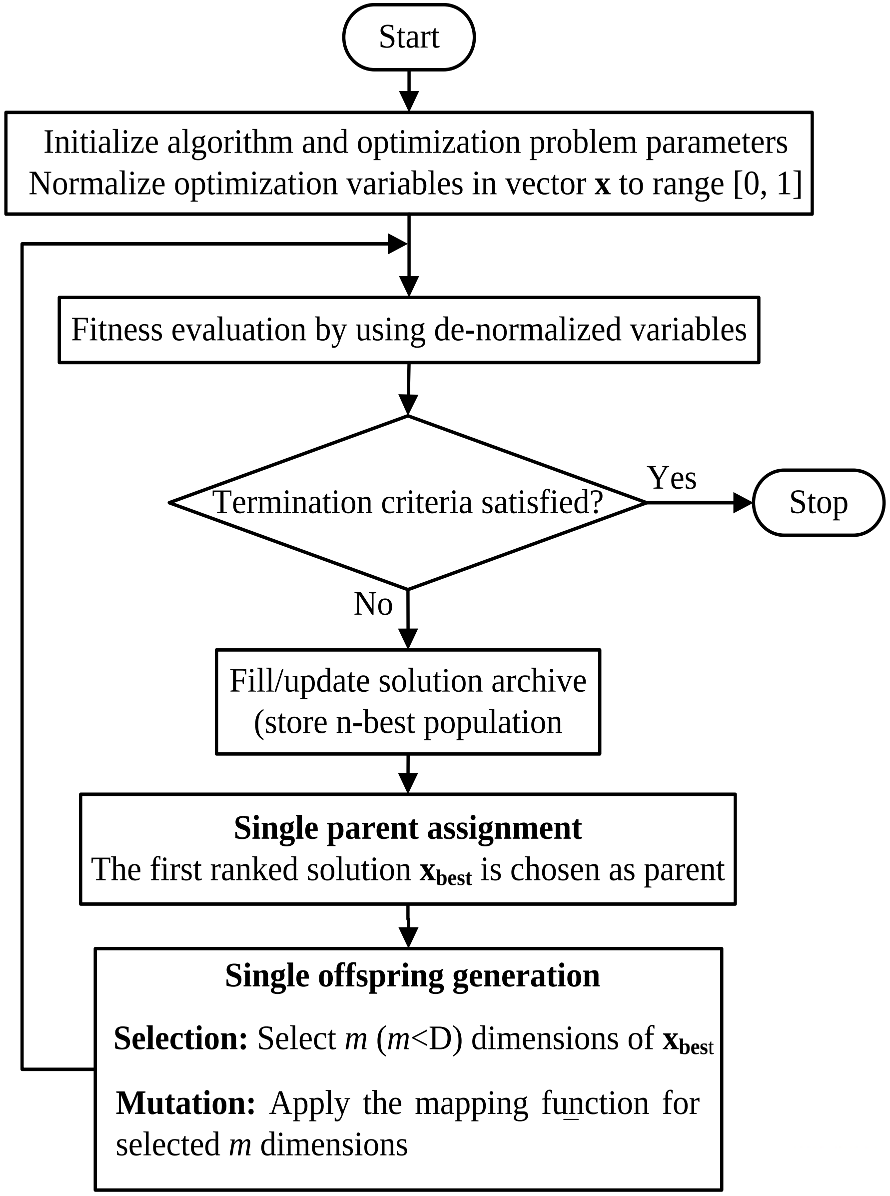
Application of Mean Variance Mapping Optimization (MVMO) to Solve OPF Problems

István Erlich
University Duisburg-Essen
Duisburg - Germany
istvan.erlich@uni-due.de

