Tutorial session

Emerging metaheuristics

Dr.ir. J.L. Rueda Torres

26th June 2015

Learning objectives (LOs)

At the end of the tutorial the attendees will be able to:

- **LO1:** Describe the basics concerning the definition of an optimization problem

- **LO2:** Describe the rationale behind metaheuristics

- **LO3:** Apply a metaheuristic algorithm to a given optimization problem

Outline

1. Problem statement
Related to LO1

2. Metaheuristics: Basic theory
Related to LO2

3. Mean-variance mapping optimization algorithm (MVMO)
Related to LO2

4. CEC2015 expensive problems
Related to LO1, LO3

1 Problem statement

Generic formulation

Minimize/maximize

Objective function $\left\{ OF = \sum_{r=1}^{p} w_r \cdot f_r\left(\mathbf{x}\right) \right.$ (1)

subject to

Constraints $\begin{cases} g_i(\mathbf{x}) \leq 0, & i = 1, \ldots, m \quad (2) \\ h_j(\mathbf{x}) = 0, & j = m+1, \ldots, n \quad (3) \end{cases}$

considering the search space given by

Bounds $\left\{ x_k^{\min} \leq x_k \leq x_k^{\max}, \quad k = 1, \ldots, \mathrm{D} \right.$ (4)

Solution vector $\left\{ \mathbf{x} = \left[x_1, x_2, \ldots, x_{\mathrm{D}} \right] \right.$ (5)

1 Problem statement

Types of optimization problems

Problem type

- Single objective/multi-objective
- Constrained/unconstrained
- Continuous (real numbers).
- Combinatorial (countable items - integer numbers).
- Mixed-integer.

Problem Complexities

- High dimensional search space
- Non-convex, discontinuous search landscape
- Multimodality
- High numerical accuracy, high nonlinearity.
- Lack of analytical expressions
- High computational burden

1 Problem statement

Optimization benchmarks

Test beds with synthetic problems can be found at

http://www.ntu.edu.sg/home/epnsugan/
(under EA Benchmarks / CEC Competitions)

- These problems are useful for the design of metaheuristics
- Performance of metaheuristic algorithms compared against state-of-the-art algorithms

Example: Composition functions

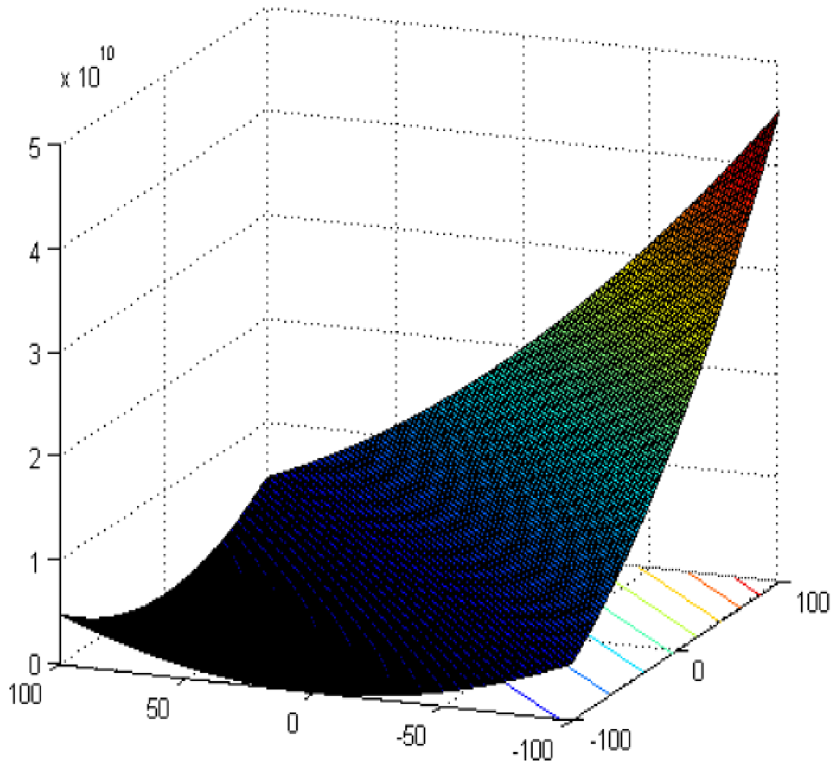
$$CF(x) = \sum_{i=1}^{n} \left[w_i \left| \left(c \frac{M_i f_i}{\lambda_i \left| f_i^{\max} \right|} \left(x - o_i - o_i^{old} \right) + b_i \right) \right| \right] + a \qquad (6)$$

1 Problem statement

Some synthetic problems:

(CEC'15 expensive optimization competition)
Source: http://www.ntu.edu.sg/home/epnsugan/

Rotated Bent Cigar Function



Properties:

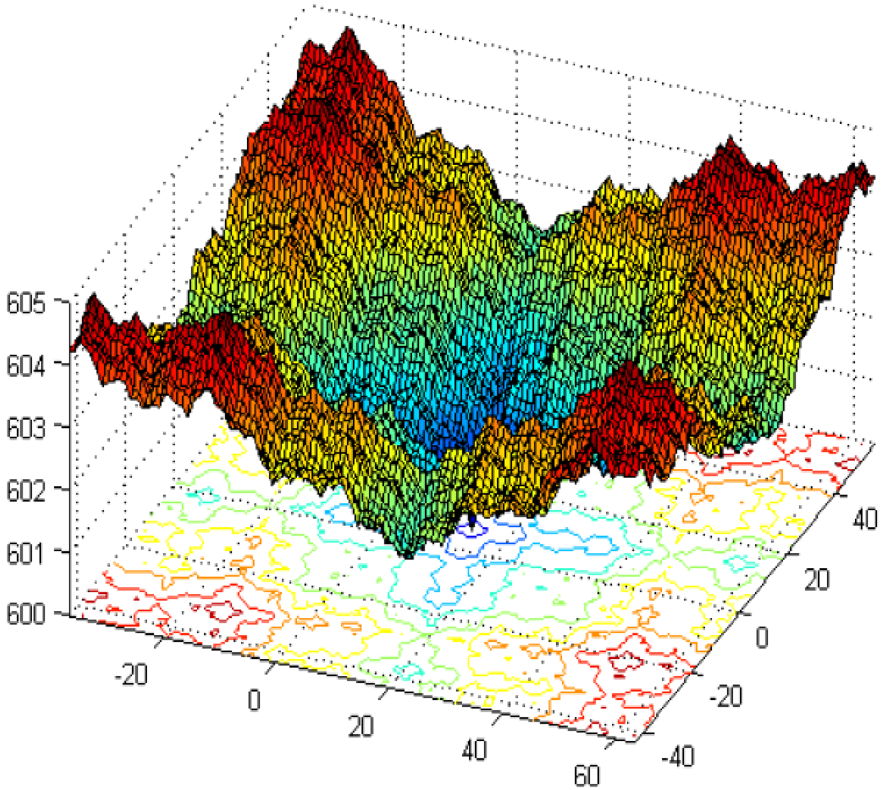
- Unimodal
- Non-separable
- Smooth but narrow ridge

1 Problem statement

Some synthetic problems:

(CEC’15 expensive optimization competition)
Source: http://www.ntu.edu.sg/home/epnsugan/

Shifted and Rotated Weierstrass Function



Properties:

- Multi-modal
- Non-separable
- Continuous but differentiable only on a set of points

