Méta-heuristiques pour l'optimisation

Differential Evolution (DE)
Particle Swarm Optimization (PSO)

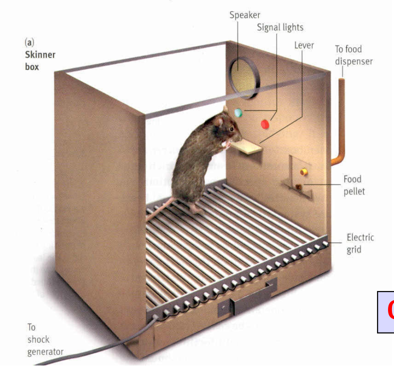
Alain Dutech

Equipe MAIA - LORIA - INRIA
Nancy, France
Web : `http://maia.loria.fr`
Mail : `Alain.Dutech@loria.fr`

3 mai 2010

Context : Neuro-Sciences

Question : influence/role of pre-frontal-cortex in reinforcement-like learning in rats ?
(Context of "Operant Conditionning", work with **Alain Marchand**, Bordeaux).



Real Data
PFC-lesion/normal
↓
SARSA .... *Policy*
State : w/o memory
↓
Fitting .... *Likelyhood*
Optimize Softmax-Policy Likelyhood

Global Optimization

- Search space $\mathcal{S}$
- Objective function to minimize

$$f : \mathcal{S} \longrightarrow \mathbb{R}$$

- f is **unknown**

- $\vec{\mathbf{x}} = (\alpha, \gamma, \beta)^T$
- f : policy likelyhood (vector ? traj ?)
- $\rightsquigarrow$ sampled.

Global Optimization

- non-convex objective function.
- unconstrained state space.
- Meta-heuristics : Simulated Annealing, Tabu Search, Ant Colony Optimization, Evolutionary Algorithm (**Differential Evolution**), **Particle Swarm Optimization**, Variable Neighborhood Search, *etc*.
- Ceci n'est **pas exhaustif** !

Differential Evolution (DE) [Storn and Price, 1995]

D-dimensionnal parameter vector

$$\vec{\mathbf{x}} = (x^0, \cdots, x^D)^T \tag{1}$$

Population of N parameter vectors at generation g

$$X[g] = \{\vec{\mathbf{x}}_i[g]\},\ i = 0, \cdots, N \tag{2}$$

Evolution of one parameter vector

1. "mutation" :

$$\vec{\mathbf{m}}_i = f(X[g]) \tag{3}$$

2. "cross-over" :

$$\vec{\mathbf{x}}_i[g+1] = \begin{cases} \vec{\mathbf{m}}_i^j & \text{for some } j \\ \vec{\mathbf{x}}_i^j[g] & \text{otherwise} \end{cases} \tag{4}$$

DE "canonical" : uninformed evolution

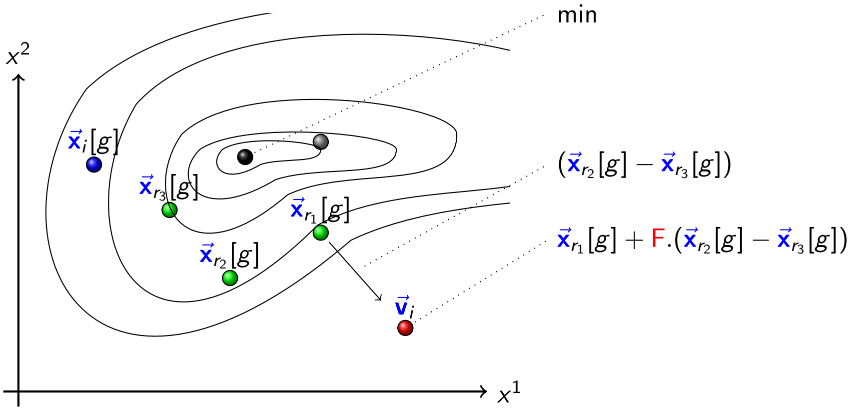
Mutation : r_1, r_2 and r_3 randomly different in $[0, \cdots, N]$

$$\vec{\mathbf{v}}_i = \vec{\mathbf{x}}_{r_1}[g] + \mathsf{F}.(\vec{\mathbf{x}}_{r_2}[g] - \vec{\mathbf{x}}_{r_3}[g]) \tag{5}$$