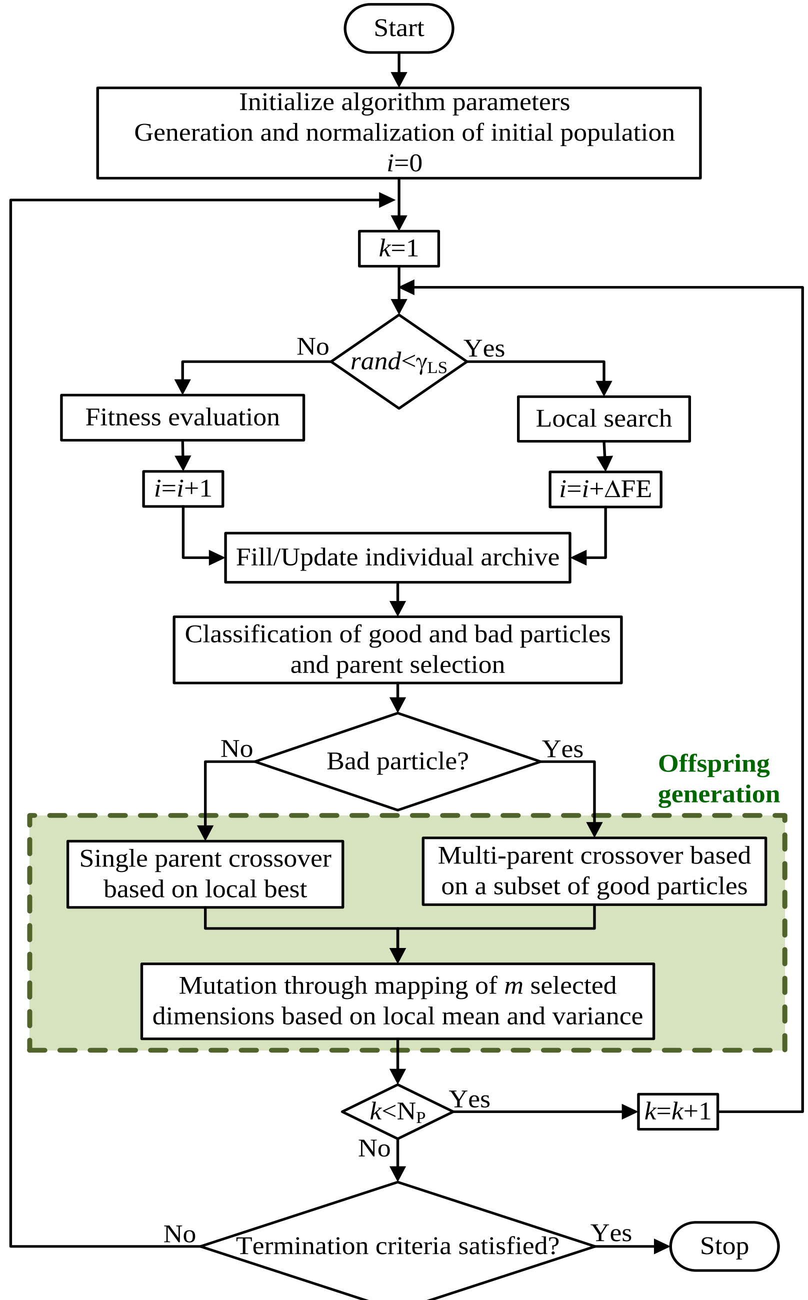
José L. Rueda
Delft University of Technology
Delft – Netherlands
j.l.ruedatorres@tudelft.nl

Rationale behind MVMO

- Introduced by I. Erlich (University Duisburg-Essen, Germany) in 2010
- Internal search range of all variables restricted to [0, 1].
- **Solution archive:** knowledge base for guiding the searching direction.
- **Mapping function:** Applied for mutating the offspring on the basis of the mean and variance of the n-best population attained so far.