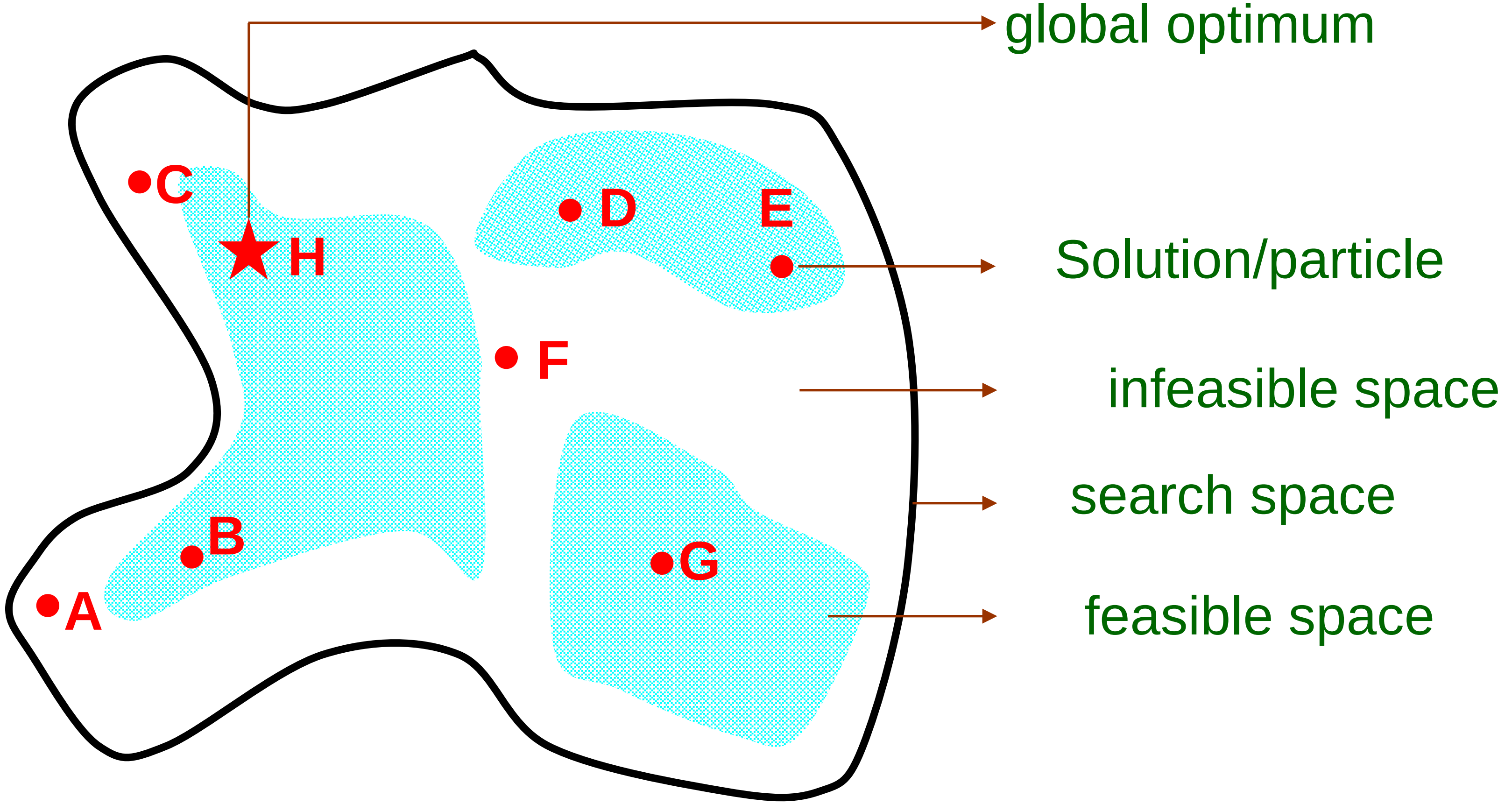
1 Problem statement

Types of solutions?

- A feasible solution satisfies all constraints.
- An optimal solution is feasible and provides the best *OF* value.
- A near-optimal solution is feasible and nearly as good as the best.

1 Problem statement

Types of solutions?



1 Problem statement

Some Parlance……

- Solution/Particle/individual: Vector of decision variables.
- Encoding: Representing the decision variables (e.g. binary coding).
- Swarm/Population: Set of candidate solutions.
- Neighborhood: Nearby solutions (in the encoding/solution space)
- Search: Constructing/improving solutions toward obtaining the optimum or near-optimum.
- Function evaluation: Determining the solution’s objective function value and feasibility (constraint fulfillment).
- Offspring creation/moving: Generating new solutions based on current best achieved solutions.

2 Metaheuristics: Basic theory

What are heuristics?

- From old Greek word 'heuriskein': the art of finding new strategies for problem solving, learning, and discovery.
- Heuristic algorithms are approximate and usually non-deterministic. They do not guarantee optimality but usually find 'good enough' solutions in reasonable time.
- Fundamental challenge: to find 'suitable shortcuts' to ease search knowledge management and decision making to speed up the process of finding a satisfactory solution.

2 Metaheuristics: Basic theory

Basic heuristic approaches

Constructive: Evaluate partial solutions. Build a solution piece by piece and terminate when a whole solution is constructed, e.g. greedy algorithms.

Improvement: Evaluate complete solutions. Sophisticated guiding scheme to explore efficiently the search space, e.g. Metaheuristics, hybrid methods.

2 Metaheuristics: Basic theory

What are metaheuristics?

- 'Meta' – Greek for 'upper layer methodology'.
- Use of high level scheme to guide the search process while subordinating and combining strategically different procedures derived from classical heuristics, artificial intelligence, or nature-inspired evolutionary techniques.
- Metaheuristic algorithms range from simple local search to complex learning processes.

2 Metaheuristics: Basic theory

What are metaheuristics?

- Main challenge: Tradeoff between exploration (diversification) and exploitation (intensification) to improve the rate of convergence and to achieve global optimality.

- Exploration: Generate diverse solutions so as to explore the search space on a global scale.

- Exploitation: Focus the search in a local region knowing that a current good solution is within this region.

2 Metaheuristics: Basic theory

Why to use metaheuristics?

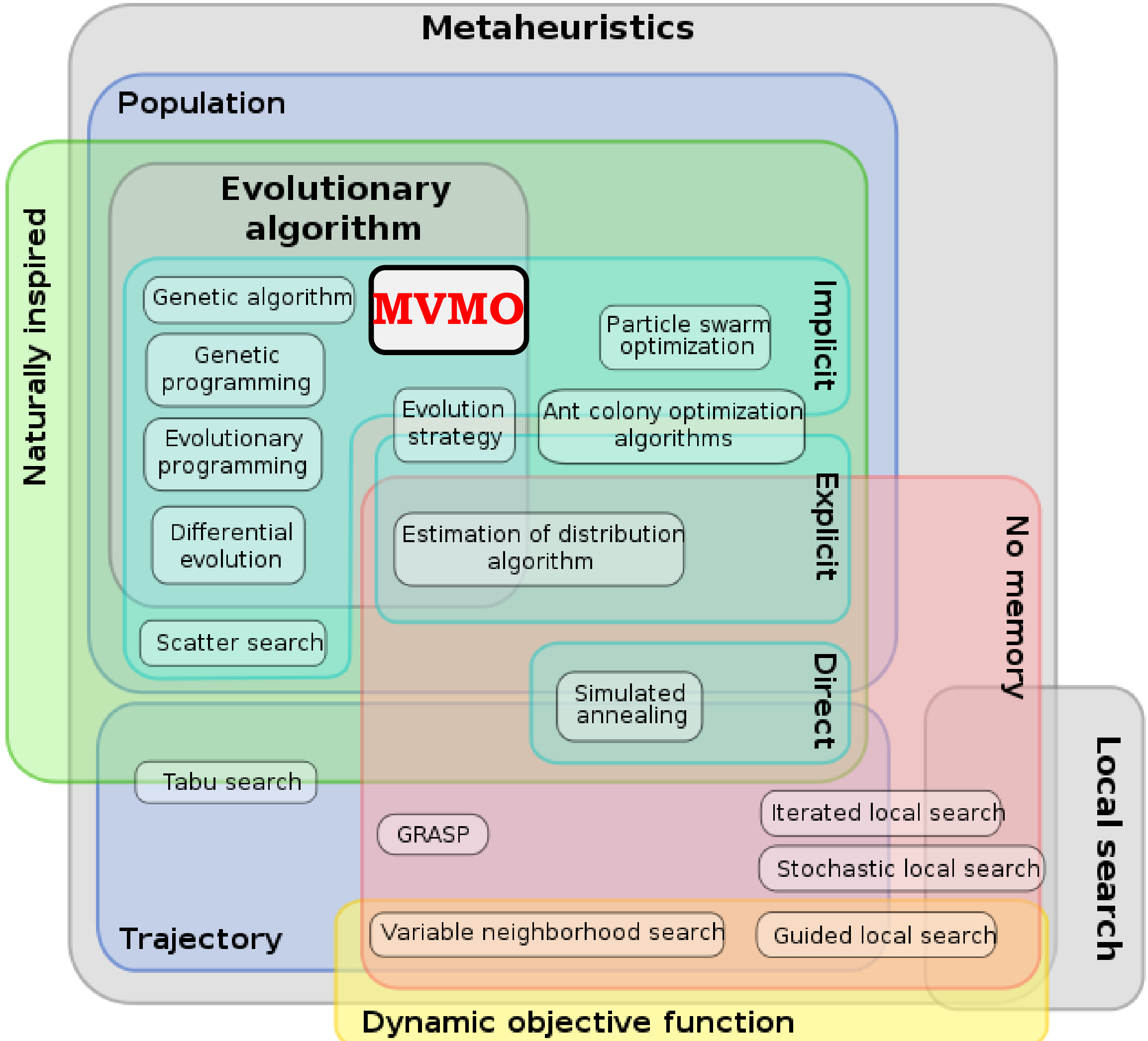
Motivated by underlying problem complexity:

- High dimensional search space (mixed-integer).
- Non-explicitly defined objective functions, differentiation is not possible.
- Nonlinear, non-convex, discontinue and multimodal landscape.

➡ **Classical optimization tools cannot be applied**

2 Metaheuristics: Basic theory

Classification



Source: Wikipedia

2 Metaheuristics: Basic theory

The pros and cons...

Advantages of metaheuristic techniques:

- Conceptual simplicity
- Easy adaptability due to open architecture
- Reduced human intervention

Potential drawbacks:

- Different solutions due to stochastic nature
- Optimality could not be guaranteed
- Sensitivity to initialization and parameter tuning

