

## Mean Variance Mapping Optimization

MVMO operates on a single solution rather than a set of solutions like in many Evolutionary Algorithms. The internal search range of all variables is restricted to  $[0, 1]$ . However, the function evaluation is carried out always in the original scales. The key feature of MVMO is a special mapping function described by the mean and shape variables. Both are derived from  $n$  the best solutions saved in the archive. The mapping function transforms a variable  $x_i^*$  varied randomly with unity distribution to another variable  $x_i$  which is concentrated around the mean value. The mapping is demonstrated in Fig. 1

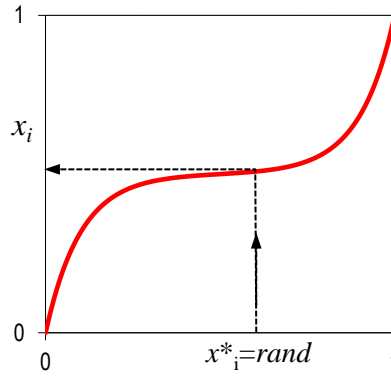


Fig. 1. Variable mapping

The distribution of the new variable  $x_i$  doesn't correspond with any of the well-known distribution functions even though there are some similarities to the Gauss function. The transformation  $x_i^* \rightarrow x_i$  is described mathematically as follows:

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

where the h-function is defined as

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

$h_x$ ,  $h_1$  and  $h_0$  are the outputs of the H-function (2) based on different inputs given by:

$$h_x = h(x = x_i^*), \quad h_0 = h(x = 0), \quad h_1 = h(x = 1) \quad (3)$$

Note that the output of (1) is always within the bounds  $[0,1]$  for every generated  $x_i^*$ .

The shape of the h-function is determined by the mean  $\bar{x}$  and the shape variables  $s_1$  and  $s_2$ . The effect of these parameters on the form of the function is illustrated in Fig. 2.

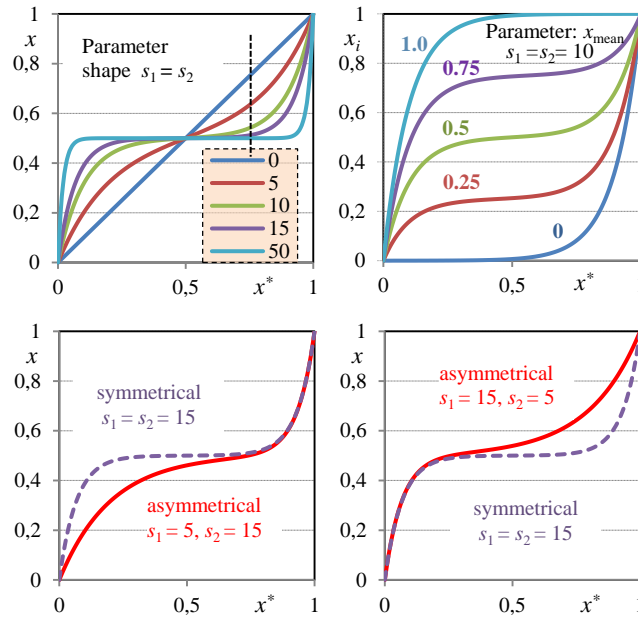


Fig. 2. Effect of parameters on the mapping function

The distinctive property of MVMO is the ability to carry out global searching but focus around the best-so-far solutions. This is shown for two variables in Fig 3.

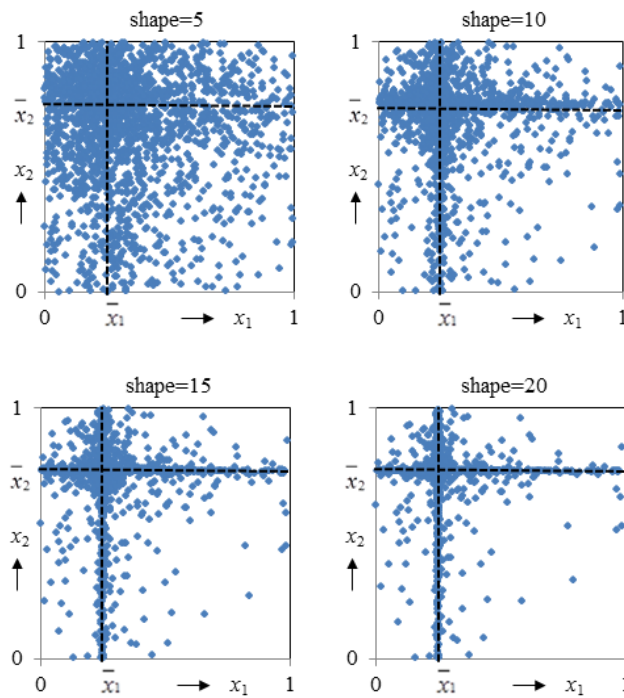


Fig. 3. Search space depending on the shape factor  
 Number of variables=2,  $\bar{x}_1=0.25$ ,  $\bar{x}_2=0.75$ , shape= $s_1=s_2$  for both variables

