Metti 5

Advanced Spring School

Thermal Measurements & Inverse Techniques

Station Biologique de ROSCOFF

June 13-18, 2011

Zero order optimization algorithms

Tutorial 2 - Emmanuel RUFFIO*, Daniel PETIT, Didier SAURY, Manuel GIRAULT*

emmanuel.ruffio@let.ensma.fr, manuel.girault@let.ensma.fr

Institut P' (UPR CNRS 3346)

CNRS, ENSMA, Université de Poitiers

Département fluides, thermique, combustion

ENSMA - BP. 40109

86961 Futuroscope Chasseneuil

Outline

1) What is an optimization problem?

2) Traditional local search algorithms

3) Metaheuristics

3) Common nature-inspired metaheuristics

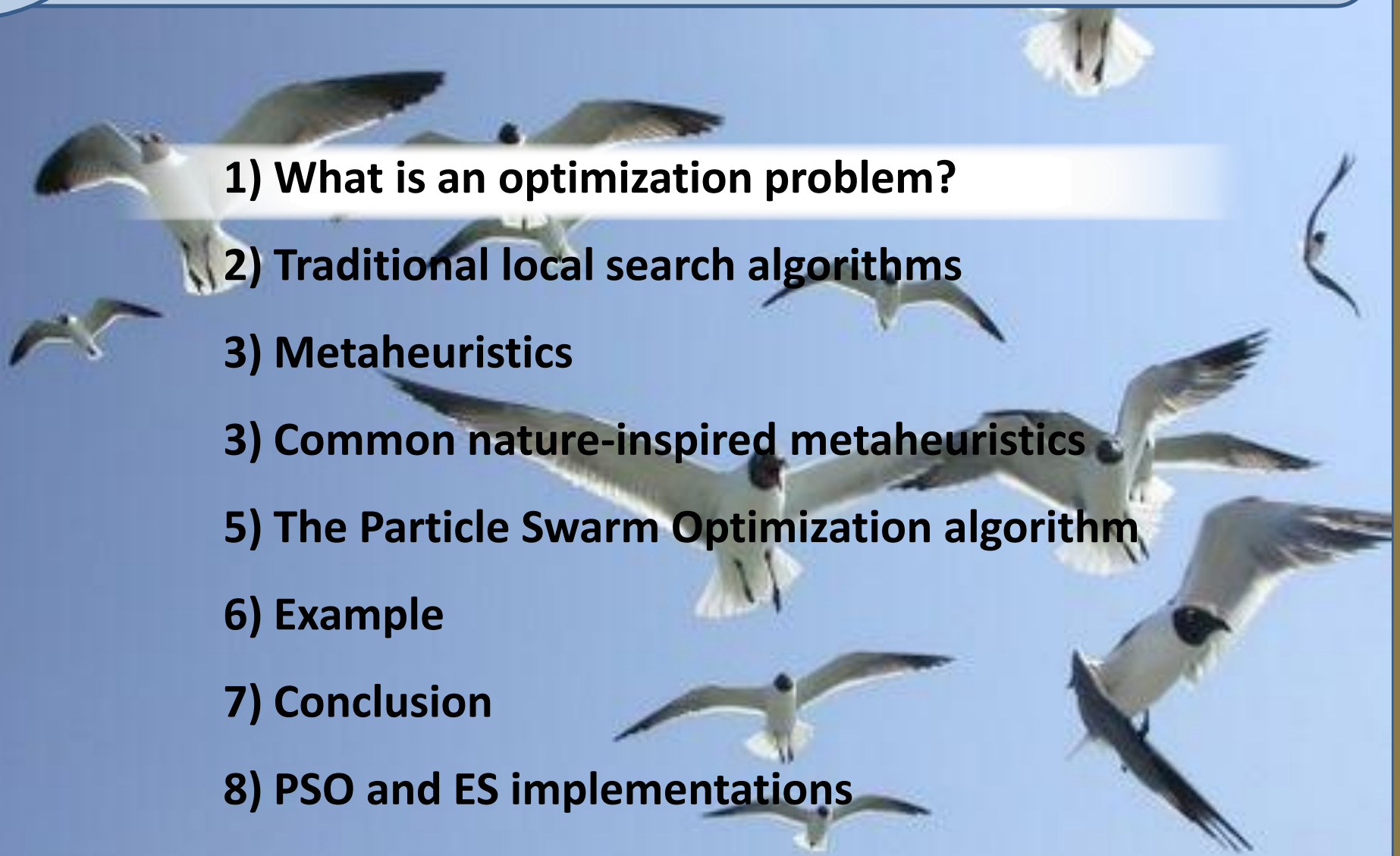
5) The Particle Swarm Optimization algorithm

6) Example

7) Conclusion

8) PSO and ES implementations

Outline



1) What is an optimization problem?

2) Traditional local search algorithms

3) Metaheuristics

3) Common nature-inspired metaheuristics

5) The Particle Swarm Optimization algorithm

6) Example

7) Conclusion

8) PSO and ES implementations

An optimization problem

Continuous optimization

$$\beta = (\beta_1, \ldots, \beta_{N_\beta})^T \in R^{N_\beta}$$

Single objective optimization

$$J(\beta) \in R$$

~~Multiobjective~~
~~Multimodale~~
~~Dynamic~~
~~Combinatorial~~

Boundary constraints

$$L_i \leq \beta_i \leq U_i \qquad \forall i \in [1; N_\beta]$$

Search space

$$S = [L_1; U_1] \times \cdots \times [L_{N_\beta}; U_{N_\beta}]$$

~~Linear constraints~~
~~$A.\beta \leq B$~~

~~Non-linear constraints~~
~~$f(\beta) = 0$~~

~~Soft constraints~~

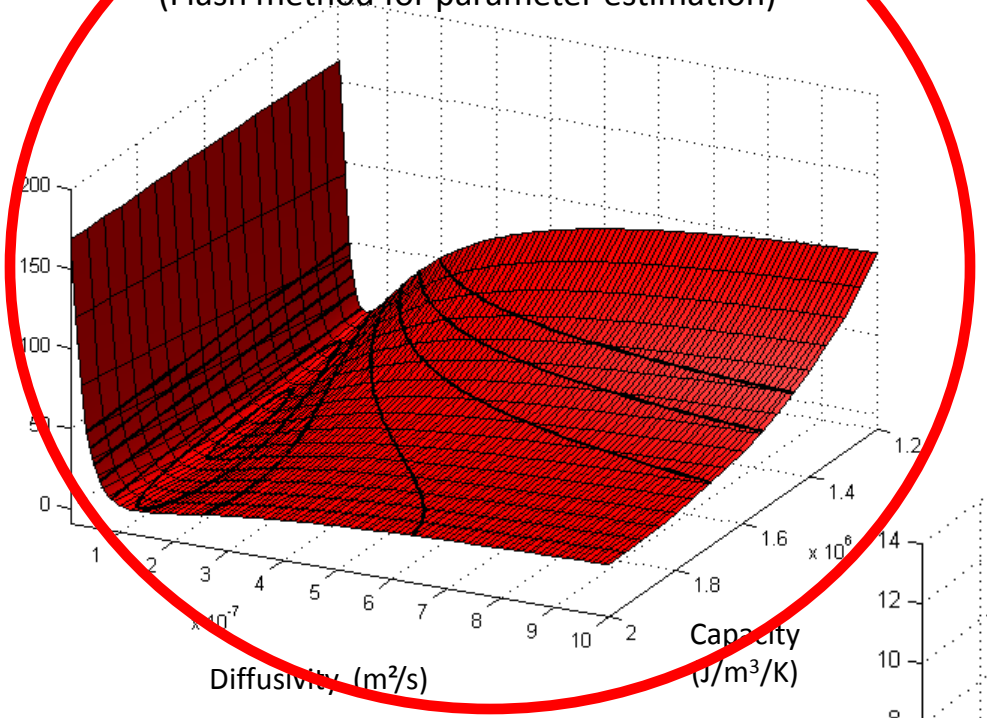
Our optimization problem

$$\hat{\beta} = \arg[\min_{\beta \in S} J(\beta)]$$

What kind of objective function?

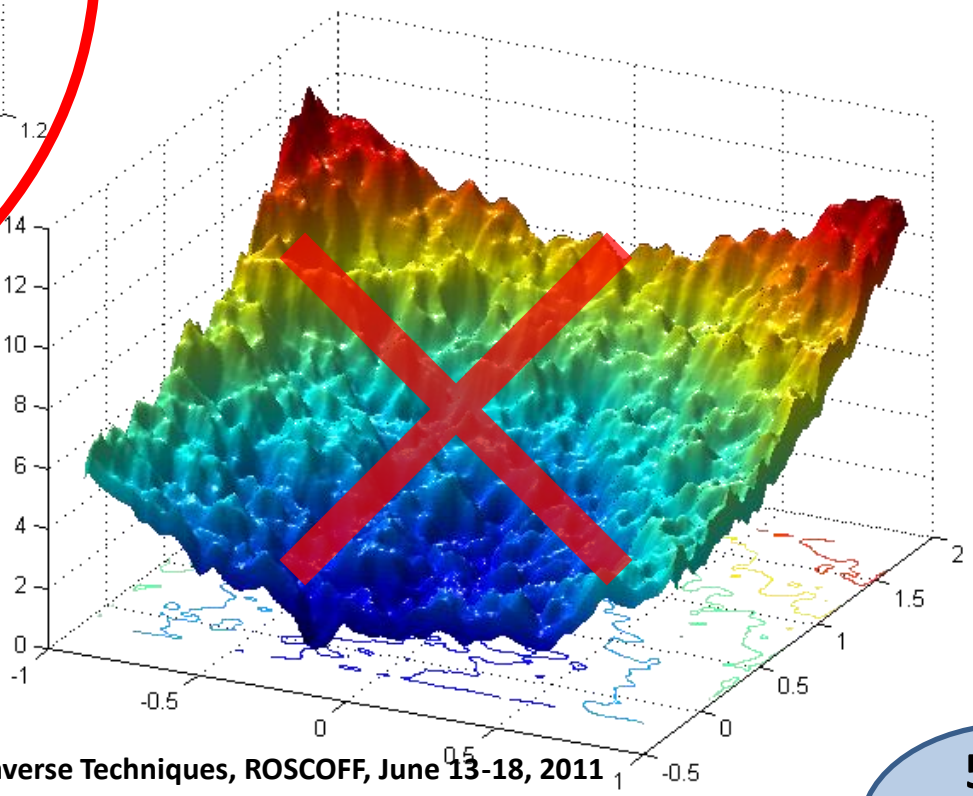
The simple case

Ex: Least squares
(Flash method for parameter estimation)

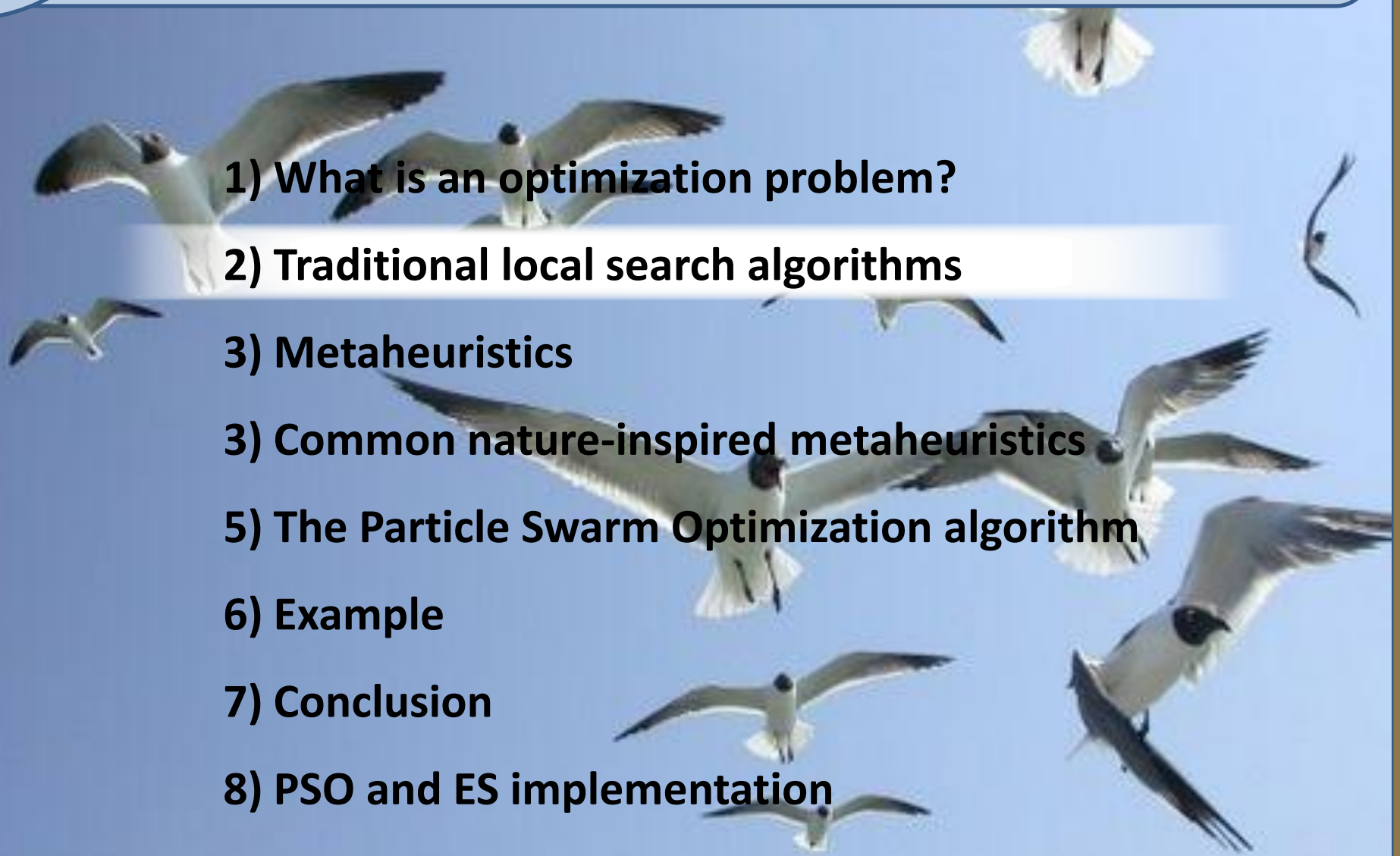


A little bit more challenging?

Ex: Least median of squares



Outline



1) What is an optimization problem?

2) Traditional local search algorithms

3) Metaheuristics

3) Common nature-inspired metaheuristics

5) The Particle Swarm Optimization algorithm

6) Example

7) Conclusion

8) PSO and ES implementation

Gradient-based methods (1/2)

Background of gradient-based methods: $J(\beta_k - \alpha_k . D_k) < J(\beta_k)$

Objective function

Step size

Descent direction

Unknown parameters

In which direction "D_k" ?

The name of a gradient-based algorithm depends on the way D_k is computed.

1) First-order method:

- Gradient : $D_k = \nabla J(\beta_k)$

2) Second-order method:

Hessian Matrix

- Newton: $D_k = [H(\beta_k)]^{-1} . \nabla J(\beta_k)$ and $\alpha_k \approx 1$

3) Pseudo second-order method:

- Quasi-Newton method: $H(\beta_k)$ is approximated (DFP, SR1, BFGS,…)
- Levenberg-Marquardt (least-squares): $J(\beta) = \sum_i (y^{\exp} - y(\beta))^2$

  $H(\beta_k)$ is approximated using the sensitivity matrix $X(\beta_k) = \frac{\partial y}{\partial \beta}(\beta_k)$

Gradient-based methods (2/2)

Background of gradient-based methods: $J(\beta_k - \alpha_k . D_k) < J(\beta_k)$

Objective function

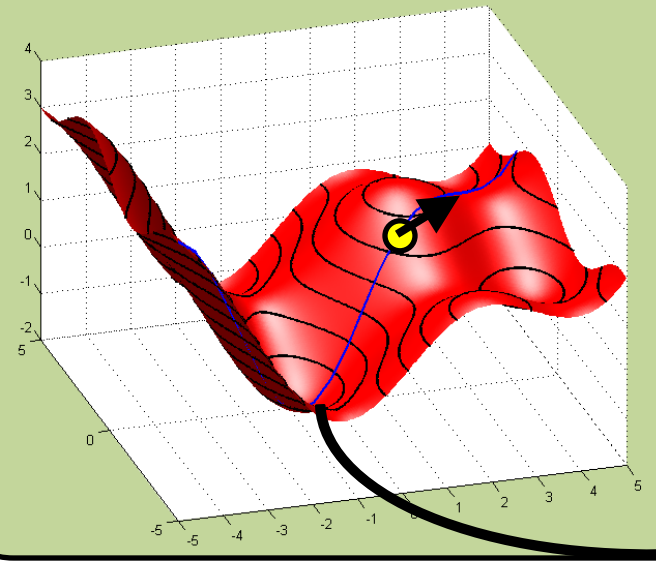
Step size

Descent direction

Unknown parameters

How far in that direction ? What is the step size "α_k" ?

