Technical Report

Solving the CEC2016 Real-Parameter Single Objective Optimization Problems through MVMO-PHM

José L. Rueda, Senior Member, IEEE
Department of Electrical Sustainable Energy
Delft University of Technology
Delft, The Netherlands
j.l.ruedatorres@tudelft.nl

István Erlich, Senior Member, IEEE
Institute of Electrical Power Systems
University Duisburg-Essen
Duisburg, Germany
istvan.erlich@uni-due

I. INTRODUCTION

This report provides a summary concerning the application of the Mean-Variance Mapping Optimization (MVMO) algorithm on two of the tests beds of the IEEE CEC 2016 Special Session & Competitions on Real-Parameter Single Objective Optimization:

i) Bound Constrained Single-Objective Computationally Expensive Numerical Optimization [1].

ii) Learning-based Real-Parameter Single Objective Optimization [2].

In order to differentiate from previous variants of MVMO, the acronym MVMO-PHM is used henceforth. P denotes the population based approach, H the hybridization of the algorithmic framework to include local search strategy, and M the adoption of a new mapping function.

The report is organised as follows: Section II recapitulates the main features behind the algorithmic procedure of MVMO and overviews a new mapping function. Sections III and IV show the numerical results concerning the solution of the computationally expensive problems, and the learning-based problems, respectively, followed by concluding remarks in section V.

II. A SHORT OVERVIEW OF MVMO-PHM

The overall algorithmic procedure of MVMO-PHM is shown in Fig. 1. For the sake of brevity, only a summary of the main features is provided below, and interested readers can find a detailed description of each step in [3].

- Firstly, the parameters of MVMO-PHM are initialized, and a set of N_P candidate solutions is randomly sampled within the min-max bounds of the optimization variables, which are simultaneously normalized from [min, max] into [0, 1], because the evolutionary mechanism of MVMO-PHM is performed within a normalized search space. The list of parameters is given in Subsection II.E.
- The optimization variables are only de-normalized for computing the fitness value or launching local search. The fitness value corresponds with the error value presented in Sections III and IV. No strategy is needed to ensure fulfilment of bound constraints, since MVMO-PHM always generates new values for each optimization variable within [0,1].
- Local search is based on interior-point method (IPM) or sequential quadratic programming (SQP) and is launched after a given number of fitness evaluations $i_{\text{loc\_start}}$ for any new solutions belonging to the group of good particles. A particle is selected for local search on a given probability γ_{LS}, that is

$$rand < \gamma_{\text{LS}} \tag{1}$$

subject to:

$$\alpha_{\text{LS\_min}} < \alpha < \alpha_{\text{LS\_max}}, \quad \alpha = i / i_{\max} \tag{2}$$

where i stands for fitness evaluation number and $\alpha_{\text{LS\_min}}, \alpha_{\text{LS\_max}}$ represents the range scheduled for local search. *rand* denotes a uniform randomly generated number within the range [0,1].
- The normalized evolutionary mechanism of MVMO-PHM comprises of: i) Filling-up and updating a set of solution archives containing evolved candidate solution from the initial population; ii) Classification of newly generated solutions into either good (to wit the smallest fitness values so far) or bad solutions (relatively higher fitness values); ii) Determination of a parent solution, from which a new (child) solution is generated; and iv) Mutation of *m* selected dimensions of the parent solution through application of the so-called mapping function.

A. Solution archive: a dynamic knowledge database

The solution archive constitutes a knowledge base that is updated whenever an improvement of the fitness value is achieved. It stores the n-best child solutions achieved so far in a descending order of fitness. The solution archive has a fixed size for the entire search process. As illustrated in Fig. 2, it records statistical measures like the mean $\overline{x}_i$, shape s_i, and d-factor d_i associated with each optimization variable. These measures are recalculated whenever an update of the archive

takes place. However, for the calculation of mean a weighting of the old and new values is used according to (3)

$$\bar{x}_{\text{update}} = 0.1 \cdot \bar{x}_{\text{old}} + 0.9 \cdot \bar{x}_{\text{new}} \quad (3)$$

The initial value $\bar{x}_{\text{ini}}$ is set in this study to 0.5, that means exactly in the middle of the search space. Other options like random or user defined definitions are also possible. Similarly, the initial values and the updating procedure of s_i, and d-factor d_i represent additional tuning factors allowing some adaptation to the function to be optimized. However, in this study the settings $s_i = 0$, $d_i = 1$ and updating these values based on the archive only without weighting old and current values, have been used. All these parameters influence the the shape of the mapping function, which allows changing the search emphasis from exploration to exploitation.

