Hybrid Single Parent-Offspring MVMO for Solving CEC2018 Computationally Expensive Problems

Rueda, José L.; Erlich, Istvan

DOI
10.1109/CEC.2018.8477807

Publication date
2018

Document Version
Accepted author manuscript

Published in
2018 IEEE Congress on Evolutionary Computation, CEC 2018

Hybrid Single Parent-Offspring MVMO for Solving CEC2018 Computationally Expensive Problems

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.de

***Abstract*— Mean-Variance Mapping Optimization (MVMO) belongs to the family of evolutionary algorithms, and has proven to be competitive in solving computationally expensive problems proposed in the Icompetitions CEC2014, CEC2015, and CEC2016. MVMO can tackle such problems by evolving a set of solutions (population based approach) or a single solution (single parent-offspring approach). The evolutionary mechanism of MVMO performs within a normalized search space in the range [0, 1]. The power of MVMO stems from its ability – based on statistical analysis of the evolving solution based on a mapping function - to adaptively shift the search priority from exploration to exploitation. This paper introduces a newly defined mapping function as well as a new rule for using an embedded local search strategy, and presents several tests conducted by using the test bed of the CEC2018 competition. Numerical results indicate significant improvements on the results obtained in CEC2016 competition.**

Keywords*— *Computationally expensive optimization; heuristic optimization; mean-variance mapping optimization; single objective optimization.

I. INTRODUCTION

Powerful optimization algorithms are needed to efficiently and successfully tackle the mathematical complexity of many real-world engineering problems. Particularly, in the new era of 'Smart Grids', the computational burden for solving problems related to real-time optimal decision making and temporal optimal operational planning is challenging. The lack of parameter information often prevents an analytical formulation of such problems, or necessitates rough approximations for using classical optimization algorithms. This is a common challenge in different types of engineering problems, e.g. optimal scheduling of distritubuted energy sources [1]. In view of this, metaheuristics, and especially evolutionary algorithms, constitute attractive solution alternatives.

Mean-variance mapping optimization (MVMO) is among emerging and promising new types of evolutionary algorithms with the ability to successfully deal with different types of complex optimization problems [2]. The evolutionary mechanism of MVMO performs within a search space normalized in the range [0, 1], and can be configured to evolve a set of solutions (population based approach) or a single solution (single parent-offspring approach) [3]. Besides, MVMO uses an archive that tracks and records the best (i.e. in terms of fitness achieved so far) evolved solutions in each function evaluation (i.e. calculation of objective function and constraints). A mapping function is applied to control the search priority from exploration to exploitation [4].

MVMO has been used to solve computationally expensive problems proposed in the competitions CEC2014 [5], CEC2015 [6], and CEC2016 [7], showing an outstanding performance in terms of convergence rate and quality of solutions found within the allowed number of function evaluations. This success has motivated further developments to adapt MVMO for smart applications in power systems, such as online optimal reactive power management of offshore wind power plants [8], [9], online optimal reconfiguration of distribution systems [10], and identification of model parameters for real-time digital simulation [11].

This paper introduces a new definition of the mapping function, which enhances the ability of the algorithm to adaptively switch the emphasis between exploration (global search) and exploitation (local search), thus lowering the likelihood of premature convergence and stagnation in local optima. Also, a new rule is introduced to judiciously call an embedded local search strategy without reducing significantly the computational budget (i.e. number of function evaluations) available for the evolutionary mechanism of MVMO. Motivated by the success achieved in CEC2016, it has been decided to configure MVMO as a single parent-offspring approach.

As was done in [12] to differentiate from previous variants of MVMO, the acronym MVMO-SHM is used in this paper. The 'S' in the acronym stands for single parent-offspring approach, H means hybrid and refers to the fact that the algorithmic framework of MVMO has an embedded local search strategy, and M denotes the use of a new mapping function. The performance of MVMO-SHM is tested by using the 15 benchmark problems defined for the IEEE-CEC 2018 competition on bound constrained single-objective computationally expensive numerical optimization [7].

The reminder of the paper is structured as follows: The theoretical background behind MVMO-SHM is presented in Section II. Section III describes the experimental setup and provides a discussion on numerical results of the study. Concluding remarks are summarized in Section IV.