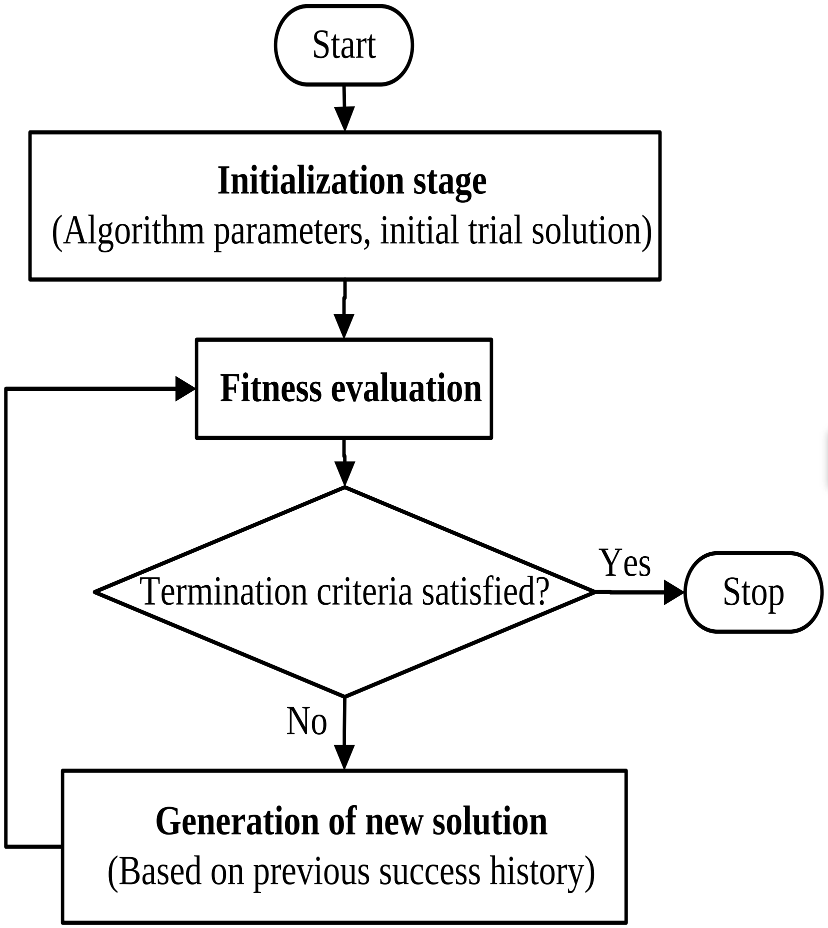
2 Metaheuristics: Basic theory

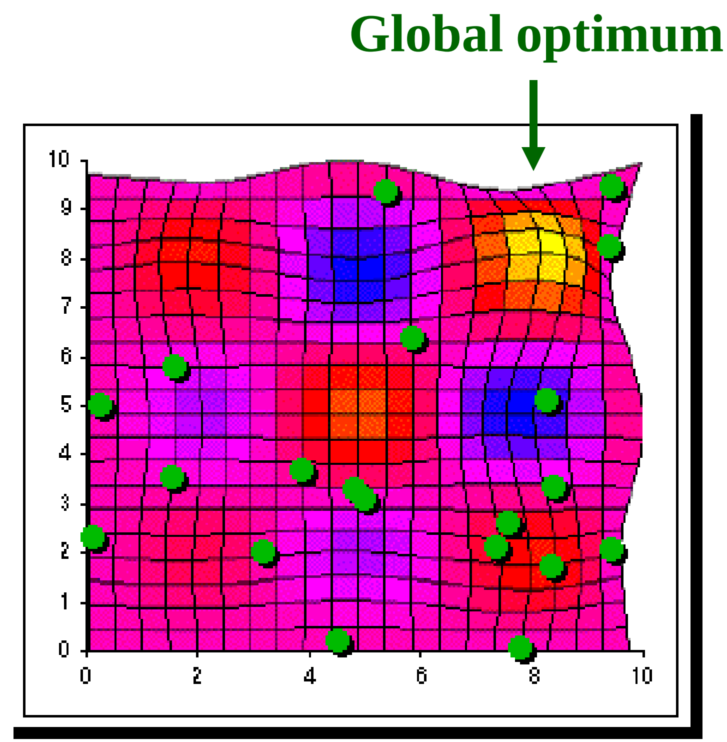
Today we focus on evolutionary algorithms (EAs)!

- Evolutionary computation was firstly introduced by I. Rechenberg. "Evolutionsstrategie - Optimierung technischer Systeme nach Prinzipien der biologischen Evolution" (PhD thesis, 1971).
- Mechanism inspired by biological evolution.
- It starts with a set of solution (population).
- The population evolves over time, with the fittest at each iteration (generation) contributing the most offspring to the next generation.
- This is motivated by a hope that the new population will be better than the old one.

2 Metaheuristics: Basic theory

A very simple evolutionary approach





2 Metaheuristics: Basic theory

Some important aspects to consider:

Adaptability

- The evolutionary mechanism of an EA can be adapted to a wide class of problems without major modifications

Randomness

- EAs include some component of randomness, e.g. initialization, selection, mutation, etc.
- A good EA performs well over a range of randomness measures.

2 Metaheuristics: Basic theory

Some important aspects to consider:

Communication

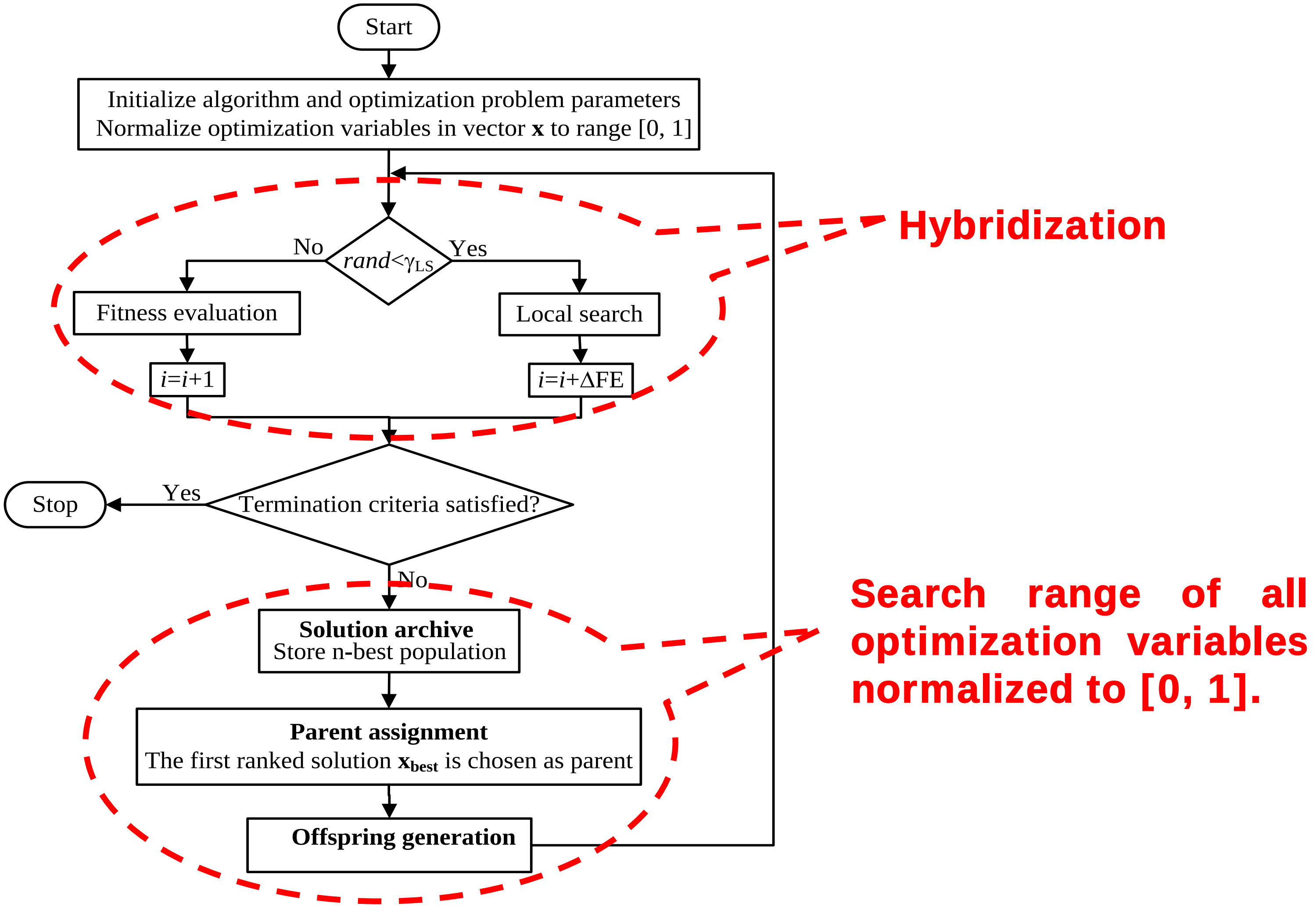
- Share of information between candidate solutions and learn from each other's successes and failures.
- Over iterations, the population of individuals evolves motivated by success measures

Balance exploration-exploitation

- Higher exploration → Higher randomness
- Higher exploitation → Lower randomness

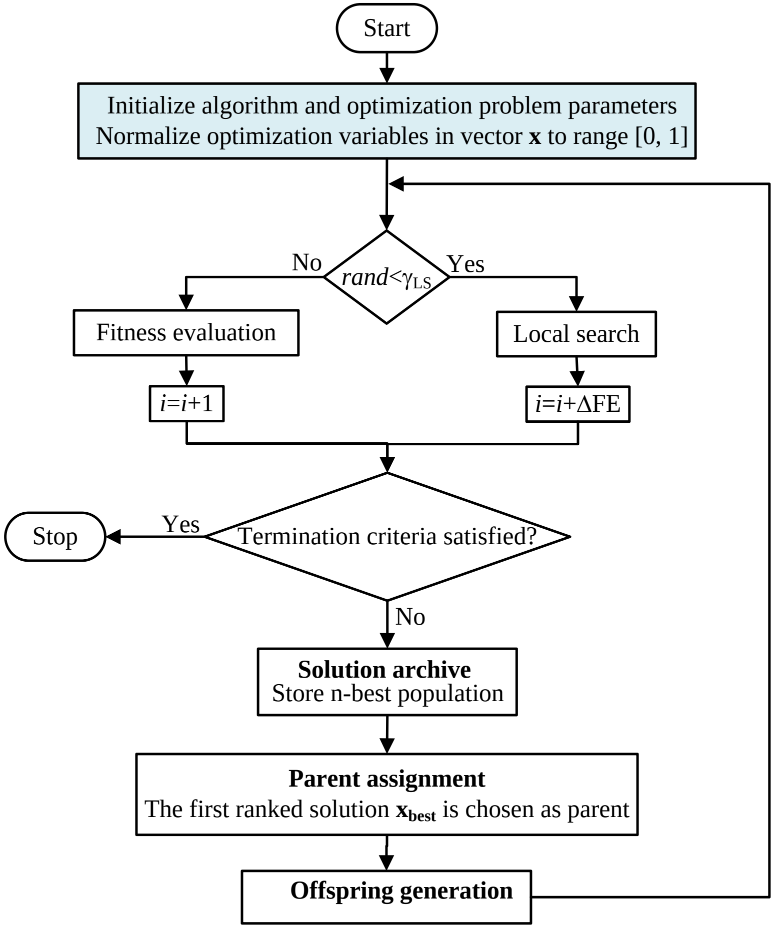
3 MVMO algorithm

EA with single parent-offspring scheme



3 MVMO algorithm

Initialization



Random selection of initial parameter values uniformly within their [min,max] intervals

$$x_j^{ini} = x_j^{min} + rand\left(x_j^{max} - x_j^{min}\right) \quad j = 1 \ldots N_{var} \quad (7)$$