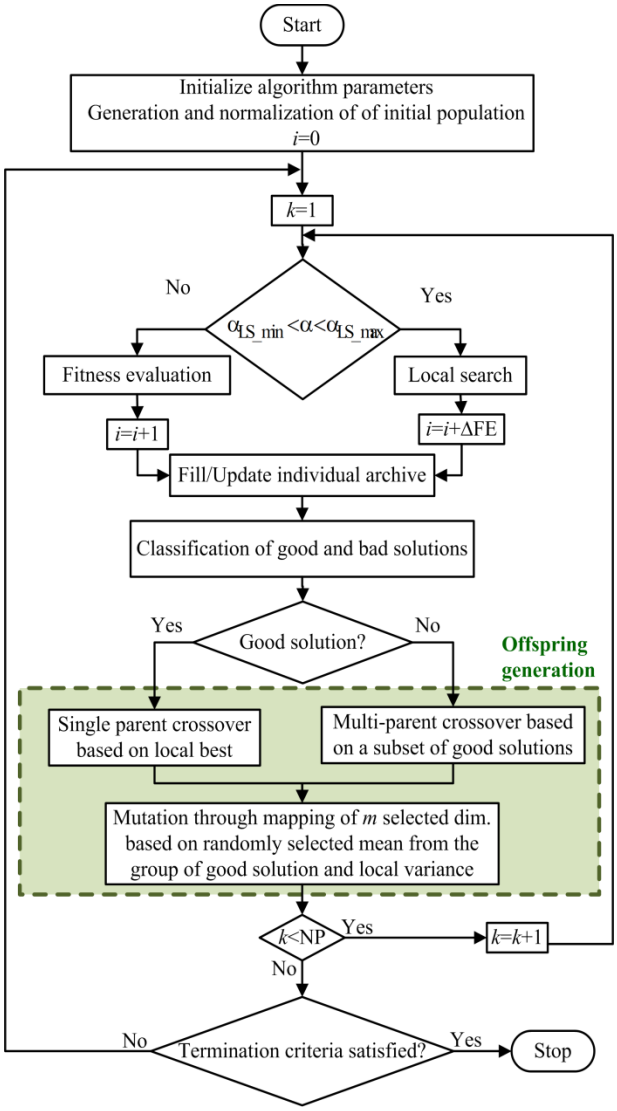

Fig. 1. Algorithmic procedure of MVMO-PHM. The fitness evaluation and candidate solution counters are denoted by *i* and *k*, whereas N_P, ΔFE, and *rand* stand for number of candidate solutions, number of fitness evaluations, and uniform random number between [0, 1], respectively.

B. *Classification and determination of parent solution*

Initially, each candidate solution is independently generated and evaluated for at least two runs. The solution with the individual best fitness achieved so far (i.e. the first ranked position in its particular solution archive) is chosen as the parent for next generation solutions.

Afterwards, the scheme shown in Fig.3 is used to classify the candidate solutions into the set of *GP* "good" solutions, or the set of *Np-GP* "bad" solutions. The classification is based on the individual best fitness achieved so far, i.e. ranking from the smallest to the largest fitness value (minimization problem).

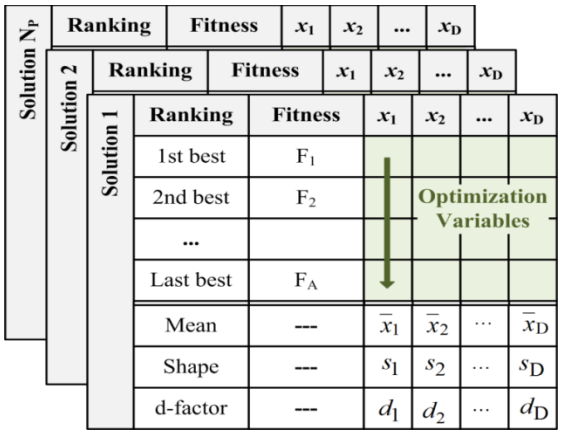

Fig. 2. Layour of the set of solution archives.

`Fig. 3 also illustrates that the evolution of each "good" solution starts by picking up the best ranked solution of the corresponding solution archive as a parent $\mathbf{x}_k^{\text{parent}}$, whereas for the evolution of a "bad" solution $\mathbf{x}_k$, a multi-parent criterion is applied in order to determine $\mathbf{x}_k^{\text{parent}}$:

$$\mathbf{x}_k^{\text{parent}} = \mathbf{x}_{\text{RG}}^{\text{best}} + \beta\left(\mathbf{x}_{\text{GB}}^{\text{best}} - \mathbf{x}_{\text{LG}}^{\text{best}}\right) \quad (4)$$

where $\mathbf{x}_{\text{GB}}^{\text{best}}$ represents one of the best solutions selected from a small group of global best solutions. The group includes initially five members which is reduced to one (global best) in the final stage. $\mathbf{x}_{\text{LG}}^{\text{best}}$ and $\mathbf{x}_{\text{RG}}^{\text{best}}$ are the last and a randomly selected intermediate solutions in the group of good solutions, respectively. In this selection the border between "good" and "bad" particles is not fixed a priori, but exhibits a small variation.

Note also in Fig. 3 that the vector of mean values associated with $\mathbf{x}_k$, which are required for subsequent mutation via the mapping function, is determined randomly from the set of good solutions. For both "good" and "bad" solutions, the shape variables s_i and d_i are synthesized based on the corresponding solution archive. The factor β is computed as follows:

$$b = 1.1 + (rand - 0.5) \cdot 2.0$$
$$\beta = \delta \cdot 3.0 \cdot b \cdot \left(1.0 + 2.5 \cdot \alpha^2\right) \cdot rand - \left(1 - \alpha^2\right) \cdot 0.8 \quad (5)$$

where δ is a tuning parameter and can vary in the range of [0.1 – 20]. The factor β is re-drawn and (5) is recalculated for any element of $\mathbf{x}_k^{\text{parent}}$ going outside the range [0, 1].

The relative number *GP* of solutions belonging to the group of good solutions is determined throughout the search process as follows:

$$GP = \text{round}\left(\text{N}_\text{P} \cdot g_p^*\right) \quad (6)$$

$$g_p^* = \text{g}_{\text{p\_ini}}^* - \alpha^2 \left(\text{g}_{\text{p\_final}}^* - \text{g}_{\text{p\_ini}}^*\right) \quad (7)$$

This value is varied in a small range of 15% around g_p^* by using a simplified normal distribution function.

Equation (7) is not calculated in the initial stage of the search process, where each solution is evaluated independently. Note that *GP* is linearly narrowed down following the decrease from $g_{\text{p\_ini}}^*$ to $g_{\text{p\_final}}^*$.