Different ways to compute the step size α_k : (1) **Armijo**, (2) **Goldstein**, (3) **Wolfe** method or (4) the traditional **linear search**:



$$\alpha_k = \arg[\min_{\alpha \geq 0}(J(\beta_k + \alpha . D_k))]$$

Function along the gradient direction

Gradient

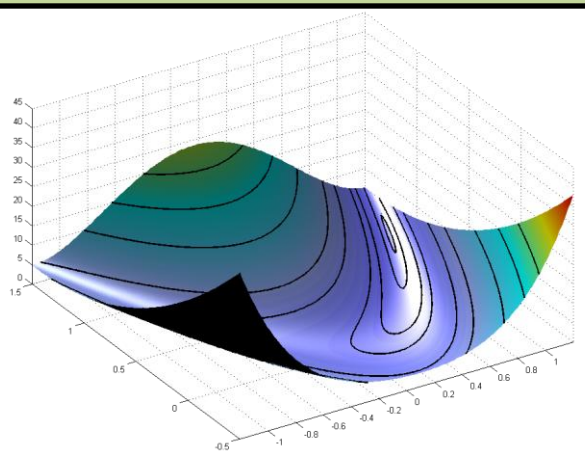
β

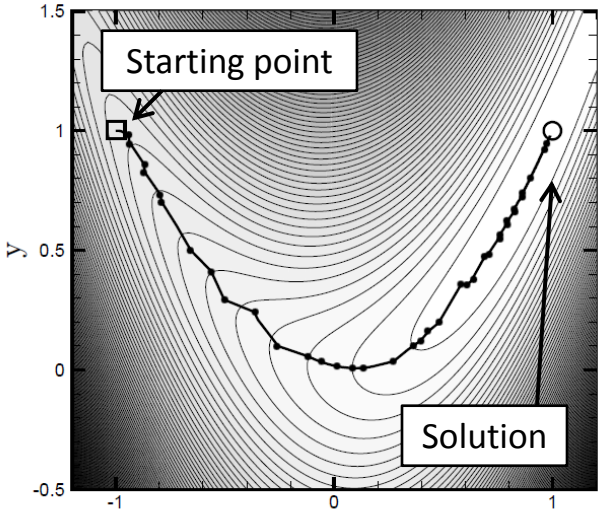
Order and efficiency

Iterative procedure: $\beta_{k+1} = \beta_k - \alpha_k . D_k$ until a convergence criterion is satisfied

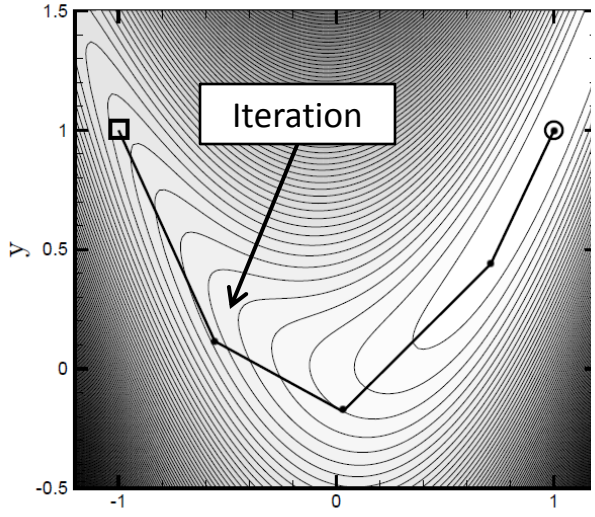
The Rosenbrock function

$$J(\beta) = \sum_{i=1}^{N_\beta - 1} \left[(1 - \beta_i)^2 + 10 . (\beta_{i+1} - \beta_i^2)^2 \right]$$

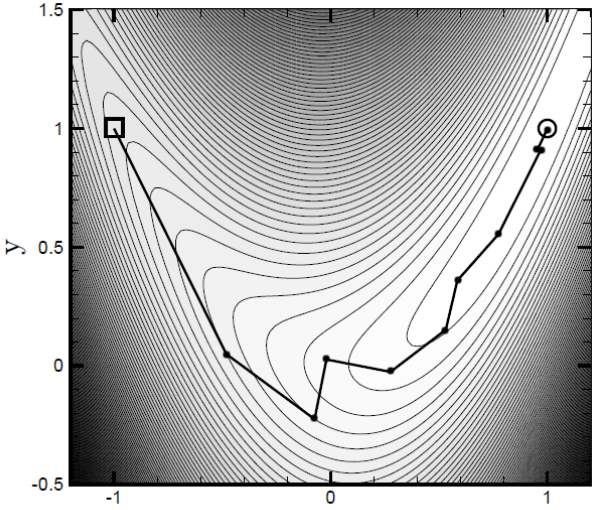




First order
Gradient method



Second order
Newton's method



Pseudo second order
Quasi-Newton's method

Simplex methods (1/2)

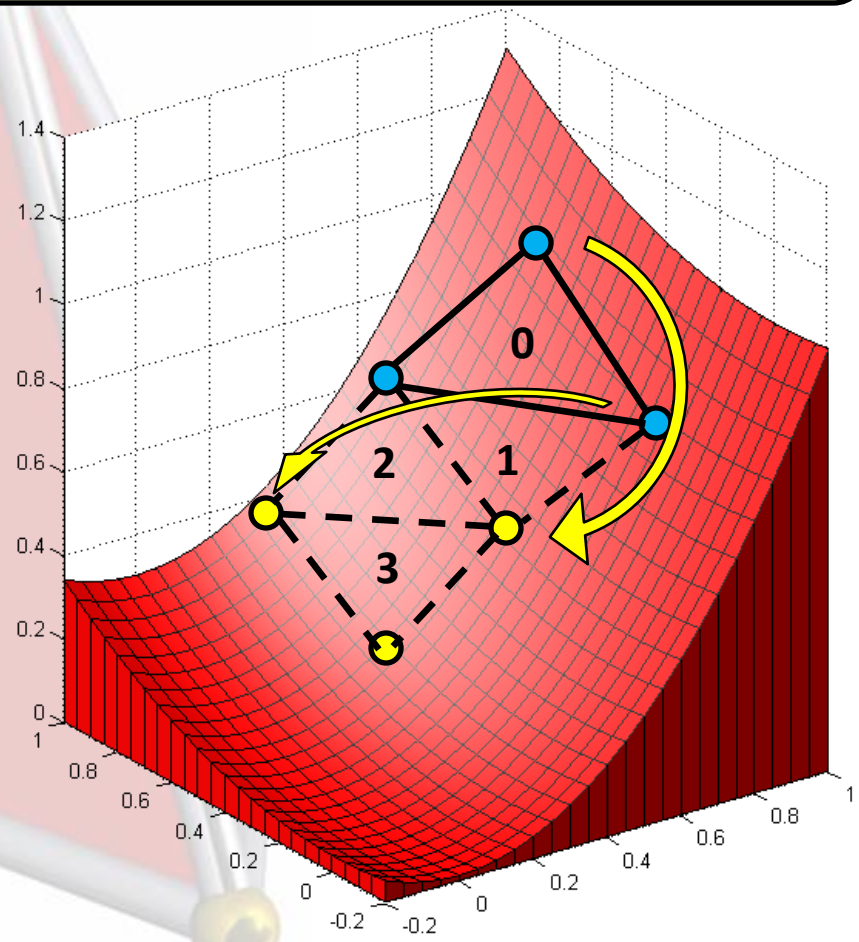
Definition : **a simplex is a generalization of a triangle to arbitrary dimension. If k is the number of dimensions, then a simplex is defined by k+1 points in this space.**

The algorithm : (George Dantzig, 1947)
-Deterministic
-Derivative-free (zero-order)
-Local search

Simple but efficient method

The worst is iteratively replaced by its symmetric with respect to the center of gravity of the other points of the simplex

-> Few rules to handle tricky situations

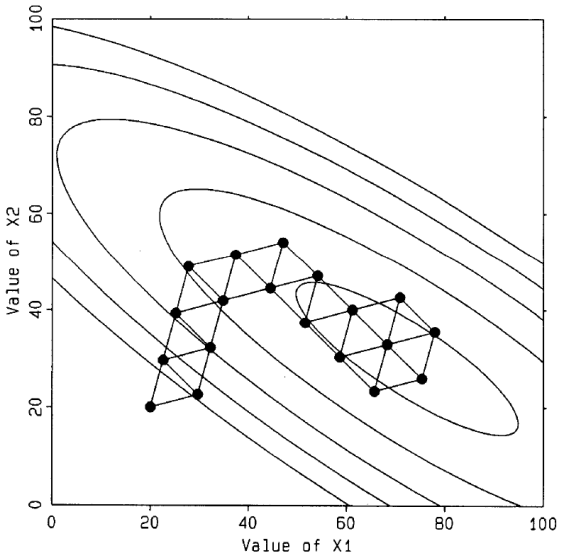


A 2D-case $J(x, y)$

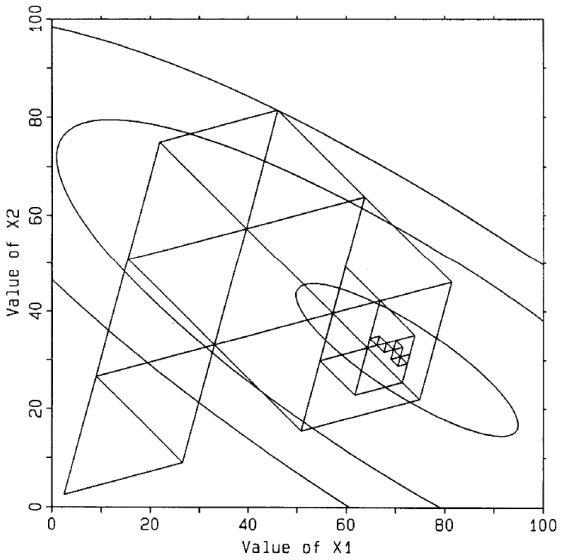
Simplex methods (2/2)

Many variants of the algorithm !

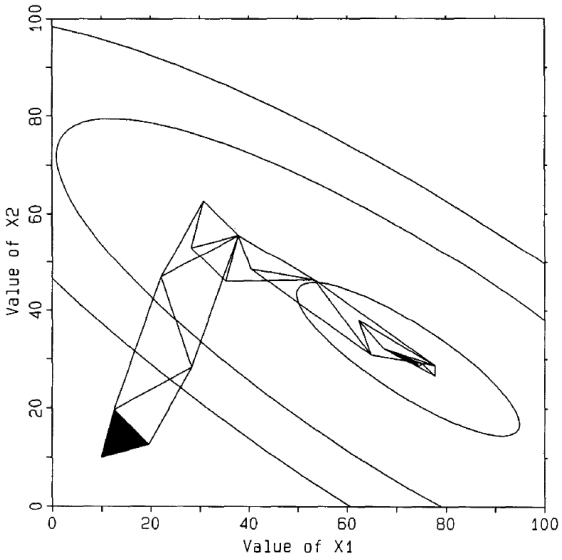
The simplest one
Fixed size



Simple but efficient
Sequential fixed size



Trickier
Variable size

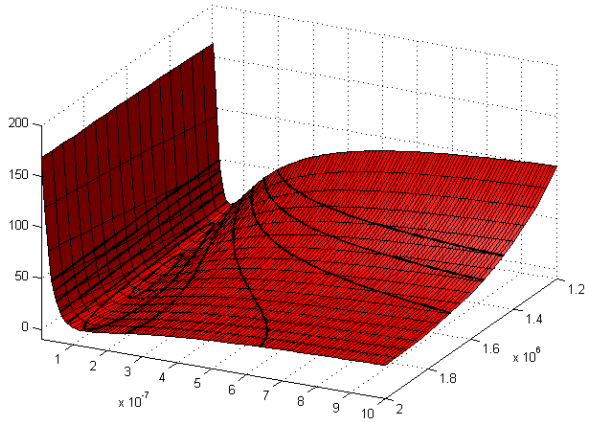


What about this one ?

The simple case

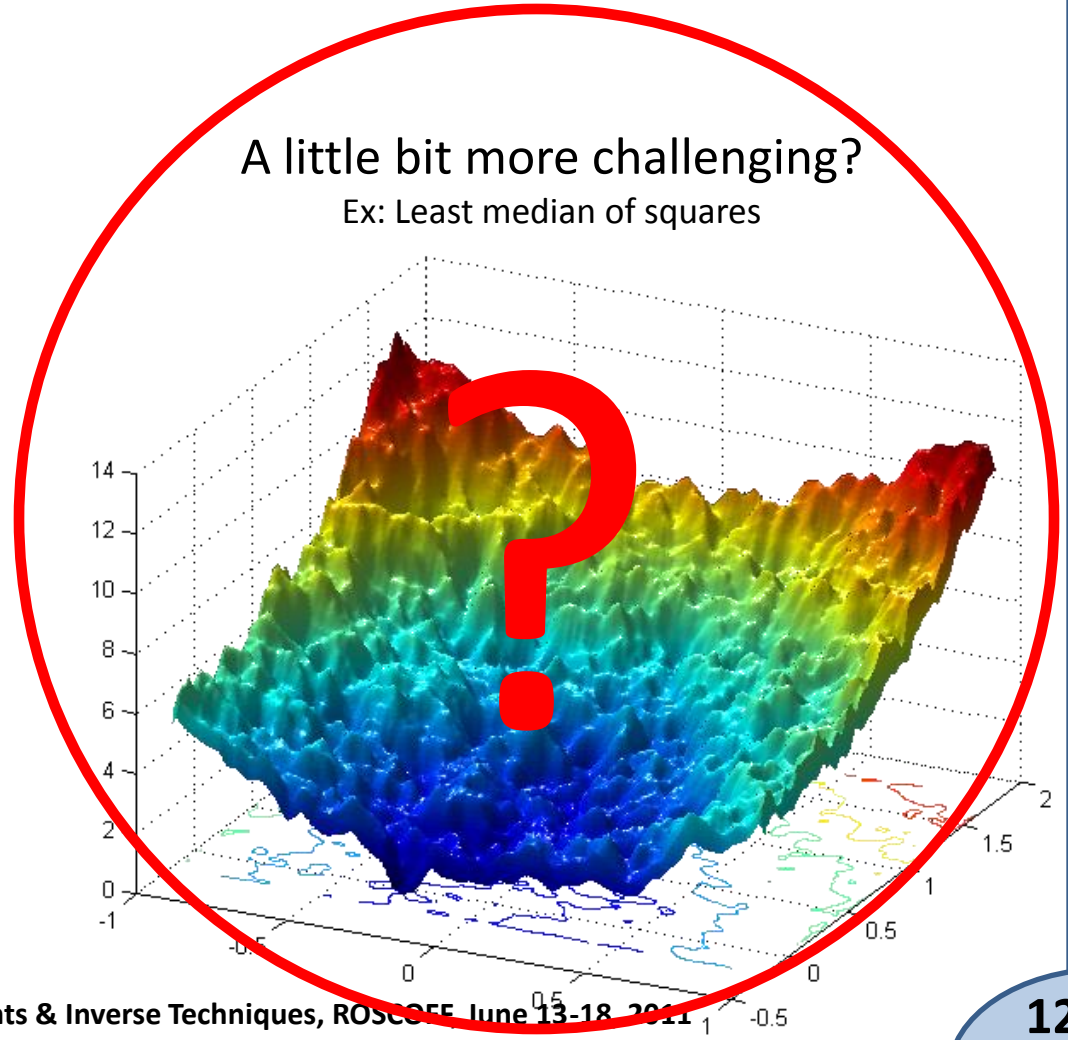
Ex: Least squares

(Flash method for parameter estimation)

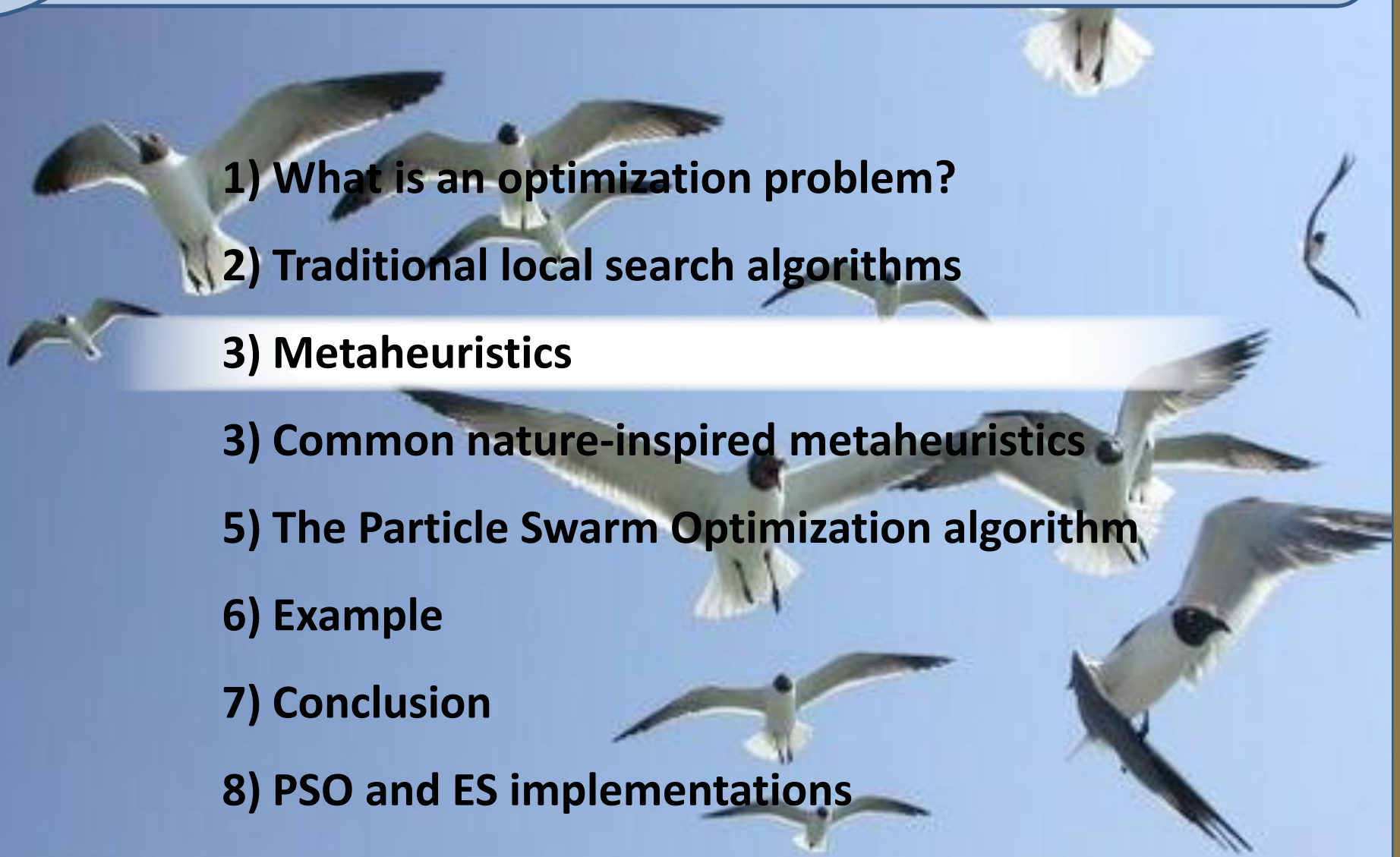


A little bit more challenging?