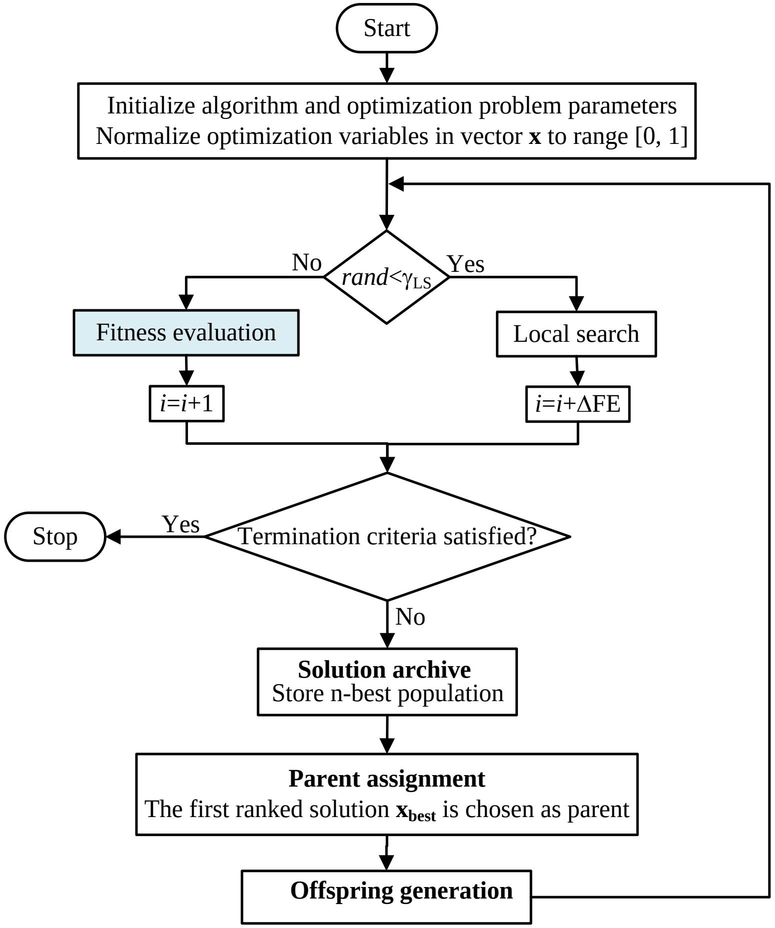
Normalization of each variable to the range [0, 1]

$$x_j^{norm} = \frac{x_j^{ini} - x_j^{min}}{x_j^{max} - x_j^{min}} \quad (8)$$

Transformation to original [min,max] intervals is performed before fitness evaluation / local search

$$x_j^{denorm} = x_j^{min} + \left(x_j^{max} - x_j^{min}\right) x_j^{norm} \quad (9)$$

3 MVMO algorithm



Fitness evaluation

Example:

$$OF = TF_i(x) - F_i^* \qquad (10)$$

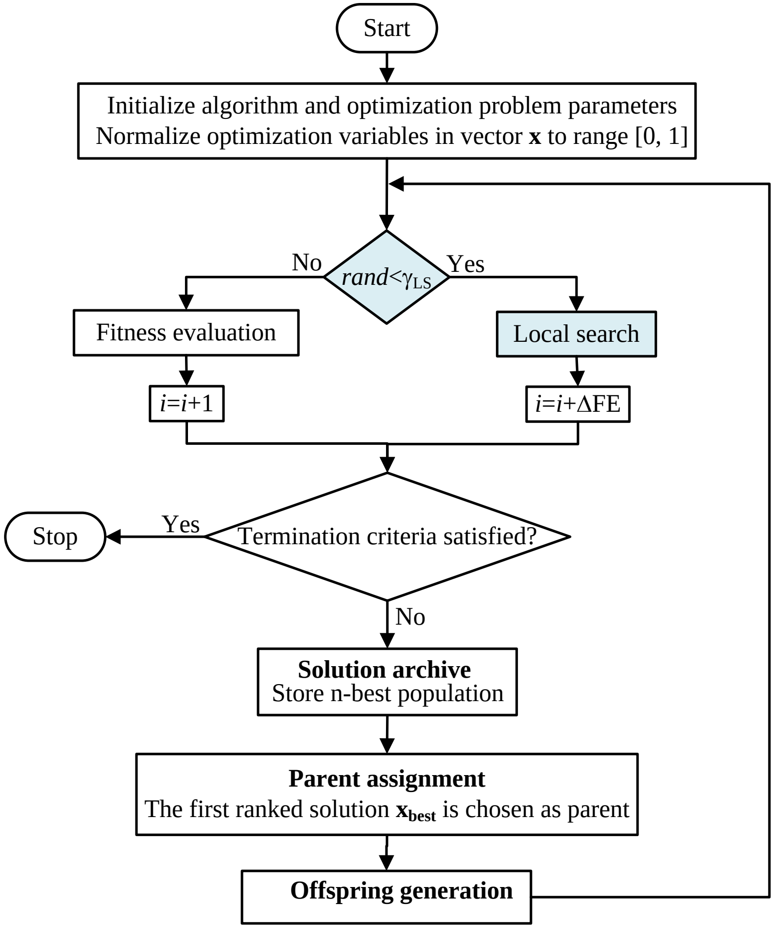
where

$TF_i(x)$: computed value of the i-th function

F_i^*: theoretical global optimum

3 MVMO algorithm

Local search



Performed according to

$$rand < \gamma_{LS} \quad (11)$$

running in the range

$$\alpha_{LS\_min} < \alpha < \alpha_{LS\_max}, \quad \alpha = i / i_{max}$$

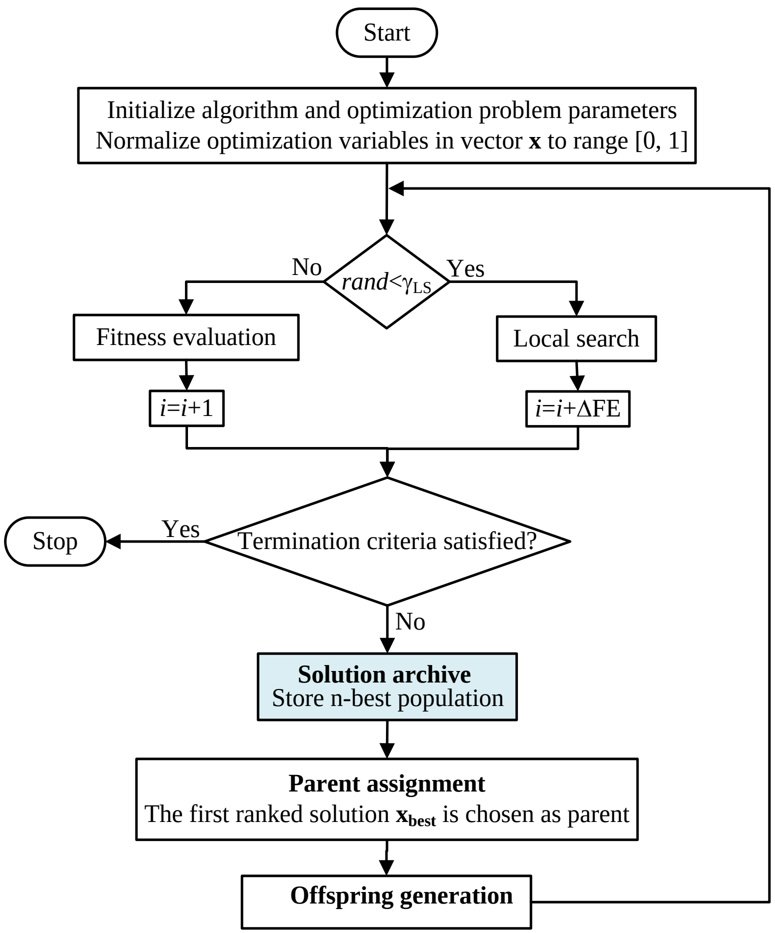
where

γ_{LS} : local search probability

Different methods can be used:

- **Classical:** Interior-Point Method (IPM)
- **Heuristic:** Hill climbing

3 MVMO algorithm



Solution archive

Ranking	Fitness	x_1	x_2	...	x_D
1st best	OF_1				
2nd best	OF_2				
...	---				
Last best	OF_A				
Mean	---	$\bar{x}_1$	$\bar{x}_2$	...	$\bar{x}_D$
Shape	---	s_1	s_2	...	s_D
d-factor	---	d_1	d_2	...	d_D