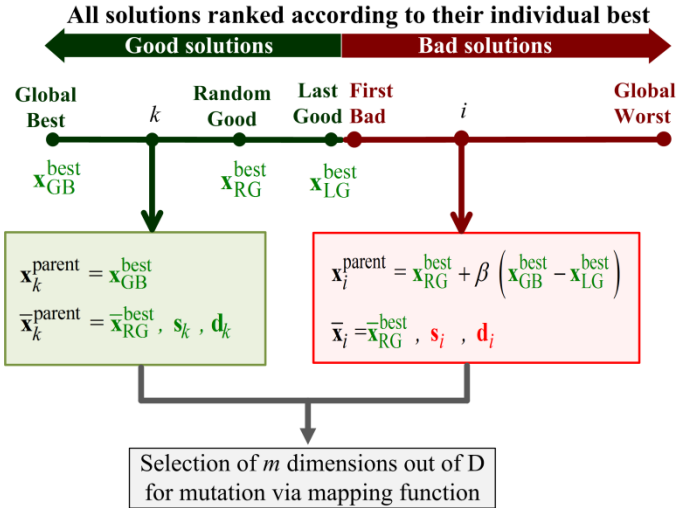

Fig. 3. Procedure for parent selection in MVMO-SH.

C. Selection of variables for mutation operation

Finally, for each candidate solution, a child vector (array) $\mathbf{x}^{\text{new}} = [x_1, x_2, x_3, \ldots, x_\text{D}]$, where D is the number of problem dimensions, is created by performing mutation operation on *m* selected dimensions of $\mathbf{x}_\text{p}^{\text{parent}}$. This operation basically consists of applying the mapping function based on the actual values of the parameters $\bar{x}_i$, s_i, and d_i associated with each solution. The value of *m* is determined throughout the search process as follows:

$$m = \text{round}\left(\text{m}_{\text{final}} + irand\left(m^* - \text{m}_{\text{final}}\right)\right) \quad (8)$$

$$m^* = \text{round}\left(\text{m}_{\text{ini}} - \alpha^2\left(\text{m}_{\text{ini}} - \text{m}_{\text{final}}\right)\right) \quad (9)$$

where *irand()* denotes a random integer number generated in the range between zero and the value given in the brackets. The selection of the *m* variables to be mutated is done by using the random-sequential strategy described in [4].

D. Mutation based on new mapping function

The new value of each selected dimension x_r of $\mathbf{x}^{\text{new}}$ is determined, based on the classical mapping function (Mapping #1 in the following), by

$$x_\text{r} = h_\text{x} + (1 - h_1 + h_0) \cdot x_\text{r}^* - h_0 \quad (10)$$

where x_r^* is a randomly generated number with uniform distribution between [0, 1]. The term *h* represents the mapping function that is used for mutation operation, whereas h_x, h_1 and h_0 are the outputs of the mapping function calculated as:

$$h_\text{x} = h(x = x_\text{r}^*), \quad h_0 = h(x = 0), \quad h_1 = h(x = 1) \quad (11)$$

where

$$h(\bar{x}, s_1, s_2, x) = \bar{x} \cdot (1 - e^{-x \cdot s_1}) + (1 - \bar{x}) \cdot e^{-(1-x) \cdot s_2} \quad (12)$$

From (10), (11), and (12), it can be noticed that x_r is always within the range [0, 1]. s_r is the shape factor calculated as follows:

$$s_\text{r} = -\ln(v_\text{r}) \cdot f_\text{s} \quad (13)$$

where v_r is the variance computed from the stored values of x_r in the solution archive, and f_s is a scaling factor.

Alternatively, two new mapping functions have been developed and used in this paper. Mapping #2 is based on Mapping #1, but the function is defined slightly differently in the two halves of the random variable x_r^*. In this way the function crosses the mean value $\bar{x}$ always at $x_\text{r}^* = 0.5$. Consequently, the probability of the optimization variable decreasing or increasing is the same in both directions.

Mapping #2

for $x_\text{r}^* < 0.5$	*for* $x_\text{r}^* \geq 0.5$
$s_1^* = s_1 \,/\, (1 - \bar{x})$	$s_2^* = s_2 \,/\, \bar{x}$
$h_\text{m} = \bar{x} \cdot \left(1 - e^{-0.5 \cdot s_1^*}\right)$	$h_\text{m} = (1 - \bar{x}) \cdot e^{-0.5 \cdot s_2^*}$
$h_\text{f} = \bar{x} \cdot \left(1 - e^{-x_\text{r}^* \cdot s_1^*}\right)$	$h_\text{b} = (1 - \bar{x}) \cdot e^{-(1 - x_\text{r}^*) \cdot s_2^*} + \bar{x}$
$h_\text{c} = (\bar{x} - h_\text{m}) \cdot 2 \cdot x_\text{r}^*$	$h_\text{c} = h_\text{m} \cdot 2 \cdot (1 - x_\text{r}^*)$
$x_\text{r} = h_\text{f} + h_\text{c}$	$x_\text{r} = h_\text{b} - h_\text{c}$

(14)

The same idea was followed in the function Mapping #3 but additionally the exponential functions were replaced by hyperbolic functions.

$$
\begin{array}{ll}
\underline{if \quad x_r^* < 0.5} & \underline{if \quad x_r^* \geq 0.5} \\
s_1^* = s_1 / \left(1-\bar{x}\right) & s_2^* = s_2 / \bar{x} \\
h_m = \bar{x} - \dfrac{\bar{x}}{\left(0.5 \cdot s_1^* + 1\right)} & h_m = \dfrac{\left(1-\bar{x}\right)}{\left(0.5 \cdot s_2^* + 1\right)} \\
h_f = \dfrac{-\bar{x}}{\left(x_r^* \cdot s_1^* + 1\right)} + \bar{x} & h_b = \dfrac{1-\bar{x}}{\left(\left(1-x_r^*\right) \cdot s_2^* + 1\right)} + \bar{x} \\
h_c = \left(\bar{x} - h_m\right) \cdot 2 \cdot x_r^* & h_c = h_m \cdot 2 \cdot \left(1-x_r^*\right) \\
x_r = h_f + h_c & x_r = h_b - h_c
\end{array}
\qquad (15)
$$

The three mapping functions are illustratively compared with one another in Fig. 4 a) and b) for different mean and shape factors. Note that by using the new function Mapping #2 and Mapping #3 the mean value is always reached at 0.5 at the x-axis. Therefore, the probability of generating a new value of x_r that is greater or smaller than the mean value is equal. This is not the case in the classical mapping (Mapping #1). For the same shape value s_{r1} and s_{r2} the slope of the Mapping #3 is higher than that of Mapping #1 and Mapping #2 around the mean value. Thus, it allows a higher global search capability. On the other hand, for performing local search Mapping #2 is more favorable. In the optimization task presented in this paper initially Mapping #3 was used. After the first local search run using SQP or IPM algorithms the mapping was switched to Mapping #2.