Ex: Least median of squares



Outline



1) What is an optimization problem?

2) Traditional local search algorithms

3) Metaheuristics

3) Common nature-inspired metaheuristics

5) The Particle Swarm Optimization algorithm

6) Example

7) Conclusion

8) PSO and ES implementations

What is a difficult problem ?

A hard problem:

The time required using any currently known algorithm **increases exponentially** with the size of the problem.

Here we are…

Then, what can we do ? three possibilities:

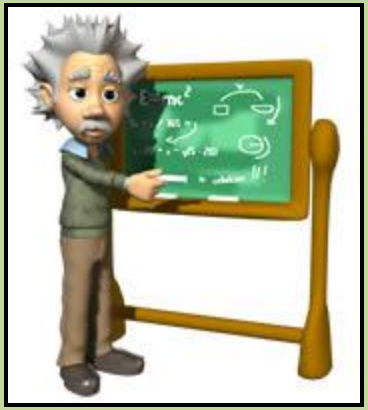
1. Use algorithms that compute **all feasible solutions** and keep best one, but it may take **exponential time**.
2. Use "approximation algorithms" that always run in **polynomial time** but they may not **always produce the optimal solution**
3. Find a new job

A metaheuristic ? (1/3)

Heuristic (from the Greek heuriskein: "**find**" or "**discover**"):
A commonsense rule (or set of rules) based on experience that aids problem solving

Examples:

•If you are **packing odd-shaped items** into a box: start with the largest remaining items, fit the smaller items into the spaces left



•If you are having difficulty **understanding** a problem, try drawing a **picture**.

A metaheuristic ? (2/3)

Heuristic in computer science:

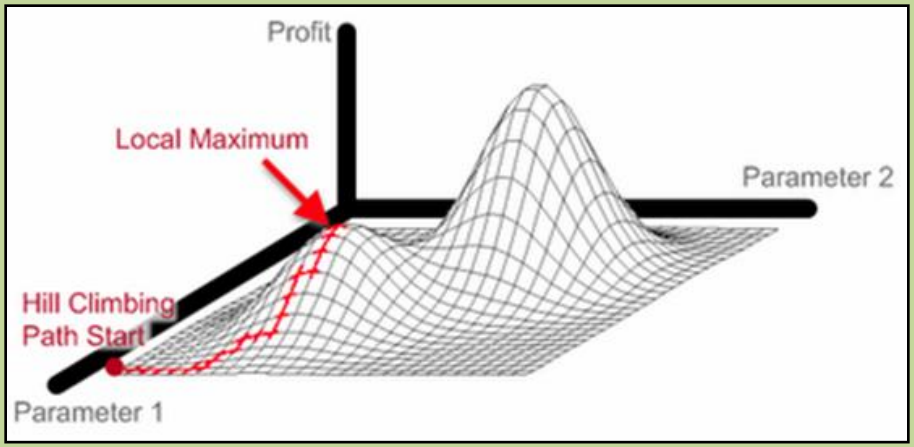
An algorithm that generally produces **acceptable solutions** with a **reasonable amount of time** in many situations, but:

-the solution can be bad

-the algorithm can be horribly slow

Example:

Greedy heuristic: to maximize profit, choose the solution of the neighborhood that maximizes local profit (Hill climbing)



A metaheuristic ? (3/3)

Meta (from Greek for "**beyond**" or "**higher**"):
a prefix used to indicate a concept which is an abstraction from another concept.

Meta-heuristic: No commonly agreed definition !

- Metaheuristics are strategies that guide the **search process**.
- Algorithms that find **acceptable solutions** of problems by using **several heuristics**.
- A **metaheuristic** is a heuristic method for solving a very general class of computational problems by combining heuristics in the hope of obtaining a more efficient or more robust procedure.

Fundamental properties of metaheuristics

- Efficiently **explore the search space**.
- May be **approximation** and **stochastic** algorithms.
- Mechanisms to avoid getting trapped **local optima**.
- Use **search experience** to be more efficient than random search.

Three principles

Exploration
(diversification, global search)
Finding areas of the search space that are still not investigated

Exploitation
(Intensification, local search)
Improving the current solutions found by performing generally small changes

Memory
(Search experience)
Store information

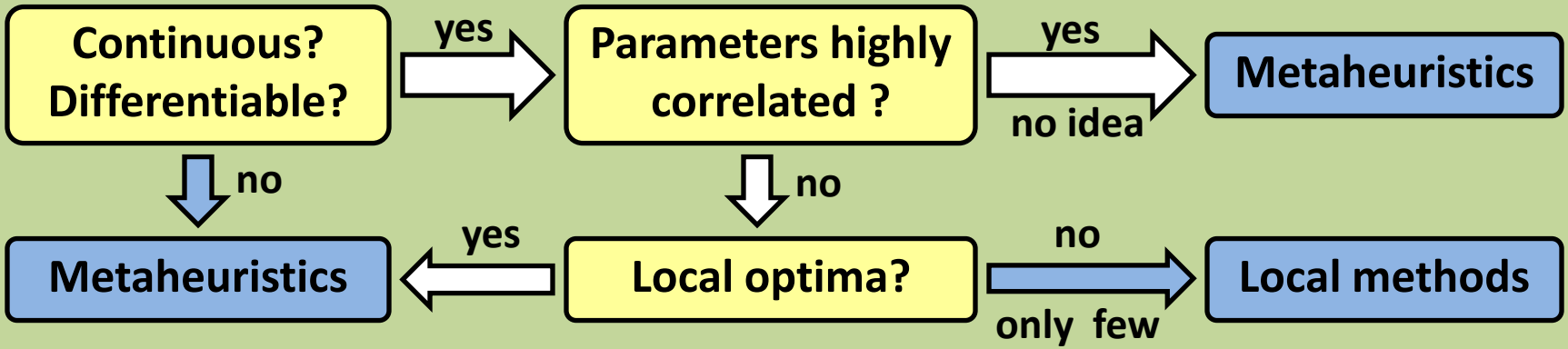
And what…?

Ok, metaheuristics are designed to find acceptable solutions of NP-hard problems

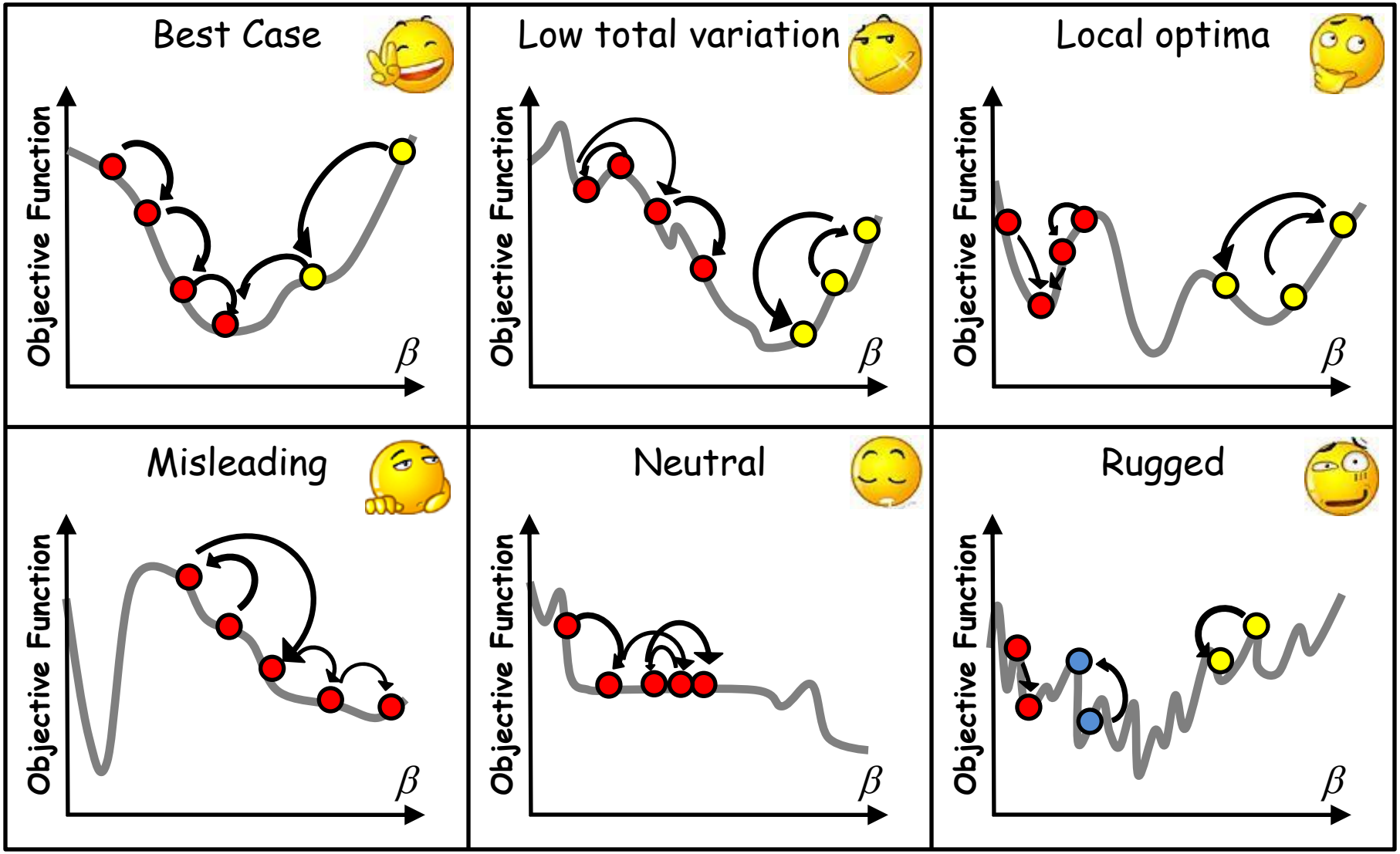
But for us, humble researchers and engineers in thermal sciences who don't understand anything to complexity computation theory, what does really matter in practice ?

1) **Which optimization algorithm can I use ?**
2) **For my problem, is this algorithm suitable ?**

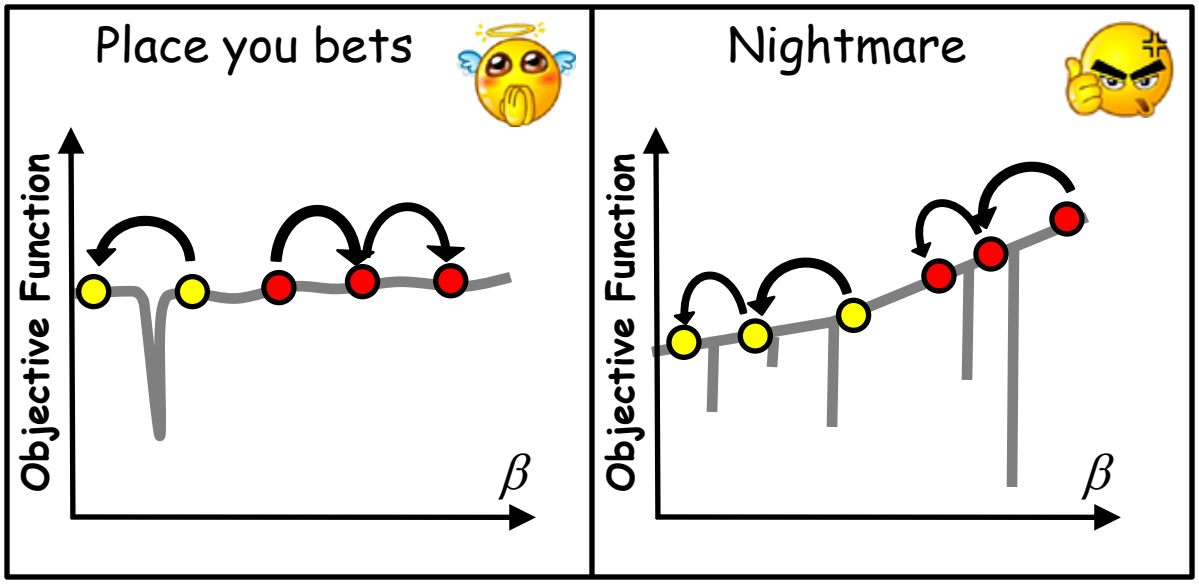
It depends … on your objective function:



Is it a hard objective function?



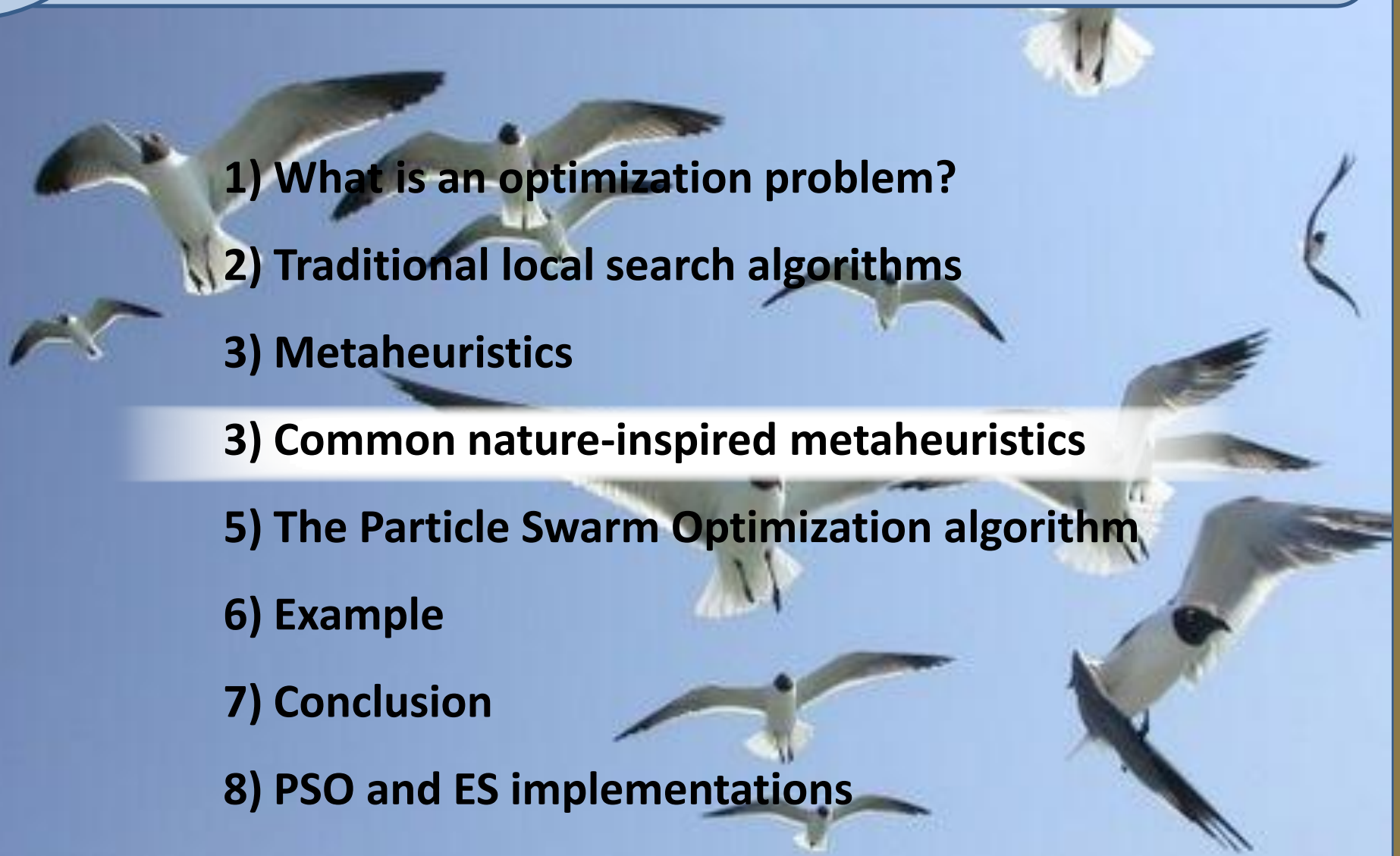
Probably not for us



-Robust optimum are often preferred

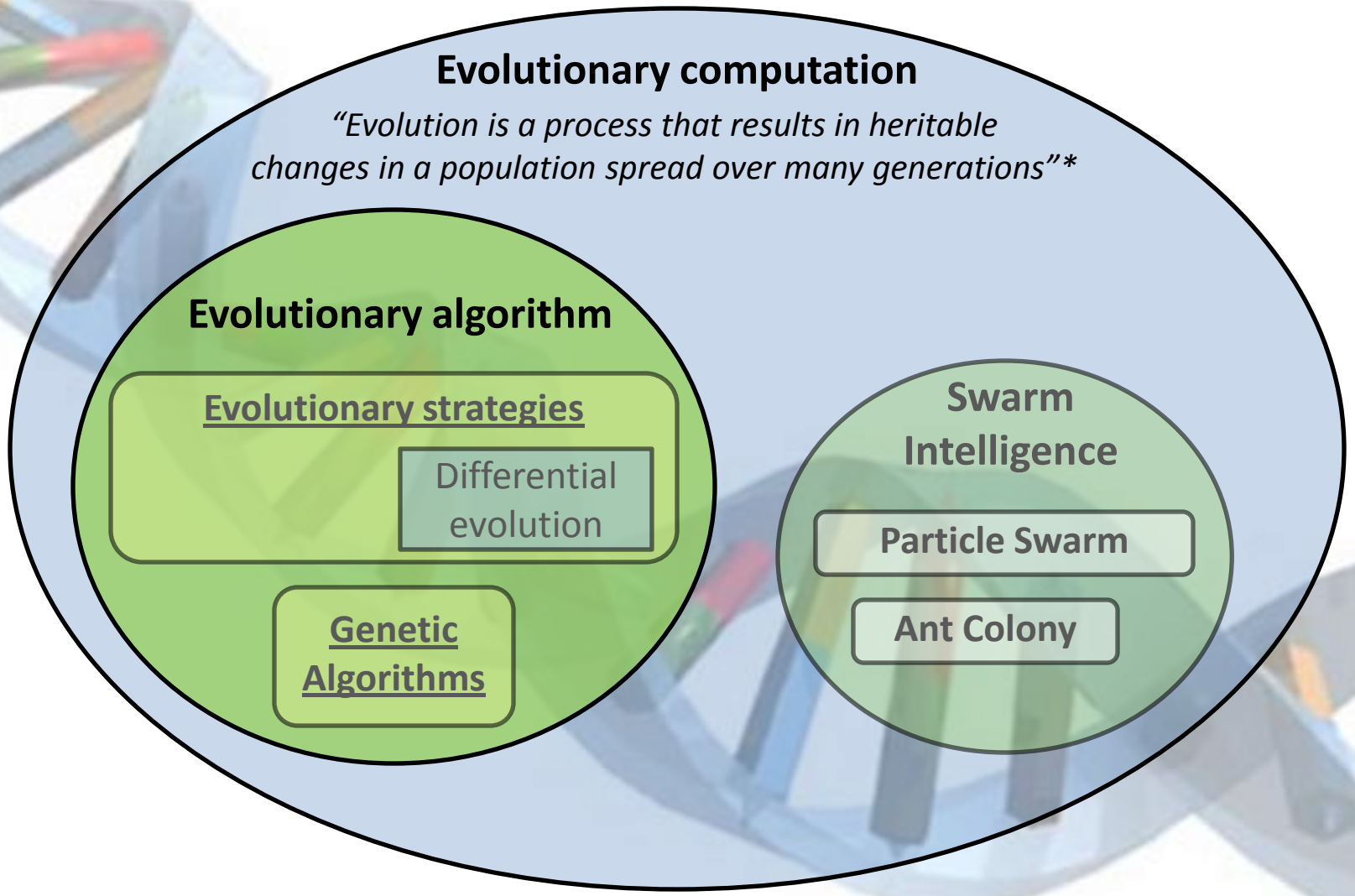
-Kind of situation hardly possible in thermal modelling