Optimization Variables

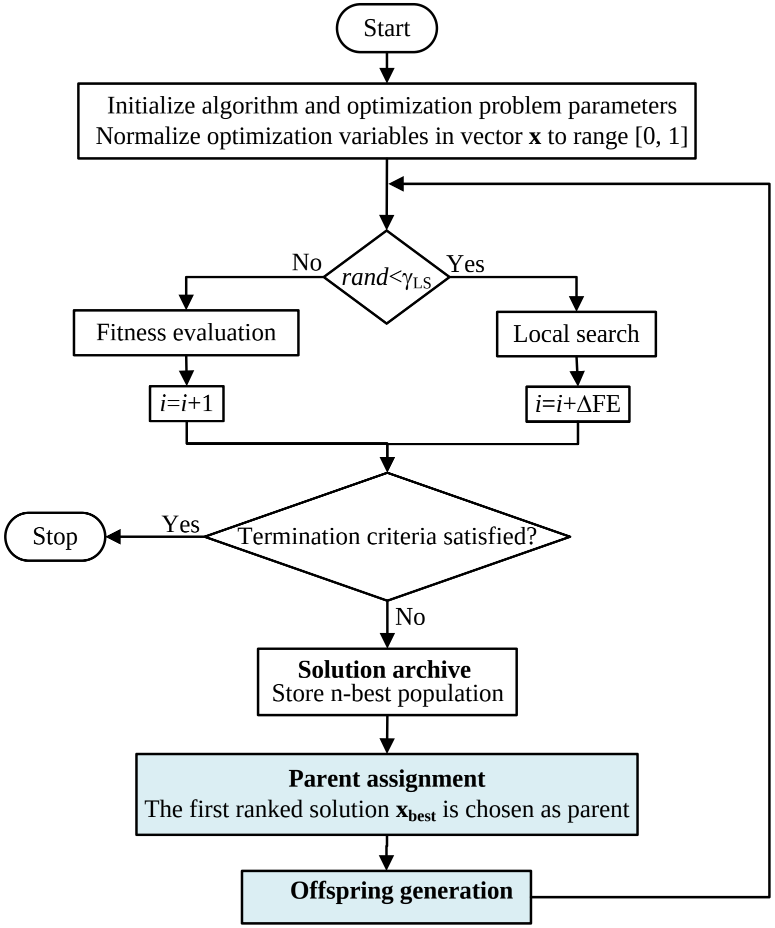
$$s_i = -\ln(v_i) \cdot f_s \quad (12)$$

$$f_s = f_s^* \left(1 + \alpha^2 \cdot rand\right) \quad (13)$$

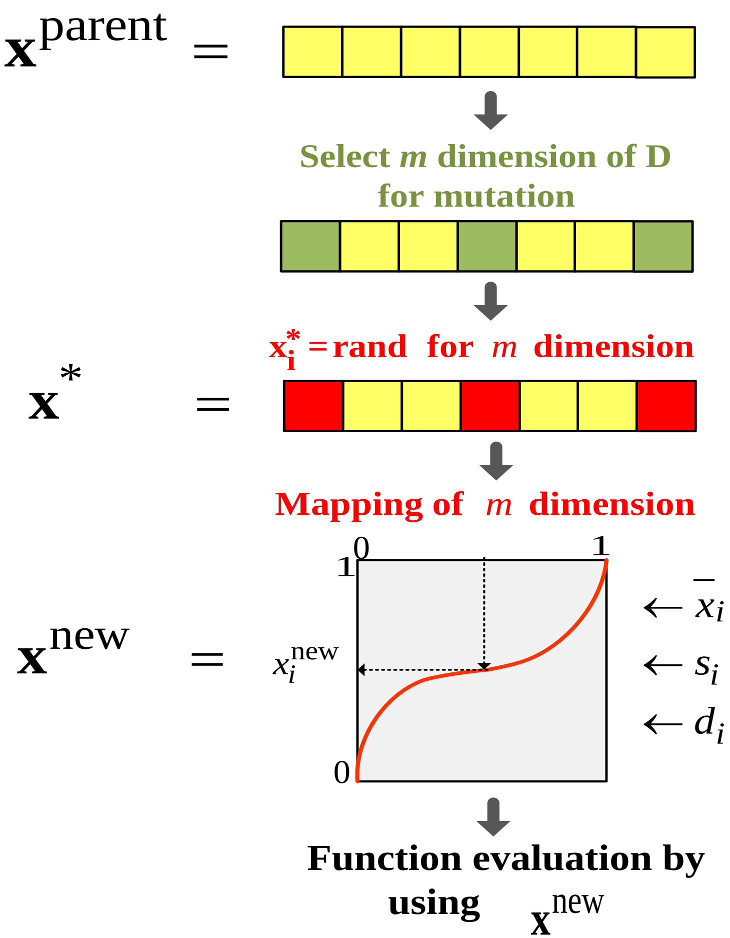
$$f_s^* = f_{s\_ini}^* + \alpha\left(f_{s\_final}^* - f_{s\_ini}^*\right) \quad (14)$$

v_i : Variance f_s : scaling factor

3 MVMO algorithm



New offspring



3 MVMO algorithm

Selection of *m* dimensions for mutation

$$m = \text{round}\left(\text{m}_{\text{final}} + \text{rand}\left(m^* - \text{m}_{\text{final}}\right)\right) \quad (15)$$

$$m^* = \text{round}\left(\text{m}_{\text{ini}} - \alpha\left(\text{m}_{\text{ini}} - \text{m}_{\text{final}}\right)\right) \quad (16)$$

$$\alpha = i / i_{\text{max}} \quad (17)$$

m_{ini} : Initial number of selected variables (e.g. $N_{var}/2$)

m_{final} : Final number of selected variables (e.g. $N_{var}/4$)

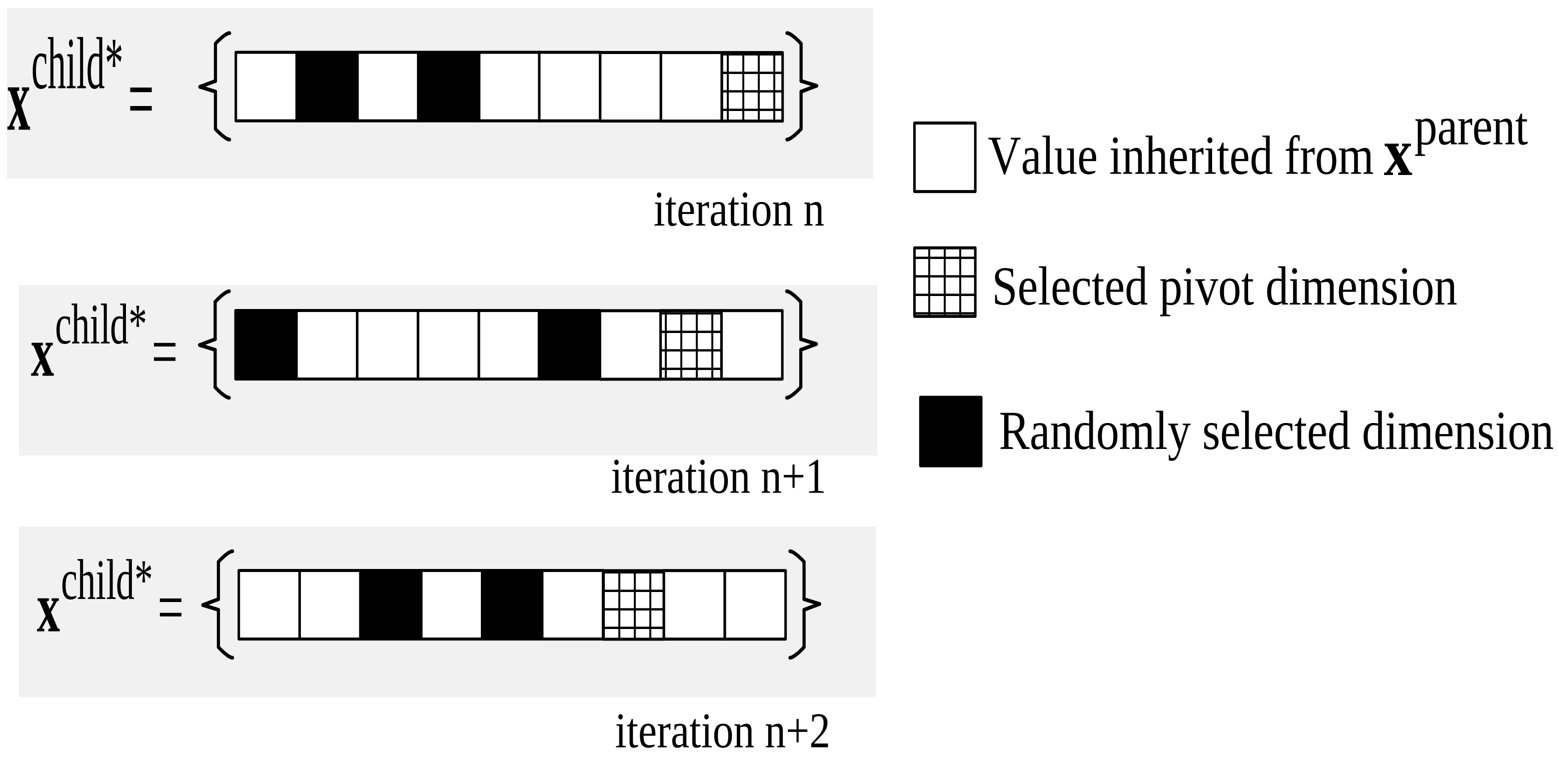
i: Function evaluation number

rand: Uniform random number within [0,1]