In the first evaluation of every solution, the mean $\bar{x}_r$ is set to the predefined initial value (0.5 in this study) and the variance v_r is set to 1.0 which corresponds with s_r=0. But as the optimization progresses, they are recalculated after every update of the particle's solution archive for each selected optimization variable. Both input and output of the mapping function cover the range [0, 1]. From (10) and (15), it can be observed that the shape of the mapping function is influenced by the mean $\bar{x}_r$ and shape factors s_{r1} and s_{r2}. So, the search diversity can be enhanced through proper variation of the shape factors. To this end, the scaling factor f_s can be additionally used to change the shape of the function. Thus, f_s is increased as the optimization progresses from a small initial value (e.g. $f_{s\_ini}^* = 1$) up to a higher final value (e.g. $f_{s\_final}^* = 20$) by using (16) and (17).

$$f_s^* = f_{s\_ini}^* + \alpha^2 \left(f_{s\_final}^* - f_{s\_ini}^* \right) \qquad (16)$$

$$f_s = f_s^* \left(1.0 + 4.0 \cdot \left(rand - 0.2\right)\right) \qquad (17)$$

The shape factors s_{r1} and s_{r2} of the variable x_r are assigned by using the procedure given in [7].

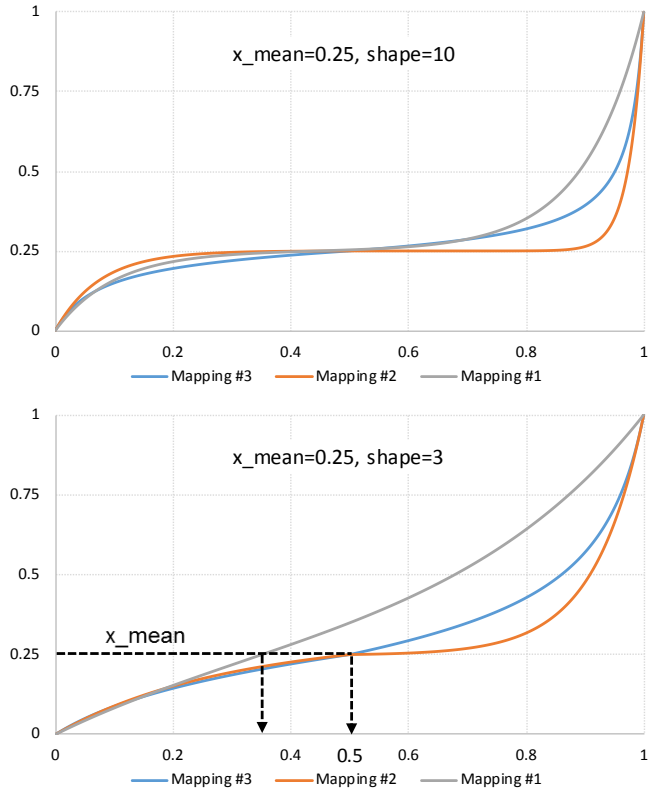

Fig. 4. Classical and the proposed mapping functions.

E. Summary of parameters of MVMO-PHM

The following set of parameters are required for the algorithm:

- Number of particles
- Archive size
- Initial and final *fs* factor $f_{s\_ini}^*$, $f_{s\_final}^*$
- Initial and final ratio of good particles $g_{p\_ini}^*$,$g_{p\_final}^*$
- Initial and final number of mutated variables (m_{ini}; m_{final})
- Local search probability γ_{LS}
- Initial/final bound of range of local search probability (α_{LS_min}, α_{LS_max})
- Factor δ in Eq. (5)

For each test function, the parameters of the algorithm were tuned by performing sensitivity analysis of the achieved fitness value under a single parameter change within 10 independent optimization runs.

The execution of local search by SQP or IPM may require performing tens, hundreds or even thousands of fitness evaluations. Thus, the use of this option is recommended for optimization problems that can be solved without considerable computing time constraints (i.e. large number of fitness evaluation budget), whereas, for optimization problems to be solved within reduced time (i.e. limited amount of function evaluations), the local search should be started earlier to have the chance of finishing the calculation within the range of the

evaluation budget, but not before MVMO-PHM was able to perform global search beforehand. The local search will always start with the best solution (first in the best solution group) and continues with the second and subsequent particles if enough evaluation budget is available.

Numerical experiments shown in the next sections III were performed on a computer with Intel® Core™ i7-3770 CPU, 3.4 GHz and 16 GB RAM, under Windows 8.1 pro, 64 bit OS. The implementation of MVMO-PHM was done in Matlab® Version R2014b and the functionalities of the Parallel Computing Toolbox are used to set a cluster with 7 cores to perform the optimization trials in a distributed manner. Stochastic integrity is guaranteed by performing independent initialization of random number streams on individual processes with respect to time plus the process identifier. The local search is performed by the IPM algorithm implemented in Matlab.

For the optimization task described in section IV the Fortran implementation of MVMO-PHM was used where the C-code for the function evaluation is embedded. In this way simulation time was reduced drastically compared with the Matlab version. For local search only a SQP algorithm implemented in the IMSL mathematical library was available.