The hybrid variant: MVMO-SH



IEEE PES Power & Energy Society®

IEEE

MVMO-SH: launching local search

Local search performed according to

$$rand < \gamma_{LS} \quad (1)$$

where

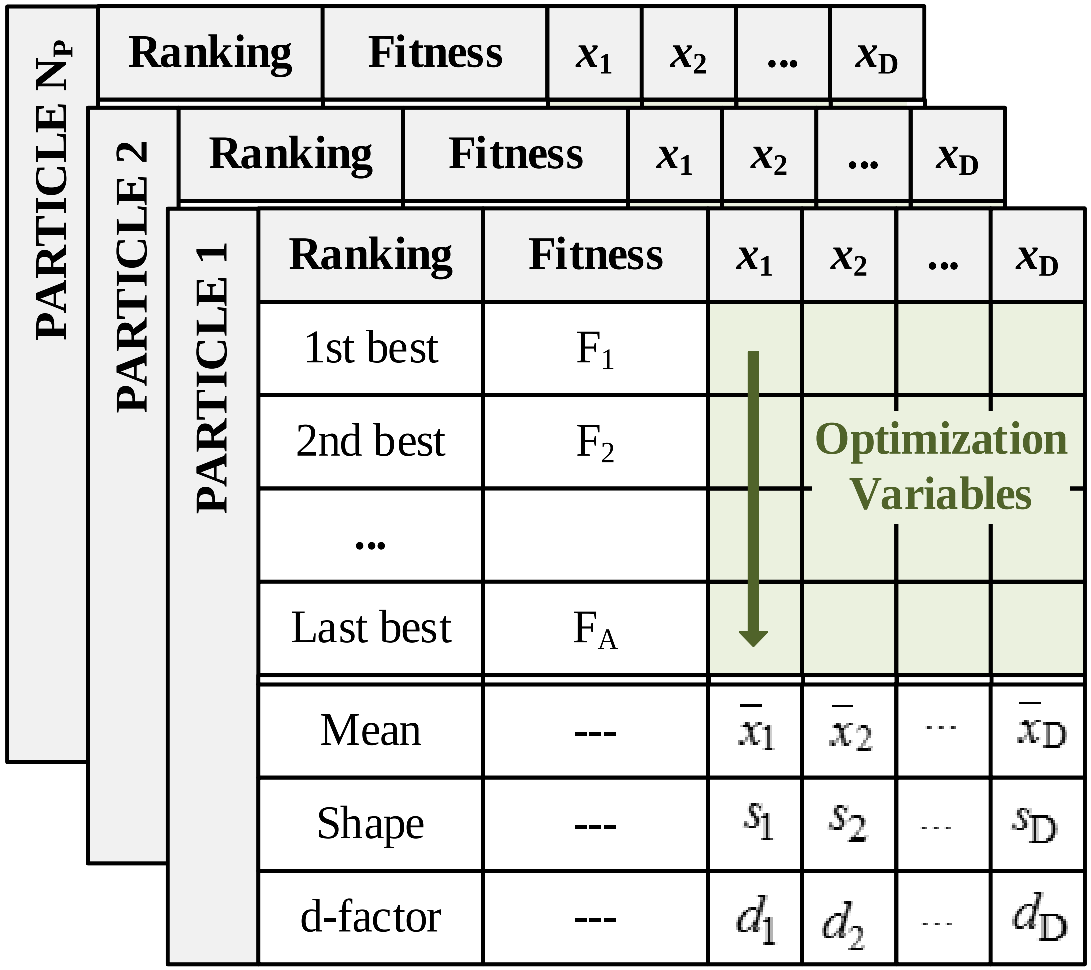
γ_{LS} : local search probability, e.g. $\gamma_{LS} = 1.5/100/D$