3 MVMO algorithm

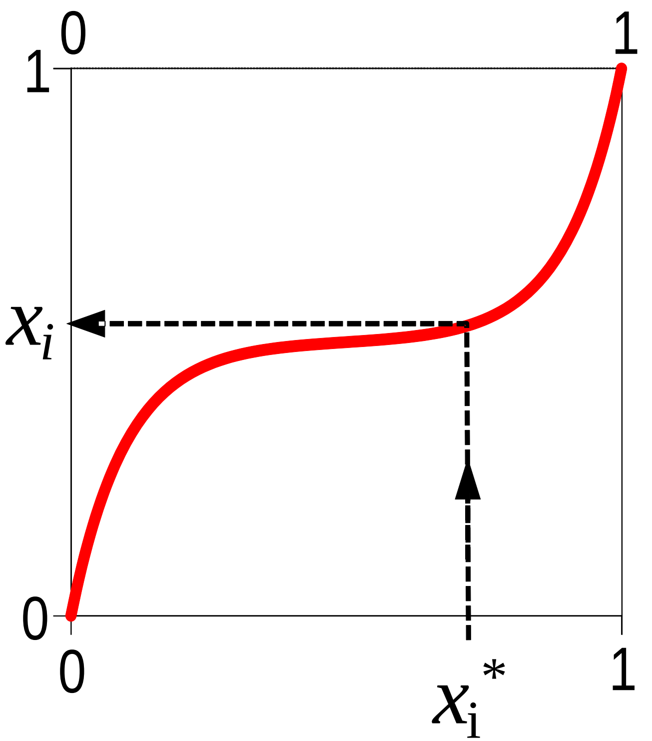
Selection of *m* dimensions for mutation

Random-sequential selection mode



3 MVMO algorithm

Mutation based on mapping function



$$h = \bar{x}_i \cdot (1 - e^{-x \cdot s_1}) + (1 - \bar{x}_i) \cdot e^{-(1-x) \cdot s_2} \quad (18)$$

$$x_i = h_x + (1 - h_1 + h_0) \cdot x_i^* - h_0 \quad (19)$$

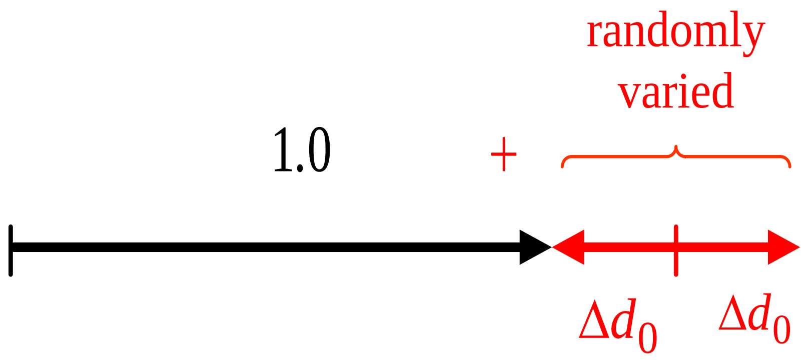
$$\begin{aligned} h_x &= h(x = x_i^*) \\ h_0 &= h(x = 0) \\ h_1 &= h(x = 1) \end{aligned} \quad (20)$$

x_i^* and x_i in the range $[0 \;\; 1]$

3 MVMO algorithm

Assignment of shape and d-factors

$s_{i1} = s_{i2} = s_i$
if $s_i > 0$ then
  $\Delta d = (1 + \Delta d_0) + 2.5 \cdot \Delta d_0 \cdot (rand - 0.5)$
  if $s_i > d_i$
    $d_i = d_i \cdot \Delta d$
  else
    $d_i = d_i / \Delta d$
  end if
  if $d_i > s_i$ then
    $s_a = d_i; \quad s_b = s_i$
  else
    $s_a = s_i; \quad s_b = d_i$
  end if
  if $\overline{x}_i < 0.5$ then
    $s_{i1} = s_a; \quad s_{i2} = s_b$
  else
    $s_{i1} = s_b; \quad s_{i2} = s_a$
  end if
end if

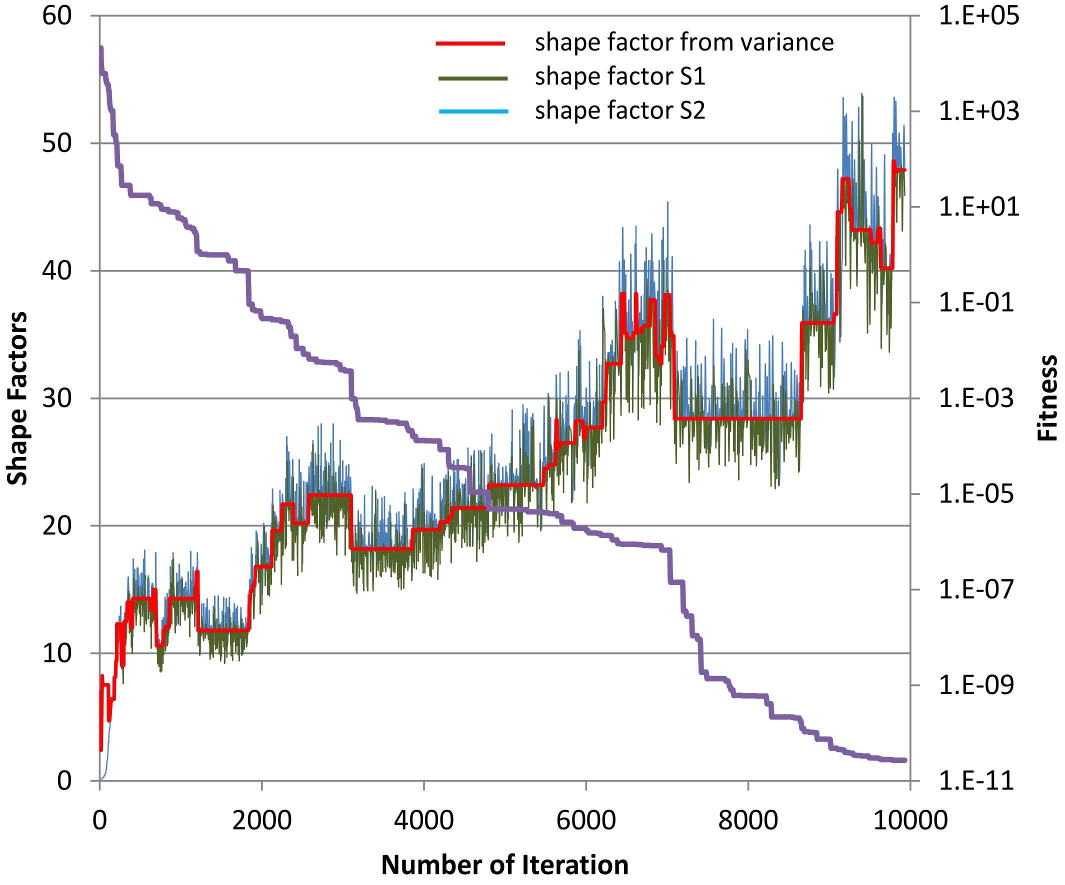


d_i is always oscillating around the shape s_i and is set to 1 in the initialization stage

$\Delta d_0 \le 0.4$

3 MVMO algorithm

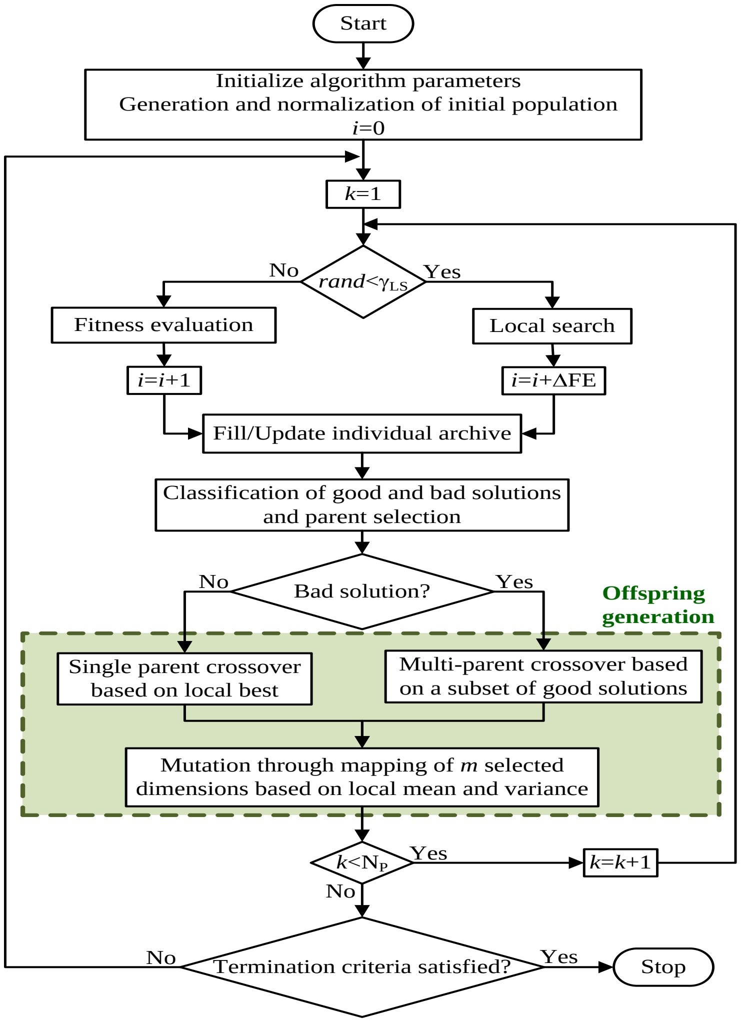
Assignment of shape and d-factors



CEC2013 function F1, MVMO-SH with 1 particle and without local search, fs=1.0, Δd_0=0.15