Outline



1) What is an optimization problem?
2) Traditional local search algorithms
3) Metaheuristics
3) Common nature-inspired metaheuristics
5) The Particle Swarm Optimization algorithm
6) Example
7) Conclusion
8) PSO and ES implementations

Genetics and evolution



* http://www.talkorigins.org/faqs/evolution-definition.html, 03/11/2009

Evolutionary algorithms (1/3)

Only the strong will survive

An evolutionary algorithm (EA):

-**generic population-based** algorithm

-**stochastic**

-inspired by the **theory of natural selection**

Technical terms:

Objective function	~~Solutions, Candidates~~	~~Parameters~~	~~Set of solutions~~
!	↓	↓	↓
Fitness (Rescaled function)	Individuals (chromosome)	Genes	Population

Evolutionary algorithms (2/3)

Main principle:

An EA manipulates a **population of individuals** that **compete** with each other through **biological inspired mechanisms** called **operators**. The fittest individuals produce more **offspring**. Generations after generations, the **population improves** and converges to a solution.

Operator:

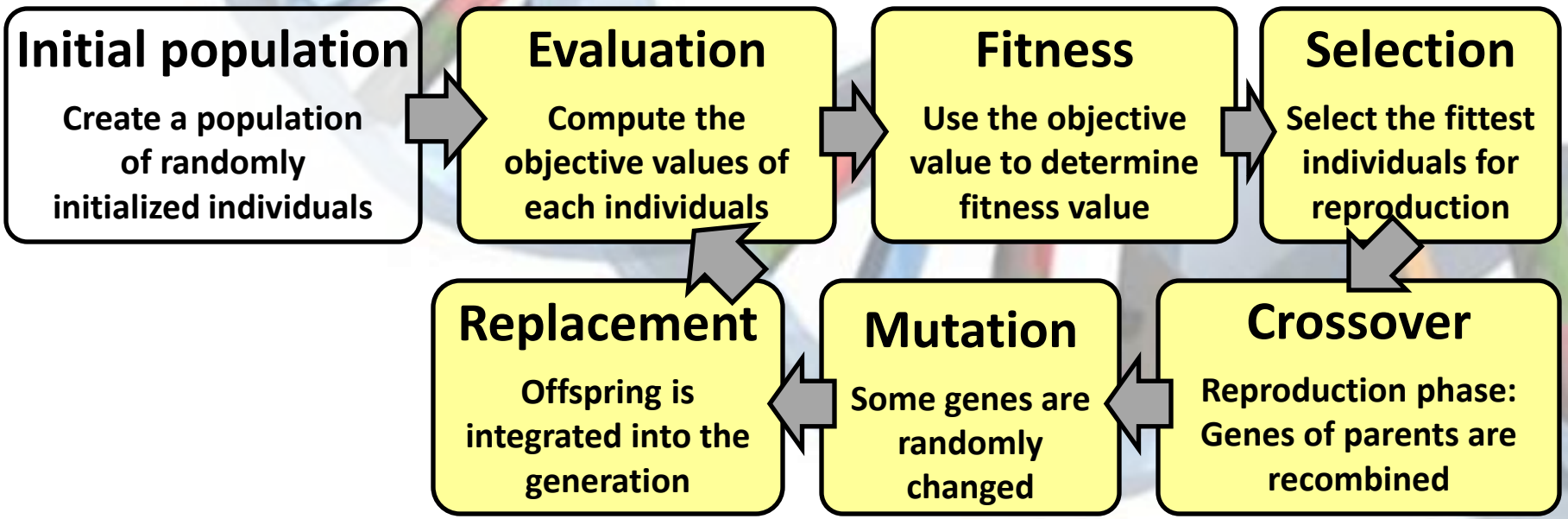
In mathematics, any symbol that indicates an operation to be performed. An operator may be regarded as a **function**, **transformation**, or map. *(Encyclopedia Britannica, 2008)*.

Evolutionary algorithms (3/3)

An algorithm is an "**evolutionary algorithm**" if it uses particular **operators** inspired by biological evolution:

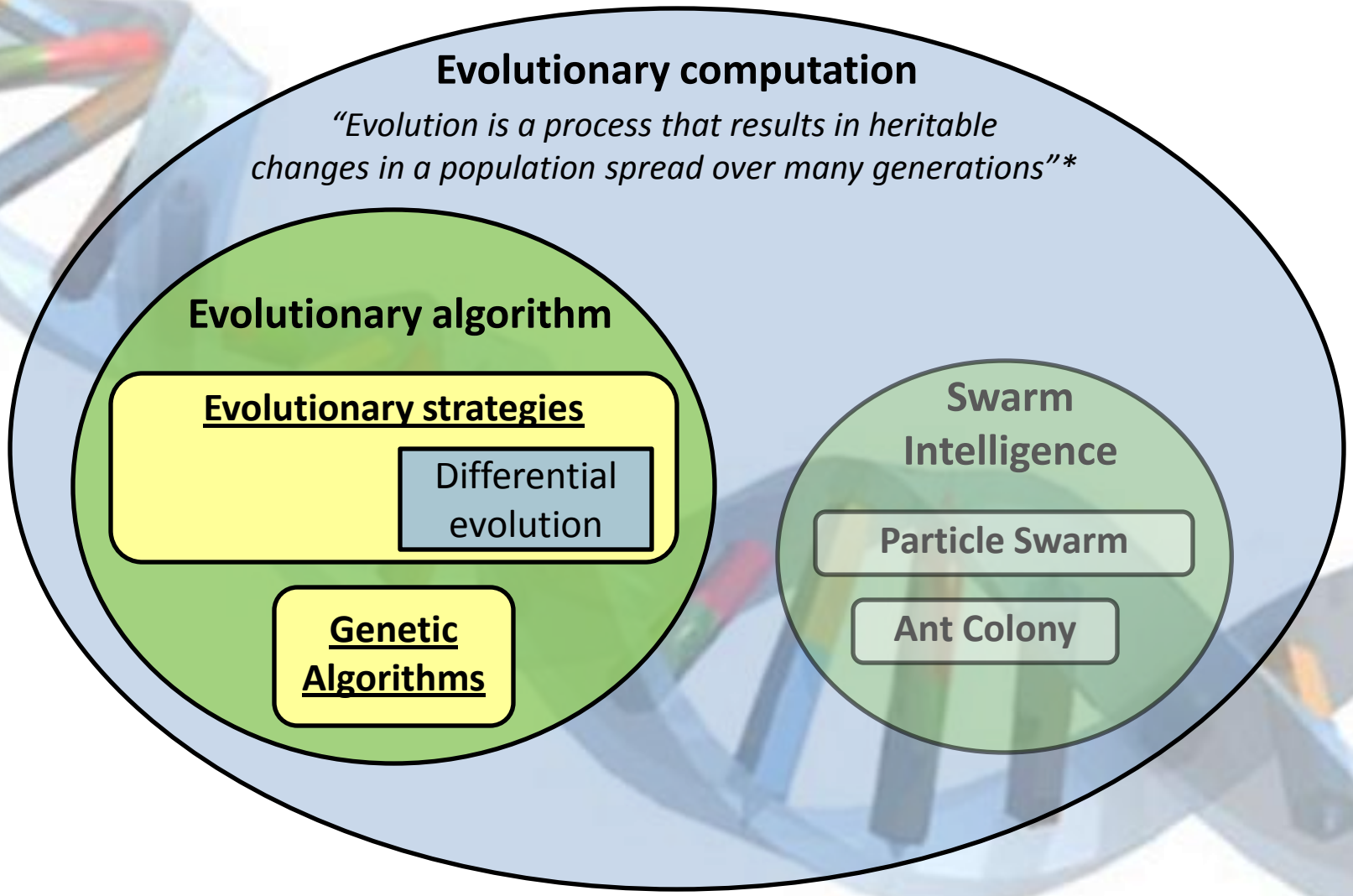
Selection, **crossover**, **mutation** and **replacement**

Generic structure



Inspired from Thomas Weise, *Global Optimization Algorithms – Theory and Application* –, e-book, 2008-05-20

Genetics and evolution



* http://www.talkorigins.org/faqs/evolution-definition.html, 03/11/2009

GA and ES (1/3)

Example: $f(x, y, z) = x^2+y^2+z^2$

(1)

F	38
x	2
y	-3
z	5

(2)

F	6
x	1
y	2
z	-1

(3)

F	13
x	3
y	2
z	2

(4)

F	53
x	0
y	6
z	3

Evolution strategies (ES) : "(4/2 + 4)-ES"

2 random parents for each children

(2)	(4)
1	0
2	6
-1	3
σ=1	σ=2

New parameter

Crossover

M
0
2
3
σ=2

Mutations create children

+ N(0, σ) =
+ N(0, σ) =
+ N(0, σ) =
X { α 50% ; 1/ α 50% }

C1
2
0
5
1.5

F=29

New population:

The 4 best individuals are kept

(2)	(3)	C2	C1	~~(1)~~	C4	~~(4)~~	C3
6	13	15	29	~~38~~	50	~~53~~	70
1st	2nd	3th	4th	~~5th~~	6th	~~7th~~	8th

× 4

F=15

F=70

F=50

In "(4/2,4)-ES" : parents are replaced by the 4 best children

GA and ES (2/3)

Example: $f(x, y, z) = x^2+y^2+z^2$

Genetic algorithm (GA)

(1)	F	38
	x	2
	y	-3
	z	5

(2)	F	6
	x	1
	y	2
	z	-1

(3)	F	13
	x	3
	y	2
	z	2

(4)	F	53
	x	0
	y	6
	z	3

Individual are ranked

1st	(2)	6
2nd	(3)	13
3th	(1)	38
4th	(4)	53

Selection probability

38%
29%
21%
12%

Individuals selected

(2)
(2)
(3)
(1)

4 couples are created

(2)+(3)
(2)+(2)
(2)+(1)
(1)+(3)

Crossover

(2)	1	2	-1
(3)	3	2	2
=			
C1	1	2	2

Crossover point

C1
C2
C3
C4

Mutations (very few)

C1
C2'
C3
C4

Replacement

(1)
(2)
(3)
(4)

GA and ES (3/3)

(1)

F	38
x	2
y	-3
z	5

(2)

F	6
x	1
y	2
z	-1

(3)

F	13
x	3
y	2
z	2

(4)

F	53
x	0
y	6
z	3

Evolution strategies (ES) : "(4/2 + 4)-ES"

2 random parents for each children

(2)	(4)
1	0
2	6
-1	3
σ=1	σ=2

Crossover

M
1
2
3
σ=2

Mutations

4 children

C1
2
0
5
1.5

Selection of the 4 best Individuals

(2)	6
(3)	13
C2	15
C1	29

(1)	38
C4	50
(4)	53
C3	70

Genetic algorithm (GA)

Selection

1st	(2)	6	38%
2nd	(3)	13	29%
3th	(1)	38	21%
4th	(4)	53	13%

4 couples are created

(2)+(3)
(2)+(2)
(2)+(1)
(1)+(3)

Crossover

C1
C2
C3
C4

Mutations

C1
C2'
C3
C4

New population

(1)
(2)
(3)
(4)

Remarks

Many similarities between ES and GA:

-Same operators and similar structure

-Same concepts (non guided mutation, selection pressure…)

"It is selection, and only selection, that directs evolution in directions that are nonrandom with respect to advantage." (Dawkins, 1986, p.312).

As all metaheuristics, many, many… and many variants:

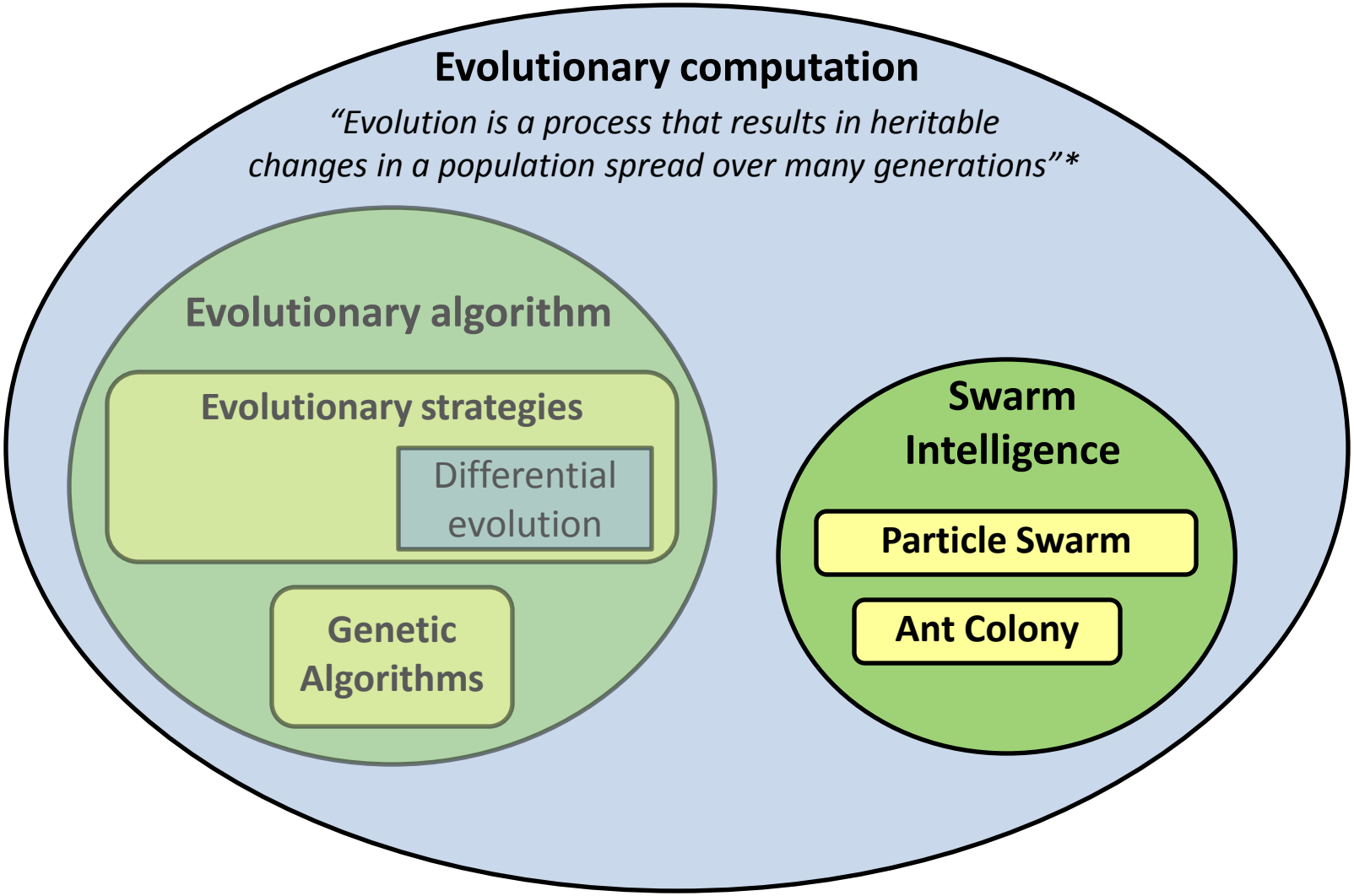
Example: different operators for EA:

Selection	Crossover	Mutation	Replacement
Uniform	1-point[1]	Permutation	Total[1]
Rank[1]	Multi-point	Random variable[2]	Elitist[2]
Tournament	Uniform[2]	Replacement[1]	Steady state
Random[2]…	Arithmetic…	…	…

1 – variant used for GA in previous slides

2 – variant used for ES in previous slides

Social behaviors



* http://www.talkorigins.org/faqs/evolution-definition.html, 03/11/2009

Swarm intelligence

A swarm:

A set of (mobile) agents which are liable to **communicate directly or indirectly** (by acting on their local environment) with each other, and which collectively carry out a distributed problem solving.

http://www.molbio.ku.dk/MolBioPages/abk/PersonalPages/Jesper/Swarm.html, 05/11/2009

Swarm intelligence (SI)

Type of artificial intelligence based on the **collective** behavior of **decentralized**, **self-organized** systems. (wikipedia)

-**Very simple rules**

-**Multiplicity, randomness, messiness**

-**No centralized control structure dictating how individual agents should behave**

-**Interactions lead to the emergence of "intelligent" global behavior**

Examples

Ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling.