As can be seen the search is focused around the mean values which are  $\bar{x}_1 = 0.25$  and  $\bar{x}_2 = 0.75$  in this example. However, there are some samples also outside the mean areas, i.e. the algorithm performs global search but the emphasis is almost around the means.

Mean and shape variables are calculated from the archive where the  $n$  best populations are stored.

$$\bar{x}_i = \frac{1}{n} \sum_{j=1}^n x_i(j) \quad (4)$$

$$s_i = -\ln(v_i) \cdot f_s \quad (5)$$

with the variance

$$v_i = \frac{1}{n} \sum_{j=1}^n (x_i(j) - \bar{x}_i)^2 \quad (6)$$

The variance is calculated only for different variables in the archive. The factor  $f_s$  can be used to change the shape of the function e.g. when the accuracy needs to be improved (increase  $f_s > 1$ ) or more global search is required (decrease  $f_s < 1$ ).

The shape variables  $s_{i1}$  and  $s_{i2}$  of  $x_i$  are determined by using the following algorithm:

$$\begin{aligned}
& s_{i1} = s_{i2} = s_i \\
& \text{if } s_i > 0 \text{ then} \\
& \quad \text{if } s_i > d_i \\
& \quad \quad d_i = d_i \cdot \Delta d \\
& \quad \text{else} \\
& \quad \quad d_i = d_i / \Delta d \\
& \quad \text{end if} \\
& \quad \text{if } rand \geq 0.5 \text{ then} \\
& \quad \quad s_{i1} = s_i ; \quad s_{i2} = d_i \\
& \quad \quad \text{else} \\
& \quad \quad \quad s_{i1} = d_i ; \quad s_{i2} = s_i \\
& \quad \quad \text{end if} \\
& \text{end if}
\end{aligned} \quad (7)$$

The initial values of  $d_i$  are set for all variables at the beginning of the optimization. Experience so far shows that values around 1-5 are suitable to guarantee good initial performance.

Zero variance can occur when all variables of  $x_i$  in the archive are identical. In this case the previous non-zero value is used further. Sometimes the variance can oscillate over a wide range. By using the factor  $d_i$  instead of  $s_i$  which is a function of variance a smoothing effect is achieved.

Furthermore, the asymmetrical properties of the mapping functions are utilized.

The mean and variance are not calculated before the archive is filled up completely. In this stage searching is performed with  $s_{i1}=s_{i2}=0$  which corresponds with a straight line between zero and one as the mapping function. Usually an archive size of 2-5 is sufficient. A larger archive size will result in a rather conservative searching with orientation on the saved best populations.

As parent of the new population the best population saved in the archive (first position) is used. Then a few variables are selected and reflected on the mapping function. Four alternative selection methods are currently implemented (Fig. 4).

- 1) Random selection
- 2) Neighbor group selection: moving the group forward in block steps
- 3) Neighbor group selection: moving the group forward in single steps
- 4) Sequential selection of the first variable and selection of the rest randomly.