II. THE MVMO-SHM SEACRH PROCEDURE

The calculation steps involved in MVMO-SHM are schematically represented in the flowchart shown in Fig. 1. First, the initial settings of the parameters necessary for the calculation steps are defined, and each of the elements of the solution vector, $\mathbf{x_0} = [x_1, x_2, \ldots, x_D]$ (containing D optimization variables) are initialized, sampled and normalized from [min, max] bounds to the range [0,1]. This entails that the creation of new solutions in MVMO-SHM involves calculations with normalized parameters. This is an advantage of MVMO-SHM, since it guarantees that any new generated solution is always within [0,1] (consequently also within [min, max]), without using additional rules to penalize or repair solutions with violation of bound constraints.

Next, an iterative loop is executed, in which fitness evaluation or local search are performed by considering the de-normalized (transformed back from [0,1] to [min, max]) candidate solution vector. In each iteration, a new solution $\mathbf{x^{new}}$ is generated based on the best solution achieved so far $\mathbf{x_{best}}$, the statistical success of previous solutions (recorded in a solution archive that is updated throughout the iterations), and a mapping function that guides the evolution of selected optimization variables based on the mean and variance computed from their values recorded in the solution archive. MVMO-SHM stops once a predefined termination criterion (e.g. achieving a maximum number of allowed fitness evaluations) is met.

A. Deciding when to launch local search

MVMO-SHM decides throughout the iterations between performing fitness evaluation and continuing with the next steps of its evolutionary mechanism, or launching a local search strategy to intensify the search within the region of the search space around the best solution found so far $\mathbf{x_{best}}$. It is worth clarifying that, for the purpose of the IEEE-CEC 2018 competition on bound constrained single-objective computationally expensive numerical optimization, the value of fitness corresponds with an error value as will be explained later in Section III.A. Unlike the previous variant of MVMO used to solve the test bed of the IEEE-CEC 2016 competition [12], the launch of local search is now defined by considering the following aspects:

- Local search is started if $i > i_{Local}$, where i is the number of fitness evaluations performed so far, and i_{Local} defines the percentage from the total number of allowed fitness evaluations from which local search is considered. This allows MVMO-SHM to ensure that its pure evolutionary mechanism focuses on search exploration at the beginning of the optimization process. At a later stage of the process, MVMO-SHM can decide whether or not to launch local search to intensify the search.
- For the first local search run, $\mathbf{x_{best}}$ is used without applying the mapping function to allow the evolution of selected optimization variables. For the following and subsequent local runs, the mapping function is applied to selected variables of $\mathbf{x_{best}}$ before starting local search.
- If the best fitness value achieved so far f_{best} is greater than a threshold f_{th}, (e.g. 1E+03), the Interior-Point method is used for local search. Otherwise, the Active-Set algorithm of Matlab is used.