D: Problem dimension

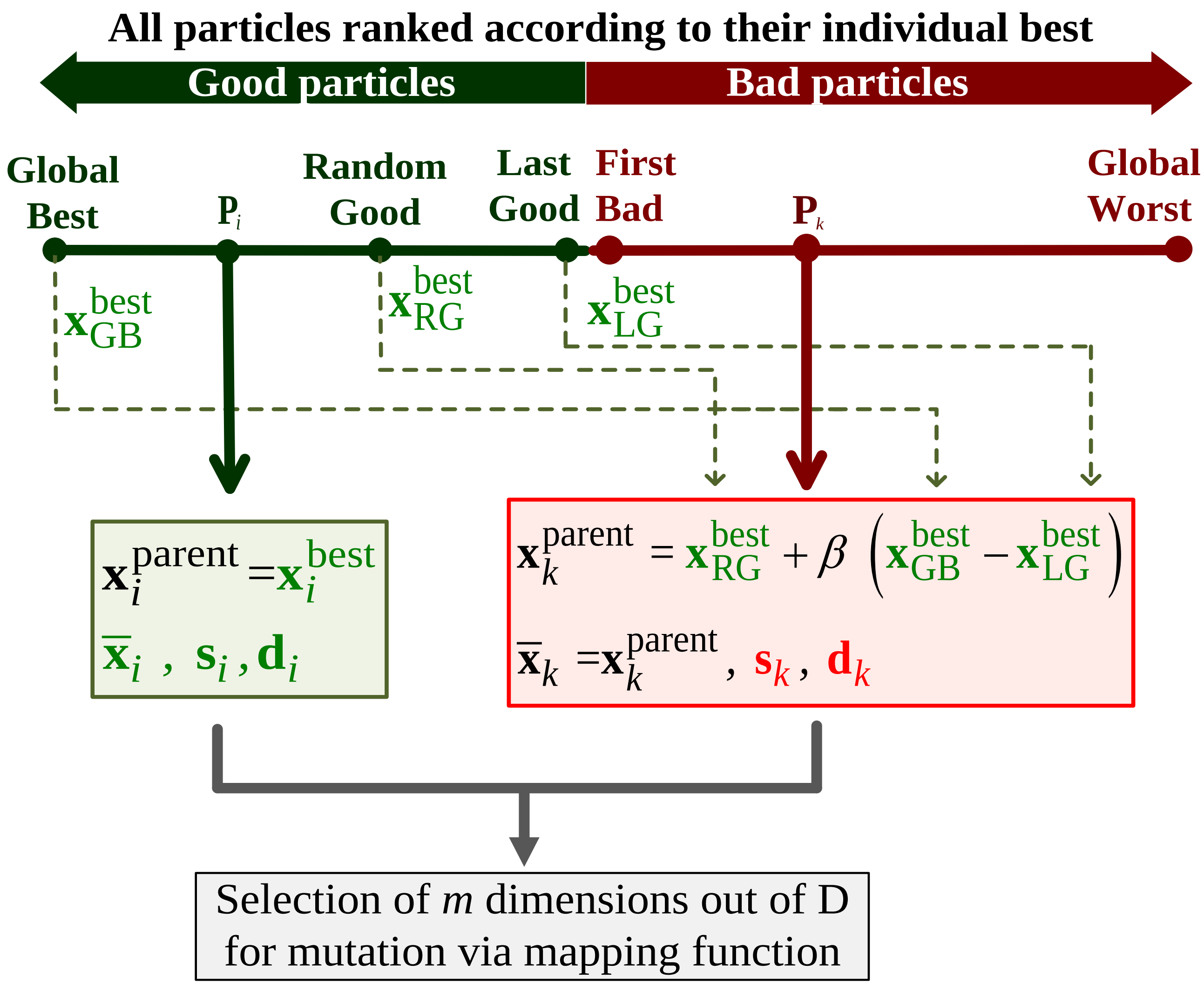
Different methods can be used:

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

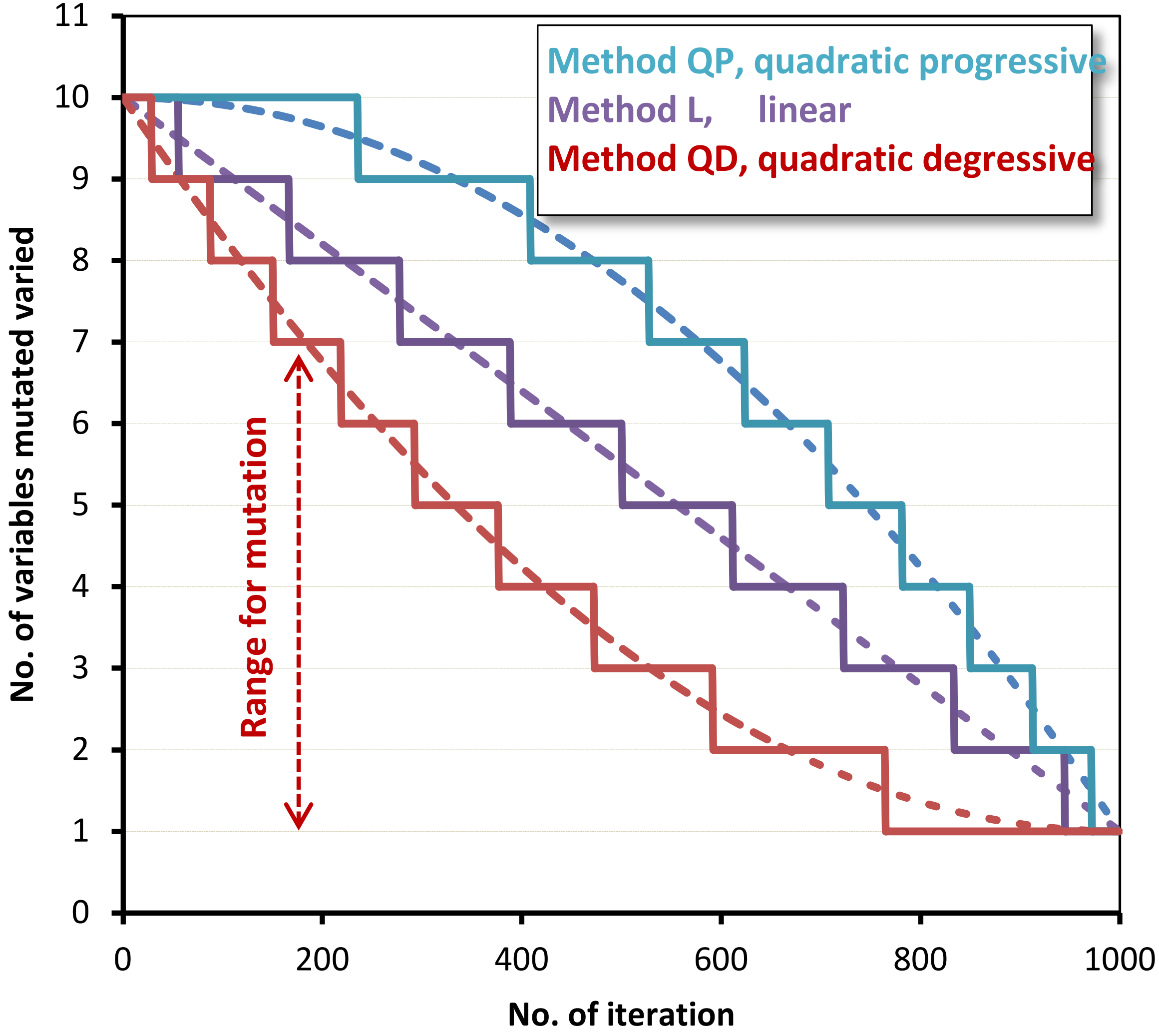
MVMO-SH: solution archive



MVMO-SH: parent selection



MVMO-SH: selection of dimensions for mutation



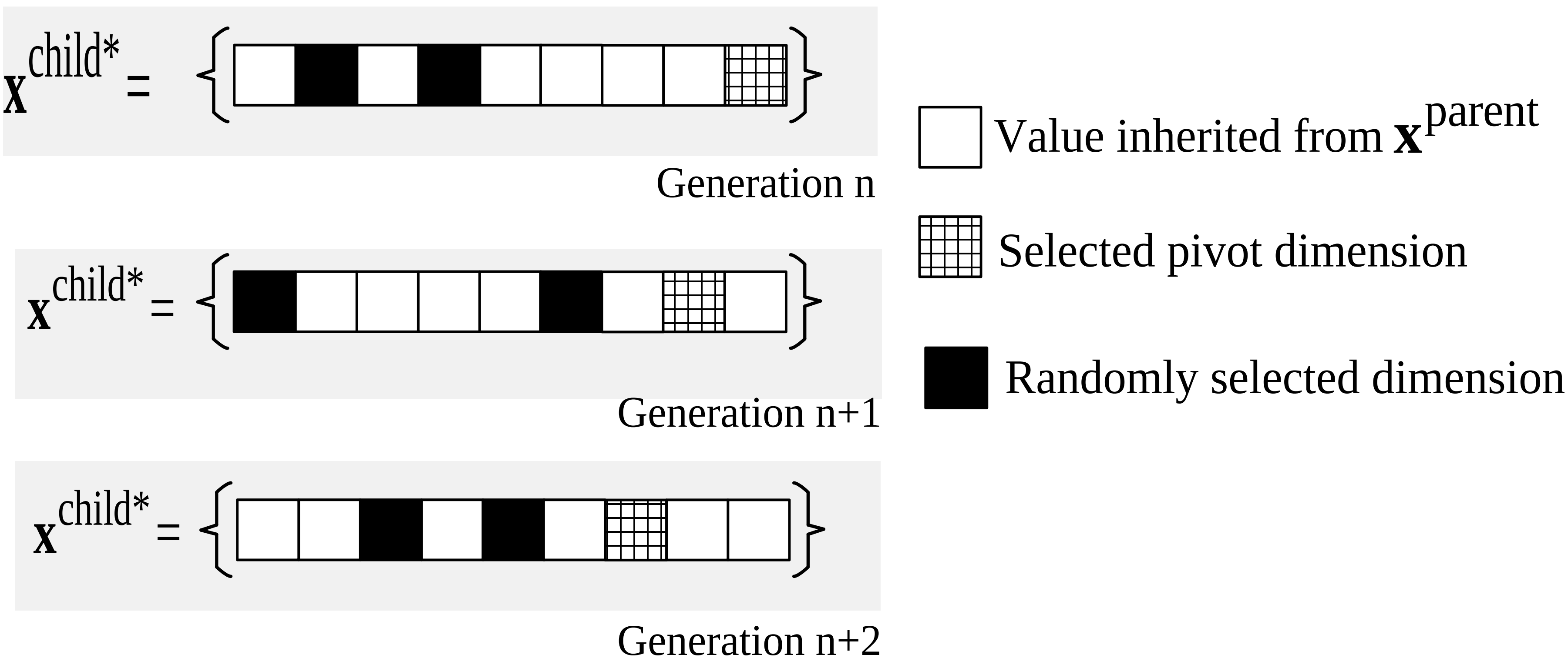
Two different strategies are available:

a) The full range corresponds with the number of mutated variables, e.g. $m = 7$

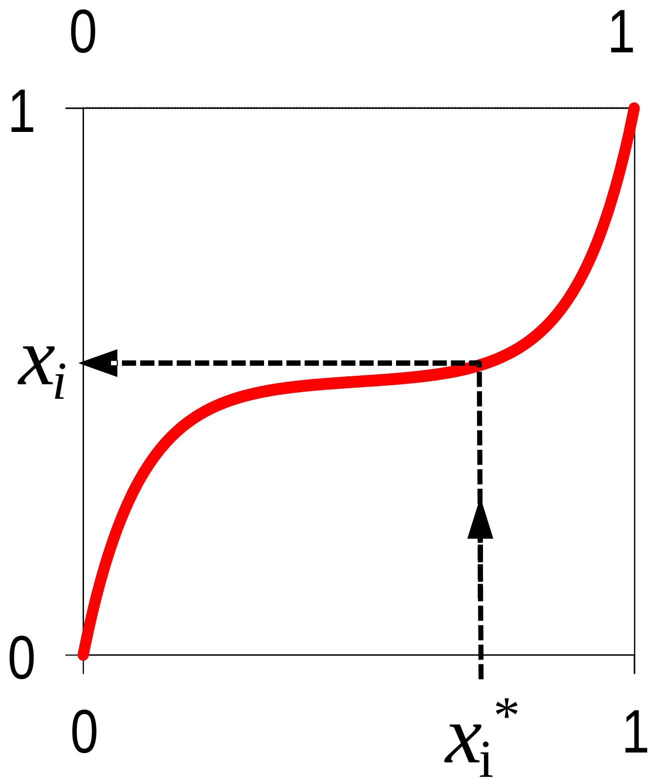
b) The number of mutated variables estimated randomly in the given range, e.g. $m = irand(7)$

