Family

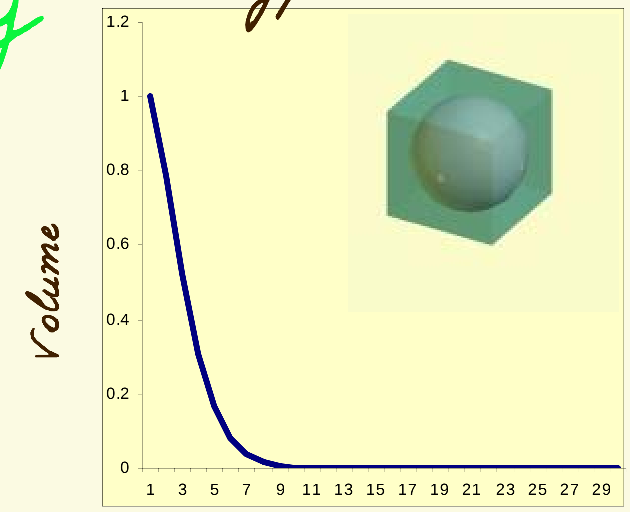


# Unbiased random proximity

Hyperparallelepiped => Biased



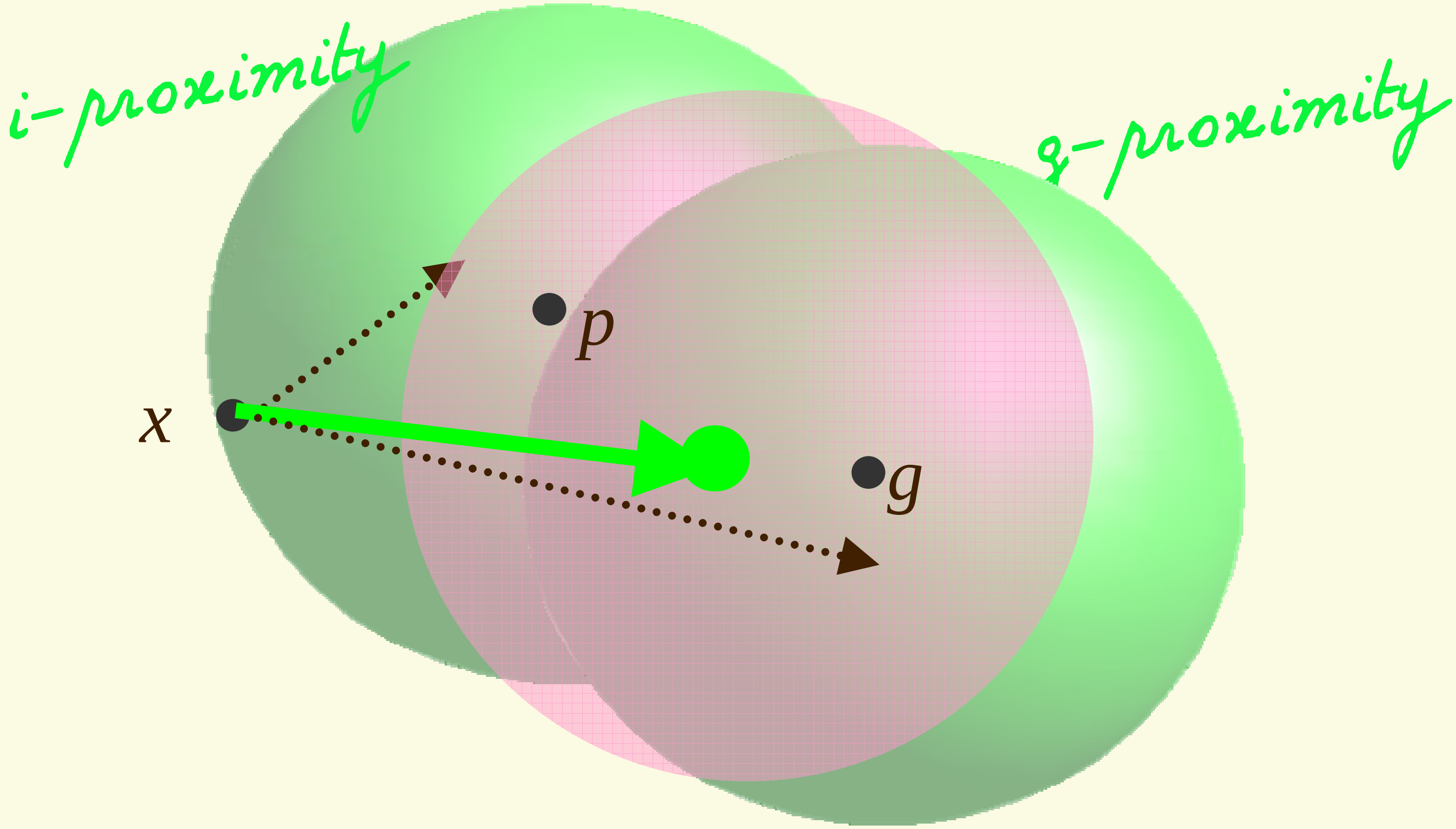

Hypersphere vs hypercube



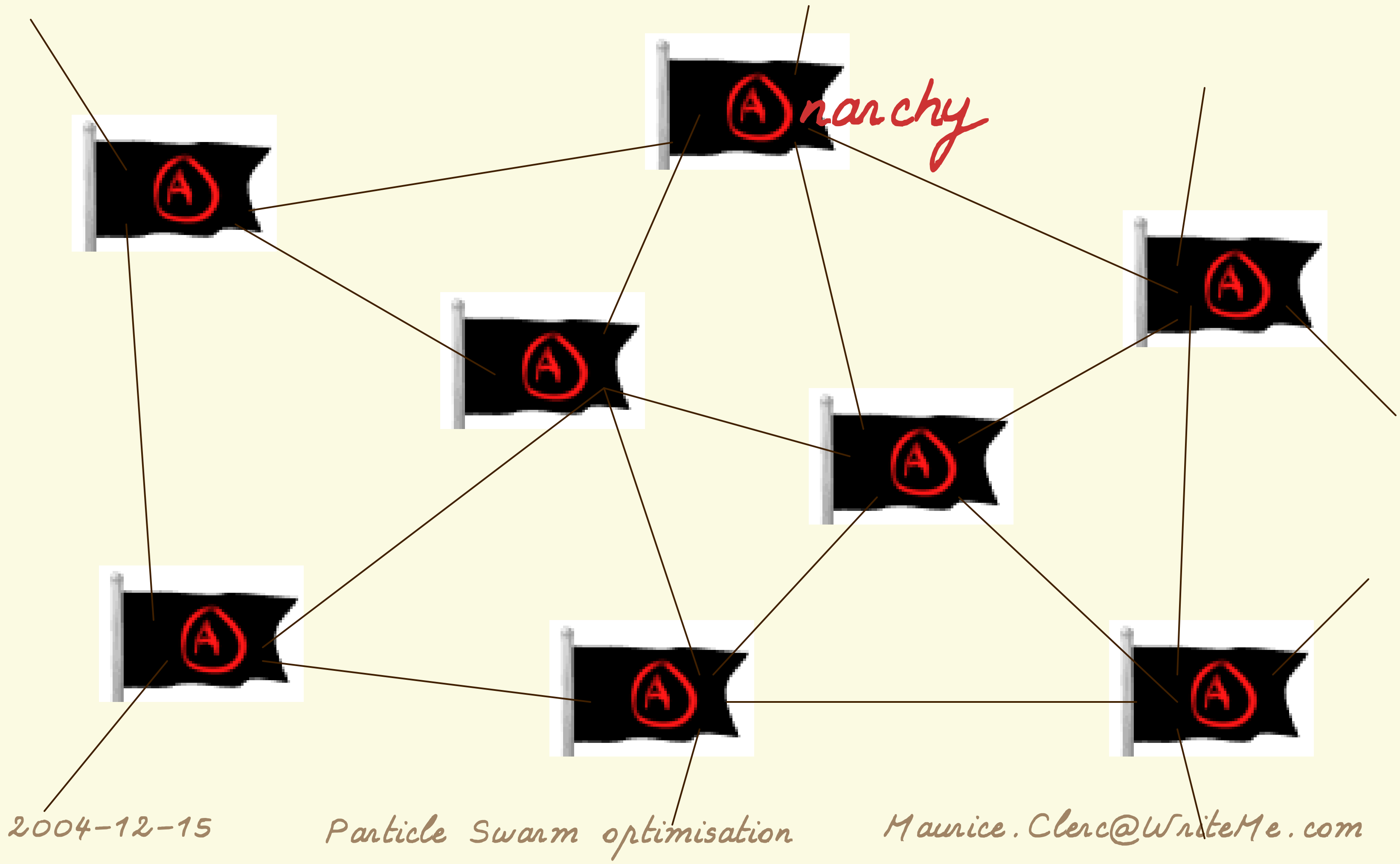

Dimension


Maurice.Clerc@WriteMe.com

# The three balls

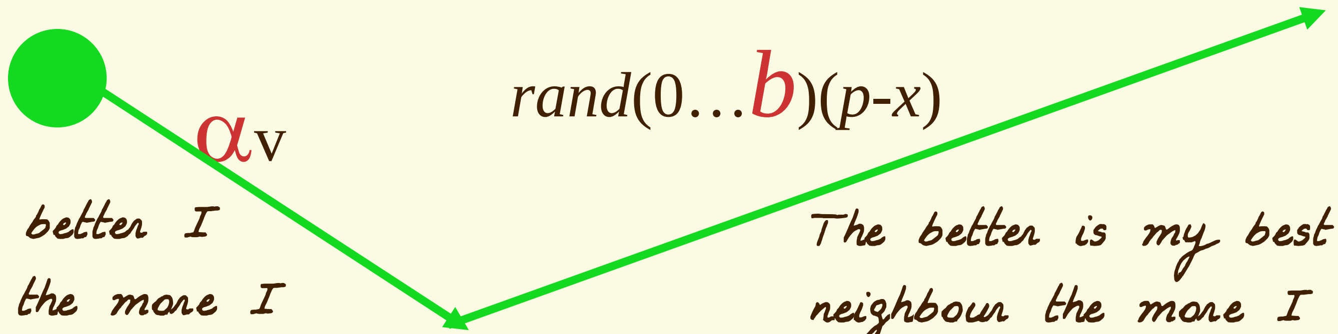


# Think locally, act locally



# Adaptive coefficients

Crisp or fuzzy rules



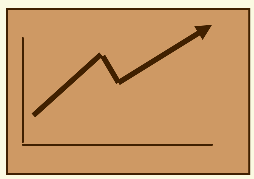

The better I am the more I follow my own way

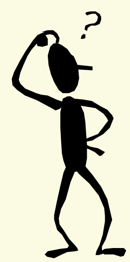
The better is my best neighbour the more I tend to go towards him