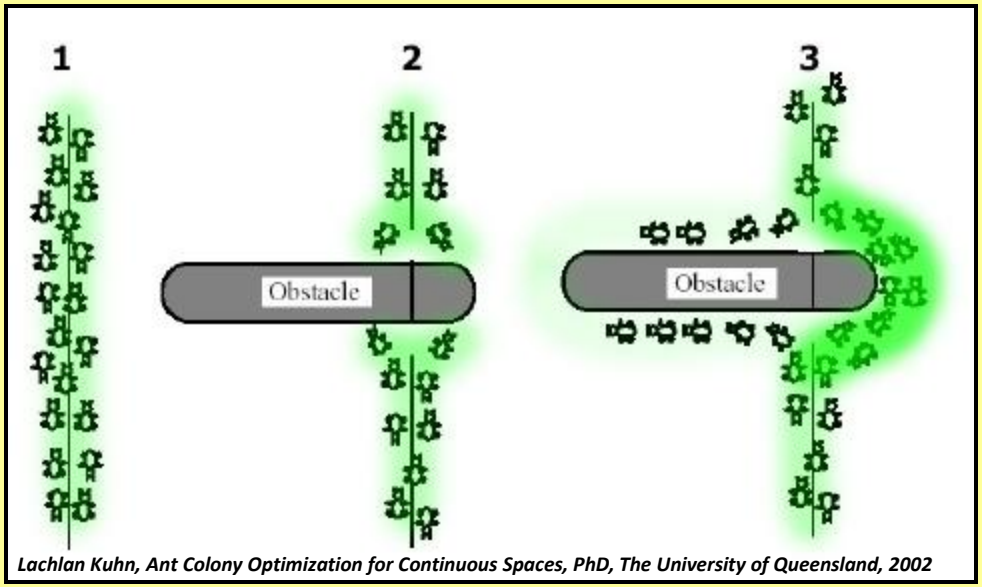
ACO algorithm

Ant colony optimization (ACO) (Dorigo 1992):

-inspired by the ants' foraging behavior (wandering in search of food)

-indirect communication using chemical pheromone trails (stigmergy)

-shortest path between food sources and their nest



Lachlan Kuhn, Ant Colony Optimization for Continuous Spaces, PhD, The University of Queensland, 2002

First and mostly applied to discrete optimization problem

Traveling salesman problem

Traveling salesman problem (TSP) (1930):

Given a list of cities and their pairwise distances, find a shortest possible tour that visits each city exactly once.

Solution found by my swarm of neurons

Algorithm:

For each iteration
- **For each** ant "k"
  - **Choose a starting city** randomly
  - **For each** non visited city
    - **Add a random one** — Intensity, Visibility
  - **End**
  - **Evaluate the tour**
  - **Ants depose pheromone**
- **End**
- **Evaporation of pheromone**

End

ACO for continuous problem ? ACO_R

Problem:

-infinite number of possible paths in the continuous domain

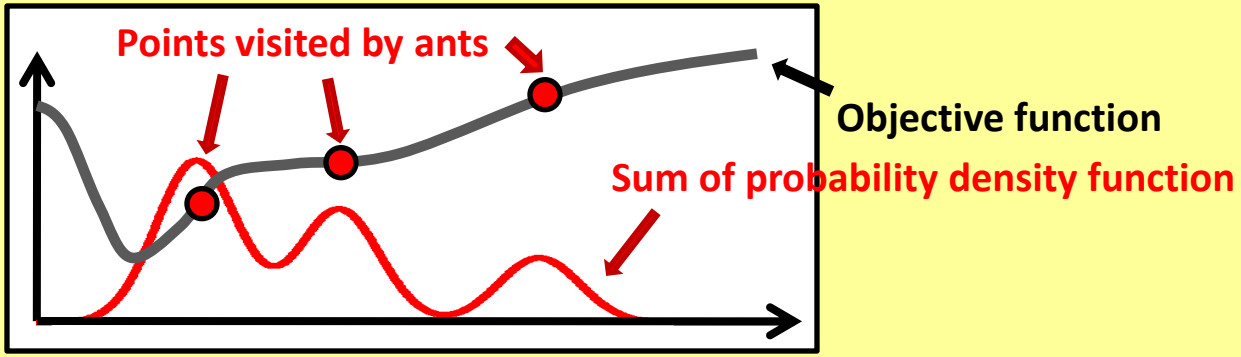
A solution:

-pheromone trails are replaced by probability density functions

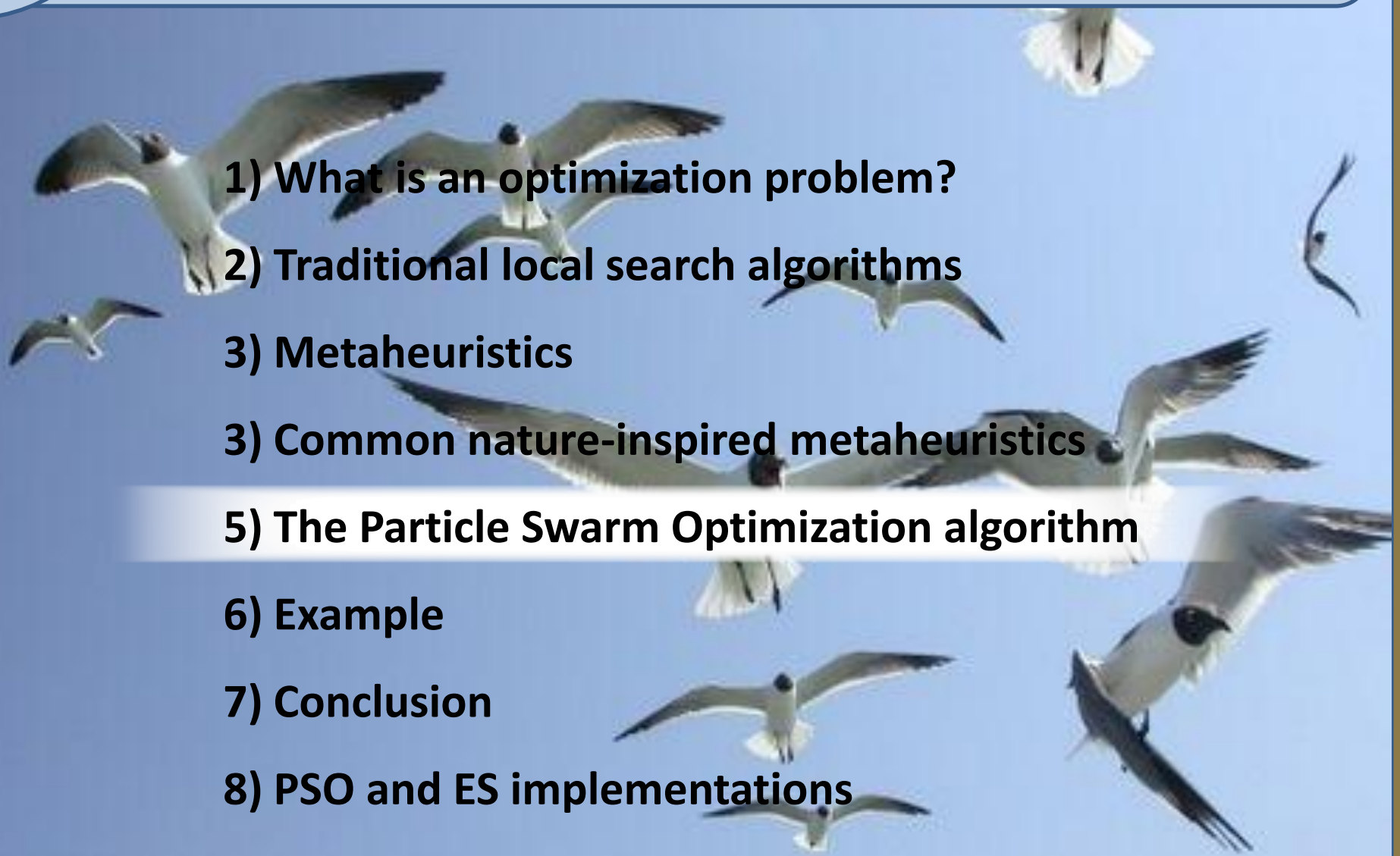
-evaporation is replaced by the forgetting the worst solutions

-ants randomly sent with respect to the sum of probability density function

Example:



Outline



1) What is an optimization problem?
2) Traditional local search algorithms
3) Metaheuristics
3) Common nature-inspired metaheuristics
5) The Particle Swarm Optimization algorithm
6) Example
7) Conclusion
8) PSO and ES implementations

PSO algorithm

Particle Swarm Optimization (PSO) (Eberhart, Kennedy 1995):

-inspired from the flocking of birds and fish
-how individuals interact
-how individuals learn from experience

Properties:

-stochastic
-derivative-free
-global

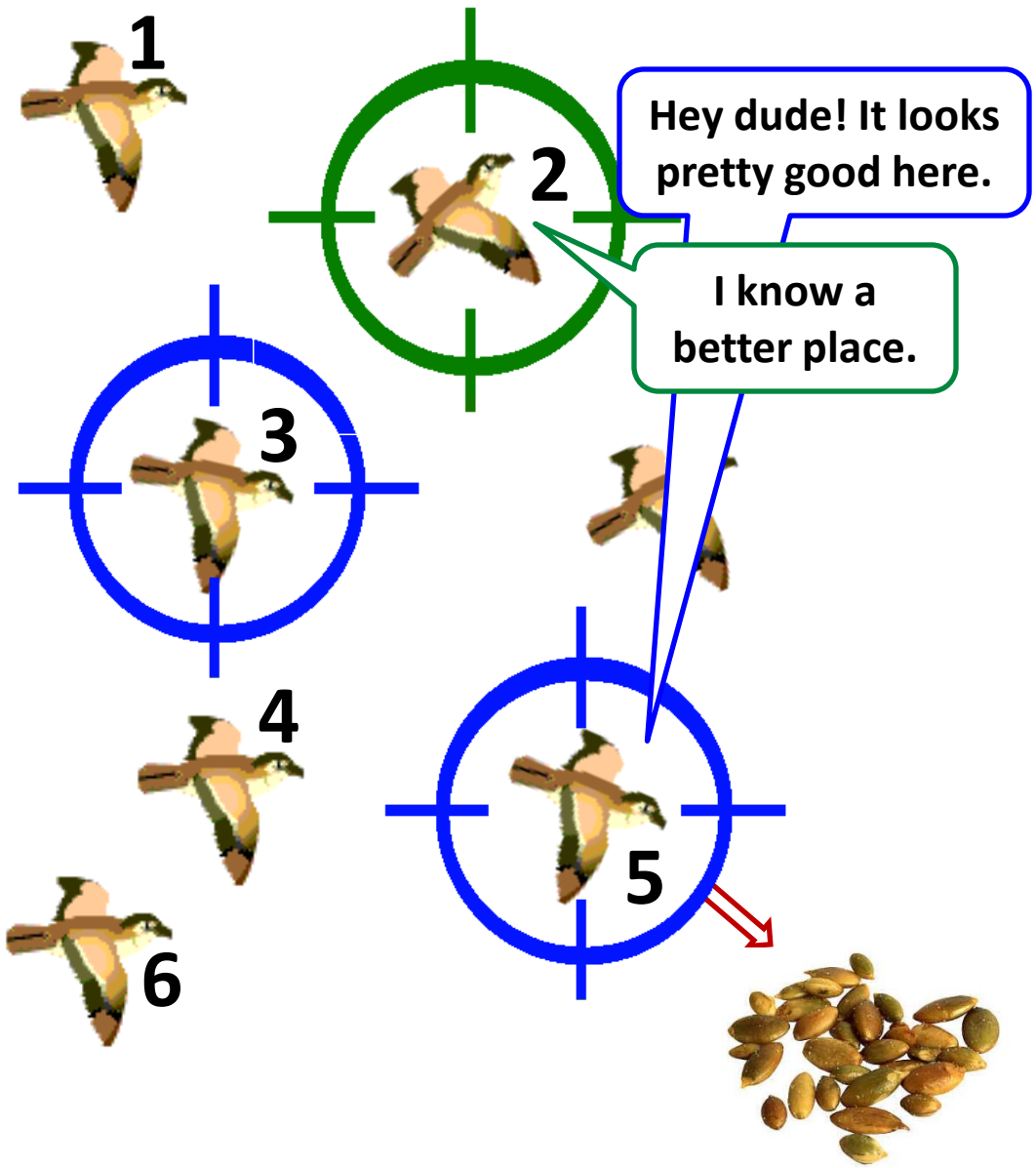
-population-based
-natured-inspired
-adapted to continuous problems

Technical term:

~~Solutions, Individuals~~

Particle

Swarm of particles



Particle 2

-Position $\beta \in S \subset R^{N_\beta}$

-Velocity $V\left(=\Delta\beta\right)$

-Performance $J\left(\beta\right) \in R$

-Best position $\hat{M}$

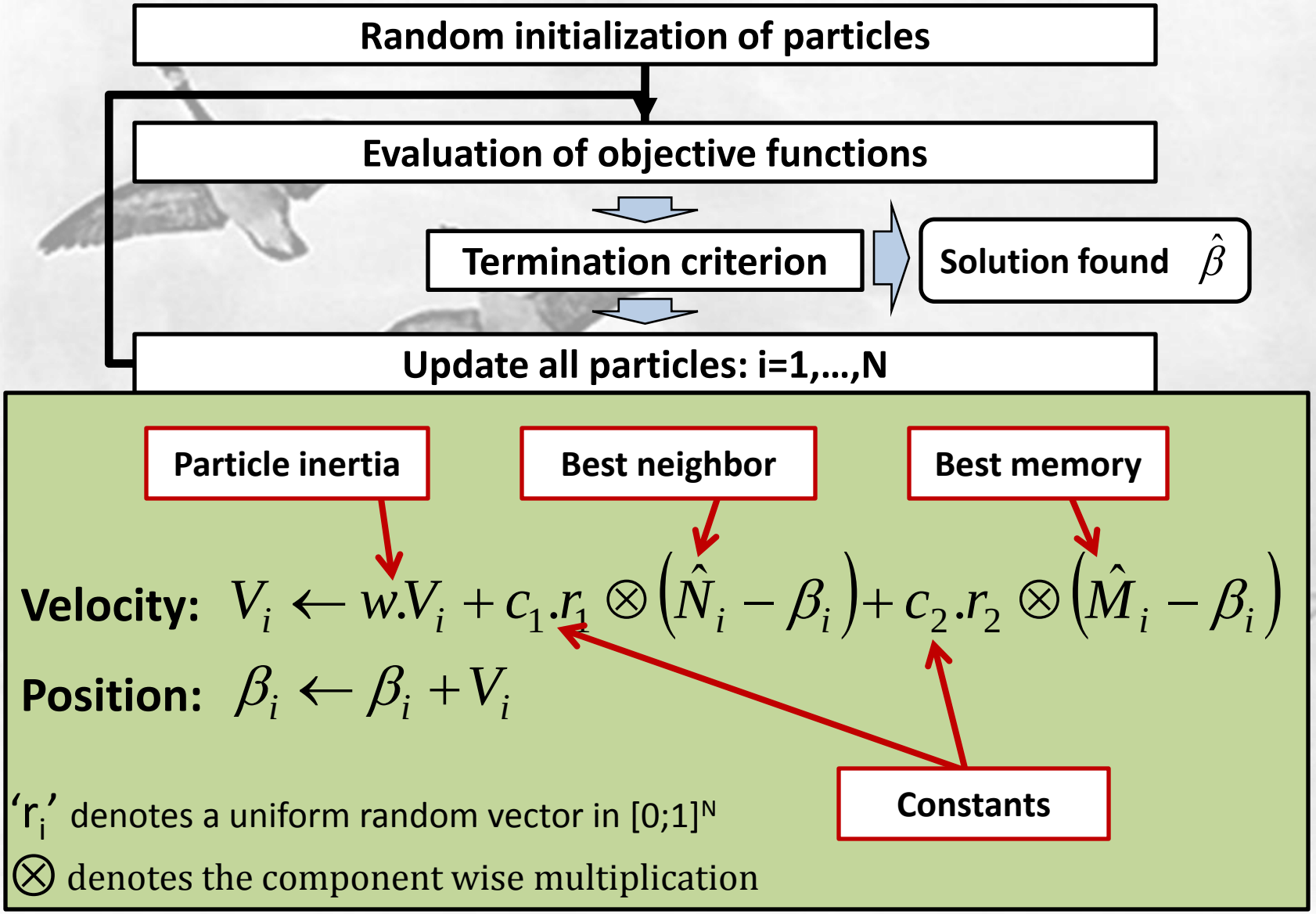
= Best experience
Best memory

-Best performance $J\left(\hat{M}\right)$

-Informants 3 and 5

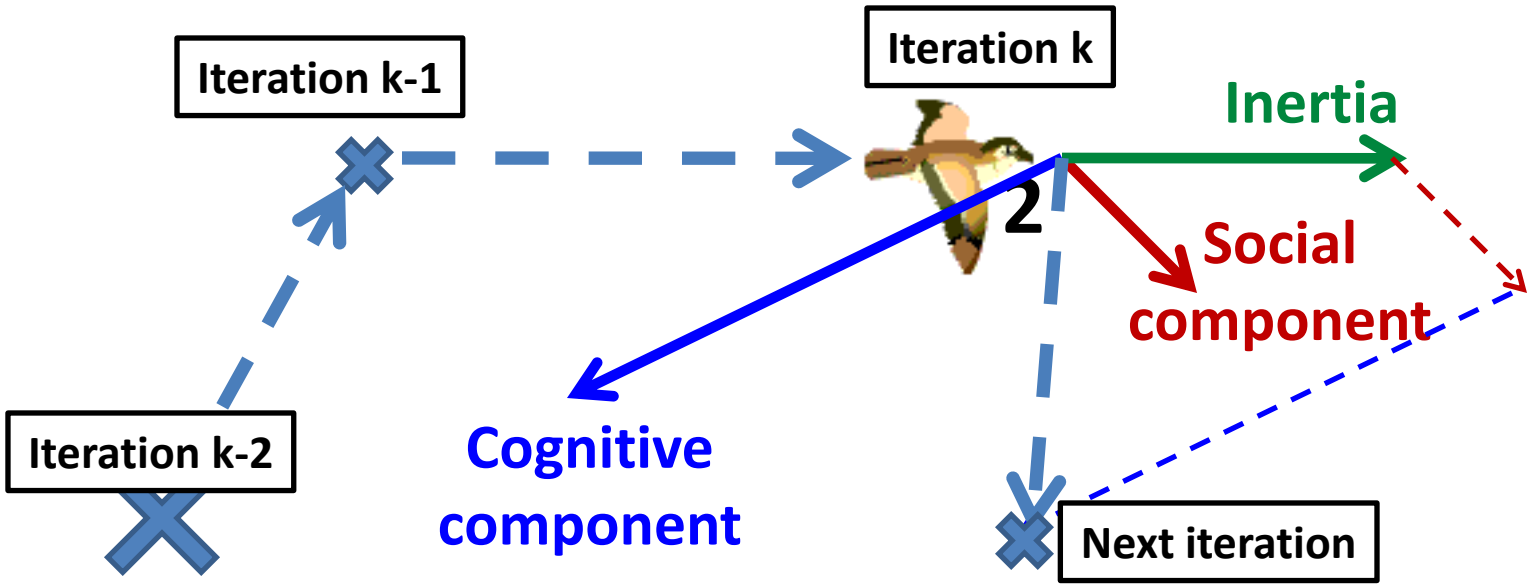
= Neighbors

The algorithm



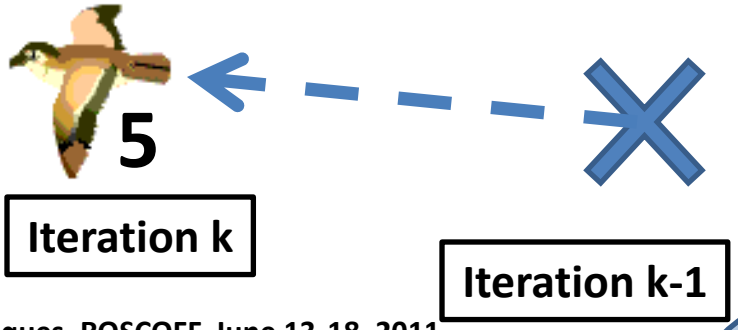
Hard to understand ? Draw a picture.