MVMO-SH: selection of dimensions for mutation

Random-sequential selection mode



MVMO-SH: mutation based on mapping function



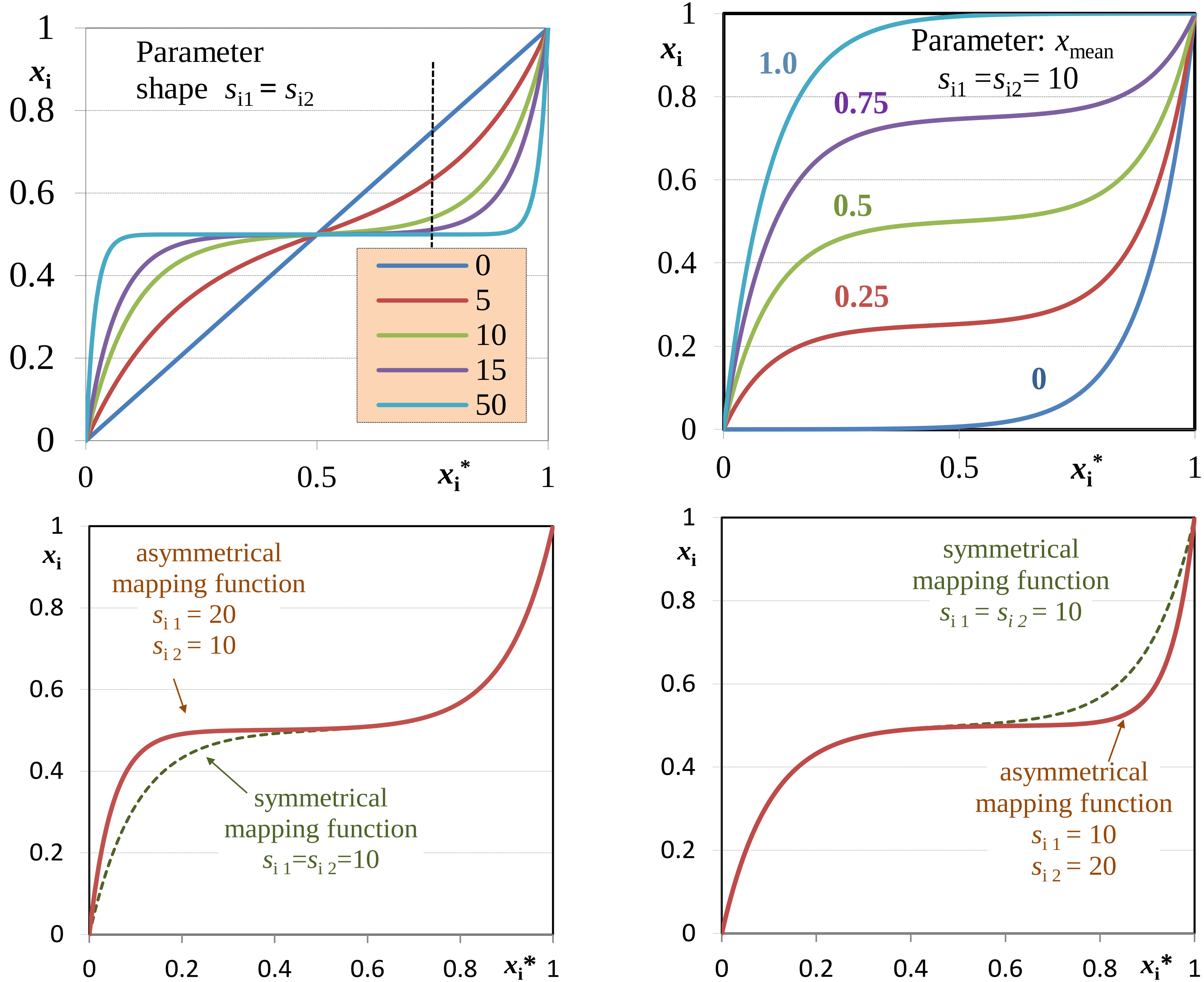
$$h = \bar{x}_i \cdot (1 - e^{-x \cdot s_1}) + (1 - \bar{x}_i) \cdot e^{-(1-x) \cdot s_2} \quad (2)$$

$$x_i = h_x + (1 - h_1 + h_0) \cdot x_i^* - h_0 \quad (3)$$

$$h_x = h(x = x_i^*)$$
$$h_0 = h(x = 0) \quad (4)$$
$$h_1 = h(x = 1)$$

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

MVMO-SH: mapping function features



MVMO-SH: assignment of shape and d-factors

$s_{i1} = s_{i2} = s_i = -\ln(v_i) \cdot f_s$

if $s_i > 0$ then

  $\Delta d = (1 + \Delta d_0) + 2 \cdot \Delta d_0 \cdot (rand - 0.5)$

  if $s_i > d_i$

    $d_i = d_i \cdot \Delta d$

  else

    $d_i = d_i / \Delta d$ (5)

  end if

  if $rand < 0.5$ then

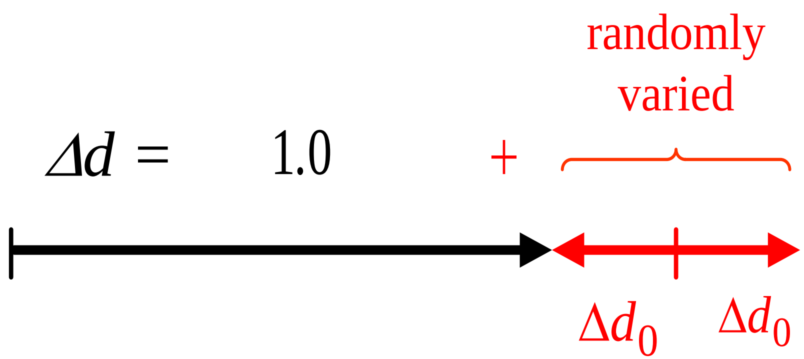
    $s_{i1} = s_i$; $s_{i2} = d_i$

  else

    $s_{i1} = d_i$; $s_{i2} = s_i$

  end if

end if



d_r is always oscillating around the shape s_r and is set to 1 in the initialization stage