# Adaptive swarm size

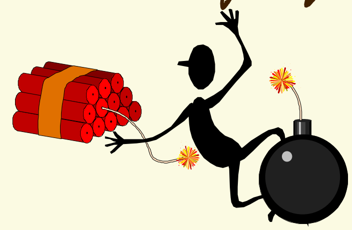
Crisp or fuzzy rules

There has been enough improvement
although I'm the worst

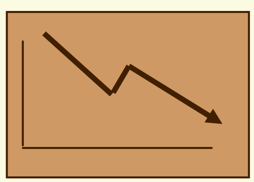




I try to kill myself



I'm the best
but there has been not enough improvement



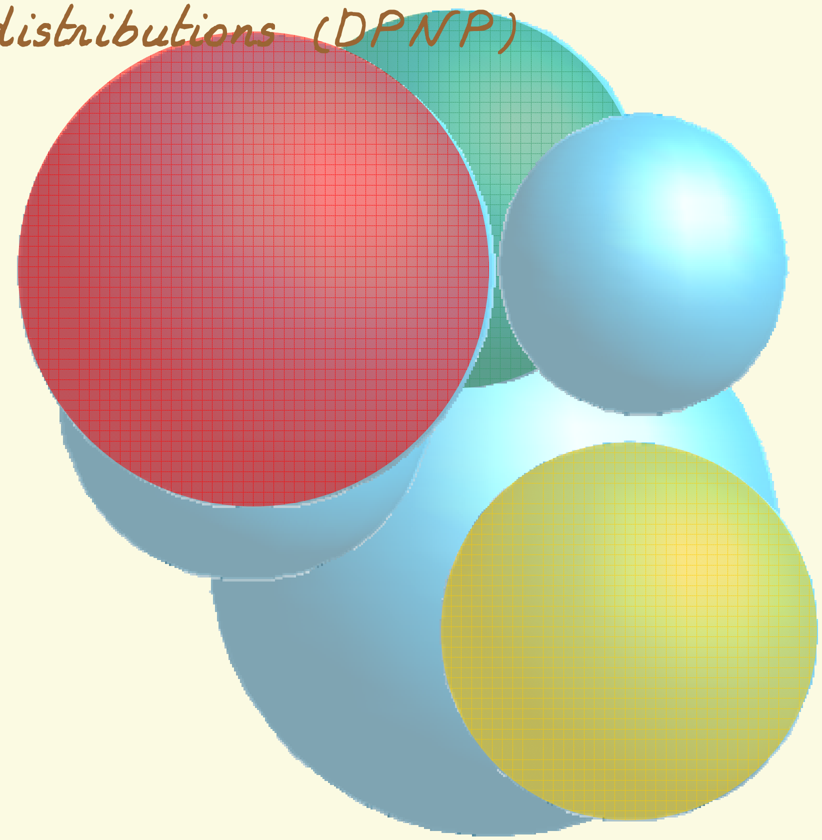
I try to generate a new particle


Maurice.Clerc@WriteMe.com

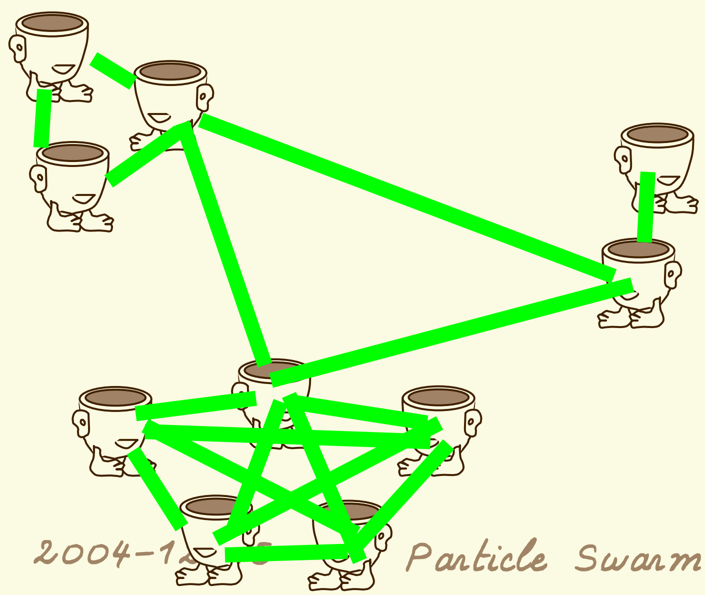
# TRIBES and strategies

TRIBES

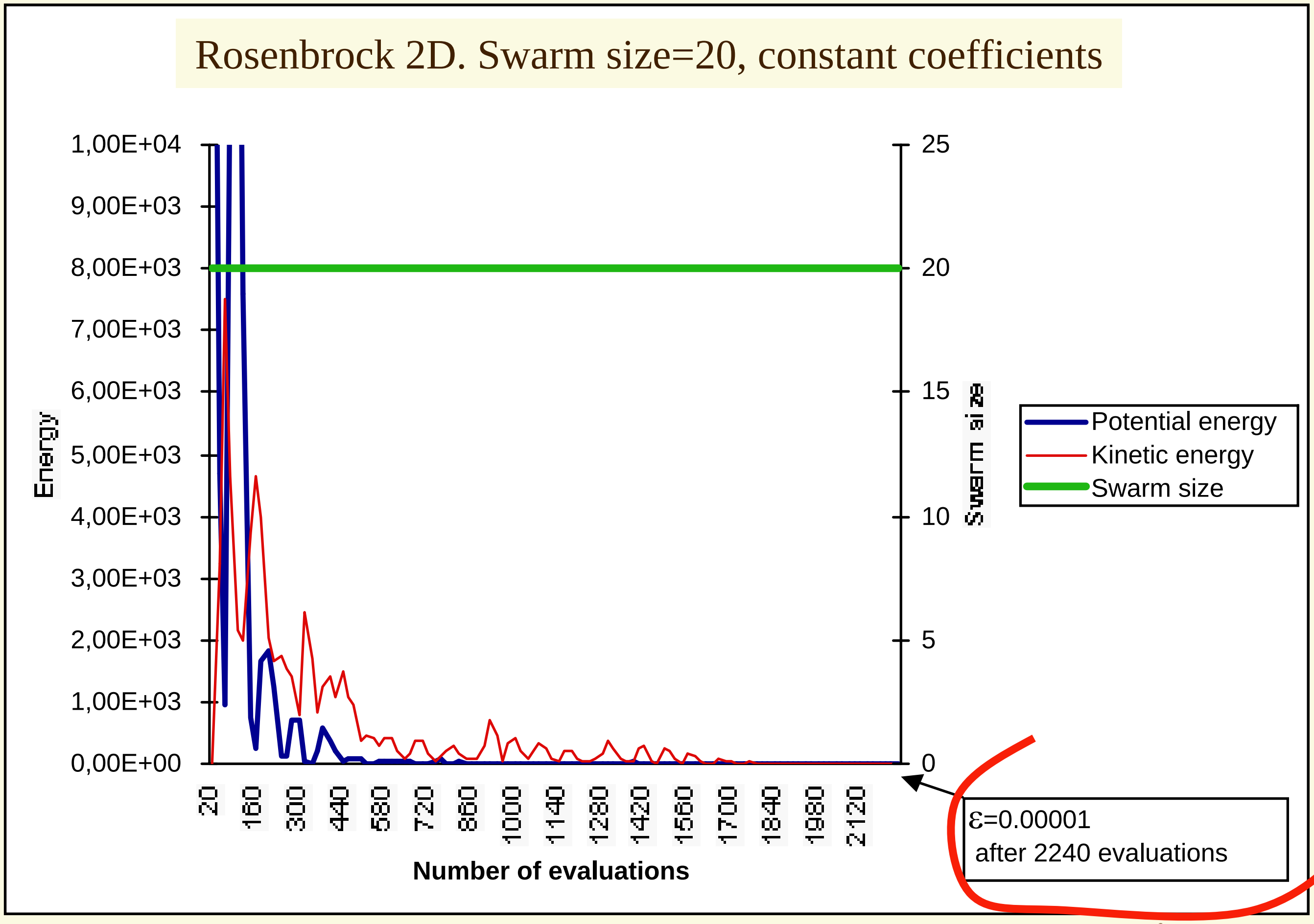
Adaptive proximity distributions (DPNP)