III. Computationally Expensive Problems

The performance of MVMO-PHM is firstly evaluated on the IEEE-CEC 2016 benchmark problems on Bound Constrained Single-Objective Computationally Expensive Numerical Optimization. The problems are summarized in Table I. Detailed description can be found in [1].

Statistical tests on convergence performance and quality of final solution provided by MVMO were carried out under the following considerations:

- Problem dimension D = 10, 30.
- Search range: $[-100,100]^D$.
- Maximum number of function evaluations: 50*D.
- Repetitions of the optimization: 20 runs.
- Uniform random initialization within the search space. The random seed is based on time, which is done using the command rand('state', sum(100*clock)) in Matlab environment.
- The objective function is defined as the error value $OF=TF_i(x)-F_i^*$, where F_i^* is the theoretical global optimum of the i-th benchmark function TF is given in Table 1. The values of OF smaller than 1E-08 are taken as zero.
- The optimization is terminated upon completion of the maximum number of function evaluations.

The parameters of MVMO-PHM were tuned by minimizing the total score measure defined in [1]:

$$\text{Total Score} = \sum_{1}^{15} mean(f_a)|_{D=10} + \sum_{1}^{15} mean(f_a)|_{D=30} + \sum_{1}^{15} median(f_a)|_{D=10} + \sum_{1}^{15} median(f_a)|_{D=30} \quad (18)$$

where

$$f_a = 0.5\cdot\left(f_{\text{MaxFEs}} + f_{0.5\text{MaxFEs}}\right) \quad (19)$$

The chosen parameters for 10D and 30D dimensions are included and highlighted in the Matlab source code, which will be made available at [5] after the completion of the CEC2016.

TABLE I. IEEE-CEC 2016 Expensive Optimization Problems

Type	No.	Description	F_i^*
Unimodal function	TF1	Rotated Bent Cigar Function	100
	TF2	Rotated Discus Function	200
Simple Multimodal functions	TF3	Shifted and Rotated Weierstrass Function	300
	TF4	Shifted and Rotated Schwefel's Function	400
	TF5	Shifted and Rotated Katsuura Function	500
	TF6	Shifted and Rotated HappyCat Function	600
	TF7	Shifted and Rotated HGBat Function	700
	TF8	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	800
	TF9	Shifted and Rotated Expanded Scaffer's F6 Function	900
Hybrid function	TF10	Hybrid Function 1 (N=3)	1000
	TF11	Hybrid Function 2 (N=4)	1100
	TF12	Hybrid Function 3 (N=5)	1200
Composition function	TF13	Composition Function 1 (N=5)	1300
	TF14	Composition Function 2 (N=3)	1400
	TF15	Composition Function 3 (N=5)	1500

In Appendix 1, the statistical attributes of the error value OF (i.e. best, worst, mean, median, and standard deviation values) calculated after 20 runs are summarized for 10D and 30D (cf. Table III and Table IV).

The numerical linearization methods used by IPM method for local search in each function were different, i.e. they were selected by performing sensitivity analysis with respect to the best achieved solution so far. For most of the functions forward linearization was sufficient. However, for TF2, TF3, TF11, TF12 and TF15 central linearization was possible (it requires

twice as many function evaluations) and resulted in considerable improvement.

As the function evaluation budget is pretty much limited, the main task for MVMO-PHM is therefore to perform global search. In the final stage local search is started with particles in the ascending sequence of their finesses archived so far.

Additional observations are summarized in the following:

- *Unimodal functions*: MVMO-PHM was effective in finding near zero error values for OF in all runs for both 10D and 30D (in the order of 1E-04 and 1E-011). Thus, the local search strategy helped MVMO-PHM to overcome the narrow ridge property of TF1 and the sensitive direction property of TF2.

- *Simple multimodal functions*: For both 10D and 30D, the evolutionary mechanism of MVMO-PHM is effective in obtaining near zero error values for OF in all runs for TF5, TF6, and TF7 (in the order of 1E-01). This success is mainly attributed to the predominance of the evolutionary mechanism of MVMO-PHM over the launching of the local search strategy.

- *Hybrid and composite functions:* For 10D and 30D, MVMO-PHM was able to find OF values in the order of 10^0, 10^1, and 10^2. This highlights the possibility of using and alternative strategy for local search (e.g. based on heuristic algorithm) to overcome the high risk of being trapped into local optima within the complex (narrow, asymmetrical and multi-modal) search space of these functions.

IV. Learning-based Problems

MVMO-PHM is also tested on the IEEE-CEC 2016 learning-based benchmark problems on Learning-based Real-Parameter Single Objective Optimization. The problems are summarized in Table II. Detailed description can be found in [2]. Statistical tests on convergence performance and quality of final solution were performed under the following considerations:

- Problem dimension D = 10, 30.
- Search range: $[-100,100]^D$.
- Max. number of function evaluations: 10000*D.
- Optimization trials per problem: 51.
- Uniform random initialization within the search space. The random seed is based on time, which is done using the command rand('state', sum(100*clock)) in Matlab environment.
- The objective function is defined as the error value $OF=TF_i(x)-F_i^*$, where F_i^* is the theoretical global optimum of the i-th benchmark function TF given in Table II. The values of OF smaller than 1E-08 are taken as zero.
- The optimization is terminated upon completion of the maximum number of function evaluations.