$\Delta d_0 \le 0.4$

The d-factors remain dynamic with the mapping even the corresponding shape doesn’t change

MVMO-SH: assignment of shape and d-factors

$s_{i1} = s_{i2} = s_i = -\ln(v_i) \cdot f_s$

if $s_i > 0$ then

$\Delta d = (1 + \Delta d_0) + 2 \cdot \Delta d_0 \cdot (rand - 0.5)$

if $s_i > d_i$

$d_i = d_i \cdot \Delta d$

else

$d_i = d_i / \Delta d$ (5)

end if

if $rand < 0.5$ then

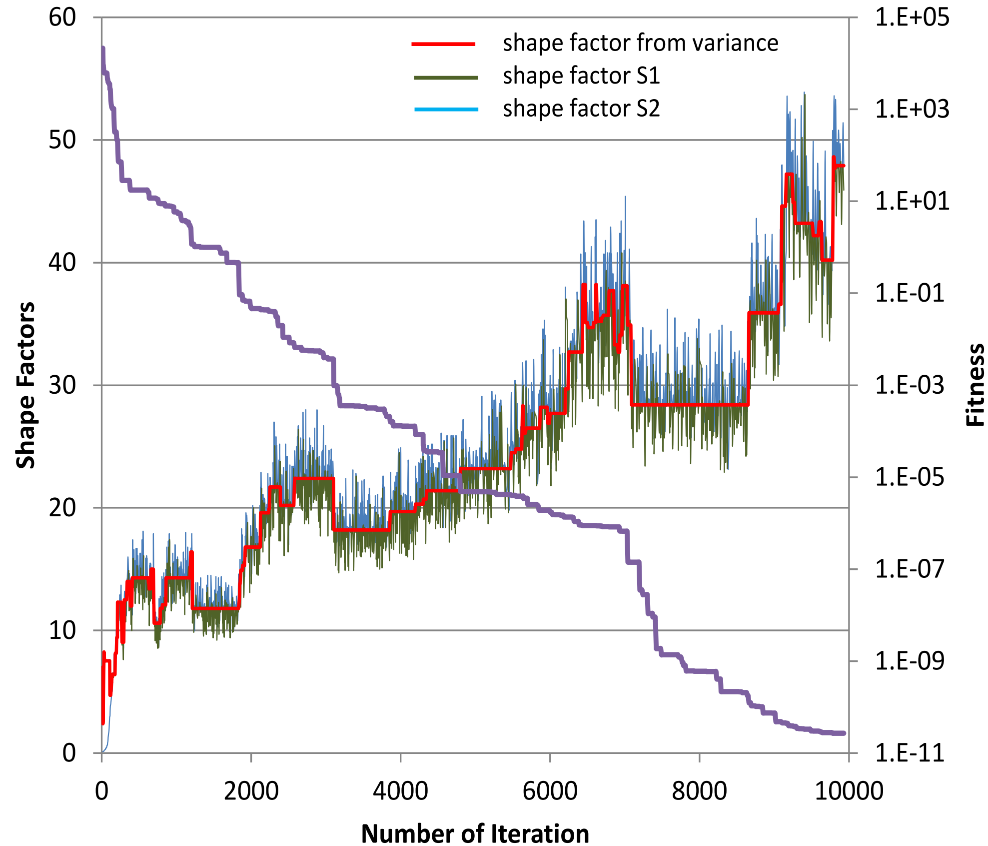
$s_{i1} = s_i$; $s_{i2} = d_i$

else

$s_{i1} = d_i$; $s_{i2} = s_i$

end if

end if



CEC2013 function F1, single particle MVMO without local search, *f*s=1.0, Δd_0=0.15

Thanks!

Dr. José L. Rueda

J.L.RuedaTorres@tudelft.nl

http://www.uni-due.de/mvmo/