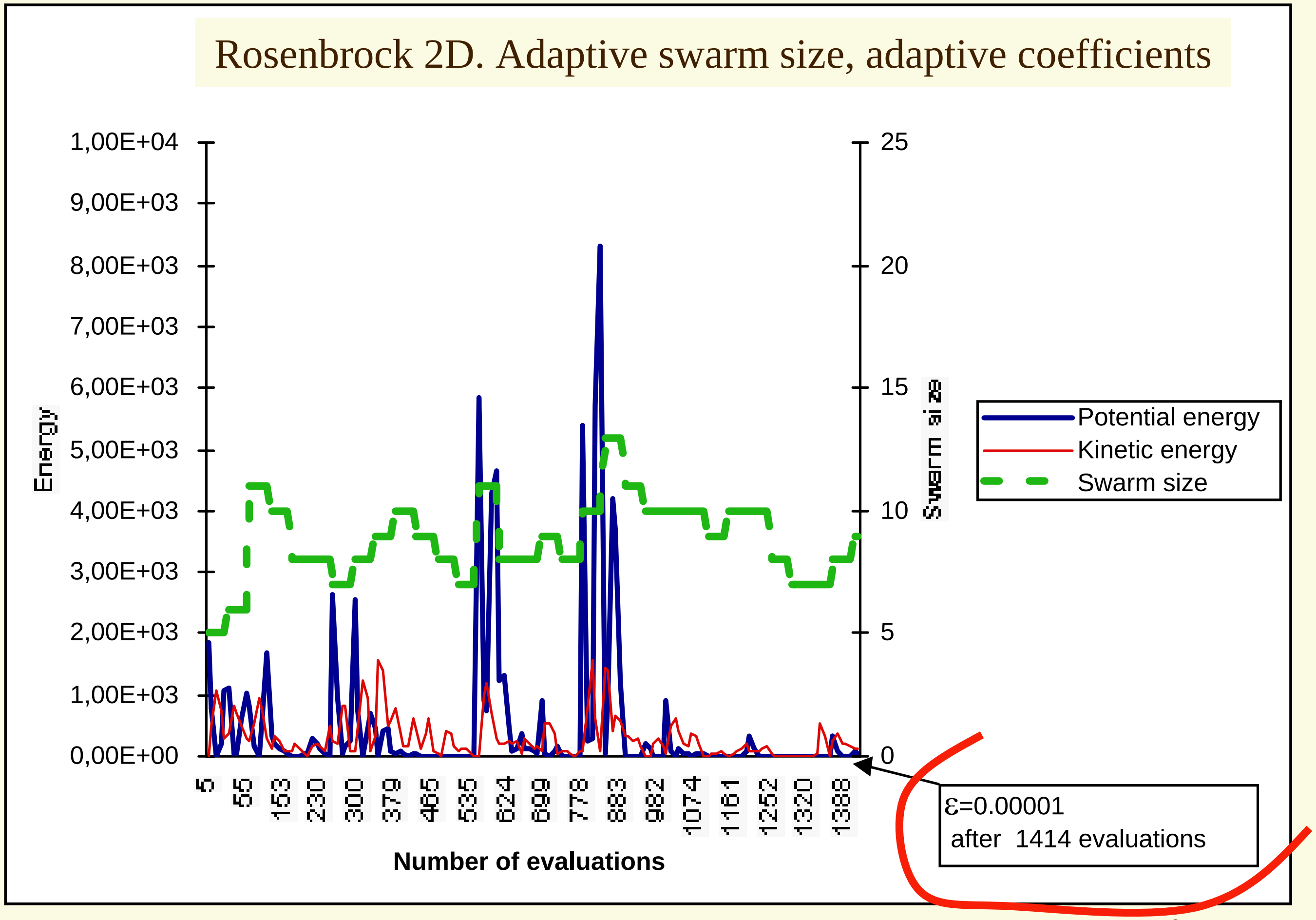
Adaptive information links



# Energies: classical process



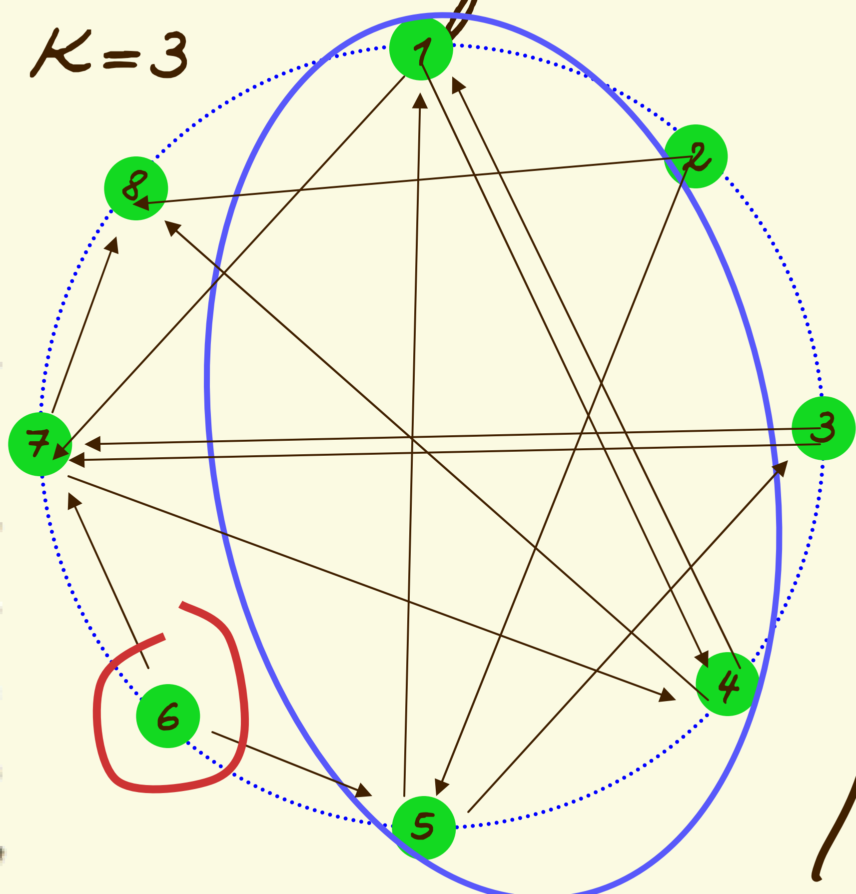
# Energies: adaptive process



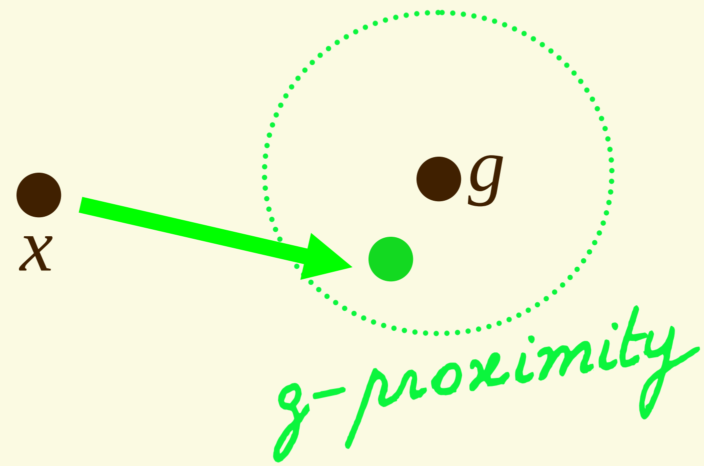
# The simplest PSO

## Random informers

K=3

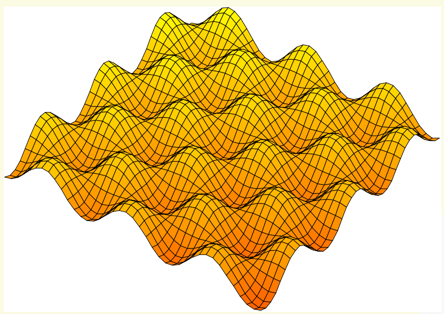


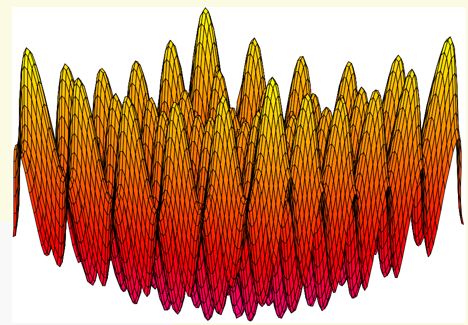

## Pivot method

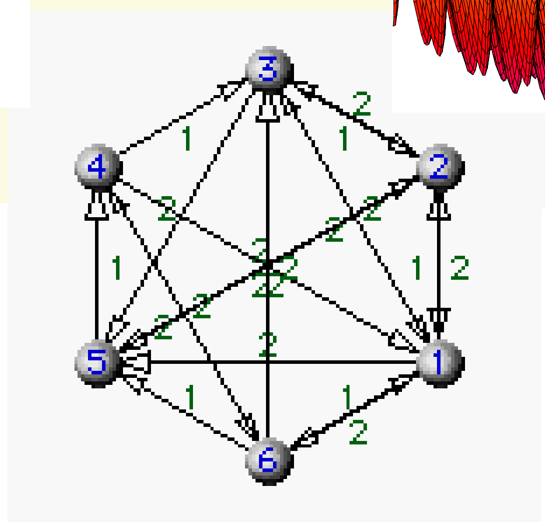


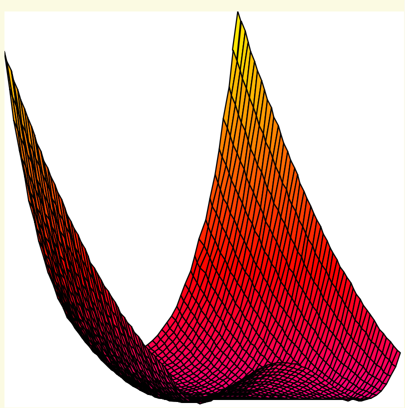
# End of Part 2

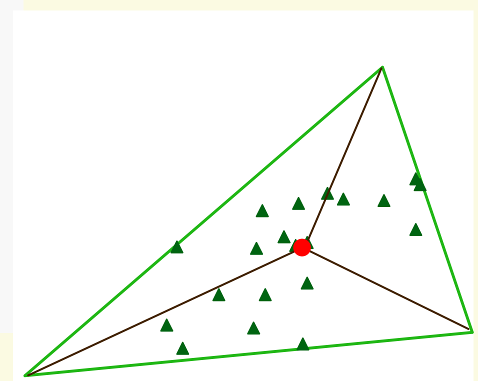
# Part 3: Story of Optimisation











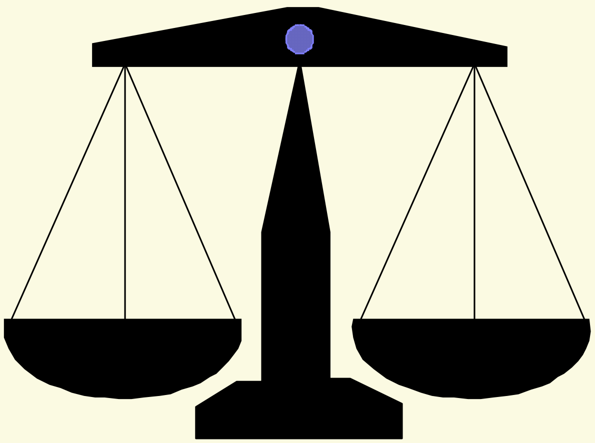
# Classical results

Optimum=0, dimension=30
Best result after 40 000 evaluations

| 30D function | PSO Type 1" | Evolutionary algo.(Angeline 98) |
|---|---|---|
| Griewank [±300] | 0.003944 | 0.4033 |
| Rastrigin [±5] | 82.95618 | 46.4689 |
| Rosenbrock [±10] | 50.193877 | 1610.359 |

# Some small problems

# Fifty-fifty



granularity=1

$$\begin{cases} x_i \in \{1 \ldots N\} \\ i \neq j \Rightarrow x_i \neq x_j \\ \sum_{1}^{D/2} x_i = \sum_{D/2+1}^{D} x_i \end{cases}$$

$N=100$, $D=20$. Search space: $[1,N]^D$

105 evaluations:

63+90+16+54+71+20+23+60+38+15

=

12+48+13+51+36+42+86+26+57+79 (=450)