The chosen parameters for 10D, 30D, 50D, and 100D dimensions are listed in separate files and will be made available after the completion at [5]. The MVMO-PHM parameters for solving the learning based problem were tuned by minimizing the total score defined as:

$$\begin{aligned}\text{Total Score} = &\sum_{1}^{15} mean(f_a)|_{D=10} + \sum_{1}^{15} mean(f_a)|_{D=30} + \\ &\sum_{1}^{15} mean(f_a)|_{D=50} + \sum_{1}^{15} mean(f_a)|_{D=100} + \\ &\sum_{1}^{15} median(f_a)|_{D=10} + \sum_{1}^{15} median(f_a)|_{D=30} + \\ &\sum_{1}^{15} median(f_a)|_{D=50} + \sum_{1}^{15} median(f_a)|_{D=100} \\ f_a = &0.5\cdot\left(f_{\text{MaxFEs}} + f_{0.5\text{MaxFEs}}\right)\end{aligned} \tag{20}$$

TABLE II. IEEE-CEC 2015 Expensive Optimization Problems

Type	No.	Description	F_i^*
Unimodal function	TF1	Rotated High Conditioned Elliptic Function	100
	TF2	Rotated Bent Cigar Function	200
Simple Multimodal functions	TF3	Shifted and Rotated Ackley's Function	300
	TF4	Shifted and Rotated Rastrigin's Function	400
	TF5	Shifted and Rotated Schwefel's Function	500
Hybrid function	TF6	Hybrid Function 1 (N=3)	600
	TF7	Hybrid Function 2 (N=4)	700
	TF8	Hybrid Function 3 (N=5)	800
Composition function	TF9	Composition Function 1 (N=3)	900
	TF10	Composition Function 2 (N=3)	1000
	TF11	Composition Function 3 (N=5)	1100
	TF12	Composition Function 4 (N=5)	1200
	TF13	Composition Function 5 (N=5)	1300
	TF14	Composition Function 6 (N=7)	1400
	TF15	Composition Function 7 (N=10)	1500

The statistical attributes of the error value OF (i.e. best, worst, mean, median, and standard deviation values), which were calculated after 51 runs, are summarized in Tables VI to IX in Appendix 2, for 10D, 30D, 50D, and 100D. The following remarks are drawn from these results:

• *Unimodal functions:* For all dimensions, MVMO-PHM is capable of finding zero error values (i.e. smaller than 1E-08) for OF in all runs. Besides, it was found out that convergence to these values was achieved shortly after the local search call. MVMO-PHM performs in this case just the global adjustment and the interior-point method used as local search function improves the optimization until zero error is reached. Therefore, MVMO-PHM constitutes a powerful tool to effectively tackle unimodal problems irrespective of dimensionality and the underlying mathematical features (e.g. asymmetrical, separable/non-separable).

• *Simple multimodal functions*: For TF3 most of the solutions converge to the local minimum around 20. However, in case of 10-D almost 50% of the solutions reach the global minimum of zero. TF4 and TF5 are successfully solved in all cases for 10D case. For the other dimensions, MVMO-PHM was able to find OF values in the order of 10^1, and 10^2.

• *Hybrid functions*: Even though the problems here are more challenging, MVMO-PHM is also capable of providing near zero error values, which are in the order of 10^{-2} and 10^{-5} for all functions in 10D case in almost all runs. The errors are in the order of 10^0 to 10^3 for the other dimensions in most optimization runs.

• *Solving composition functions*: For the TF11, TF13, and TF14, MVMO-PHM is capable of providing small fitness values in the range of 10^0 and 10^{-2} for both dimensions. For the remaining functions and evaluations in other dimensions, the errors are in the order of 10^2, and 10^3. TF15 converges always to the local minimum of 100.

V. CONCLUSIONS

This paper overviewed the features of MVMO-PHM, which is a new variant of MVMO algorithm. The upgraded version includes a new mapping function to enhance the capability of the algorithm to adaptively shift the search emphasis from exploration to exploitation throughout the search space. Numerical results on the test beds of the IEEE CEC 2016 Special Session & Competitions on Real-Parameter Single Objective Optimization evidenced the potential of MVMO-PHM to successfully tackle formulations of the test problems in different dimensions for both computationally expensive case (with limited computing budget) and learning based case (without limited computing budget). The addition of local search helped to improve the quality of obtained results. Nevertheless, it was also found out that this finding does not hold for every problem and every dimension. Thus, current research work is directed toward the development and addition of an alternative local search strategy into MVMO-PHM.

REFERENCES

[1] 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, Nov. 2014. [Online]. Available at: http://www.ntu.edu.sg/home/epnsugan/

[2] J.J. Liang1, B.Y. Qu, and P.N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2015 Competition on Learning-based Real-Parameter Single Objective Optimization," Technical Report, Nanyang Technological University (Singapore) and Zhengzhou University (China), Nov. 2014. [Online]. Available at: http://www.ntu.edu.sg/home/epnsugan/

[3] J.L. Rueda, Rueda, and I. Erlich, "Testing MVMO on learning-based real-parameter single objective benchmark optimization problems," in Proc. IEEE Congress on Evolutionary Computation (CEC), pp.1025-1032,Sendai, Japan, May 2015.

[4] I. Erlich, G. K. Venayagamoorthy, and W. Nakawiro, "A mean-variance optimization algorithm," 2010 IEEE Congress on Evolutionary Computation, pp.1-6, July 2010.