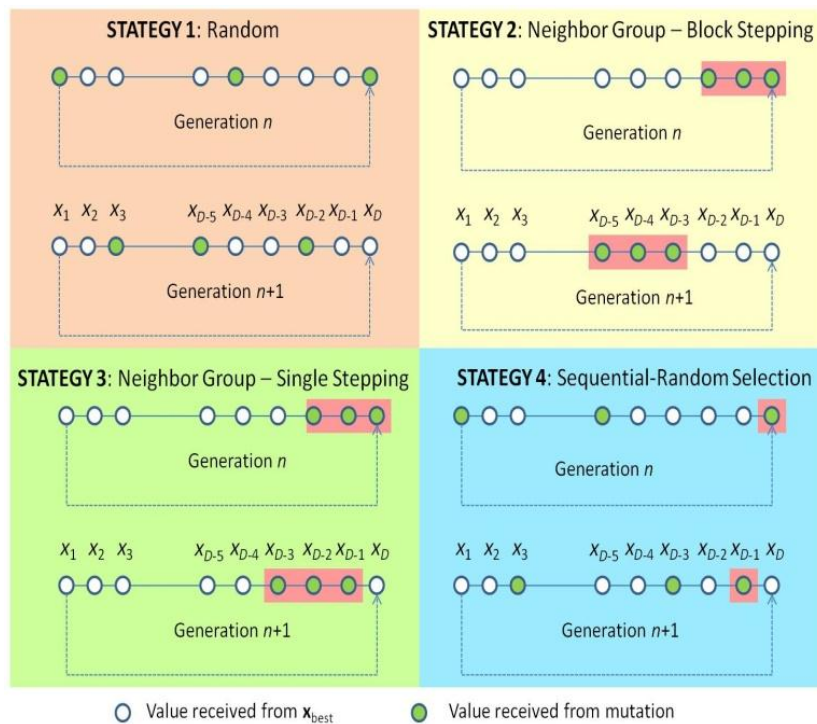


Fig. 4. Selection strategies

The flowchart of the MVMO algorithm is shown in Fig. 5.

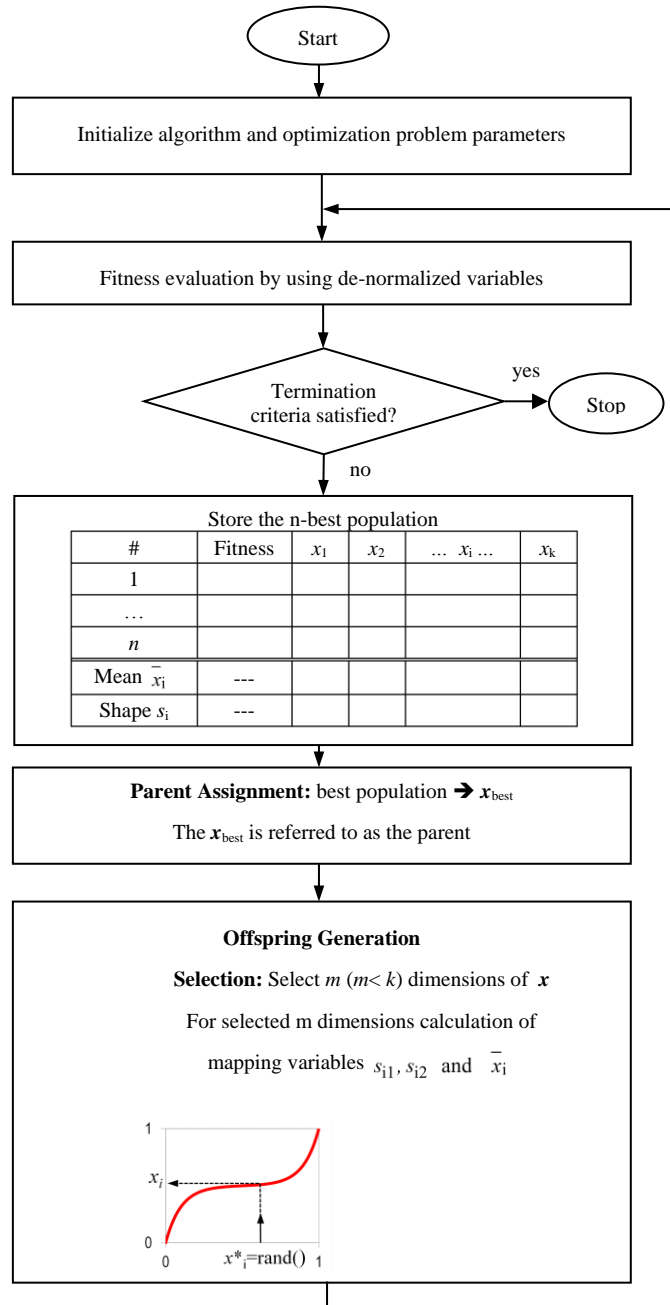


Fig. 5. Flowchart of the MVMO algorithm

Initial parameter settings required for MVMO are:

- Size of archive
- Number of variables selected simultaneously
- Selection method
- FS\_Factors for shape scaling
- Increment  $\Delta d$
- Initial value of  $d_i$
- maximum number of iterations

## New extensions to MVMO:

The following extensions have included into the MVMO-algorithm:

### Variable FS factor:

$$f_s = f_s^* \cdot (1 + rand) \quad (8)$$

where  $f_s^*$  denotes the smallest value of  $f_s$  and  $rand$  is a random number in the range [0, 1]. When the accuracy of the optimization needs to be improved, the following extension can also be used in order to allow a progressive increase of  $f_s^*$ :

$$f_s^* = f_{s\_ini}^* + \left( \frac{i}{i_{final}} \right)^2 (f_{s\_final}^* - f_{s\_ini}^*) \quad (9)$$

The variable  $i$  represents the iteration number. For  $f_{s\_final}^* = f_{s\_ini}^*$  the  $fs$  factor is fixed. The recommended range is:  $f_{s\_ini}^* = 0.9 \dots 1.0$   $f_{s\_final}^* = 1.0 \dots 3.0$

### Variable Increment $\Delta d$ :

The factor  $\Delta d$  is used in the MVMO algorithm as show below

$$\begin{aligned} & s_{i1} = s_{i2} = s_i \\ & \text{if } s_i > 0 \text{ then} \\ & \Delta d = (1 + \Delta d_0) + 2 \cdot \Delta d_0 \cdot (rand - 0.5) \\ & \quad \text{if } s_i > d_i \\ & \quad \quad d_i = d_i \cdot \Delta d \\ & \quad \text{else} \\ & \quad \quad d_i = d_i / \Delta d \\ & \quad \text{end if} \\ & \text{if } rand \geq 0.5 \text{ then} \\ & \quad s_{i1} = s_i ; \quad s_{i2} = d_i \\ & \quad \text{else} \\ & \quad \quad s_{i1} = d_i ; \quad s_{i2} = s_i \\ & \quad \text{end if} \\ & \text{end if} \end{aligned} \quad (10)$$

At every iteration, each  $d_i$  is scaled up or down with the factor  $\Delta d$ . If  $d_i > s_i$ , the current  $d_i$  is divided by  $\Delta d$  which is always larger than 1.0 and thus leads to reduced value of  $d_i$ . In case  $d_i < s_i$ ,  $d_i$  will be multiplied by  $\Delta d$  resulting in increased  $d_i$ . In this way  $d_i$  will always oscillate around the current

shape factor  $s_i$ . Moreover,  $\Delta d$  is varied randomly around the value  $(1+\Delta d_0)$  with decreasing amplitude of  $\Delta d_0$  according to.

$$\Delta d_0 = \Delta d_0^{\text{ini}} + \left( \frac{i}{i_{\text{final}}} \right)^2 \left( \Delta d_0^{\text{final}} - \Delta d_0^{\text{ini}} \right) \quad (11)$$

The recommended range for the bandwidth factor  $\Delta d_0$  is  $0.01 \leq \Delta d_0 \leq 0.4$ .

## **Hybrid variant: MVMO-SH**

### **General procedure**

The motive behind the hybrid variant of the Mean-Variance Optimization Method, denoted by MVMO-SH, is to further exploit the innate power of global searching of the former MVMO. The flowchart of MVMO<sup>S</sup> is schematically depicted in Fig. 6. The overall search process is initiated by a set of NP particles, each one having its own memory represented by the corresponding archive and mapping functions. As shown in the figure, a set of two nested loops is entered after the initialization stage. The outer loop is terminated once the termination criterion is satisfied. For each particle, the inner loop encompasses the steps corresponding to fitness evaluation or local search, update of the solution archive, fitness-based classification of good and bad particles, parent selection, and offspring generation.

### **Fitness evaluation and local search**

The elements of the candidate solution vector (i.e. optimization variables) are de-normalized from [0, 1] range to their original [min, max] boundaries before fitness evaluation or local search is performed. In case of unconstrained optimization problems, the fitness corresponds with the value of the objective function associated to the candidate solution being evaluated, whereas for constrained problems, it should include a penalty value resulting from constraint violation.

An interior-point method (IPM) based strategy is included in this step as a local improvement option. After a given number of fitness evaluations, IMP is performed with a probability  $\gamma$  for any child of the population. Experience achieved so far suggest using  $\gamma = 1/100/D$ , where D is the problem dimension.

### **Solution archive**

Each particle has an associated compact and continually updated solution archive, which stores its n-best achieved offsprings in a descending order of fitness and serves as the knowledge base for guiding the search direction (i.e. adaptive memory). The archive size is fixed for the entire process.

For each particle, and after every fitness evaluation/local search, an update of its archive takes place only if the new solution is better than those in the archive. An illustration of the archive is shown in Fig. 7.