# Knapsack

granularity=1

$$\begin{cases} x_i \in \{1 \ldots N\} \\ i \neq j \Rightarrow x_i \neq x_j \\ \displaystyle\sum_{i\in I, |I|=D, I\subset\{1,N\}} x_i = S \end{cases}$$

N=100, D=10, S=100,

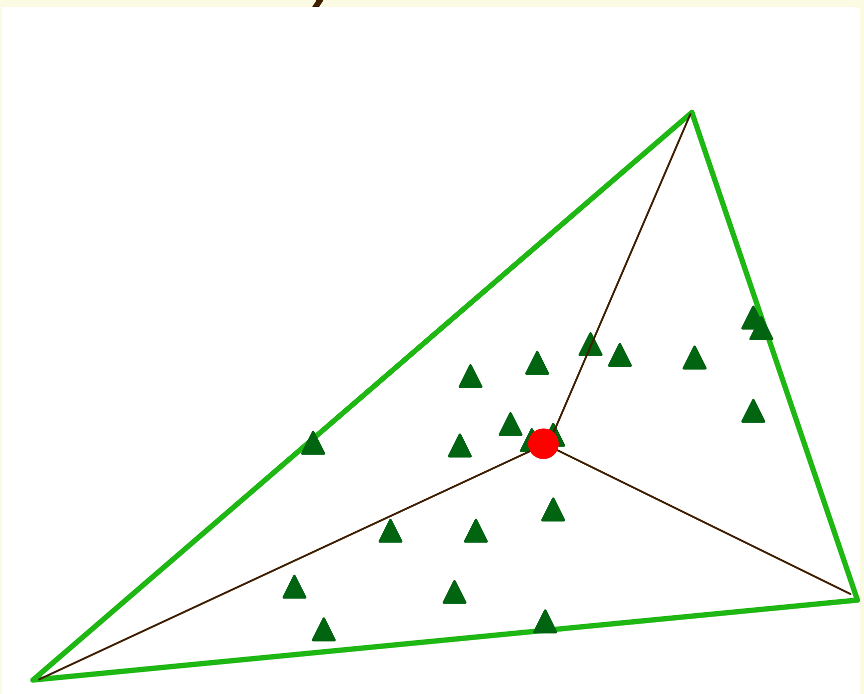
870 evaluations:

run 1 => (9, 14, 18, 1, 16, 5, 6, 2, 12, 17)

run 2 => (29, 3, 16, 4, 1, 2, 6, 8, 26, 5)

# Apple trees

- Best position



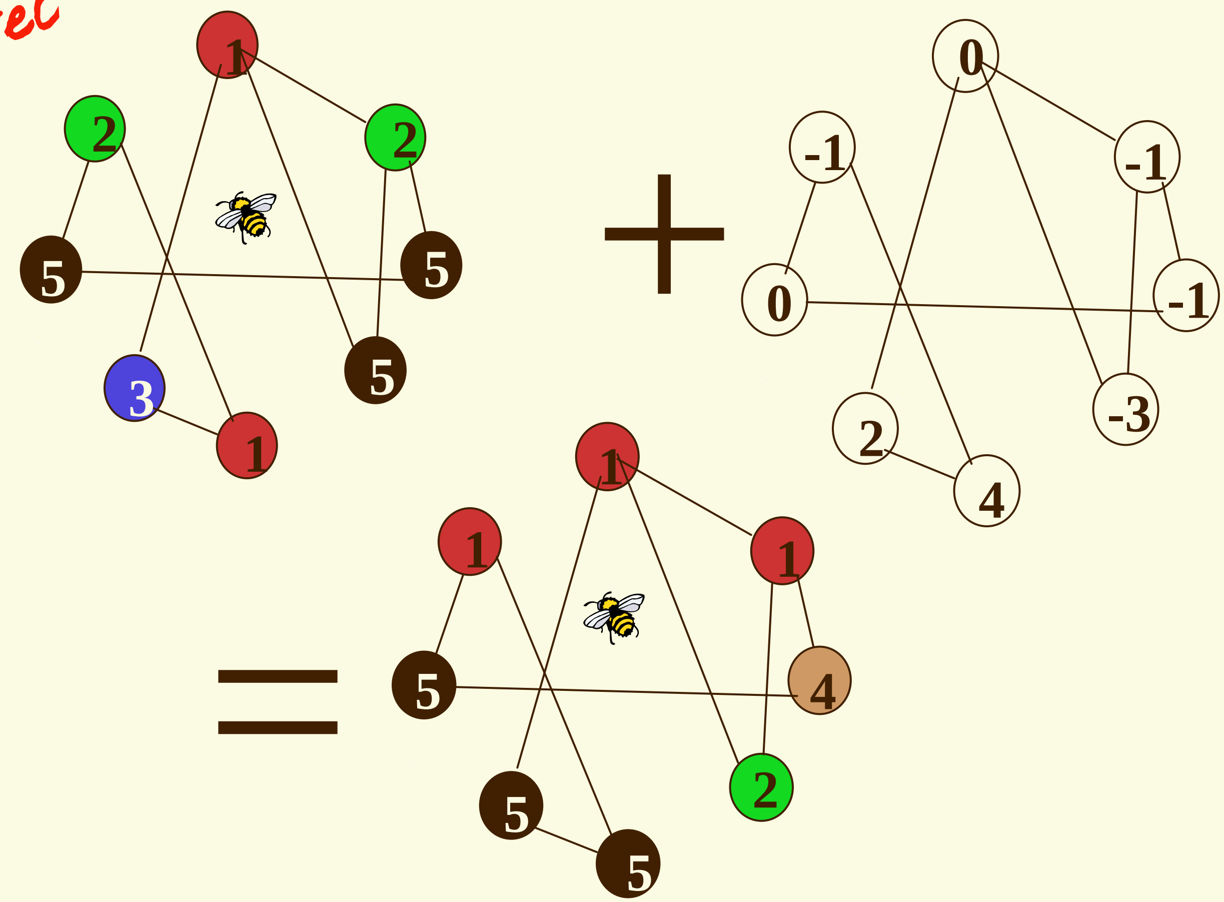
$$f=\left(n_1-n_2\right)^2+\left(n_2-n_3\right)^2$$

Swarm size=3

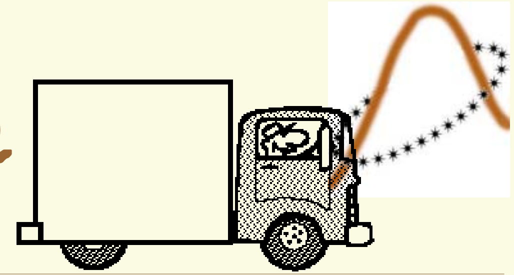
| Evaluation | n1 | n2 | n3 |
|---|---|---|---|
| 0 | 3 | 0 | 17 |
| 1 | 6 | 4 | 10 |
| 2 | 3 | 11 | 6 |
| 3 | 7 | 7 | 6 |

# Graph Coloring Problem

pos_plus_vel

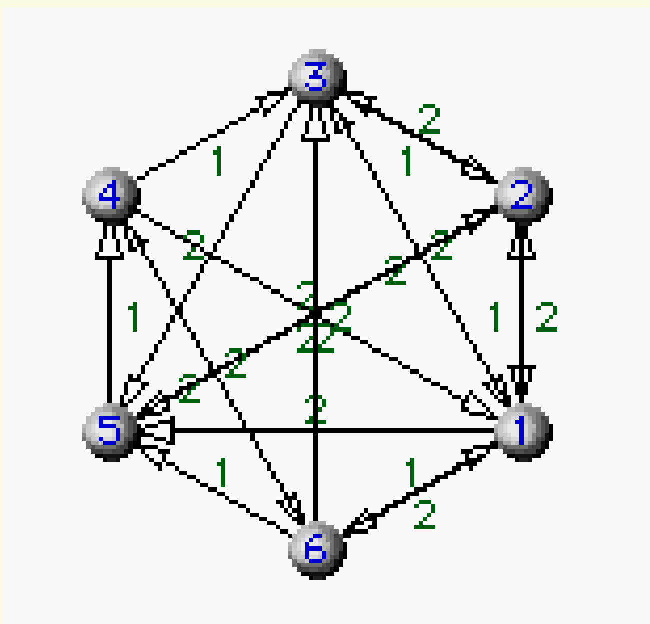


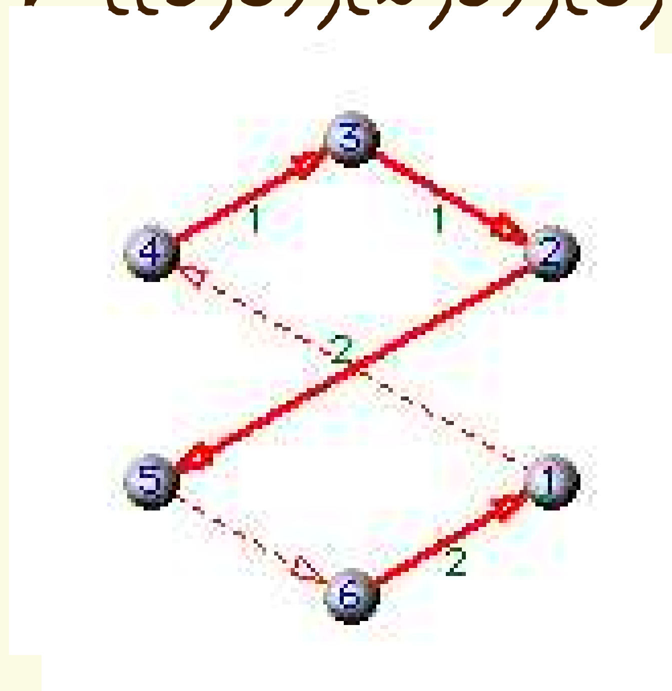
# The Tireless Traveller



Example of position: $X=(5,3,4,1,2,6)$

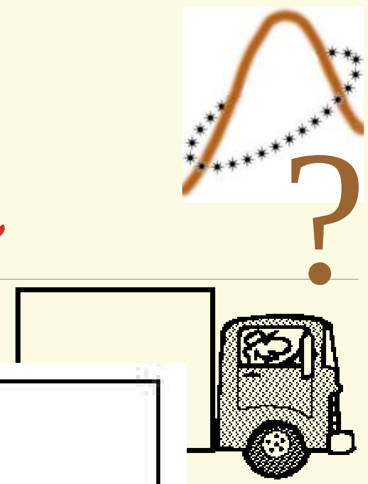
Example of velocity: $v=((5,3),(2,5),(3,1))$