Cross-Over : n and L randomly chosen

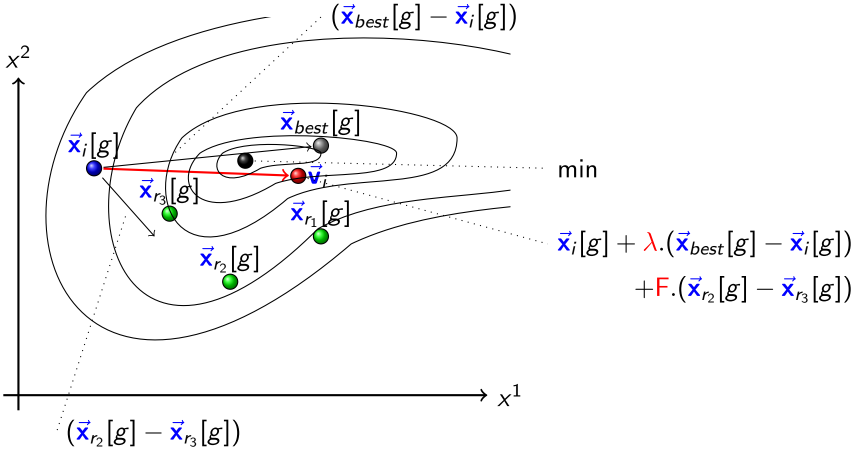
$$\vec{\mathbf{x}}_i[g+1] = \begin{cases} \vec{\mathbf{v}}_i^j & \text{for } j = \langle n \rangle_D, \cdots, \langle n + L - 1 \rangle_D \\ \vec{\mathbf{x}}_i^j[g] & \text{otherwise} \end{cases} \tag{6}$$

DE "canonical" : uninformed mutation



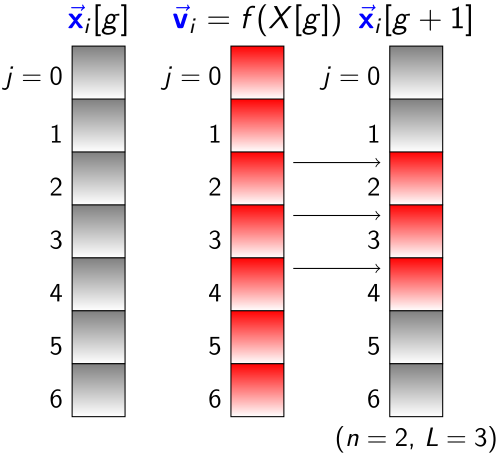
$$\vec{v}_i = \vec{x}_{r_1}[g] + F.(\vec{x}_{r_2}[g] - \vec{x}_{r_3}[g])$$

DE “canonical” : informed mutation



$$\vec{\mathbf{v}}_i = \vec{\mathbf{x}}_i[g] + \lambda.(\vec{\mathbf{x}}_{best}[g] - \vec{\mathbf{x}}_i[g]) + F.(\vec{\mathbf{x}}_{r_2}[g] - \vec{\mathbf{x}}_{r_3}[g])$$

DE "canonical" : cross-over



Particle Swarm Optimisation (PSO) [Kennedy and Eberhart, 1995]

D-dimensionnal parameter vector and **velocity**

$$\vec{\mathbf{x}} = (x^0, \cdots, x^D)^T \quad (7)$$

$$\vec{\mathbf{v}} = (v^0, \cdots, v^D)^T \quad (8)$$

$$(9)$$

Population of N parameter vectors and velocity at generation g

$$X[g] = \{(\vec{\mathbf{x}}_i[g], \vec{\mathbf{v}}_i[g])\},\ i = 0, \cdots, N \quad (10)$$

Initialisation

Evolution of one parameter vector

1. "velocity" :

$$\vec{\mathbf{v}}_i[g+1] = f(X[g]) \quad (11)$$

2. "position" :

$$\vec{\mathbf{x}}_i[g+1] = \vec{\mathbf{x}}_i[g] + \vec{\mathbf{v}}_i[g+1] \quad (12)$$

PSO "canonical"

Velocity :

- φ_1 and φ_2 are accelaration constants
- R_1 and R_2 are uniformly chosen in $[0, 1]$
- $\vec{\mathbf{x}}_{i\text{best}}$ best **past** position of i
- $\vec{\mathbf{x}}_{i,\text{nbest}}$ best position in **neighbors** of i (past ??)

$$\vec{\mathbf{v}}_i[g+1] = \vec{\mathbf{v}}_i[g] + \varphi_1.R_1.(\vec{\mathbf{x}}_{i\text{best}}[g] - \vec{\mathbf{x}}_i[g]) + \varphi_2.R_2.(\vec{\mathbf{x}}_{i,\text{nbest}}[g] - \vec{\mathbf{x}}_i[g]) \tag{13}$$