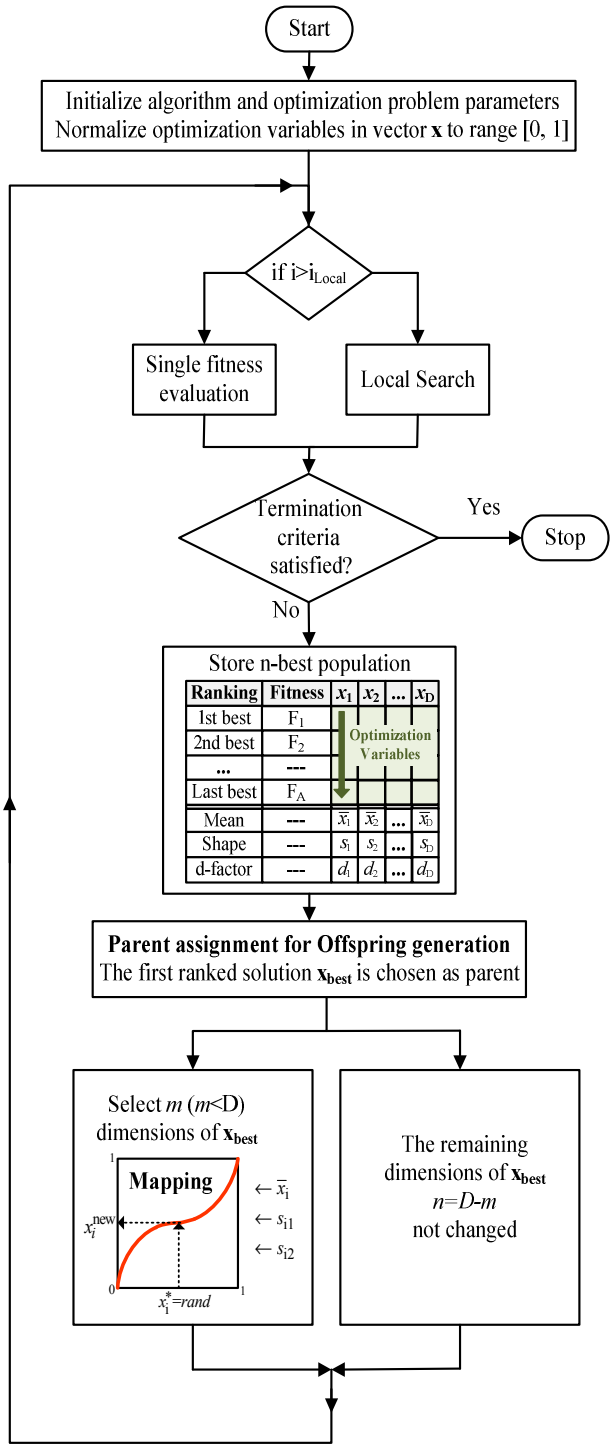

Fig. 1. Algorithmic procedure of MVMO-SHM. The fitness evaluation counter is denoted by i.

B. Recording successful solutions in an archive

As shown in Fig. 2, MVMO-SHM stores and sorts the n_{sol}-best solutions found so far based on their fitness values. As a result, the most attractive solutions found by MVMO-SHM across the whole search space are kept. In this way, the archive constitutes a data base that is continuously updated, which on the one hand allows the definition of the starting point for the generation of a new solution, by taking the first ranked solution $\mathbf{x}_{best}$ as the parent solution, and on the second hand allows computing the mean and variance for each optimization variable, which are inputs to the mapping function used to generate new values for selected optimization variables (elements) of $\mathbf{x}_{best}$.

Ranking	Fitness	x_1	x_2	...	x_D
1st best	F_1	Optimization Variables			
2nd best	F_2				
...	---				
Last best	F_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

Fig. 2. The solution archive of MVMO-SHM.

The size of the archive is defined in the initialization stage of MVMO-SHM, and is kept fixed throughout the optimization search process. All through the iterations, after each execution of either fitness evaluation or local search, the archive is updated only if a new (child) generated solution has a better fitness value than the fitness values associated to the solutions stored in the archive. This entails that the last ranked solution will be removed, and the ranking and sorting of the solutions in the archive will be updated to account for the fitness value of the new successful solution. For instance, if the fitness of the new generated solution is better than the fitness of the first ranked solution $\mathbf{x}_{best}$, then the new generated solution will become $\mathbf{x}_{best}$ (is placed in the first position), whereas the last ranked solution is removed from the archive and the other remaining solutions will be downgraded w.r.t. their positions in the archive.

Furthermore, the archive computes and stores the mean $\bar{x}_i$, shape factor s_i, and d-factor d_i associated with each optimization variable (cf. columns associated to each optimization variable in Fig. 2). These parameters are recalculated only if there is an update of the archive in a given iteration. In addition, in this new version of MVMO-SHM, to avoid unnecessary computing effort, the mean and the shape are calculated only if the distance between the values of the optimization variables in the archive is larger than the tolerance defined in (1).

$$\Delta x = 10^{-(3.5+k\cdot 5)} \tag{1}$$

where k represents the relative fitness evaluation counter (i.e. actual number of fitness evaluations performed so far divided by the maximum allowed number of fitness evaluations).

$\bar{x}_i$, s_i, and d_i are inputs of the mapping function, so they play a crucial role in MVMO-SHM to shift the search focus from global search in the initial stage of the optimization process towards more local search in the final stage of the optimization process.

After performing several optimization trials with the test bed of the IEEE-CEC 2018 competition, it has been decided to set the initial mean value of each optimization variable, $\bar{x}_{ini}$, to 0.5. This means that $\bar{x}_{ini}$ is exactly in the middle of the search space. Other options like defining random initial mean values, or user defined mean values can be used as an alternative option. Interested readers are kindly invited to try other alternatives when applying MVMO-SHM to the test bed of the IEEE-CEC 2018 competition or any other engineering problem.