[5] http://www.ntu.edu.sg/home/epnsugan/

APPENDIX 1

TABLE III. RESULTS FOR 10D – COMPUTATIONALLY EXPENSIVE PROBLEMS

Type	Function	Best	Worst	Median	Mean	Std.
Unimodal functions	TF1	8.3211409E-02	2.8728670E-01	2.8505850E-01	2.5430055E-01	5.8003068E-02
	TF2	4.0330406E-11	1.2880426E-07	6.1277206E-11	1.6324708E-08	3.9157388E-08
Simple multimodal Functions	TF3	5.9060976E+00	9.6624855E+00	7.4263857E+00	7.5004925E+00	1.0016074E+00
	TF4	7.3368619E+01	4.6181011E+02	2.4769440E+02	2.6887479E+02	9.2222606E+01
	TF5	6.3665921E-01	1.8129485E+00	1.2461696E+00	1.1560639E+00	3.0766503E-01
	TF6	1.0385587E-01	7.1486387E-01	1.8770458E-01	2.6989016E-01	2.0604179E-01
	TF7	2.4625742E-01	1.0404678E+00	4.8502620E-01	5.4635208E-01	2.5928454E-01
	TF8	3.5381389E+00	4.3220248E+02	2.7482157E+01	6.7403535E+01	1.0623392E+02
	TF9	3.1944964E+00	4.2620638E+00	3.7730652E+00	3.7051547E+00	3.3805877E-01
Hybrid functions	TF10	1.5050711E+02	1.0117779E+03	5.6238626E+02	5.8750163E+02	2.3006930E+02
	TF11	6.2035435E+00	1.4113582E+01	9.5903659E+00	9.9614404E+00	2.0415214E+00
	TF12	5.1866707E+01	3.7442588E+02	2.6373327E+02	2.6566261E+02	8.3934150E+01
Composition functions	TF13	3.1537583E+02	3.1797911E+02	3.1683096E+02	3.1668844E+02	6.0639648E-01
	TF14	1.9158277E+02	2.0535900E+02	1.9699979E+02	1.9801811E+02	3.7411054E+00
	TF15	1.0170334E+01	5.9665913E+02	4.4609015E+02	4.1544852E+02	1.8272531E+02

TABLE IV. RESULTS FOR 30D – COMPUTATIONALLY EXPENSIVE PROBLEMS

Type	Function	Best	Worst	Median	Mean	Std.
Unimodal functions	TF1	2.4508713E-01	4.5354479E+02	5.0342901E+01	9.5961638E+01	1.3334466E+02
	TF2	6.3380412E-11	9.1904437E-07	2.2860121E-08	2.2860995E-07	3.4584672E-07
Simple multimodal Functions	TF3	1.9536502E+01	2.8419218E+01	2.4483993E+01	2.4728202E+01	2.2897036E+00
	TF4	4.5992486E+02	1.6658622E+03	1.0688505E+03	1.1718018E+03	3.6346325E+02
	TF5	8.1906523E-01	2.6612810E+00	1.4093392E+00	1.5234015E+00	4.2070096E-01
	TF6	1.3554694E-01	4.6837339E-01	4.1381007E-01	3.7958659E-01	9.9580431E-02
	TF7	3.3693884E-01	4.9970283E-01	4.4061227E-01	4.3328325E-01	6.4591599E-02
	TF8	7.1122001E+01	9.7627659E+02	3.1042876E+02	3.1063179E+02	2.6333317E+02
	TF9	1.1321078E+01	1.3543539E+01	1.3359465E+01	1.2998498E+01	6.8644221E-01
Hybrid functions	TF10	8.0364140E+02	8.1575535E+05	3.5386855E+03	1.1623419E+05	2.2539217E+05
	TF11	2.4050431E+01	1.3425059E+02	8.6649848E+01	8.0579215E+01	4.3257353E+01
	TF12	1.7199864E+02	7.7590854E+02	4.4869950E+02	3.9447239E+02	1.5164278E+02
Composition functions	TF13	3.4073786E+02	3.6769244E+02	3.5244270E+02	3.5409712E+02	8.4609156E+00
	TF14	2.4072486E+02	2.9560615E+02	2.6120415E+02	2.6183924E+02	1.4560999E+01
	TF15	7.6605495E+02	1.1207065E+03	9.4236409E+02	9.2841203E+02	1.1778148E+02

TABLE V. COMPUTATIONAL COMPLEXITY – COMPUTATIONALLY EXPENSIVE PROBLEMS $\hat{T}_1/\mathrm{T}_0$

Function	D=10	D=30
TF1	5.8663333e+01	1.0653750e+02
TF2	6.5066667e+01	1.2661875e+02
TF3	6.9066667e+01	1.4720000e+02
TF4	7.2266667e+01	1.6789375e+02
TF5	7.6233333e+01	1.8425000e+02
TF6	8.0256667e+01	2.0526875e+02
TF7	8.4433333e+01	2.2223750e+02
TF8	8.8570000e+01	2.3934375e+02
TF9	9.1963333e+01	2.5743750e+02
TF10	9.7100000e+01	2.7127500e+02
TF11	1.0131333e+02	2.8812500e+02
TF12	1.0484000e+02	3.0490000e+02
TF13	1.0843667e+02	3.2310625e+02
TF14	1.1170667e+02	3.4259375e+02
TF15	1.1549333e+02	3.6477500e+02

APPENDIX 2

TABLE VI. RESULTS FOR 10D - LEARNING-BASED PROBLEMS

Type	Function	Best	Worst	Median	Mean	Std.
Unimodal function	TF1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	TF2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Simple Multimodal functions	TF3	0.0000E+00	2.0000E+01	2.0000E+01	1.8753E+01	4.7593E+00
	TF4	4.9748E+00	1.2934E+01	8.9546E+00	8.8571E+00	1.9624E+00
	TF5	1.8736E-01	1.2210E+02	4.9630E+00	1.1242E+01	2.3380E+01
Hybrid function	TF6	5.0388E-05	1.4113E+00	3.7201E-02	1.8724E-01	3.2337E-01
	TF7	0.0000E+00	1.2362E-01	4.6557E-02	5.7009E-02	3.3812E-02
	TF8	4.8330E-02	1.1068E+00	4.6603E-01	4.5112E-01	2.5333E-01
Composition function	TF9	1.0011E+02	1.0021E+02	1.0017E+02	1.0017E+02	2.5078E-02
	TF10	2.1657E+02	2.1936E+02	2.1686E+02	2.1700E+02	4.4007E-01
	TF11	2.3533E+00	1.2002E+01	4.0322E+00	4.4207E+00	1.8370E+00
	TF12	1.0038E+02	1.0085E+02	1.0057E+02	1.0060E+02	1.0229E-01
	TF13	3.0424E-02	3.0530E-02	3.0530E-02	3.0499E-02	4.8427E-05
	TF14	1.6507E-01	1.0001E+02	1.0000E+02	8.4545E+01	3.6202E+01
	TF15	1.0000E+02	1.0000E+02	1.0000E+02	1.0000E+02	6.7424E-09