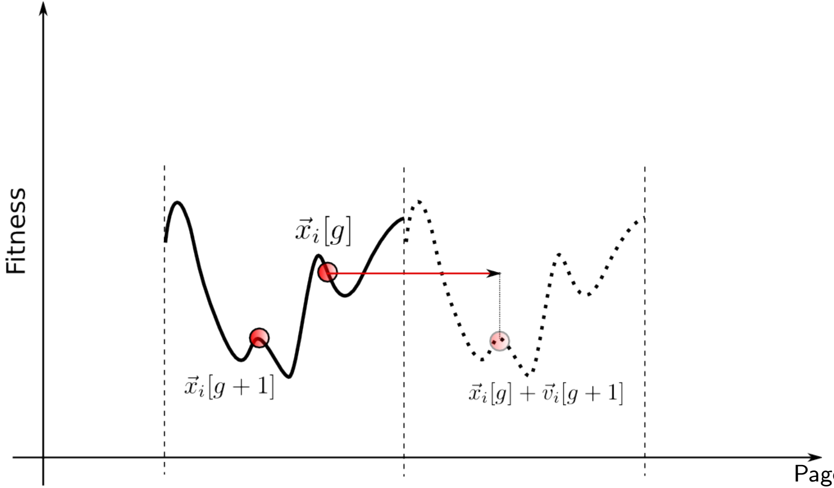
Position :

$$\vec{\mathbf{x}}_i[g+1] = \vec{\mathbf{x}}_i[g] + \vec{\mathbf{v}}_i[g+1] \tag{14}$$

Constraints

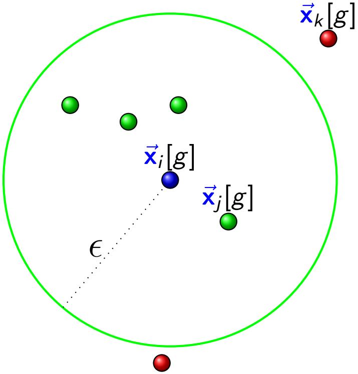
Enforce constraints on particles

- Projection into constrained space
- Periodic Search Space [Zhang et al., 2004]



Neighbors

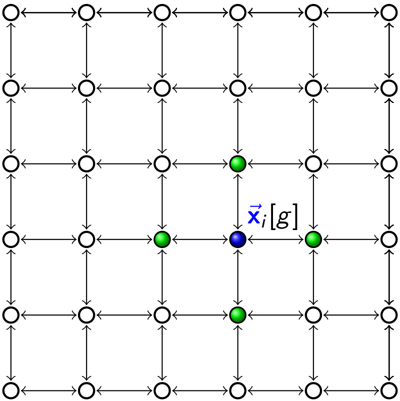
What is $\vec{\mathbf{x}}_{i,\text{nbest}}[g]$?



Sphere, radius ϵ

Neighbors

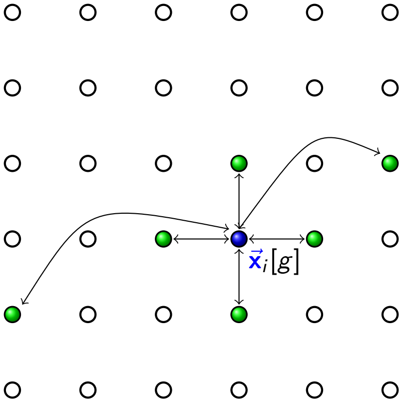
What is $\vec{\mathbf{x}}_{i,\text{nbest}}[g]$?



Small world

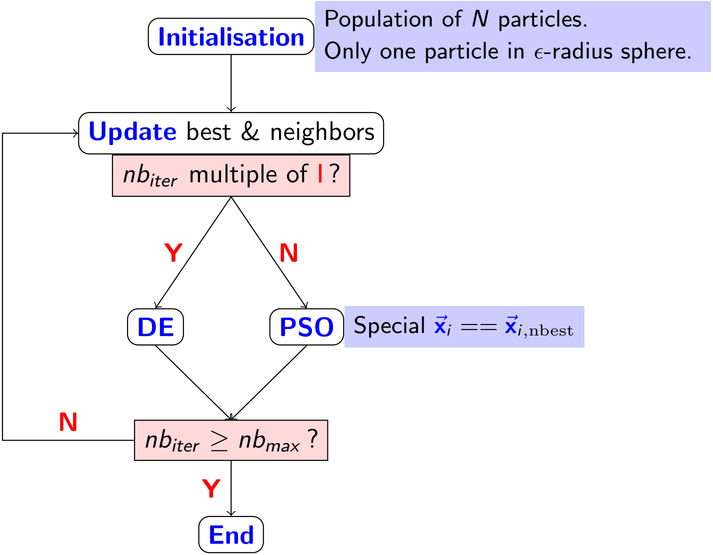
Neighbors

What is $\vec{\mathbf{x}}_{i,\mathrm{nbest}}[g]$?