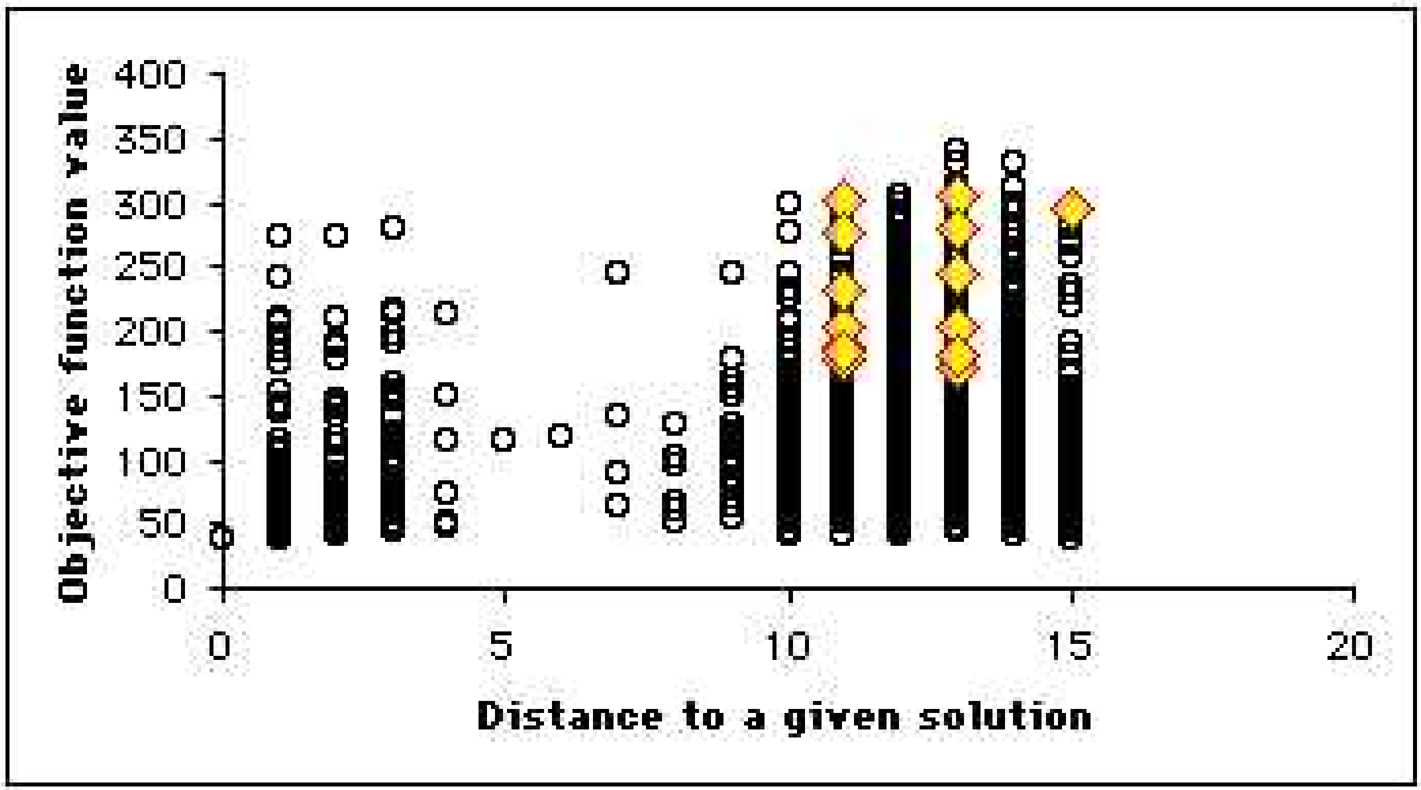




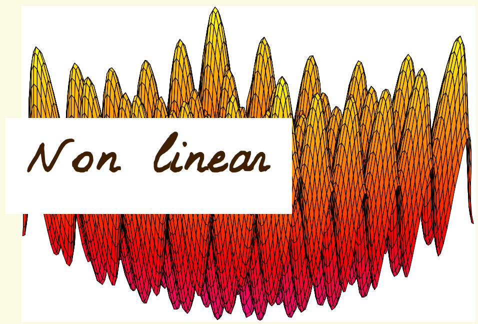
# BR17, the movie

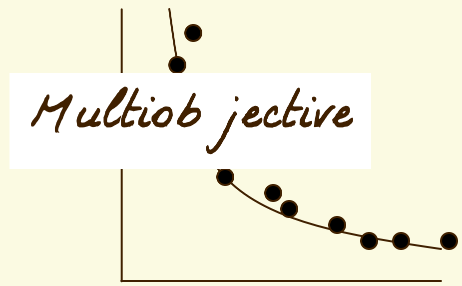
Structured search space

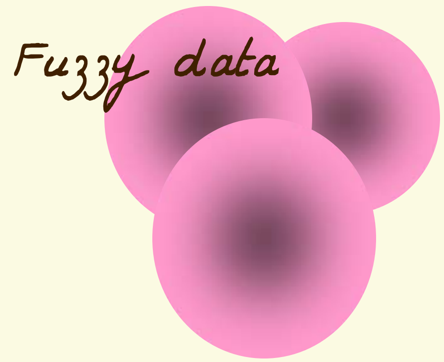


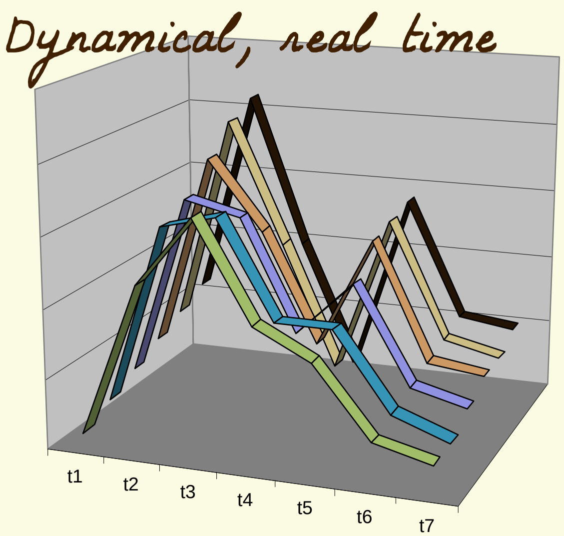


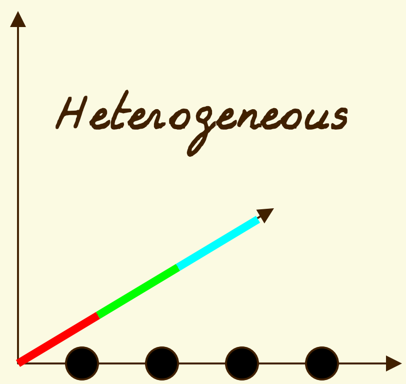
# Ecological niche



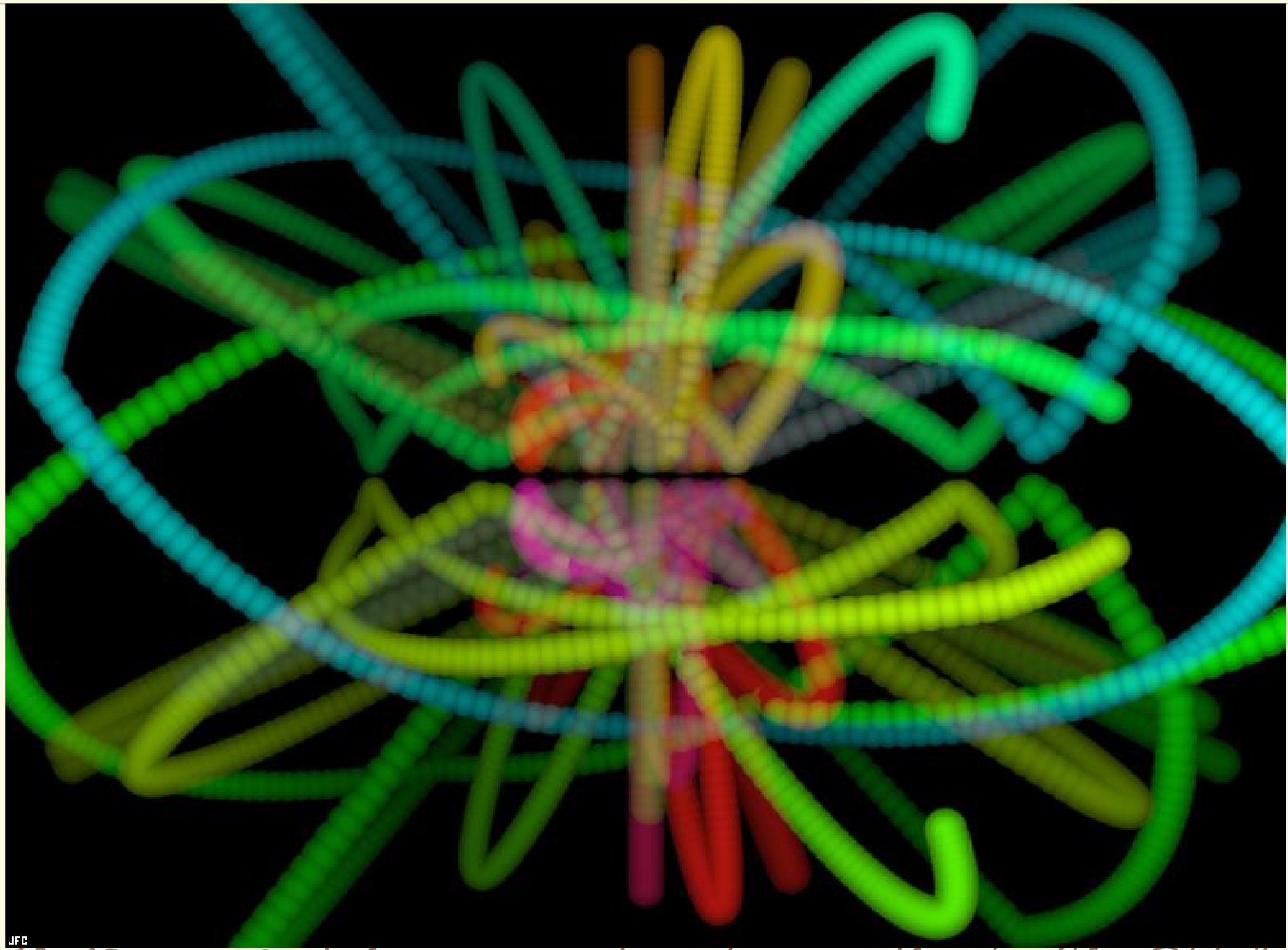



# End of Part 3

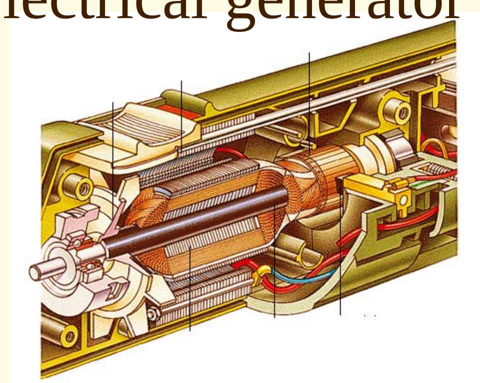


# Part 4: Real applications

Medical diagnosis

Industrial mixer

Electrical generator



Electrical vehicle

# Applications (1)

Salerno, J. Using the particle swarm optimization technique to train a recurrent neural model. IEEE International Conference on Tools with Artificial Intelligence, 1997, p. 45-49, 1997.