TABLE VII. RESULTS FOR 30D - LEARNING-BASED PROBLEMS

Type	Function	Best	Worst	Median	Mean	Std.
Unimodal function	TF1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	TF2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Simple Multimodal functions	TF3	2.0000E+01	2.0000E+01	2.0000E+01	2.0000E+01	9.2423E-06
	TF4	5.0743E+01	7.6612E+01	6.4672E+01	6.4731E+01	4.9645E+00
	TF5	3.9109E+02	1.5479E+03	9.2756E+02	9.2639E+02	2.8564E+02
Hybrid function	TF6	2.2559E+01	5.8824E+02	1.6542E+02	2.0603E+02	1.4476E+02
	TF7	1.6538E+00	4.6952E+00	2.9792E+00	2.9787E+00	6.2297E-01
	TF8	3.4020E+00	1.8667E+02	3.7396E+01	5.3847E+01	4.8097E+01
Composition function	TF9	1.0225E+02	1.0269E+02	1.0249E+02	1.0248E+02	1.0243E-01
	TF10	1.5067E+02	7.3032E+02	4.0158E+02	4.0462E+02	1.5044E+02
	TF11	3.0075E+02	3.0225E+02	3.0120E+02	3.0126E+02	3.2430E-01
	TF12	1.0284E+02	1.0369E+02	1.0329E+02	1.0329E+02	2.0163E-01
	TF13	2.5733E-02	2.6533E-02	2.6102E-02	2.6012E-02	2.4567E-04
	TF14	1.0004E+02	1.0682E+02	1.0073E+02	1.0113E+02	1.2044E+00
	TF15	1.0000E+02	1.0000E+02	1.0000E+02	1.0000E+02	2.4776E-09

TABLE VIII. RESULTS FOR 50D - LEARNING-BASED PROBLEMS

Type	Function	Best	Worst	Median	Mean	Std.
Unimodal function	TF1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	TF2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Simple Multimodal functions	TF3	2.0000E+01	2.0000E+01	2.0000E+01	2.0000E+01	1.0641E-06
	TF4	7.2350E+00	1.7909E+01	1.1940E+01	1.1919E+01	2.5241E+00
	TF5	1.6763E+03	3.5109E+03	2.5017E+03	2.5701E+03	4.8603E+02
Hybrid function	TF6	1.5431E+02	1.5200E+03	8.7363E+02	8.9832E+02	2.8725E+02
	TF7	6.1937E+00	7.8602E+01	9.9262E+00	2.2399E+01	1.9434E+01
	TF8	1.7823E+01	7.6933E+02	2.5984E+02	2.9790E+02	1.9116E+02
Composition function	TF9	1.0364E+02	1.0432E+02	1.0392E+02	1.0393E+02	1.3631E-01
	TF10	8.2123E+02	1.7482E+03	1.1439E+03	1.1732E+03	1.9946E+02
	TF11	3.0339E+02	3.1458E+02	3.0624E+02	3.0652E+02	2.1144E+00
	TF12	1.0541E+02	1.0671E+02	1.0592E+02	1.0599E+02	3.0089E-01
	TF13	7.0535E-02	8.2876E-02	7.6515E-02	7.6399E-02	3.9205E-03
	TF14	1.0044E+02	1.8948E+02	1.1618E+02	1.2211E+02	1.8517E+01
	TF15	1.0000E+02	1.0000E+02	1.0000E+02	1.0000E+02	1.9510E-09

TABLE IX. RESULTS FOR 100D - LEARNING-BASED PROBLEMS

Type	Function	Best	Worst	Median	Mean	Std.
Unimodal function	TF1	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	TF2	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Simple Multimodal functions	TF3	2.0000E+01	2.0000E+01	2.0000E+01	2.0000E+01	3.8817E-06
	TF4	3.3829E+01	5.7708E+01	4.8753E+01	4.8857E+01	5.1027E+00
	TF5	6.8683E+03	9.9922E+03	8.1923E+03	8.2841E+03	7.5778E+02
Hybrid function	TF6	2.0366E+03	4.8072E+03	3.3964E+03	3.4239E+03	5.7921E+02
	TF7	2.5342E+01	1.6832E+02	1.2026E+02	1.2097E+02	4.4926E+01
	TF8	8.7651E+02	2.7128E+03	1.6358E+03	1.6830E+03	4.4559E+02
Composition function	TF9	1.0632E+02	1.0760E+02	1.0694E+02	1.0699E+02	2.5313E-01
	TF10	2.3402E+03	4.0964E+03	3.1021E+03	3.1876E+03	4.6233E+02
	TF11	7.8725E+02	1.3156E+03	9.9640E+02	1.0035E+03	1.0788E+02
	TF12	1.1230E+02	2.0041E+02	1.1318E+02	1.1655E+02	1.7113E+01
	TF13	6.0247E-02	6.4212E-02	6.2115E-02	6.2232E-02	1.0243E-03
	TF14	1.0051E+02	4.3353E+03	2.0763E+03	1.9911E+03	1.1167E+03
	TF15	1.0000E+02	1.0000E+02	1.0000E+02	1.0000E+02	2.8000E-09

TABLE X. COMPUTATIONAL COMPLEXITY - LEARNING-BASED PROBLEMS

Dimension	T_0	T_1	$\hat{T}_2$	$\left(\hat{T}_2 - T_1\right)/T_0$
D=10	1.0920070e-01	5.5043393e+02	6.8640440e-02	-5.0399429e+03
D=30	1.0920070e-01	1.9822735e+03	2.7456176e-01	-1.8150057e+04
D=50	1.0920070e-01	2.3468790e+02	2.3306549e+00	-2.1278000e+03
D=100	1.0920070e-01	5.7030846e+02	1.4445693e+00	-5.2093429e+03