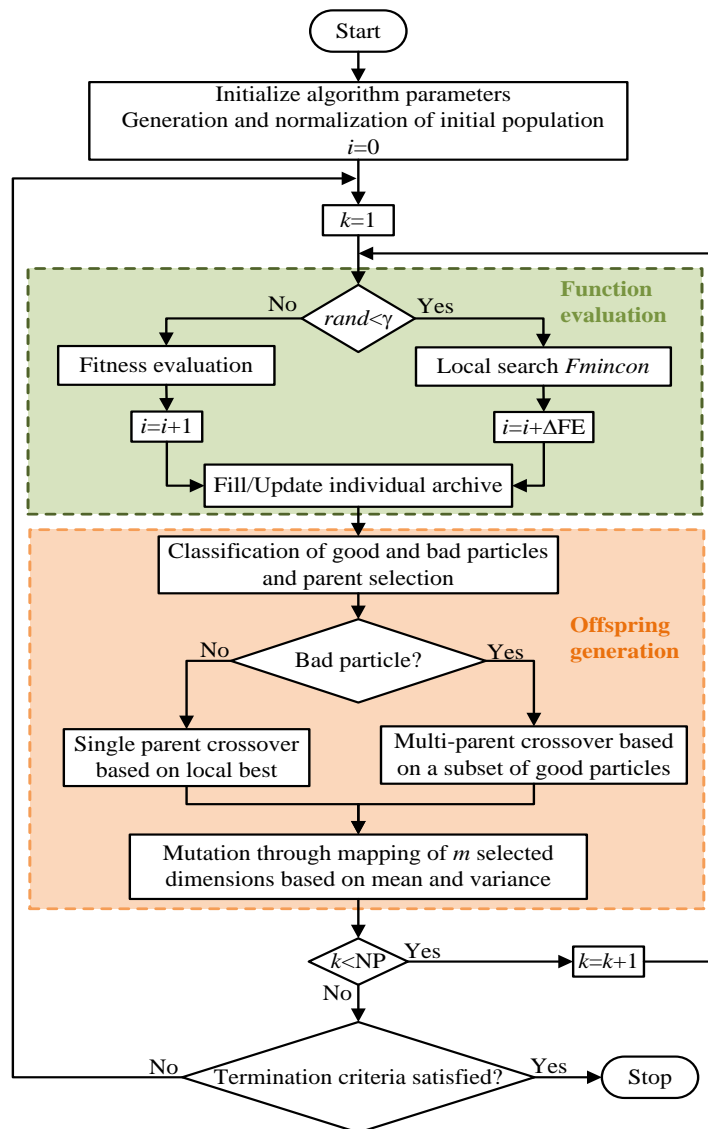


Fig. 6. MVMO-SH overall procedure. The fitness evaluation and particle counters are denoted by  $i$  and  $k$ , whereas NP and  $\Delta FE$  represent total number of particles and number of fitness evaluations required by local search, respectively. The term  $rand$  represents a random uniform number between 0 and 1.

Ranking	Fitness	$x_1$	$x_2$	...	$x_D$
1st best	$F_1$				
2nd best	$F_1$		<b>Optimization Variables</b>		
...					
Last best	$F_A$				
Mean	---		$\bar{x}_1$	$\bar{x}_2$	...
Shape	---	$s_1$	$s_2$	...	$s_D$

Fig. 7. Structure of the solution archive.



## Parent selection

All NP particles are ranked according to their local best fitness and classified into two groups: A set of  $GP$  “good particles”, and the set of NP- $GP$  “bad particles”.  $GP$  is dynamically determined throughout the search process as follows:

$$GP = \text{round}\left(NP \cdot g_p^*\right) \quad (12)$$

$$g_p^* = g_{p\_ini}^* + \frac{i}{i_{\text{final}}} \left( g_{p\_final}^* - g_{p\_ini}^* \right) \quad (13)$$

where  $i$  denotes fitness evaluation number. The equation (12) is not calculated in the initial stage of the search process, where each particle is evaluated independently. In this way,  $GP$  is linearly narrowed down following the decrease from  $g_{p\_ini}^*$  to  $g_{p\_final}^*$ . For the current application, it is considered that  $g_{p\_ini}^*=0.7$  and  $g_{p\_final}^*=0.2$ .