Small world with random added links [Saxena and Vora, 2008]

DE+PSO Hybrid algorithm [Zhang et al., 2009]



DE in hybrid algorithm

Mutation : j randomly different in $[0, \cdots, N]$

- $\vec{\mathbf{x}}_{i\mathrm{best}}$ best **past** position of i
- $\vec{\mathbf{x}}_{i,\mathrm{nbest}}$ best position in **neighbors** of i (past ? ?)

$$\vec{\mathbf{v}}_i = \vec{\mathbf{x}}_{i,\mathrm{nbest}}[g] + \mathsf{F}.(\vec{\mathbf{x}}_{i\mathrm{best}}[g] - \vec{\mathbf{x}}_{j\mathrm{best}}[g]) \quad (15)$$

Cross-Over : D_i randomly chosen$[0, \cdots, N]$, C_R in $[0, 1]$

$$\vec{\mathbf{x}}_i[g+1] = \begin{cases} \vec{\mathbf{v}}_i^j & \text{if } j = D_i \quad \text{or} \quad \text{with proba } C_r \\ \vec{\mathbf{x}}_i^j[g] & \text{otherwise} \end{cases} \quad (16)$$

PSO in hybrid algorithm

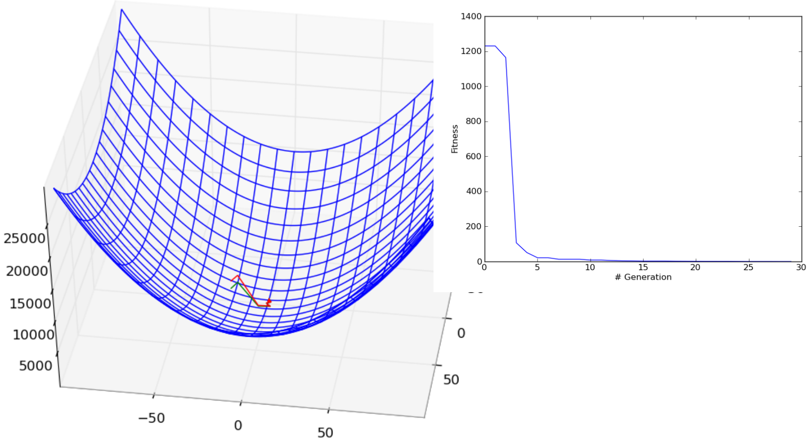
Canonical version of PSO (small world + 2 rnd links).
Prevent beeing stuck with null-velocity when ($\vec{\mathbf{x}}_i == \vec{\mathbf{x}}_{i,\mathrm{nbest}}$)

$$\vec{\mathbf{x}}_i[g+1] = \vec{\mathbf{x}}_{i,\mathrm{nbest}}[g] + \delta.\vec{\mathrm{rand}} \quad (17)$$

$$\vec{\mathbf{v}}_i[g+1] = \vec{\mathbf{x}}_i[g+1] - \vec{\mathbf{x}}_i[g] \quad (18)$$