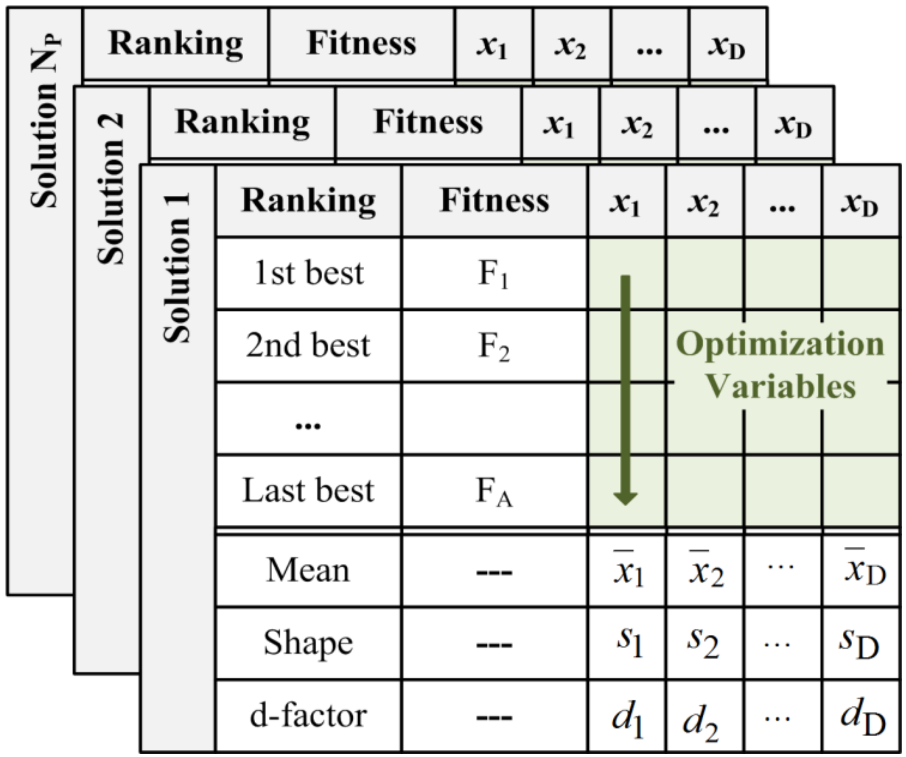
3 MVMO algorithm

Population based approach



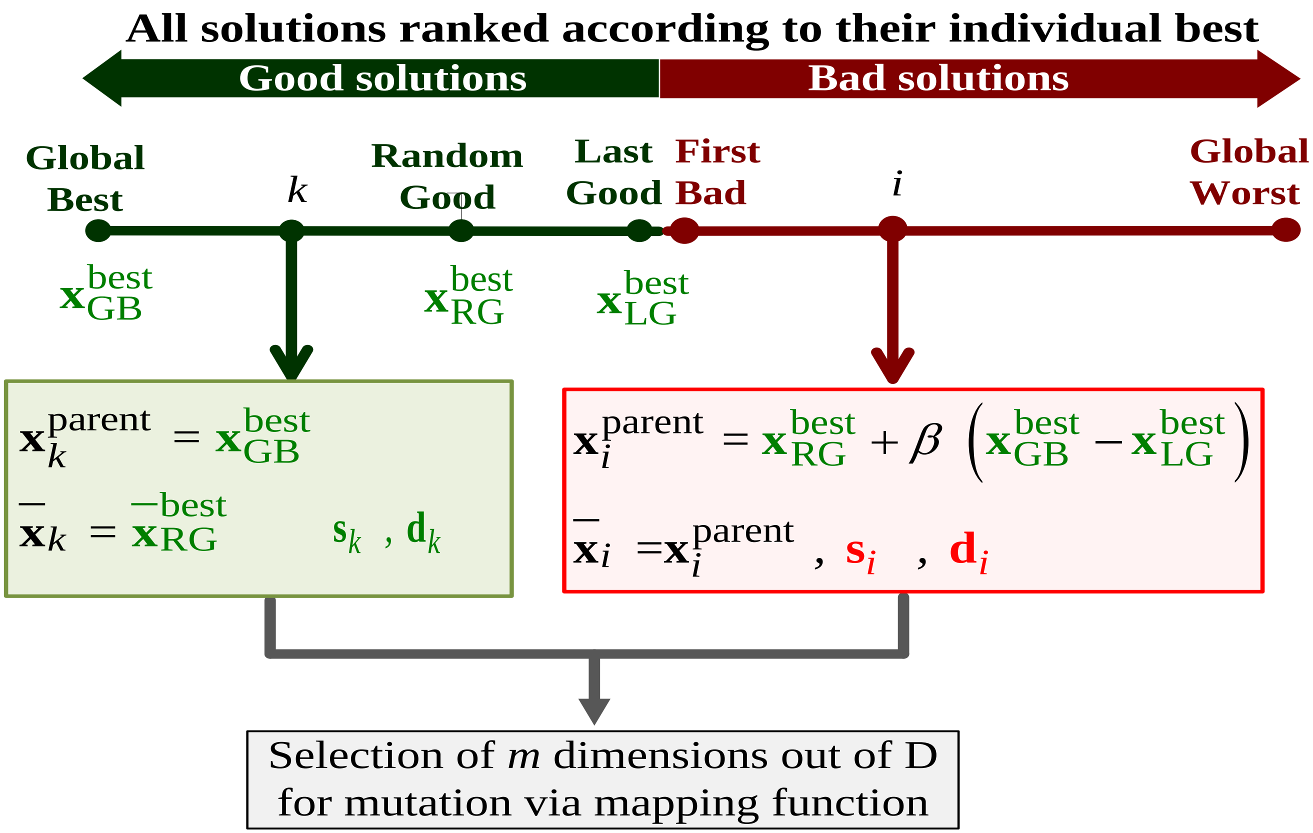
3 MVMO algorithm

Solution archive



3 MVMO algorithm

Parent selection



$$\beta = 2.5\left(rand + 0.25 \cdot \alpha^2 - 0.5 \right) \quad (21)$$

3 MVMO algorithm

Parent selection

All particles ranked according to their local best

Dynamic Good/Bad Particle Selection:

➔ The ranking is re-calculated after any function evaluation

➔ The border between Good/Bad particles is shifted downwards with the progress of iteration

$$GP = \text{round}\left(N_p \cdot g_p^{*}\right) \quad (22)$$

$$g_p^{*} = g^{*}_{p\_ini} - \alpha^2 \left(g^{*}_{p\_final} - g^{*}_{p\_ini}\right) \quad (23)$$

N_P: number of particles

i: function evaluation counter

3 MVMO algorithm

MVMO demo:

https://www.uni-due.de/mvmo/download

3 Other algorithms

1. Covariance Matrix Adaptation Evolution Strategy (CMA-ES)
https://www.lri.fr/~hansen/cmaesintro.html

2. Linearized Biogeography-based Optimization (LBBO)
http://academic.csuohio.edu/simond/bbo/

3. Fireworks Algorithm (FWA)
http://www.cil.pku.edu.cn/research/fa/

4. Firefly algorithm, cuckoo search, and bat algorithm
http://www.mathworks.com/matlabcentral/profile/authors/2652824-xin-she-yang

5. Teaching-Learning-Based Optimization (TLBO)
https://sites.google.com/site/tlborao/

4 CEC2015 Expensive Problems

1. Test bed description:

Q. Chen, B. Liu, Q. Zhang, J.J. Liang, P. N. Suganthan, and B.Y. Qu, "Problem Definition and Evaluation Criteria for CEC 2015 Special Session and Competition on Bound Constrained Single-Objective Computationally Expensive Numerical Optimization," Technical Report, Oct. 2013

2. Results and codes for different algorithms

http://www.ntu.edu.sg/home/epnsugan/

(under EA Benchmarks / CEC Competitions)

4 CEC2015 Expensive Problems

- **MVMO parameters:** $\mathrm{N_p} = 1;\ \text{Archive size} = 7$

  $f^*_{\mathrm{s\_ini}} = 1;\ f^*_{\mathrm{s\_final}} = 2;\ \Delta d_0 = 0.1$

  $m_{\mathrm{ini}} = 1;\ m_{\mathrm{final}} = 4$

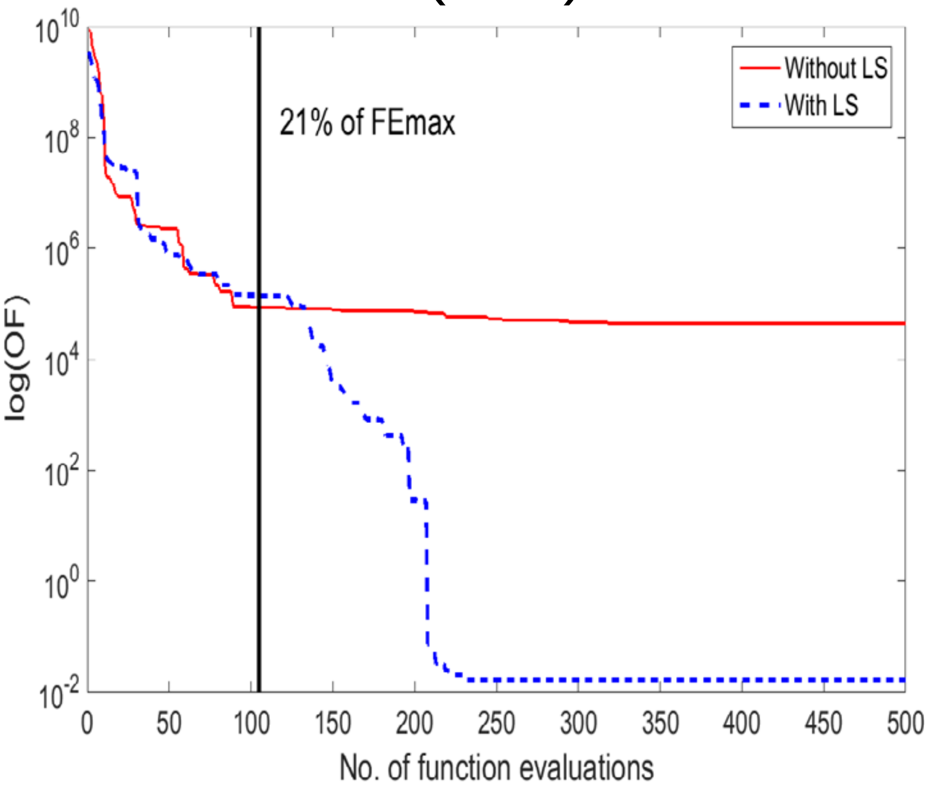
- **10D problems:** Only one local search call was allowed directly after 21 % of Femax, γ_{LS}=100%