C. Generation of a new solution: new mapping function

The first ranked solution $\mathbf{x}_{best}$ in the solution archive in the current iteration is chosen as parent solution to generate a new solution $\mathbf{x}^{new}=[x_1, x_2, x_3, \ldots, x_D]$ for the next iteration. Basically, m out of D elements of $\mathbf{x}^{new}$ are replaced by new values generated based on the mapping function. The strategies for determining m and the random-sequential selection of m variables among all D variables, which were proposed in [12], are also used in this paper.

In the original version of MVMO, the new value corresponding to each selected dimension x_i of $\mathbf{x}^{new}$ is computed by applying the formula of the classical mapping function as defined in (2).

$$x_i = h_x + (1 - h_1 + h_0)\cdot x_i^* - h_0 \tag{2}$$

Also, in the original version of MVMO, x_i^* constitutes a randomly chosen number that is sampled from a uniform continuous distribution in the interval [0, 1]. However, for the test bed of the IEEE-CEC 2018 competition, it has been decided, after performing sensitivity analysis with few optimization trials (to solve all problems of the test bed), to sample x_i^* from a normal distribution around the mean value of x_i equal to 0.5 (from the middle of the search space to x_i). Furthermore, the variance $\sigma_{x_i^*}$ of the normally distributed x_i^* becomes smaller throughout the iterations. In addition, the normal function $N(0.5, \sigma_{x_i^*})$ is programmed such that x_r^* is always within the interval [0, 1].

In (2), the term h corresponds with the mathematical definition of the mapping function, which is expressed in (3).

$$h(\bar{x}, s_1, s_2, x) = \bar{x}\cdot(1 - e^{-x\cdot s_1}) + (1-\bar{x})\cdot e^{-(1-x)\cdot s_2} \tag{3}$$

As shown in (4), the terms h_x, h_1 and h_0 of (2) stand for the outputs of h, when considering $x = x_i^*$, $x = 0$, and $x = 1$, respectively.

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

Recalling Fig. 2, it is worth pointing out that the shape factor s_i is defined to take into account the variance v_i of x_i, which is computed from the recorded values that are stored in the column of the solution archive corresponding to x_i :

$$s_i = -\ln(v_i) \cdot f_s \tag{5}$$

The scaling factor f_s is a parameter of MVMO-SHM that allows to automatically adjust s_i throughout the iteration. This is important in cases in which v_i is close to zero (i.e. the search is stagnated in the direction associated to x_i). The automatic adjustment of f_s is implemented such that it can vary, as the optimization progresses, between a small initial value and a higher final value. For instance, it can be decided to have $f_{s\_ini} = 1$ to not influence s_i , and $f_{s\_final} = 20$ to have a strong influence on s_i . The variation of f_s is done according to (6) and (7).

$$f_s = \left| f_{s0} \left[4 + 1.65 \left(rand - 0.15 \right) \right] \right| \tag{6}$$

$$f_{s0} = f_{s\_ini} + k \left(f_{s\_final} - f_{s\_ini} \right) \tag{7}$$

where k denotes the relative fitness evaluation counter (i.e. actual number of fitness evaluations performed so far divided by the maximum allowed number of fitness evaluations). The values 4 and 1.65 in (6) were determined by performing sensitivity analysis w.r.t. the obtained error values of the problems of the IEEE-CEC 2018 competition. The suitability of these parameters for other types of optimization problems can be evaluated in a similar fashion, and is out of the scope of the study presented in this paper.

It is worth recalling that $\bar{x}_{ini}$ is set to 0.5 in the first iteration of MVMO-SHM. Besides, s_i is set to 0. Next, throughout the iteration, $\bar{x}_i$ and s_i are recalculated after any update of the solution archive.

For the IEEE-CEC 2018 competition, a (new) mapping function, defined in (6), is proposed and used in MVMO-SHM:

$if \quad x_i^* < 0.5$	$if \quad x_i^* \geq 0.5$
$s_{i-1}^* = s_{i-1} / (1 - \bar{x}_i)$	$s_{i-2}^* = s_{i-2} / \bar{x}_i$
$h_m = \bar{x}_i - \dfrac{\bar{x}_i}{(0.5 \cdot s_{i-1}^* + 1)}$	$h_m = \dfrac{(1 - \bar{x})}{(0.5 \cdot s_{i-2}^* + 1)}$
$h_f = \bar{x} \cdot \left(1 - e^{-x_i^* \cdot s_{i-1}^*}\right)$	$h_b = (1 - \bar{x}_i) / \left((1 - x_i^*) \cdot s_{i-2}^* + 1 \right) + \bar{x}_i$
$h_c = (\bar{x} - h_m) \cdot 2 \cdot x_i^*$	$h_c = h_m \cdot 2 (1 - x_i^*)$
$x_i = h_f + h_c$	$x_i = h_b - h_c$