"B2" problem

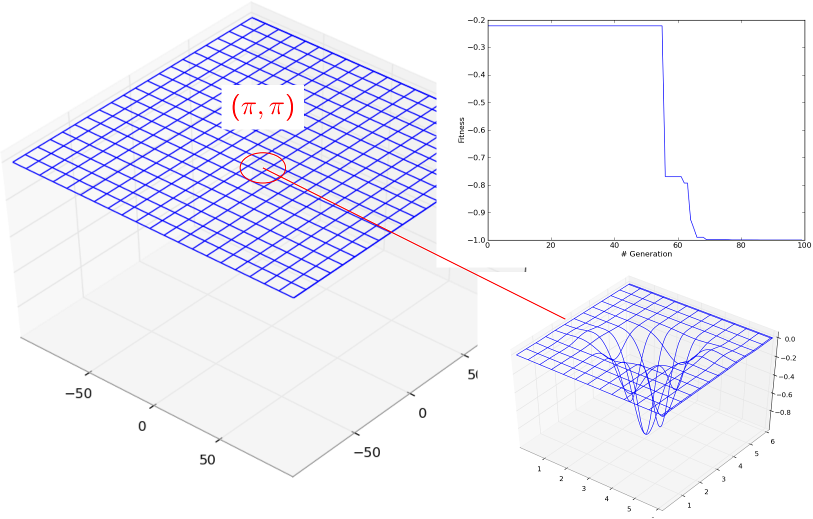
One global minimum of 0 at $(0, 0)$



$$B2(x_1, x_2) = x_1^2 + 2x_2^2 - 0.3\cos(3\pi x_1) - 0.4\cos(4\pi x_2) + 0.7$$
$$-100 < x_j < 100, j = 1, 2$$

“ES” problem

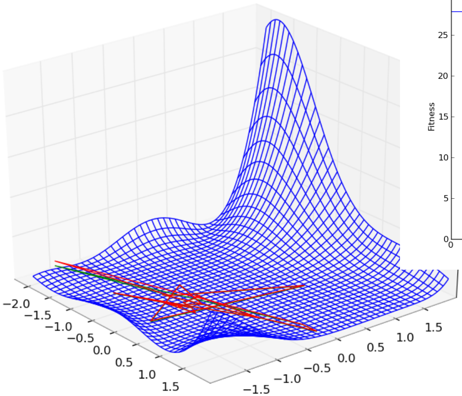
One global minimum of -1 at (π, π)

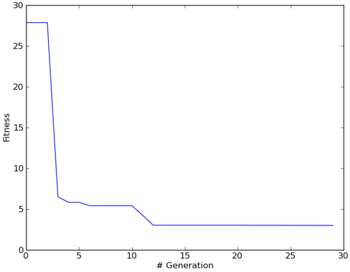


$$Es(x_1, x_2) = -\cos(x_1)\cos(x_2).\exp(-((x_1 - x_2)^2 + (x_2 - \pi)^2))$$

"GP" problem

One global minimum of 3 at $(0, -1)$



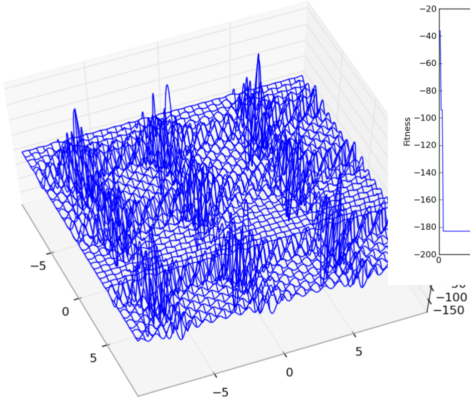


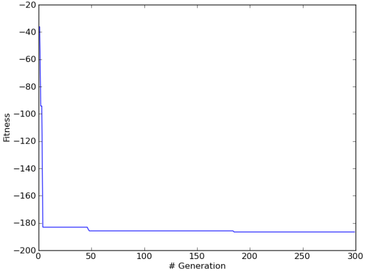
$$GP(x_1, x_2) = [1 + (x_1 + x_2 + 1)^2.(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2.(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$$

$$-2 < x_j < 2, \quad j = 1, 2$$

"SH" problem

760 local minima, 18 global minima of −186.73.





$$SH(x_1, x_2) = \sum_{j=1}^{5} j \cos[(j+1)x_1 + j] \times \sum_{j=1}^{5} j \cos[(j+1)x_2 + j]$$

$-10 < x_j < 10, \quad j = 1, 2$

Références I

Kennedy, J. and Eberhart, R. (1995).
Particle swarm optimization.
In *Proc. of the IEEE Int. Conf. on Neural Networks (ICNN'95)*, volume 4, pages 1942–1948.

Saxena, A. and Vora, M. (2008).
Novel approach for the use of small world theory in particle swarm optimization.
In *16th Int. Conf. on Advanced Computing and Communications (ADCOM 2008)*, pages 363–366.

Storn, R. and Price, K. (1995).
Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces.
Technical Report TR-95-012, International Computer Science Institute.

Références II

Zhang, C., Ning, J., Lu, S., Ouyang, D., and Ding, T. (2009).
A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization.
Operations Research Letters, 37 :117–122.

Zhang, W.-J., Xie, X.-F., and Bi, D.-C. (2004).
Handling boundary constraints for numerical optimization by particle swarm flying in periodic search space.
In *Congress on Evolutionary Computation (CEC)*, pages 2307–2311, Prtland, OR.