- **30D problems:** Local search launched 5 times after 21 % of FEmax with a probability γ_{LS}=35%

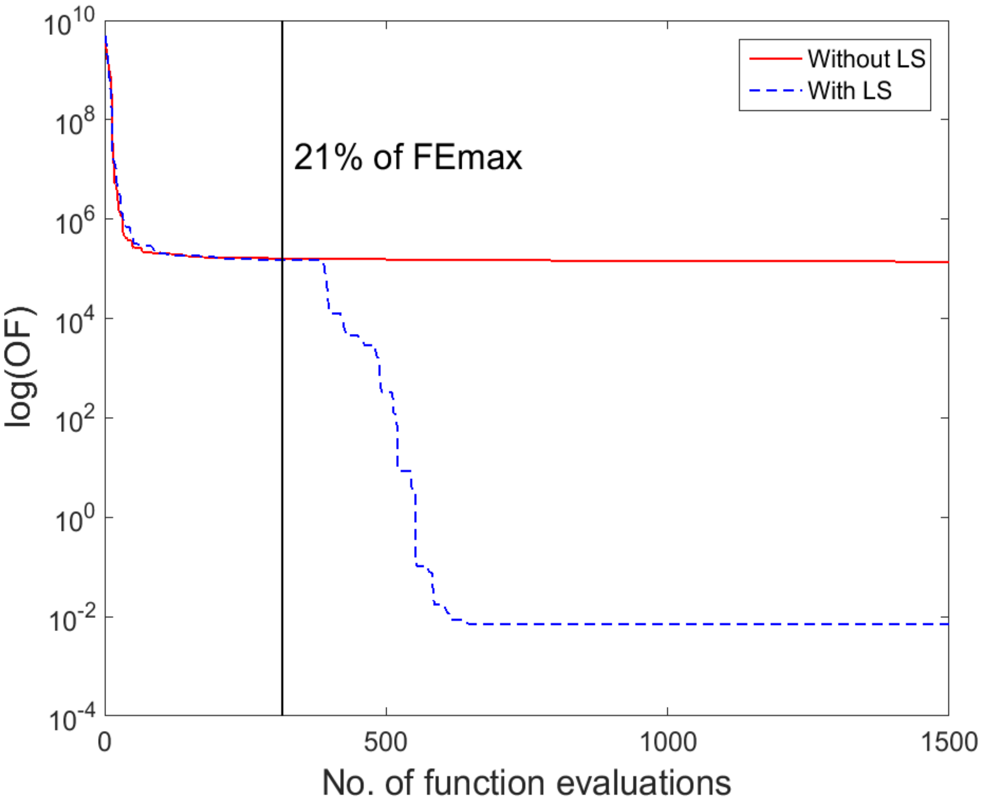
4 CEC2015 Expensive Problems

Unimodal, hybrid and composition functions

TF2 (10D)

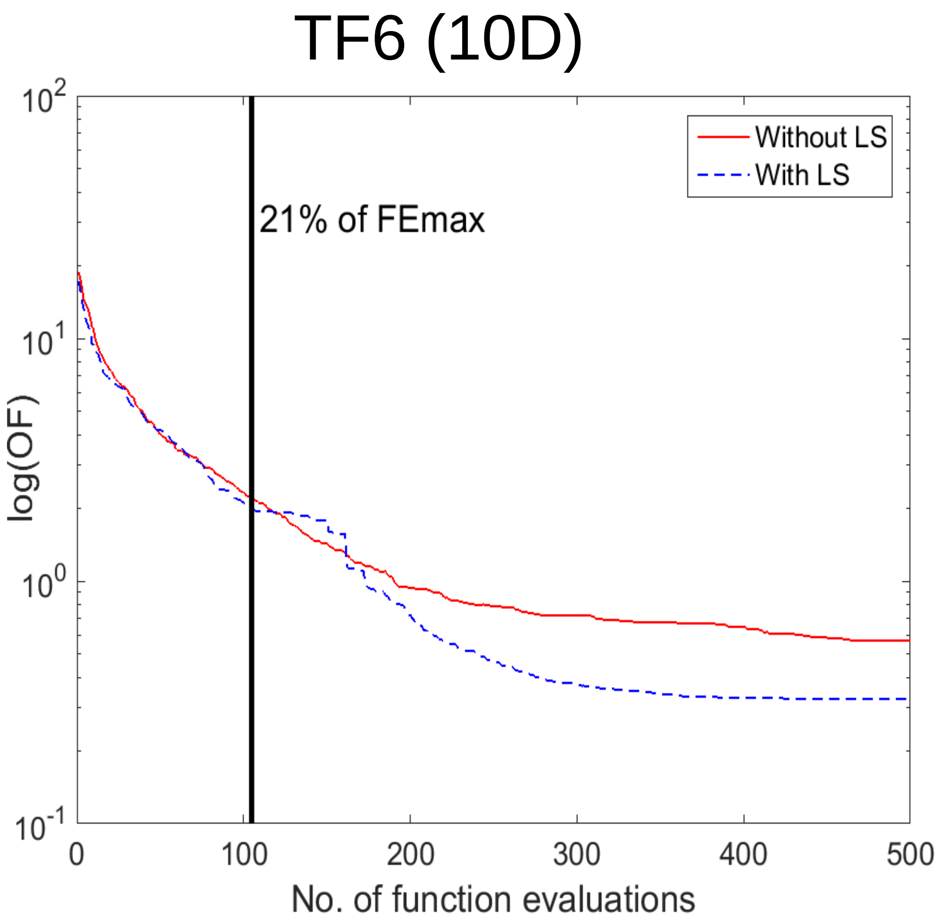


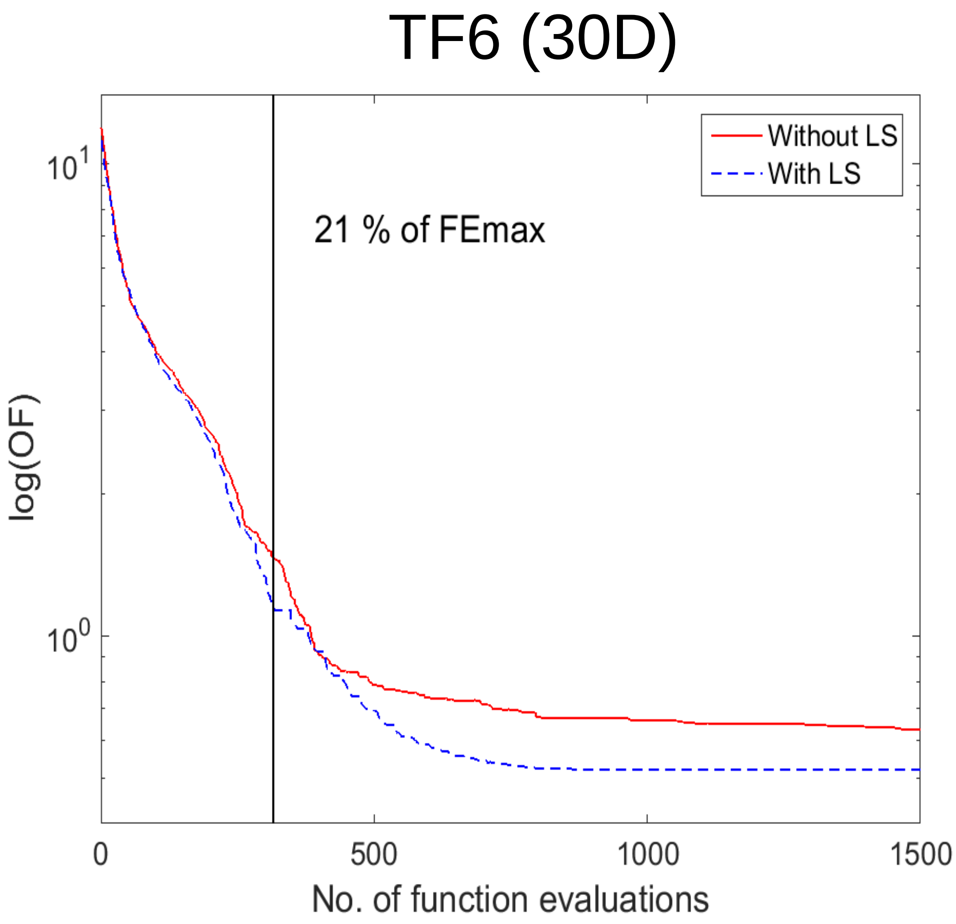
TF2 (30D)



Including LS entails a dramatic decrease of the error values

4 CEC2015 Expensive Problems
Multimodal functions





Evolutionary mechanism of MVMO is more effective

4 CEC2015 Expensive Problems

Important findings

- Single parent-offspring MVMO approach proves effective for computationally expensive optimization problems
- IPM-based LS strategy helped to improve the performance on unimodal functions
- LS entails minor improvements for hybrid and composition functions
- Use of LS led to adverse effect when solving simple multimodal functions.

Thanks for your attention!

Questions?