$$V_i \leftarrow w.V_i + c_1.r_1 \otimes \left(\hat{N}_i - \beta_i\right) + c_2.r_2 \otimes \left(\hat{M}_i - \beta_i\right)$$
$$\beta_i \leftarrow \beta_i + V_i$$



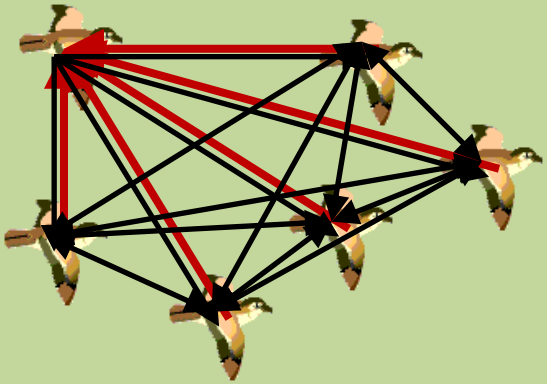
Note: Effect of component wise multiplication

In fact, cognitive and social components are not necessarily oriented toward the best memory and the best neighbor respectively.

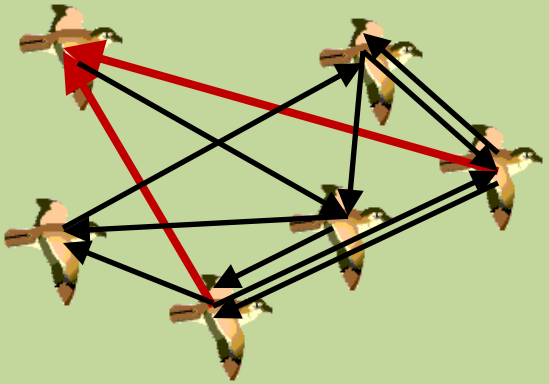


The neighborhood

All particles are neighbors

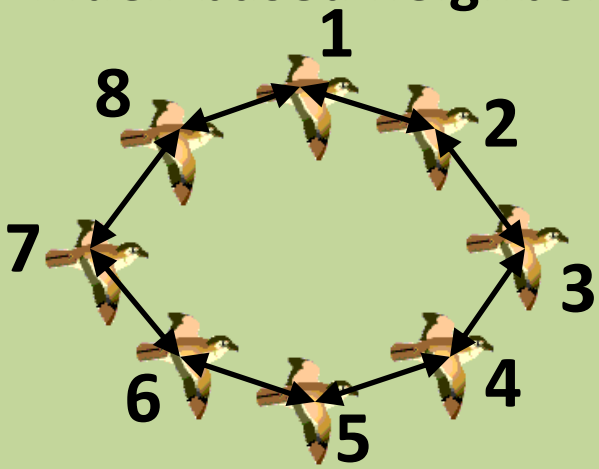


Random neighbors

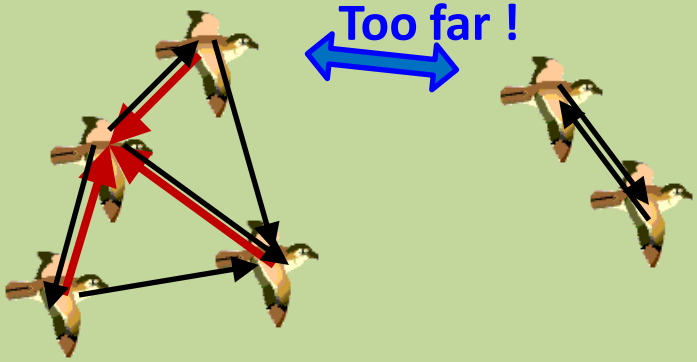


Update each iteration !

Index-based neighbors



Geographical neighborhood



Change each iteration !

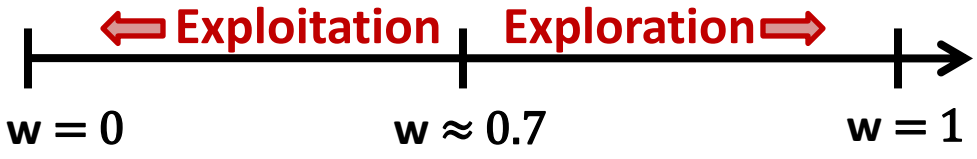
Algorithm adjustments

Population size = 20 particles

Neighborhood type = random or ring (index-based)

Neighborhood size = 3 or 4 particles

Particle inertia (≈friction)



Velocity: $V_i \leftarrow w.V_i + c_1.r_1 \otimes \left(\hat{N}_i - \beta_i\right) + c_2.r_2 \otimes \left(\hat{M}_i - \beta_i\right)$

Position: $\beta_i \leftarrow \beta_i + V_i$

Velocity

Often clamped to avoid explosion:

$-V_{max} \leq V_i \leq V_{max}$

Acceleration coefficients ($c_1 = c_2$)

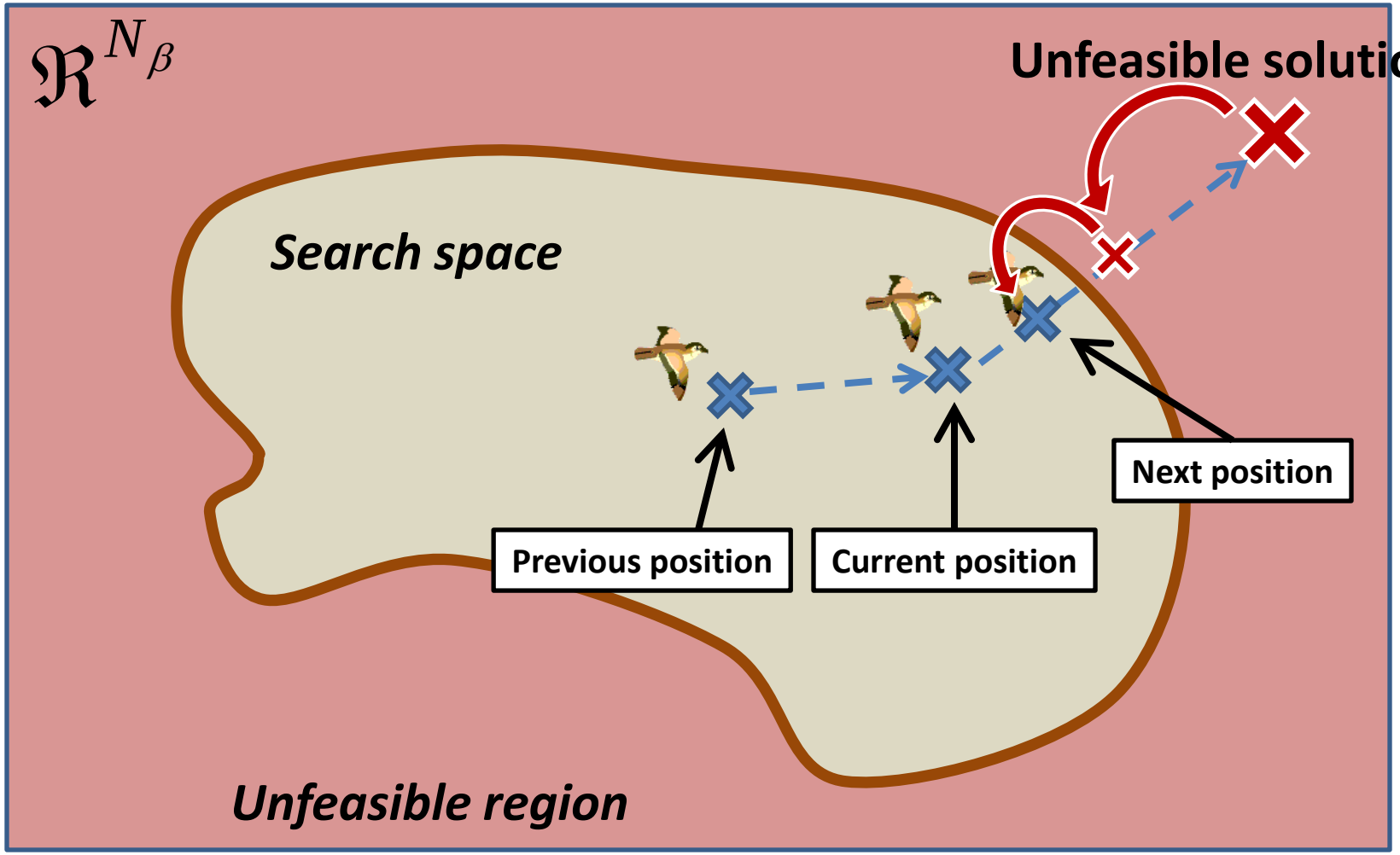
Not independent of inertia

Same compromise exploitation/exploration

c ≈ 1.5

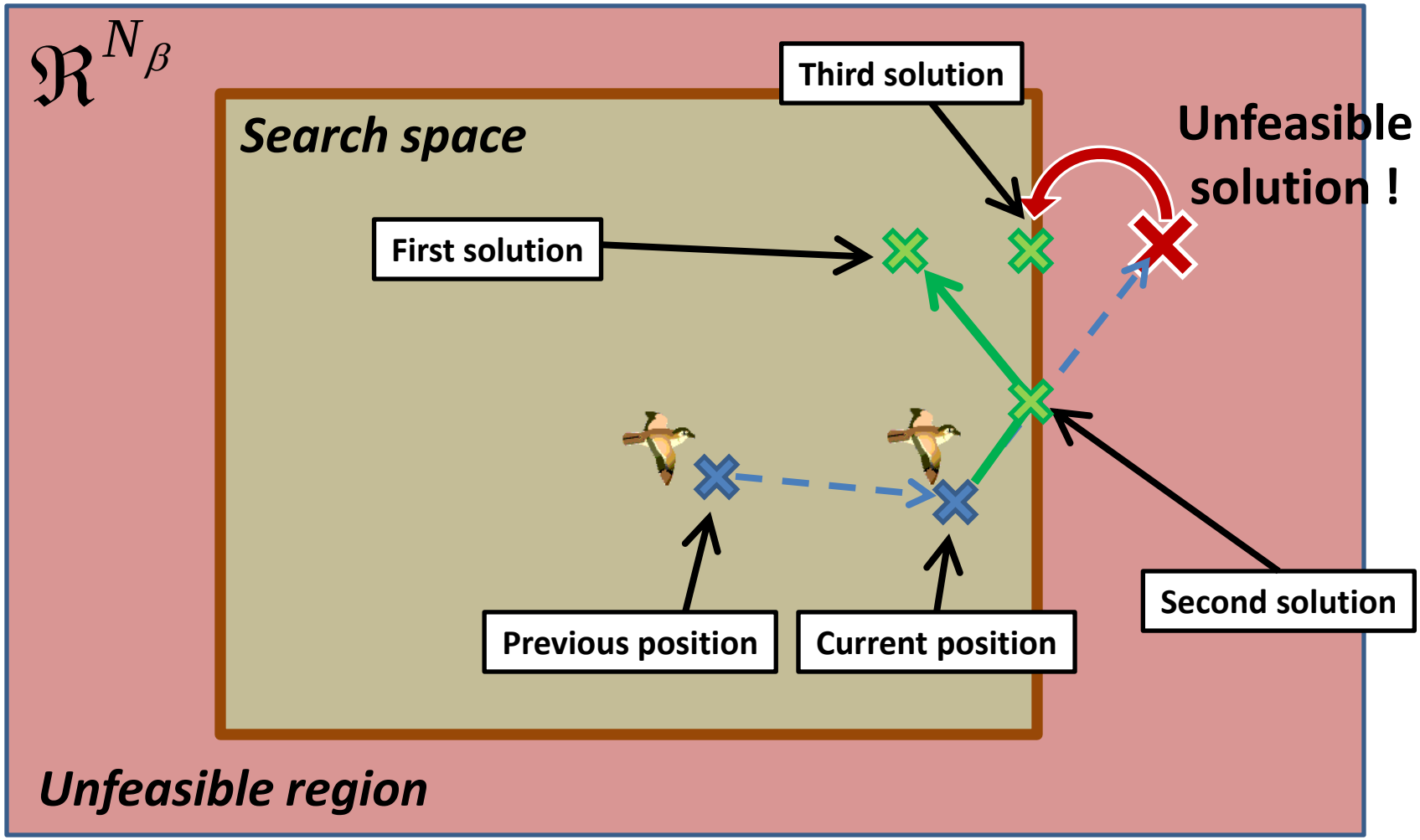
Handling hard constraints (1/2)

A method for the general case



Handling hard constraints (1/2)

In our case, simple boundary constraints



Variants

Constriction coefficient (Clerc, Eberhart, Kennedy):

$$V_i \leftarrow \chi.\left[V_i + \frac{\varphi}{2}.r_1 \otimes \left(\hat{N}_i - \beta_i\right) + \frac{\varphi}{2}.r_2 \otimes \left(\hat{M}_i - \beta_i\right)\right]$$
$$\beta_i \leftarrow \beta_i + V_i$$

with $\chi. = \dfrac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}}$

Time dependant:

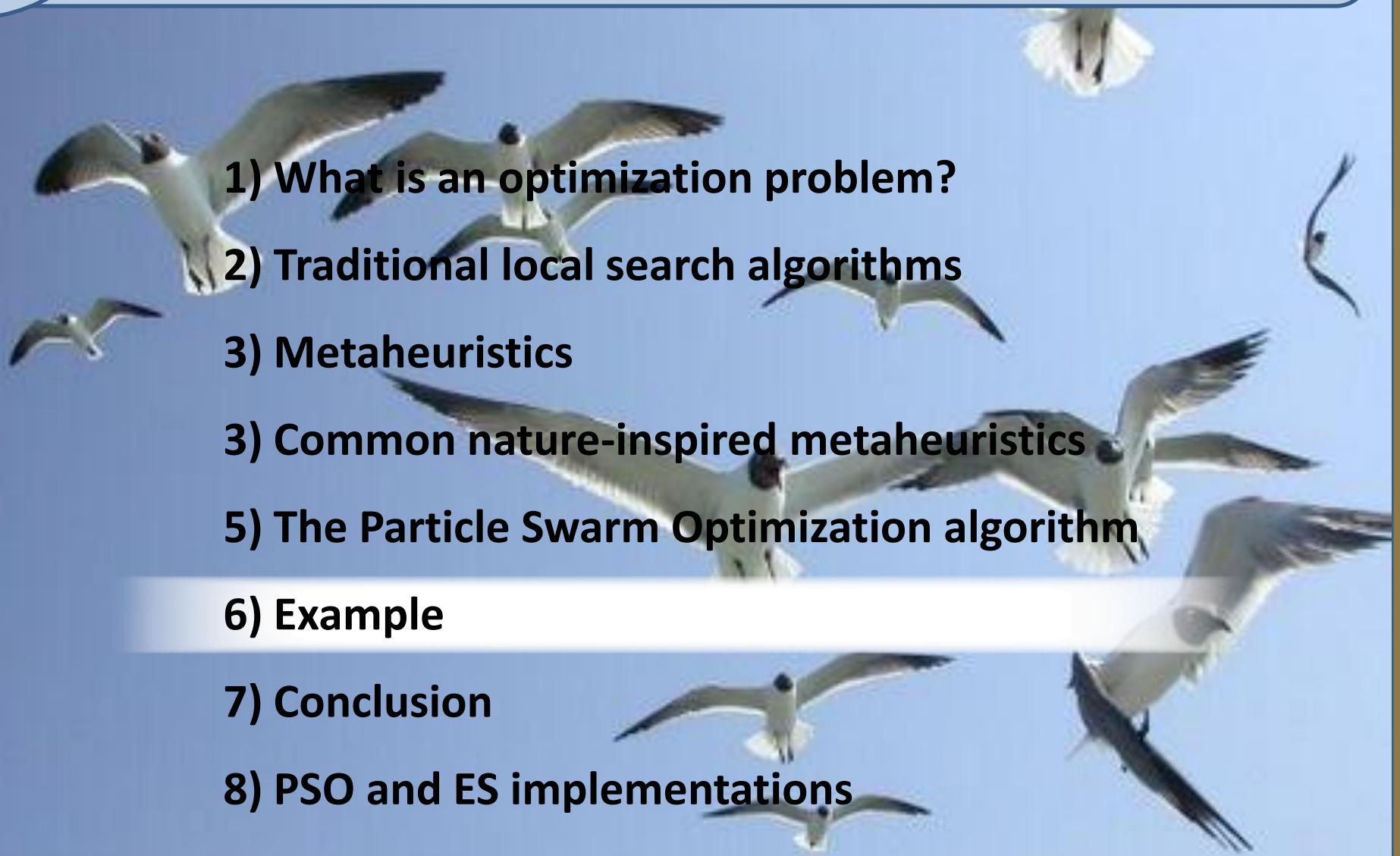
$$V_i \leftarrow w(t).V_i + c_1.r_1 \otimes \left(\hat{N}_i - \beta_i\right) + c_2.r_2 \otimes \left(\hat{M}_i - \beta_i\right)$$
$$\beta_i \leftarrow \beta_i + V_i$$