He Z., Wei C., Yang L., Gao X., Yao S., Eberhart R. C., Shi Y., "Extracting Rules from Fuzzy Neural Network by PSO", *IEEE IEC*, Anchorage, Alaska, USA, 1998.

Secrest B. R., Traveling Salesman Problem for Surveillance Mission using PSO, AFIT/GCE/ENG/01M-03, Air Force Institute of Technology, 2001.

Yoshida H., Kawata K., Fukuyama Y., "A PSO for Reactive Power and Voltage Control considering Voltage Security Assessment", *IEEE TPS*, vol. 15, 2001, p. 1232-1239.

Krohling, R. A., Knidel, H., and Shi, Y. Solving numerical equations of hydraulic problems using PSO. Proceedings of the IEEE CEC, Honolulu, Hawaii USA. 2002.

# Applications (2)


Kadrovach, B.A., and Lamont G., A particle swarm model for swarm-based networked sensor systems, ACM symposium on Applied computing, Madrid, Spain, p. 918-924, 2002

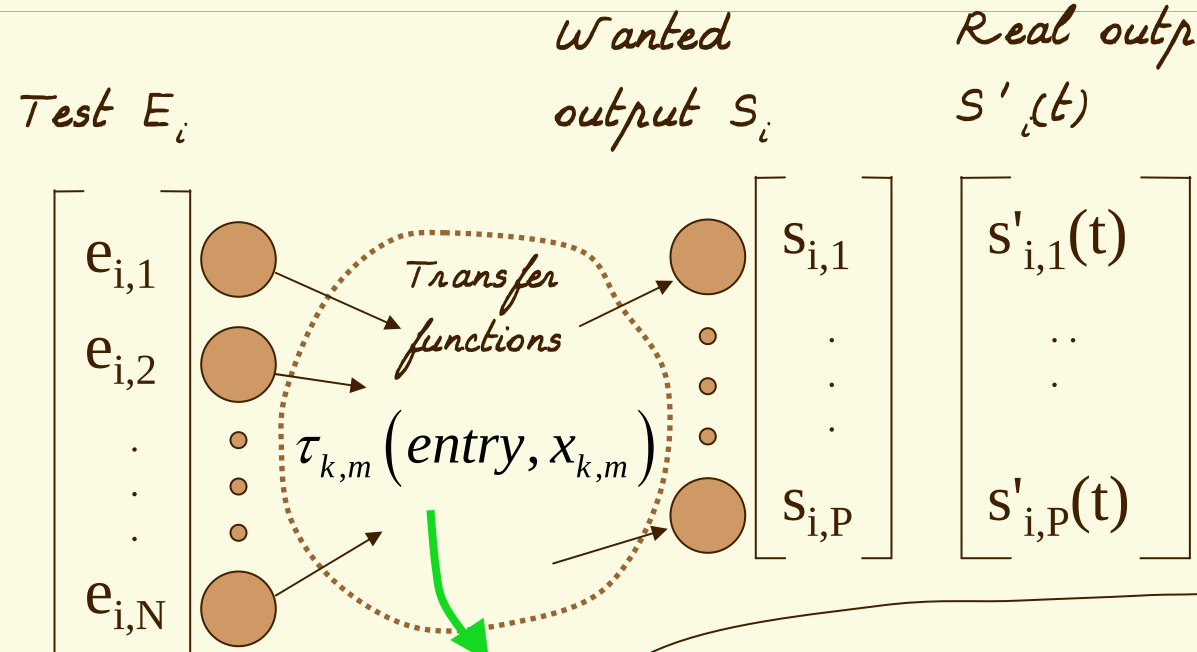
Omran, M., Salman, A., and Engelbrecht, A. P. Image classification using PSO. Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning 2002 (SEAL 2002), Singapore. p. 370-374, 2002.

Coello Coello, C. A., Luna, E. H., and Aguirre, A. H. Use of PSO to design combinational logic circuits. LNCS No. 2606, p. 398-409, 2003.

Onwubolu, G. C. and Clerc, M., "Optimal path for automated drilling operations by a new heuristic approach using particle swarm optimization," International Journal of Production Research, vol. 4, p. 473-491, 2004.

Onwubolu G.C., TRIBES application to the flowshop scheduling problem, New Optimization Techniques in Engineering. Heidelberg, Germany, Springer: p. 517-536, 2004
.

# Neuronal network




$$\frac{1}{1+e^{x_{k,m}entry}}$$

$$\begin{cases} E = \left(E_1 \ldots E_{nb\_tests}\right) \\ X(t) = \left(x_{k,m}(t)\right) \\ S = \left(S_1 \ldots S_{nb\_tests}\right) \\ S'(t) = \left(S'_1(t) \ldots S'_{nb\_tests}(t)\right) \end{cases}$$

Function to minimise

$$f(X) = \left\| S - S' \right\|$$

# To know more

THE site:

Particle Swarm Central, http://www.particleswarm.info

Kennedy, J., R. Eberhart, et al. (2001). Swarm Intelligence, Morgan Kaufmann Academic Press.

Self advert

2005 IEEE TEC award

Clerc M., Kennedy J., "The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex space", *IEEE Transaction on Evolutionary Computation*, 2002,vol. 6, p. 58-73

# More self ad.

My PSO site: http://clerc.maurice.free.fr/pso/index.htm

## If you read French

Clerc M., "L'optimisation par essaim particulaire. Principes et pratique", *Hermès, Techniques et Science de l'Informatique*, 2002. Article de 25 p.

Clerc M., L 'optimisation par essaims particulaires. http://www.editions-hermes.fr/fr/. Parution février 2005