During the early stage of the search process, each particle is independently evaluated for a few (at least two) function evaluations, and the solution that produced the local best fitness achieved so far (i.e. the one corresponding to the first position in its particular solution archive) is chosen as the parent for the next offspring. Thereafter, the local best-based parent assignment is further adopted for each particle classified as good, whereas, by contrast, for each bad particle  $\mathbf{x}_p$ , the parent  $\mathbf{x}_p^{\text{parent}}$  is synthesized by using a multi-parent strategy as follows:

$$\mathbf{x}_p^{\text{parent}} = \mathbf{x}_k + \beta \left( \mathbf{x}_i - \mathbf{x}_j \right) \quad (14)$$

where  $\mathbf{x}_i$ ,  $\mathbf{x}_j$  and  $\mathbf{x}_k$  represent the first (global best), the last, and a randomly selected intermediate particle of the group of good particles, respectively. The vector of mean values associated to  $\mathbf{x}_p$ , which are required for mutation and mapping by the so called h-function is also set to  $\mathbf{x}_p^{\text{parent}}$ . The factor  $\beta$  is a normally distributed number (with mean  $\mu$  and  $\sigma$  standard deviation), which is re-drawn if any element of the vector  $\mathbf{x}_p^{\text{parent}}$  is outside the range  $[0, 1]$ .

## Crossover

In the next generation, the particle's child vector  $\mathbf{x}^{\text{new}} = [x_1, x_2, x_3, \dots, x_D]$  is created by combining a subset of  $D-m$  directly inherited dimensions from  $\mathbf{x}^{\text{parent}}$  vector (i.e. crossover) and  $m$  selected dimensions that undergo mutation operation through mapping function based on the means and variances calculated from the particle's solution archive. The above-described procedure for parent assignment and offspring creation is graphically illustrated in Fig. 8.

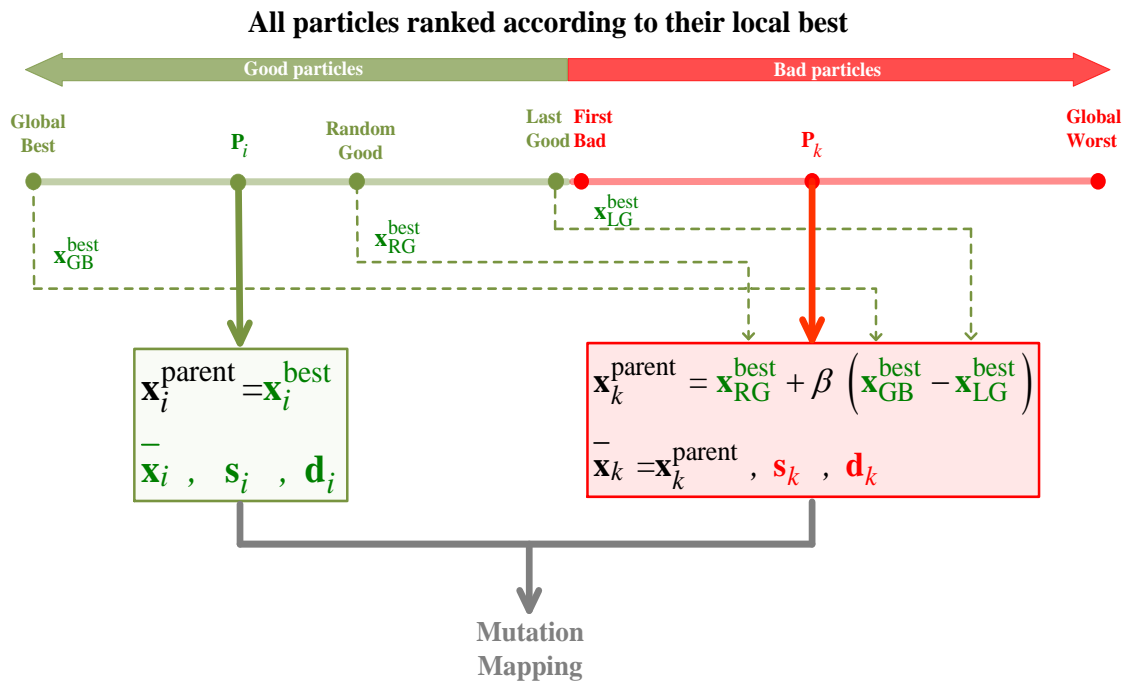


Fig. 8 Offspring creation in MVMO-SH.

**Consider: if the number of initial particles is chosen as NP=1, MVMO-SH will perform as standard MVMO.**