In [12], the shape factors s_{i-1} and s_{i-2} were assigned to vary around s_i by considering perturbations introduced by d_i. However, sensitivity analysis evidenced that $s_i = s_{i-1} = s_{i-2}$ is more suitable (e.g. better fitness) for the problems of the IEEE-CEC 2018 competition. Therefore, $s_i = s_{i-1} = s_{i-2}$ was used always for all runs of MVMO-SHM for all optimization problems.

From (2) and (8), it is worth highlighting that the shape of the mapping function is influenced by $\bar{x}_i$ and s_i . Indeed, these parameters constitute statistical measures of the success so far achieved during the search performed by MVMO-SHM. In this way, these parameters allow automatically switching the search diversity between exploration and exploitation. From (2) and (8), note also that the new generated value of x_i will always be within the interval [0, 1], which means that search boundaries of x_i will never be violated throughout the iterations.

For sake of illustration, the classical and the proposed mapping function are illustratively compared in Fig. 3 a) and b) for different mean and shape factors. Note that by using the new mapping function (denoted in the figure as Mapping #2) the mean value is always reached at 0.5 at the x-axis. Hence, there is equal probability of generating a new value of x_i that is smaller or greater than $\bar{x}_i$. By contrast, this does not happen if the classical mapping function (denoted in the figure as Mapping #1). Note also in the figure that the slope of the new mapping function is higher than that of the classical mapping function around the mean value. This entails that a better global search capability can be achieved with the new mapping function (i.e. less risk of stagnation around $\bar{x}_i$).

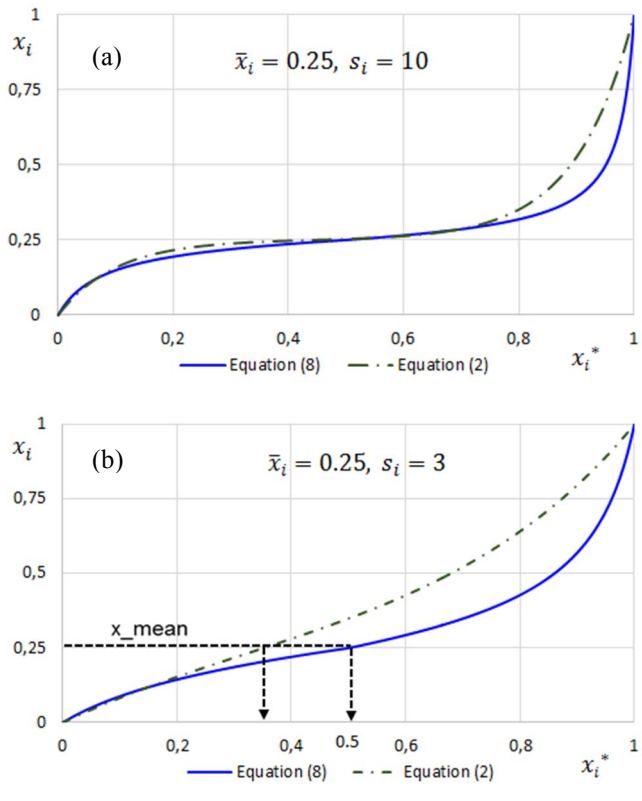

Fig. 3. Classical and the proposed mapping functions.

D. Defining the parameter settings in MVMO-SHM

The set of parameters of MVMO-SHM, which should be defined to solve all problems (test functions) of the IEEE-CEC 2018 Competition on Expensive Problems, are listed below:

P1: Number of particles.

P2: Archive size.

P3: Initial-final *fs* factor ($f^*_{s\_ini} - f^*_{s\_final}$).

P4: Number of iteration from which local search starts.

MVMO-SHM is tuned by performing sensitivity analysis of the achieved fitness value under a single parameter change within 10 independent optimization runs. The actual parameters used to solve all problems of the IEEE-CEC 2018 Competition on Expensive Problems are provided in the Matlab scripts submitted to the organizers of the competition. The scripts can be downloaded from [13].