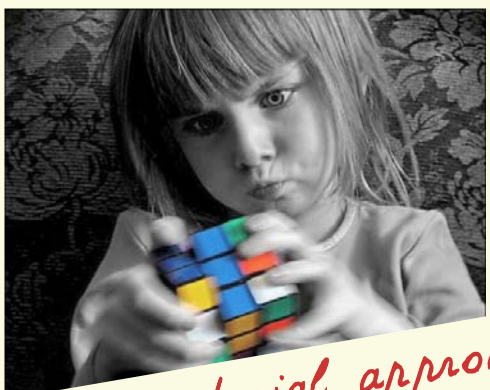
# PSO in the world

eXtended Particle Swarms (XPS) project

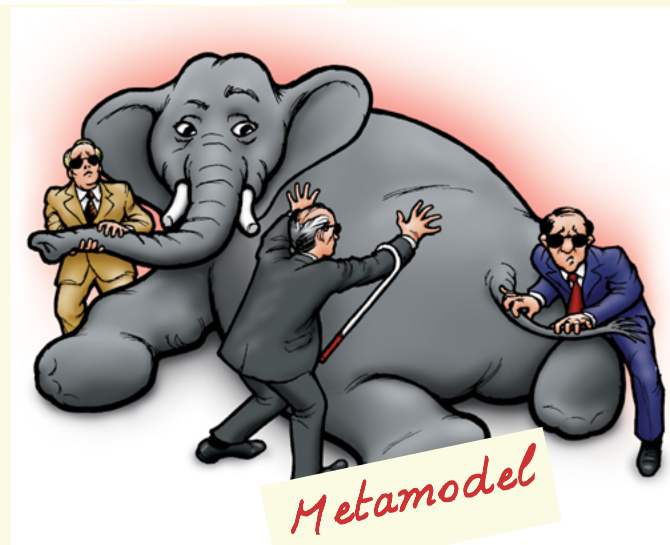
# Some open questions

New mathematical ideas to model particle interactions

Adaptive weighted relationships

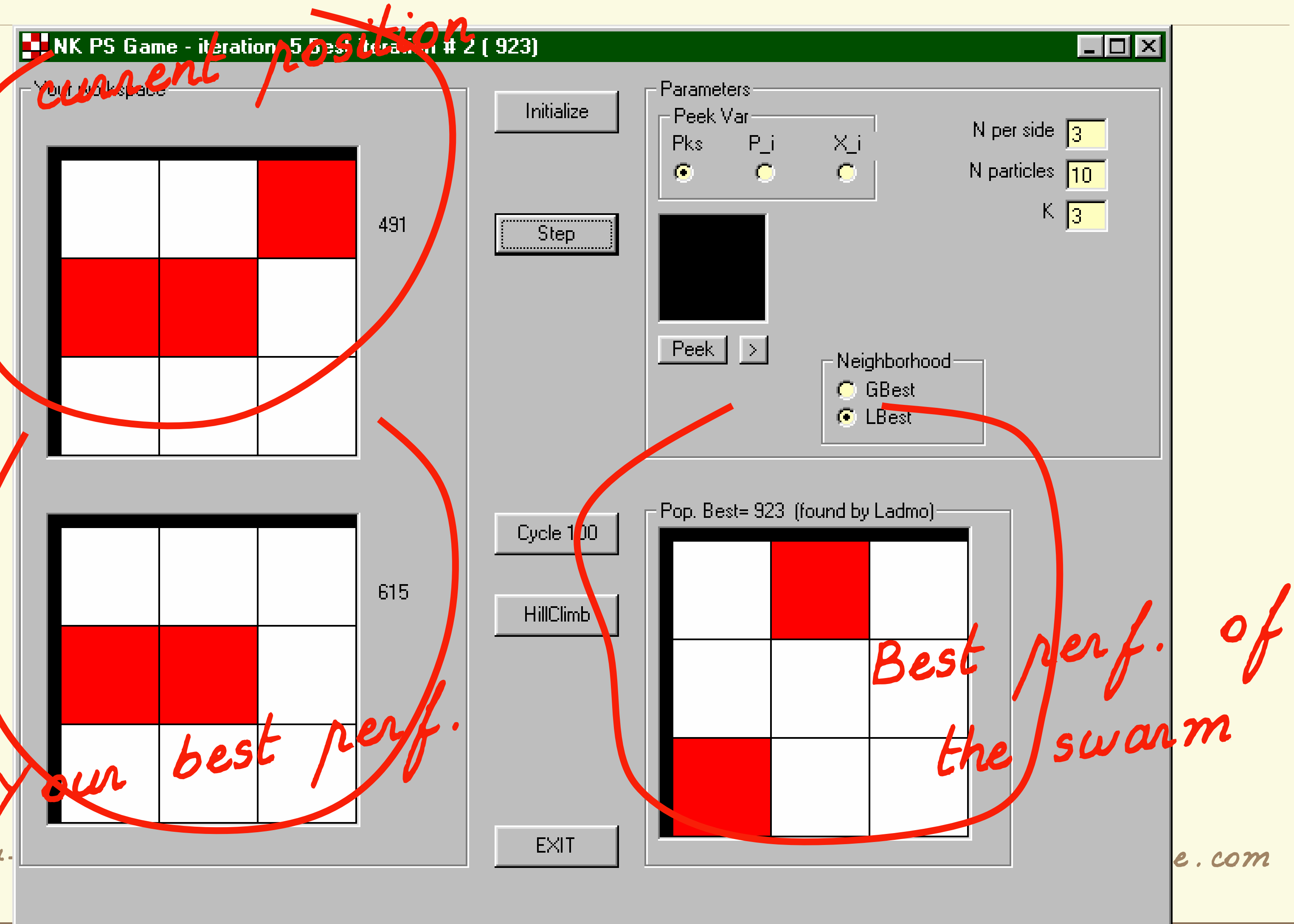


Better combinatorial approaches

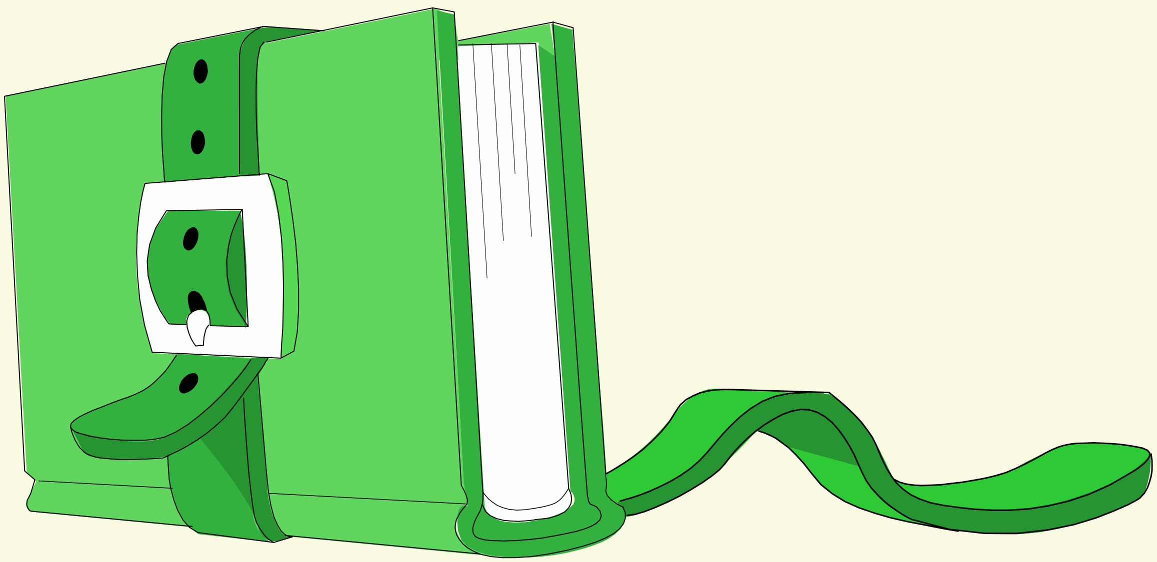


Metamodel

# Beat the swarm!



# APPENDIX



# Canonical form

$$\begin{cases} v(t+1) = v(t) + \varphi\left(q - x(t)\right) \\ x(t+1) = x(t) + v(t+1) \end{cases}$$

M

$$y(t) = q - x(t) \qquad \begin{bmatrix} v(t+1) \\ y(t+1) \end{bmatrix} = \begin{bmatrix} 1 & \varphi \\ -1 & 1-\varphi \end{bmatrix} \begin{bmatrix} v(t) \\ y(t) \end{bmatrix}$$

Eigen values $e_1$ and $e_2$

$$\begin{cases} v(t+1) = \alpha v(t) + \beta \varphi y(t) \\ y(t+1) = -\gamma v(t) + (\delta - \eta\varphi) y(t) \end{cases}$$

# Constriction

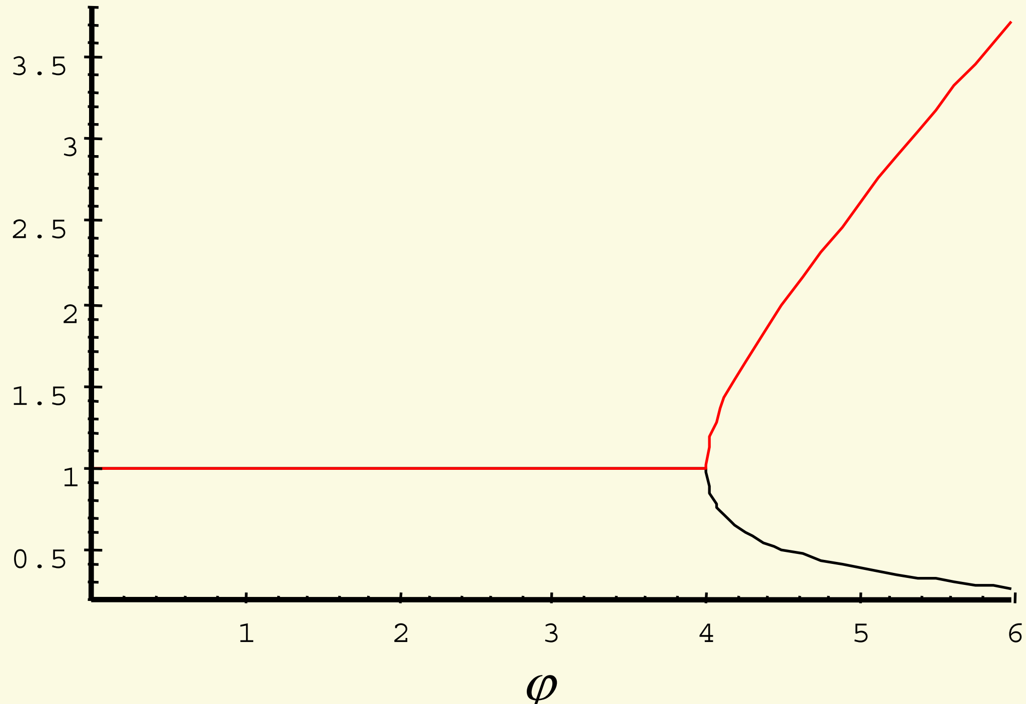
## Constriction coefficients

$$\begin{cases} \chi_1 = \dfrac{\alpha+\delta-\eta\varphi+\sqrt{(\eta\varphi)^2+2\varphi(\alpha\eta-\delta\eta-2\beta\gamma)+(\alpha-\delta)^2}}{2-\varphi+\sqrt{\varphi^2-4\varphi}} \\[2ex] \chi_2 = \dfrac{\alpha+\delta-\eta\varphi-\sqrt{(\eta\varphi)^2+2\varphi(\alpha\eta-\delta\eta-2\beta\gamma)+(\alpha-\delta)^2}}{2-\varphi+\sqrt{\varphi^2-4\varphi}} \end{cases}$$

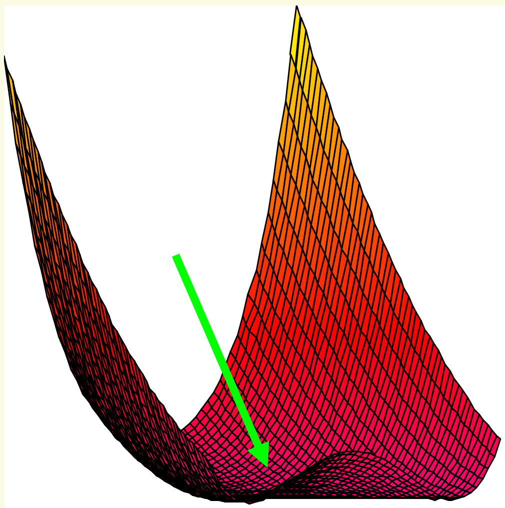
# Convergence criterion

$$\begin{cases} |\chi_1 e_1| < 1 \\ |\chi_2 e_2| < 1 \end{cases} \Leftarrow \begin{cases} |\chi_1| < 1 \\ |\chi_2 e_2| < 1 \end{cases}$$