Fully informed (Mendes, Kennedy, Neves) :

$$V_i \leftarrow \chi.\left[V_i + \frac{\varphi}{K_i}\sum_{k}^{K_i} r \otimes \left(N_k - \beta_i\right)\right]$$
$$\beta_i \leftarrow \beta_i + V_i$$

With K_i the neighborhood size

Outline



1) What is an optimization problem?
2) Traditional local search algorithms
3) Metaheuristics
3) Common nature-inspired metaheuristics
5) The Particle Swarm Optimization algorithm
6) Example
7) Conclusion
8) PSO and ES implementations

Thermal parameter estimation

Context

-Anisotropic material

-Temperature dependant parameters

$$\begin{cases} \lambda_x(T) = \lambda_{x0} + \lambda_{x1}.T \\ \lambda_y(T) = \lambda_{y0} + \lambda_{y1}.T \\ C(T) = C_0 + C_1.T \end{cases} \Rightarrow$$

Unknown parameters

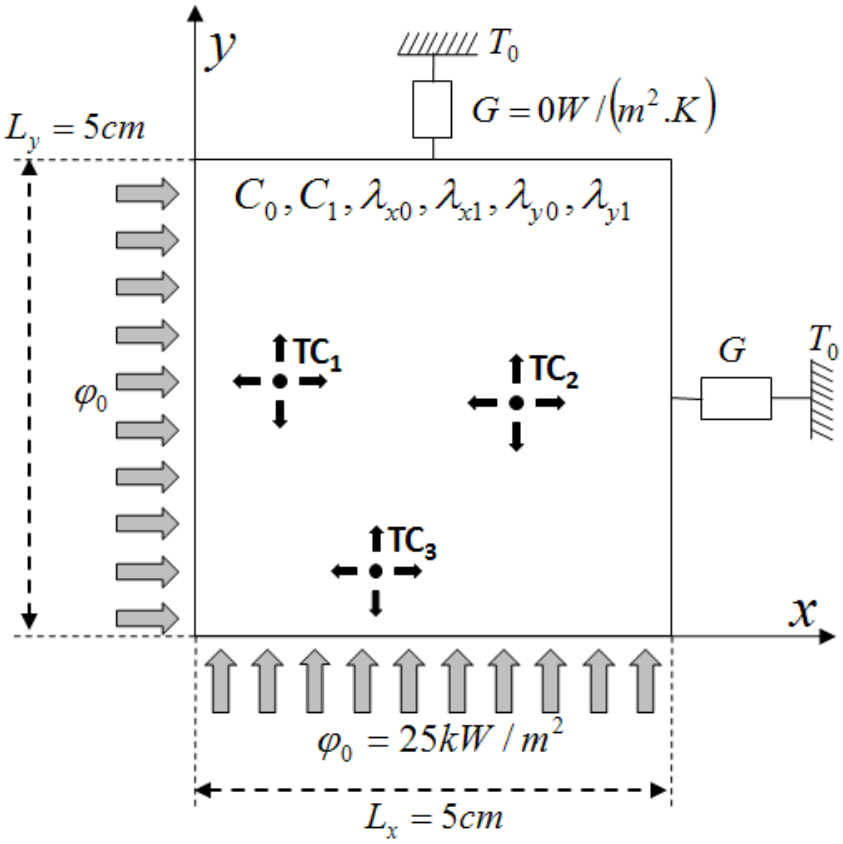
$$\beta = \left(\lambda_{x0}, \lambda_{x1}, \lambda_{y0}, \lambda_{y1}, C_0, C_1\right)$$

Objective

Estimate β with the precision

The numerical experiment

- 2D-sample of an orthotropic media (5x5cm)
- Insulated on the right and top boundaries
- Heated on the left and bottom with a constant heat flux



Jarny, An inverse analysis to estimate linearly temperature dependent, 1995

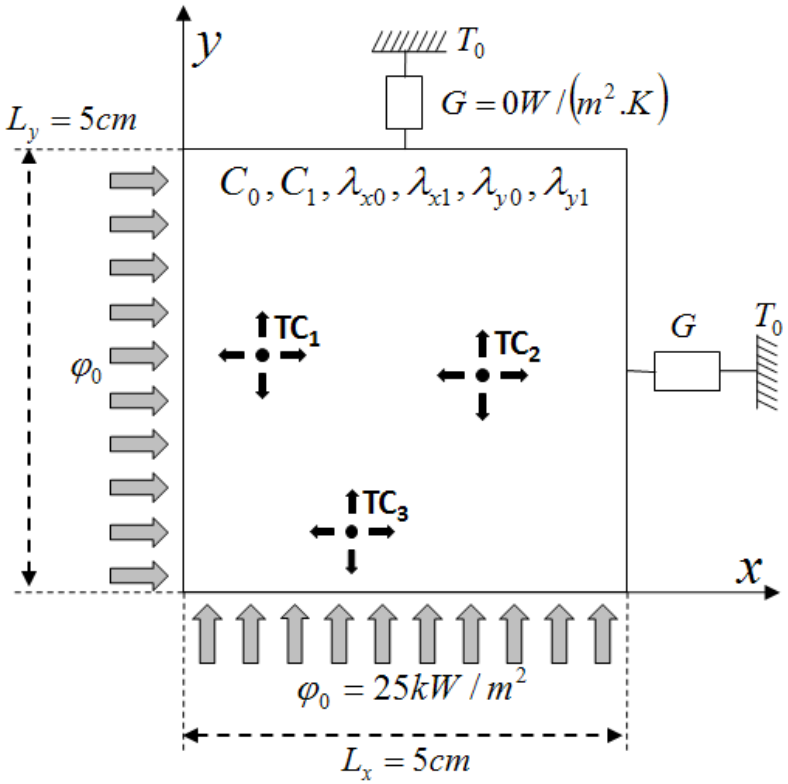
Optimal experiment design (1/3)

Uncertainty of estimated parameters:

-Effect of measurement noise: $\mathrm{cov}_\beta(\sigma_m) = \sigma_m^2 \left[X_\beta^T . X_\beta\right]^{-1}$

-Effect of uncertain parameters:

$$\mathrm{cov}_\beta(\sigma_\gamma) = \left[X_\beta^T . X_\beta\right]^{-1} X_\beta^T X_\gamma \, \mathrm{cov}(e_\gamma) X_\gamma^T X_\beta \left[X_\beta^T . X_\beta\right]^{-1}$$



Optimization problem

Where should I put thermocouple to estimate β with the best precision ?

Optimality criterion

$$J(P) = \sum_k \left(\frac{\sigma_{\beta_k}}{\beta_k} \right)^2$$

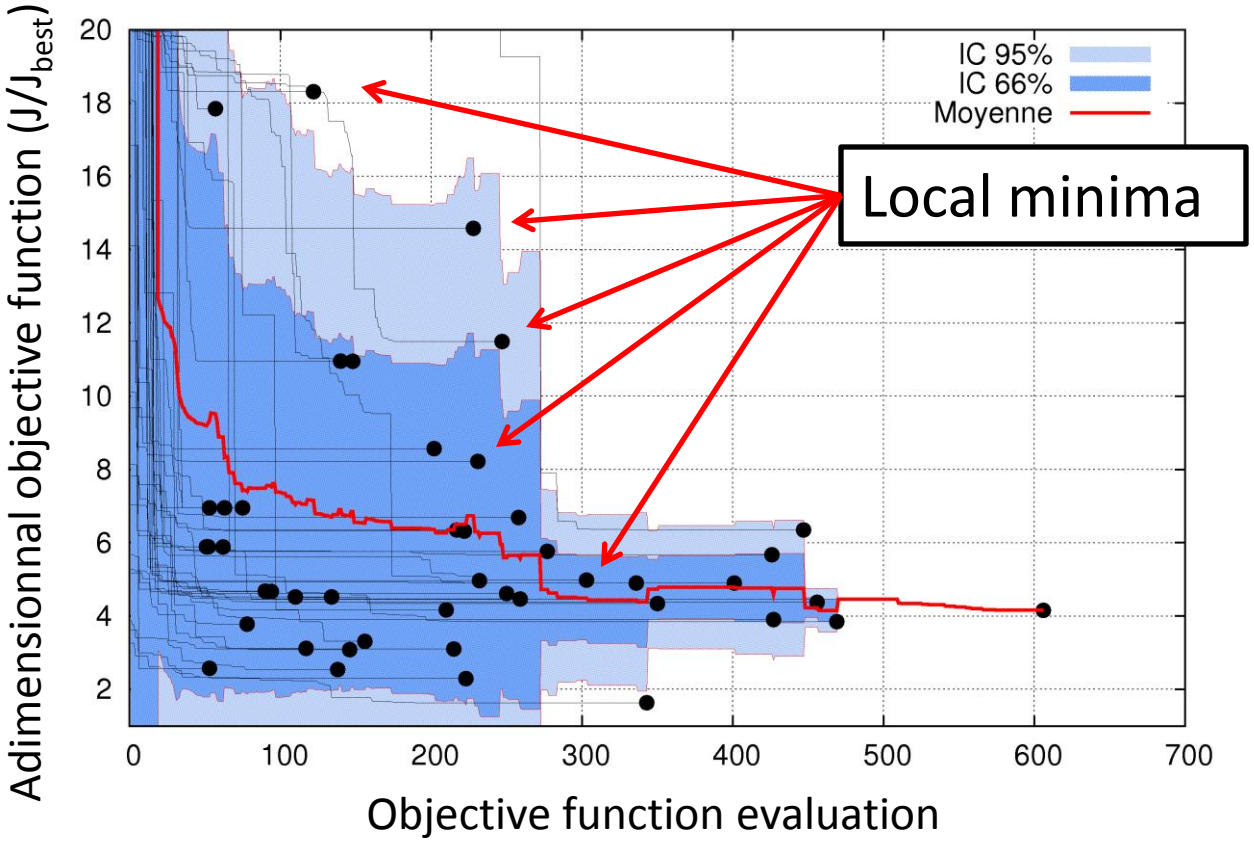
With P the adjustable parameters

Solution

$$\hat{P} = \arg\left[\min_P \left(J(P)\right)\right]$$

Optimal experiment design (2/3)

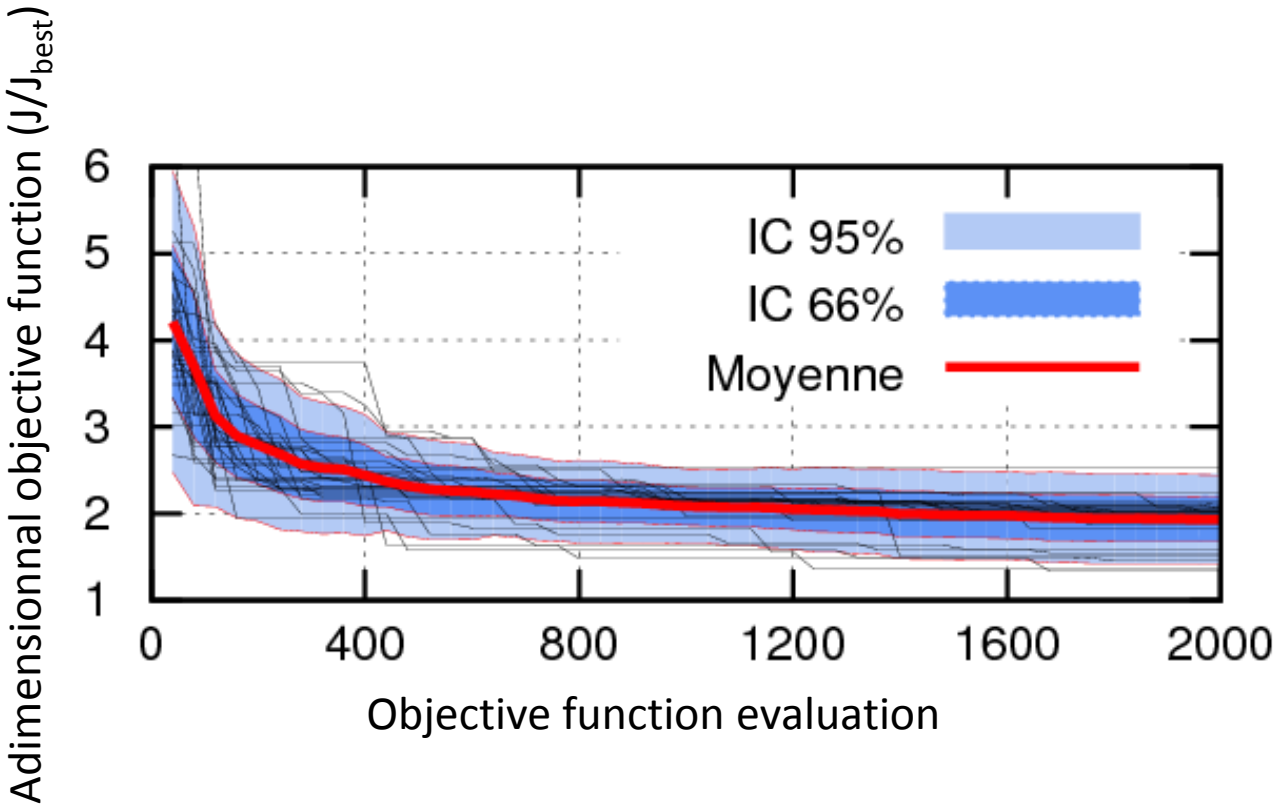
Minimization using gradient-based methods



Several runs give several very different results

Optimal experiment design (3/3)

Minimization using PSO or EA

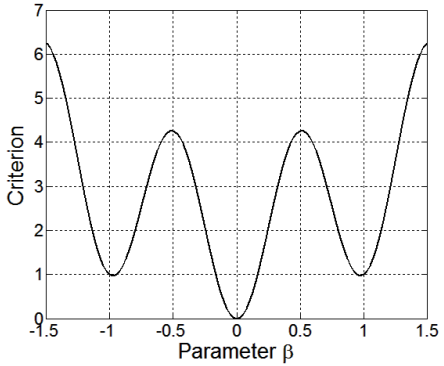


Several runs give always a good result

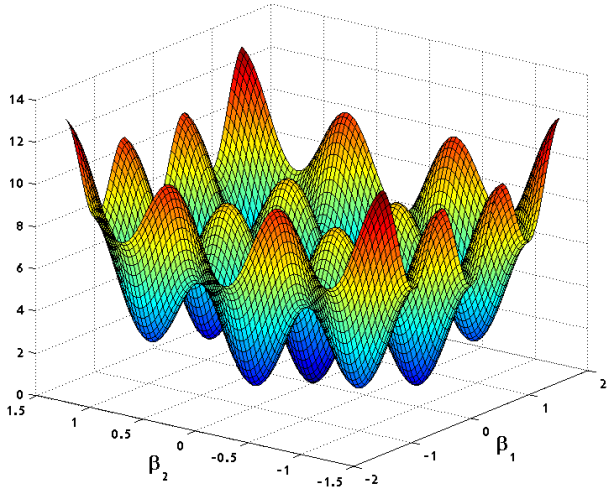
Rastrigin function (1/2)

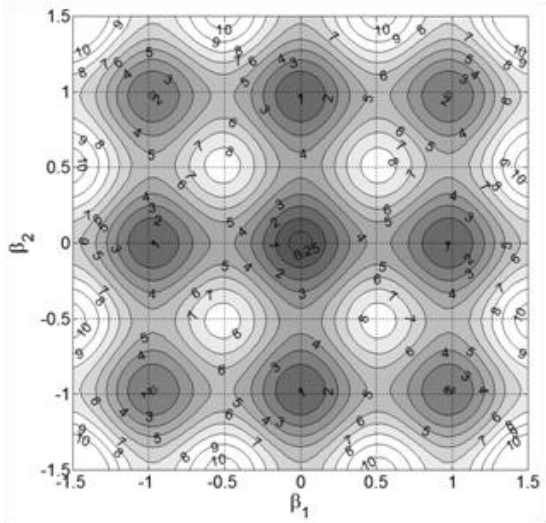
Definition: $J(\beta) = A\,N + \sum_{i=1}^{N}\left[\beta_i^2 - A\cos(2\pi\beta_i)\right]$

With N the dimension and A a constant.