III. Experimental Results and Discussion

The performance of MVMO-SHM on the test bed of the IEEE-CEC 2018 Competition on Expensive Problems was tested by using 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-SHM was done in Matlab® Version R2017b, 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.

A. CEC2018 test bed

Table I overviews the list of IEEE-CEC 2018 expensive optimization test problems. The mathematical formulation and properties of these problems can be found in [7]. The rules of the competition state that the problems are treated as black-box problems, and also that a single set of parameters shall be defined for the contestant algorithm to solve all problems in 10D and 30D.

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

- Problem dimension D = 10, 30.
- Search range: $[-100,100]^D$.
- Maximum number of function evaluations (MaxFEs): 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 indicated in Table I. The values of OF smaller than 1E-03 are taken as zero.
- The optimization is terminated upon completion of the maximum number of function evaluations.

TABLE I. IEEE-CEC 2018 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 the Appendix, the statistical attributes of the error value OF (i.e. best, worst, mean, median, and standard deviation values) calculated after 20 runs are summarized. Since the function evaluation budget is pretty much limited, the main task for MVMO-SHM is to perform global search. Local search is launched close to the final stage of MVMO-SHM by following the procedure indicated in Section II.A. The following conclusions are drawn from Table II and Table III:

- *Unimodal functions*: MVMO-SHM has a satisfactory performance and allowed to find near to zero error values for OF in all runs for both 10D and 30D (in the order of 1E-01 and 1E-04). This is mainly attributed to the evolutionary mechanism of MVMO-SHM. Local search strategy supported MVMO-SHM to effectively narrow the ridge property of TF1 and the sensitive direction property of TF2.

- *Simple multimodal functions*: MVMO-SHM was able to find near to zero error values for OF in all runs for TF5,

TF6, and T7 (in the order of 1E-01) in both, 10D and 30D. For TF3, TF8, and TF9, the errors are between 1E+00 and 1E+01. This is an important enhancement w.r.t. the results obtained in the CEC2016 competition [12]. The new mapping function and strategy used to launch local search in MVMO-SHM can successfully tackle the complexities of these functions.

- *Hybrid functions:* MVMO-SHM was able to find errors between 1E+00 and 1E+02 in both, 10D and 30D. An optimal tuning of the parameters of MVMO-SHM is expected to further enhance its performance.

- *Composition functions:* MVMO-SHM was able to find errors between 1E+01 and 1E+02 in both, 10D and 30D. An optimal tuning of the parameters of MVMO-SHM is expected to further enhance its performance.

Two score measures, which are indicated in [7], were calculated in Matlab according to (9) and (10), respectively. The score measures were calculated for 10 repetitions of the application of MVMO-SHM to solve the IEEE-CEC 2018 expensive optimization test problems, for both, 10D and 30D. The calculated scores for each repetition are shown in Table IV. According to [6], the total score (Score 2 for 10D + Score 2 for 30D) obtained by MVMO (reported in [12]) in the CEC2016 Competition on Expensive Optimization was 3.062550E+06. Note from Table IV, that MVMO-SHM leads to a significant reduction, cf. total score (Score 2 for 10D + Score 2 for 30D), which can be in the order of 1E+04.

$$\text{Score1} = \sum_{1}^{15} mean(\text{OF})\,|_{\text{Dim}} + \sum_{1}^{15} median(\text{OF})\,|_{\text{Dim}} \tag{9}$$

$$\text{Score2} = \sum_{1}^{15} mean(f_a)\,|_{\text{Dim}} + \sum_{1}^{15} median(f_a)\,|_{\text{Dim}} \tag{10}$$

$$f_a = 0.5 \cdot \left(OF_{\text{MaxFEs}} + OF_{0.5\text{MaxFEs}}\right)$$

where Dim can be 10D or 30D.

The measures of computational complexity, defined in [7], are shown in Table V.