$$\boxed{\kappa = |\chi_2 e_2|}$$

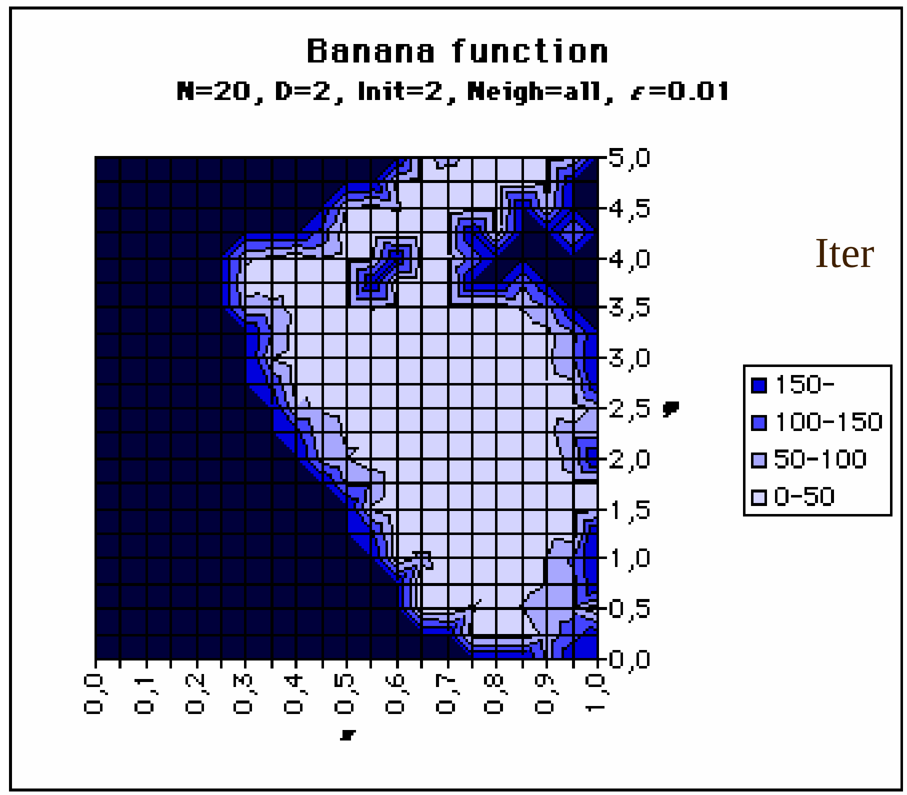


# Robustness



*Performance map :*

*NeededIterations(κ,φ)*



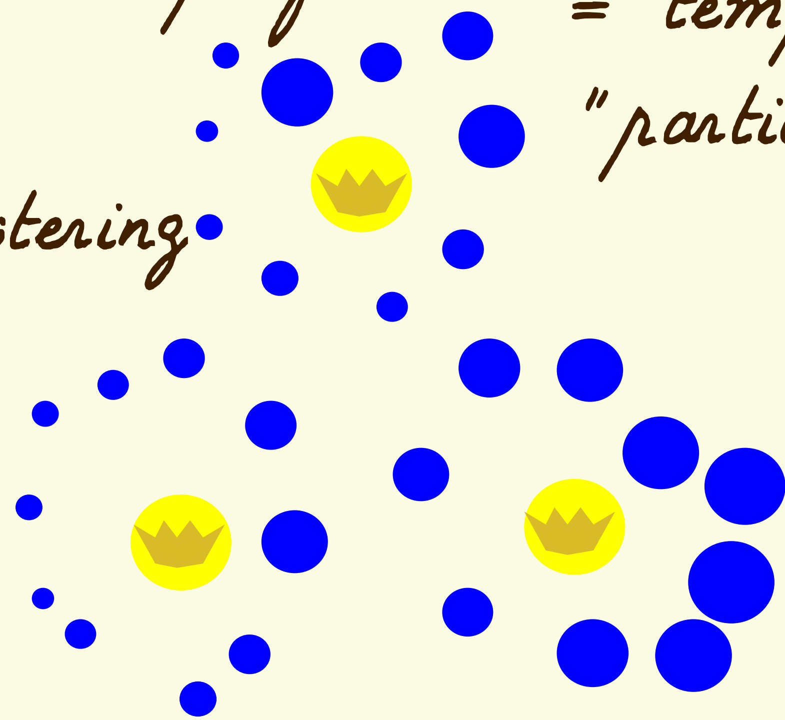

$$f(x_1, x_2) =$$
$$100\left(x_2 - x_1^2\right) + \left(1 - x_1\right)^2$$

# Clusters and queens

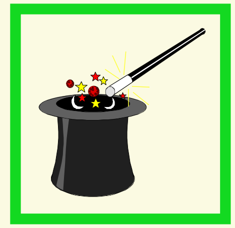
Each particle is weighted by its perf.

Centroids = queens = temporary new "particles"

Dynamic clustering



# Magic Square (1)

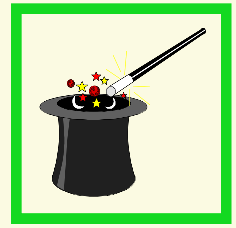


$$\begin{bmatrix} m_{1,1} & \cdots & m_{1,\sqrt{D}} \\ \vdots & m_{i,j} & \vdots \\ m_{\sqrt{D},1} & \cdots & m_{\sqrt{D},\sqrt{D}} \end{bmatrix}$$

$$\begin{cases} m_{i,j} = x_{j+(i-1)\sqrt{D}} \\ m_{i,j} \in \{1 \cdots N\} \\ m_{i,j} \neq m_{k,l} \end{cases}$$

$$\sum_{i=1}^{\sqrt{D}-1}\left(\sum_{j=1}^{\sqrt{D}}\left(m_{i,j}-m_{i+1,j}\right)\right)^2 + \sum_{j=1}^{\sqrt{D}-1}\left(\sum_{i=1}^{\sqrt{D}}\left(m_{i,j}-m_{i,j+1}\right)\right)^2 = 0$$

# Magic Square (2)



| | | |
|---|---|---|
| 55 | 30 | 68 |
| 42 | 49 | 62 |
| 56 | 74 | 23 |

| | | |
|---|---|---|
| 30 | 61 | 53 |
| 89 | 32 | 23 |
| 25 | 51 | 68 |

| | | |
|---|---|---|
| 80 | 3 | 30 |
| 22 | 72 | 19 |
| 11 | 38 | 64 |

| | | |
|---|---|---|
| 50 | 43 | 67 |
| 58 | 55 | 47 |
| 52 | 62 | 4 |

| | | |
|---|---|---|
| 43 | 51 | 78 |
| 75 | 33 | 64 |
| 54 | 88 | 30 |

D=3x3, N=100

10 runs

13430 evaluations

→ 10 solutions

| | | |
|---|---|---|
| 65 | 28 | 64 |
| 63 | 55 | 39 |
| 29 | 74 | 54 |

| | | |
|---|---|---|
| 27 | 96 | 39 |
| 73 | 40 | 49 |
| 62 | 26 | 74 |

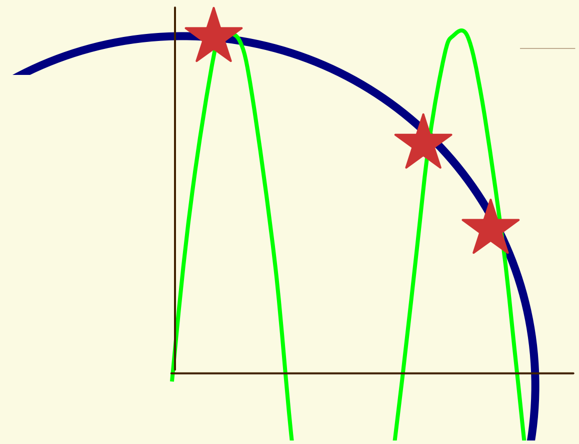
| | | |
|---|---|---|
| 22 | 70 | 58 |
| 40 | 75 | 35 |
| 88 | 5 | 57 |

| | | |
|---|---|---|
| 18 | 25 | 59 |
| 32 | 53 | 17 |
| 52 | 24 | 26 |

| | | |
|---|---|---|
| 50 | 65 | 68 |
| 69 | 42 | 72 |
| 64 | 76 | 43 |

# Non linear system

$$\begin{cases} x_1^2 + x_2^2 - 1 = 0 \\ \sin(10x_1) - x_2 = 0 \end{cases}$$



Search space $[0,1]^2$

1 run
143 evaluations → 1 solution

10 runs
1430 evaluations → 3 solutions

# Model fitting (ARMA + AIC)

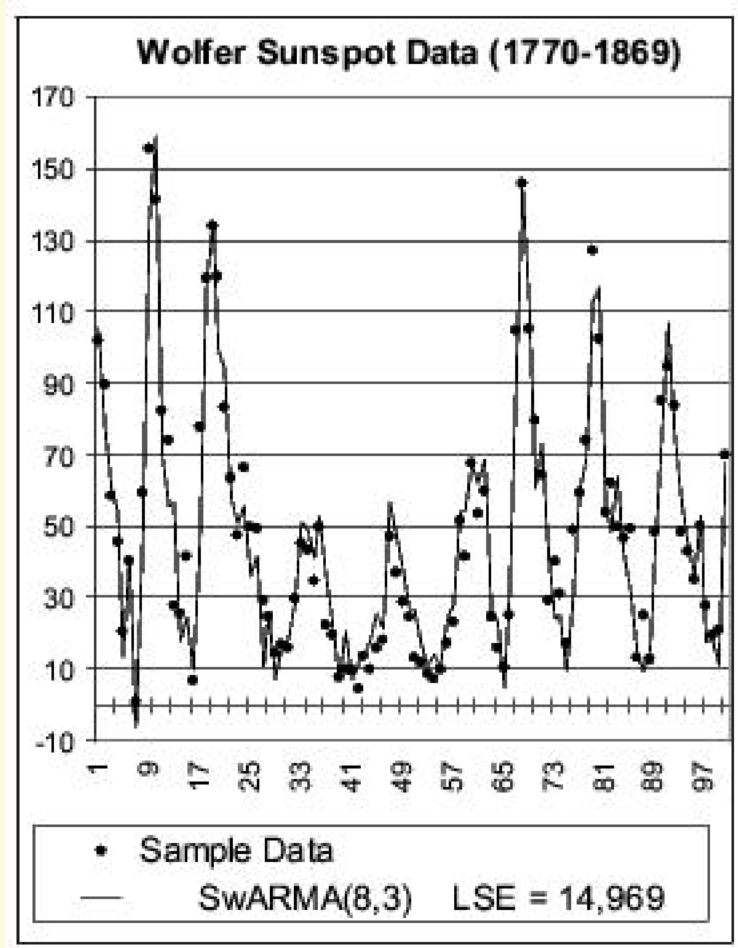
Autoregressive Moving Average
+ Akaike's Information Criterion

$$\sum_{i=0}^{n} \phi_i y_{t-i} = \sum_{j=0}^{m} \theta_j a_{t-j}$$

$$\sigma^2 = \frac{\sum_{i=1}^{N} \left( y_{t_{data}} - y_{t_{ARMA}} \right)^2}{N}$$

heterogenuous

$$f = n \log\left(\sigma^2\right) + 2(n+m)$$



# A binary PSO C code

*Information links are modified at random if there has been no improvement*

```
// Pivot method -----------------------------------------
// Works pretty well on some problems .. and pretty bad on some others
 P[s]=P_m[g]; // Initialise the new position of particle s
              // at the position of the best known around

  dist=log(D);  // We suppose here D>=2

  r=alea(1,dist); // Radius for DPNP

 for (k=0;k<r;k++)// Switch at random some bits
  {
    d=alea_integer(0,D-1);
    P[s][d]=1- P[s][d][d]; // Around g
  }
```

# End of ANNEXE