1d-Rastrigin function

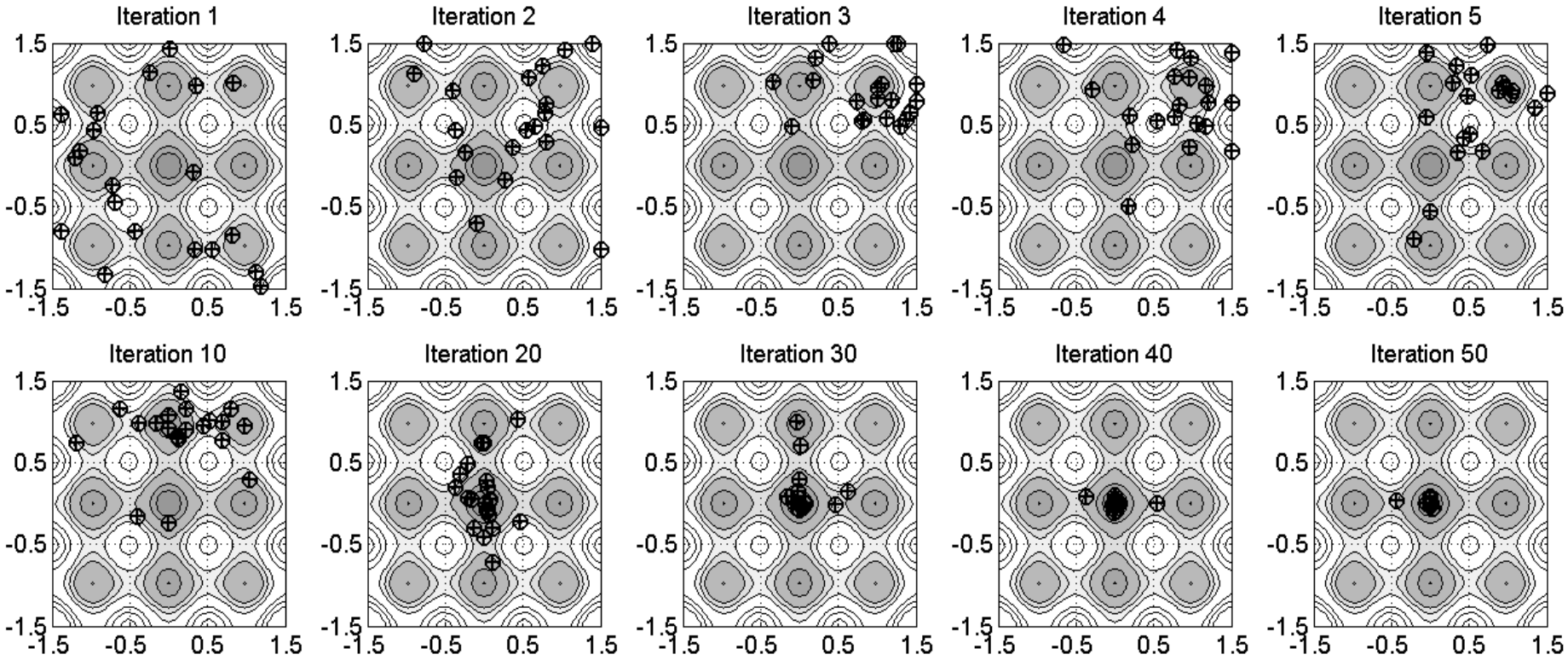




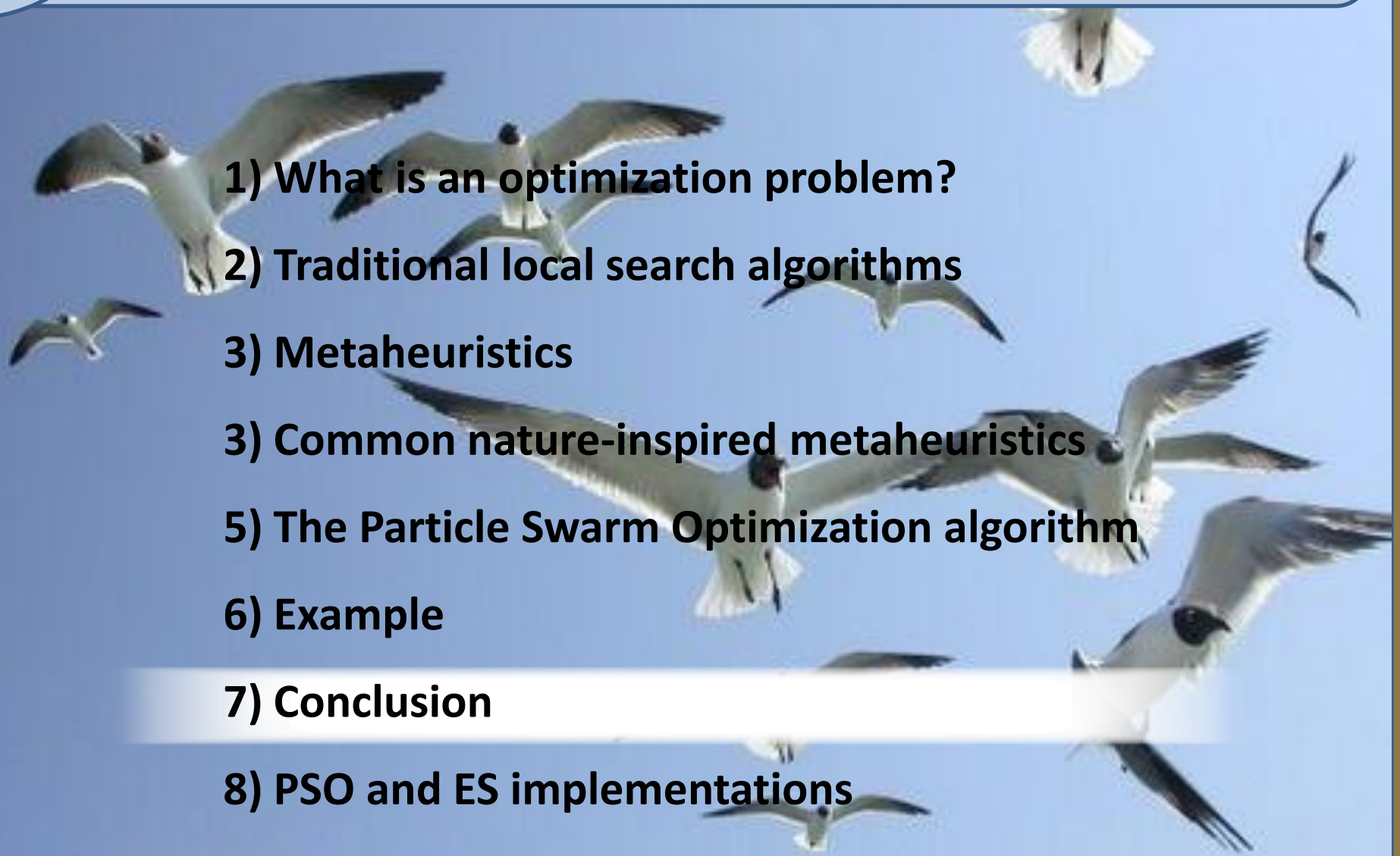
2d-Rastrigin function

Rastrigin function (2/2)

Particle positions with respect to the iteration



Outline



1) What is an optimization problem?
2) Traditional local search algorithms
3) Metaheuristics
3) Common nature-inspired metaheuristics
5) The Particle Swarm Optimization algorithm
6) Example
7) Conclusion
8) PSO and ES implementations

Conclusion (1/3)

Metaheuristics

- are designed to solve hard optimization problem
- try to find a compromise exploration/exploitation
  (somewhere between pure random and deterministic search)
- find good solutions in reasonable time ("approximation algorithms")
- are often natured-inspired, mostly population-based, nearly always stochastic
- are often global, derivative-free

Remarks

- Very few assumptions on the objective function shape
- The more information you provide, the more efficient the algorithm will be
- One never knows if the current best solution is the optimal one
- The choice of a termination criterion is not straightforward

Conclusion (2/3)

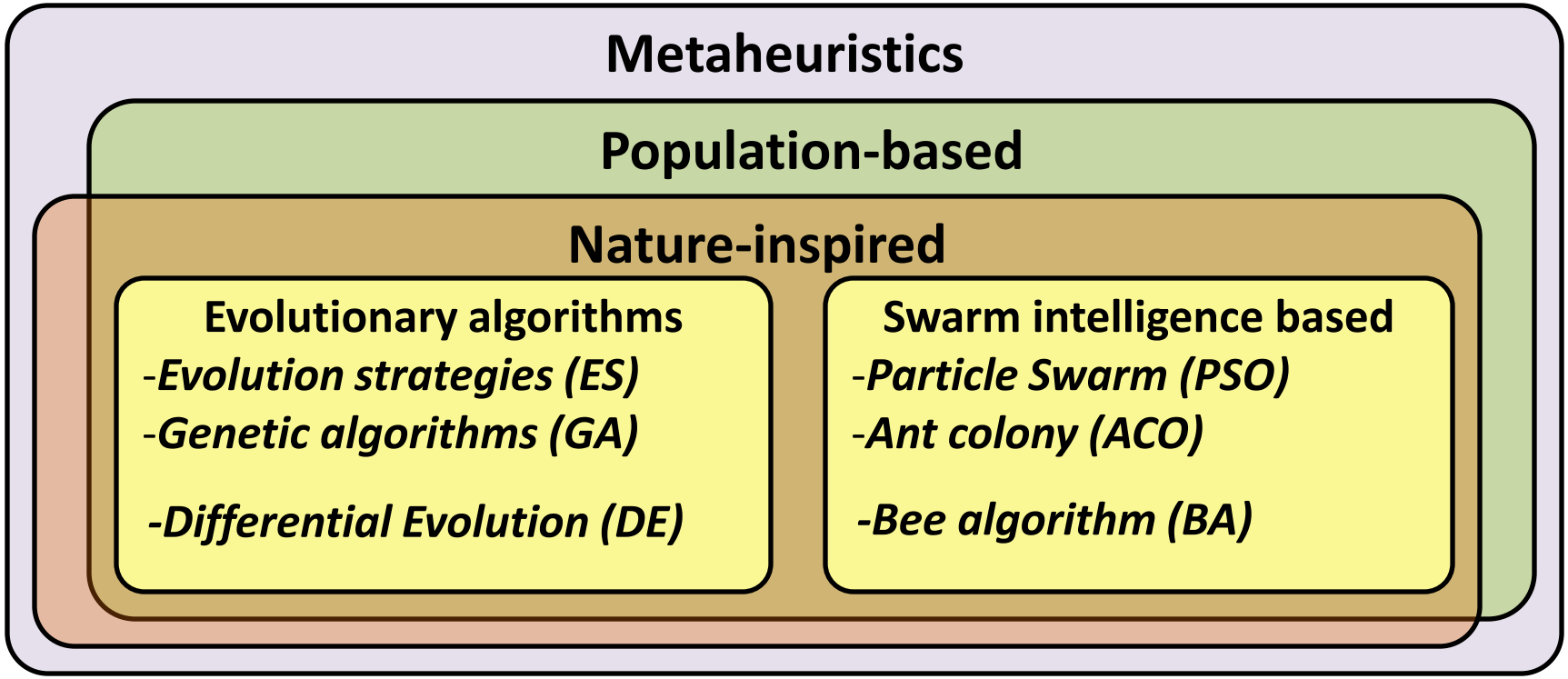
Good metaheuristic ?

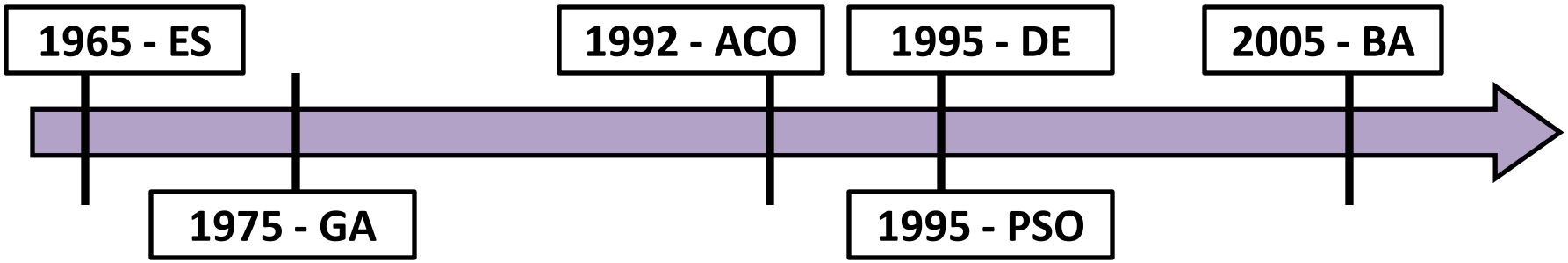
- Not (too much) problem specific
- Easy to apply, easy to implement
- With only few parameters

Best metaheuristic ?

- Such a thing doesn't exist
- Compared on all objective functions, all algorithms are equivalent ("No free lunch theorem")
- A algorithm can be more efficient on a specific set of objective functions
- Hybrid methods (Ex: PSO + Gradient)

Conclusion (3/3)





References

Simplex

- C. Porte, Méthodes directes d'optimisation – Méthodes à une variable et Simplex, Techniques de l'Ingénieur

Global optimization

- Weise, Global optimization algorithms, e-book, 2006
- J. Dréo, A. pétrowski, P. Siarry, E. Taillard. Métaheuristiques pour l'optimisation difficile. Paris : Editions Eyrolles, 2003 pp.70, 154. 2-212-11368-4

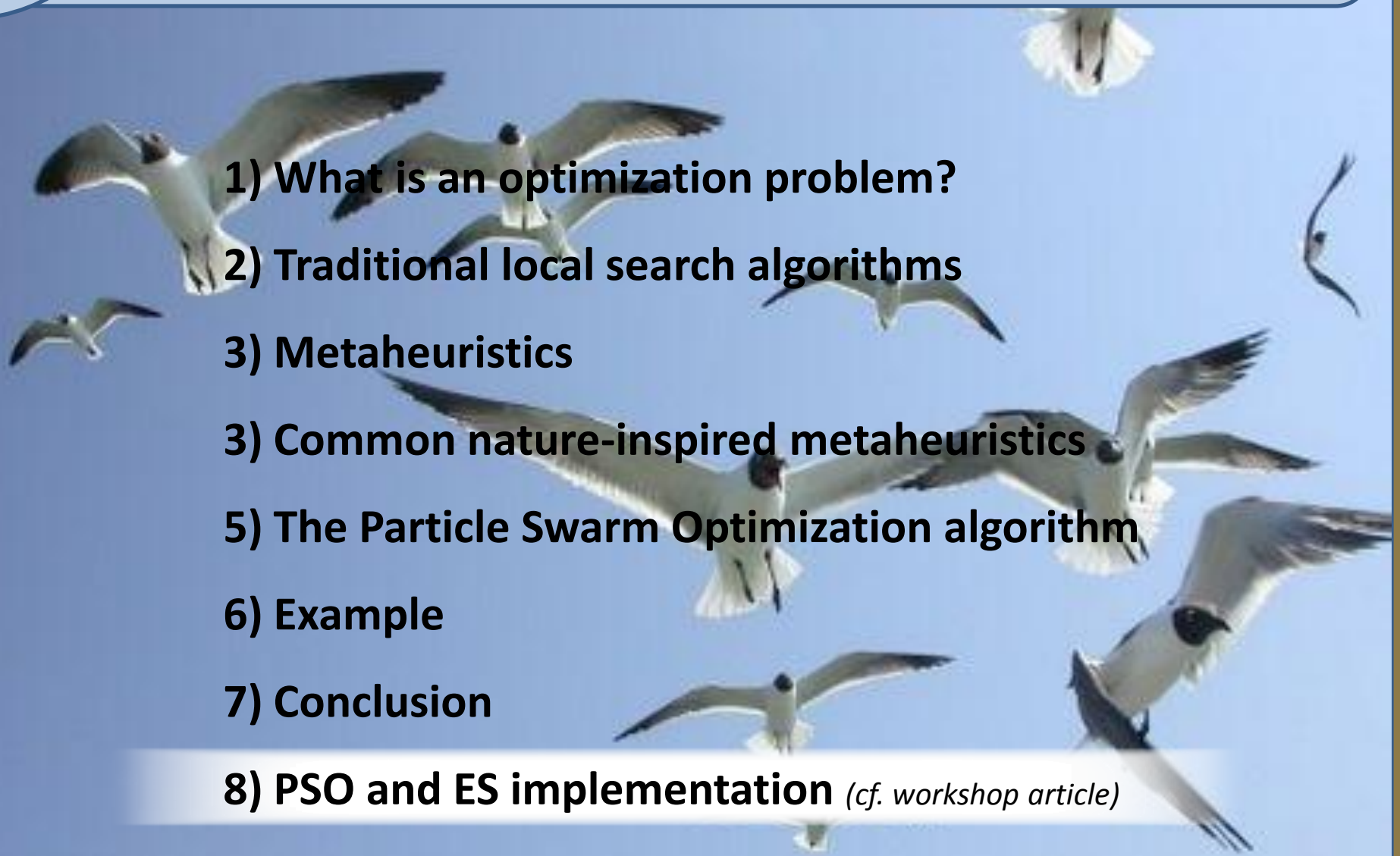
Evolutionary algorithm

- Renders, J-M. Algorithmes génétiques et réseaux de neurones. Paris : Editions Hermès, 1995. 2-86601-467-7

Swarm Intelligence based algorithm

- F. Van Den Bergh, An analysis of Particle Swarm Optimizers, PhD, University of Pretoria, November 2001
- J. Kennedy, Swarm Intelligence, Chapter 6, ook
- D. Merkle, M. Middendorf, Swarm Intelligence, Chapter 14, book
- F. Van Den Bergh, A.P. Engelbrecht, A study of particle swarm optimization particle trajectories, Information Sciences 176 (2006) 937-971
- C. Blum, Ant colony optimization: Introduction and recent trends, Physics of Life Reviews 2 (2005) 353-373
- G. Gilchev, I.C. Parmee, The Ant Colony Metaphor for searching Continuous Design Spaces, Proceedings of the AISB Workshop on Evolutionary Computation, April 3-4, 1995

Outline



1) What is an optimization problem?
2) Traditional local search algorithms
3) Metaheuristics
3) Common nature-inspired metaheuristics
5) The Particle Swarm Optimization algorithm
6) Example
7) Conclusion
8) PSO and ES implementation *(cf. workshop article)*