IV. CONCLUSIONS

This paper presented a new variant of MVMO, termed as MVMO-SHM. This variant has a new mapping function and incorporates a new rule for using an embedded local search strategy. MVMO-SHM was applied to the IEEE-CEC 2018 competition test suite on Bound Constrained Single-Objective Computationally Expensive Numerical Optimization. Basically, MVMO-SHM is executed as a single parent-offspring algorithm, i.e. it evolves a single solution throughout the iterations. The evolution is done within a normalized search space, i.e. the optimization variables are treated in a search spaced bounded in the interval [0,1]. This interval is equivalent to the original [min, max] bounds. This is one advantage of MVMO-SHM over other evolutionary algorithms, since it does not need any additional strategy to repair or replace solutions that violate the bounds of the optimization variables. A new mapping function has been proposed. The slope of this mapping function is automatically adjusted throughout the iterations to enhance the global search capability of MVMO-SHM. Numerical results show an improvement in the order of 100 times decrease in the score measure used to rank the algorithms in CEC2016 competition test suite on Bound Constrained Single-Objective Computationally Expensive Numerical Optimization.

REFERENCES

[1] F. Lezama, J. Soares, Z. Vale, and Jose Rueda, "Test bed: Optimal scheduling of distributed energy resources considering uncertainty of renewables, EVs, load forecast and market prices," IEEE WCCI 2018 Competition Evolutionary Computation in Uncertain Environments: A Smart Grid Application. December 2017. [Online]. Available at http://sites.ieee.org/psace-mho/2018-evolutionary-computation-in-uncertain-environments-competition/.

[2] J.L. Rueda, and I. Erlich, "MVMO for Bound Constrained Single-Objective Computationally Expensive Numerical Optimization", in Proc. 2015 IEEE Congress on evolutionary Computation, pp. 1-7, Sendai, Japan, May 2015..

[3] J.L. Rueda, and I. Erlich, "Testing MVMO on Learning-based Real-Parameter Single Objective Benchmark Optimization Problems", in Proc. 2015 IEEE Congress on evolutionary Computation, pp. 1-7, Sendai, Japan, May 2015..

[4] J.C. Cepeda, J.L. Rueda, I. Erlich, A.W. Korai, and F.M. Gonzalez-Longatt, "Mean–Variance Mapping Optimization Algorithm for Power System Applications in DIgSILENT PowerFactory .", PowerFactory Applications for Power System Analysis, Power Systems, Springer International Publishing Switzerland, pp. 267-295, Jan. 2015..

[5] Qin Chen, "Review of the CEC2014 Computational Expensive Optimization Competition". 2014 IEEE WCCI. July 2014. [Online]. Available at http://www.ntu.edu.sg/home/epnsugan/

[6] P.N. Suganthan, "Competition on Real-Parameter Single Objective Computationally expensive Optimization", 2015 IEEE WCCI. May 2015. [Online]. Available at http://www.ntu.edu.sg/home/epnsugan/

[7] 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/

[8] A. M. Theologi, M. Ndreko, J.L. Rueda, M. A. M. M. van der Meijden, and F. González-Longatt, "Optimal management of reactive power sources in far-offshore wind power plants," 2017 IEEE Manchester PowerTech, Manchester, June 2017, pp. 1-6.

[9] J. L. R. Torres, A. M. Theologi, M. Ndreko, I. Erlich, and P. Palensky, "Metaheuristic approach for online optimal reactive power management in near-shore wind power plants," 2017 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America), Quito, Ecuador, Sep. 2017, pp. 1-6.

[10] A. Gbadamosi, J.L. Rueda, D. Wang, and P. Palensky, "Application of mean-variance mapping optimization for parameter identification in real-time digital simulation," 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), Prague, Sep. 2017, pp. 11-16.

[11] J.L. Rueda, R. Loor, and I. Erlich, "MVMO for Optimal Reconfiguration in Smart Distribution Systems," in Proc. 9th IFAC Symposium on Control of Power and Energy Systems, pp. 1-6, New Delhi, India, Dec. 2015.

[12] J.L. Rueda, and I. Erlich, "Solving the CEC2016 Real-Parameter Single Objective Optimization Problems through MVMO-PHM". 2016 IEEE World Congress on Computational Intelligence, Vancouver Canada, pp. 1-10, July 2016. [Online]. Available at http://www.ntu.edu.sg/home/epnsugan/.

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

APPENDIX

TABLE II. RESULTS FOR 10D – MVMO-SHM

Type	Function	Best	Worst	Median	Mean	Std.
Unimodal functions	TF1	2.74910E-04	2.85512E-01	1.87481E-01	1.79024E-01	9.77837E-02
	TF2	6.74949E-03	1.81420E-02	1.47218E-02	1.47325E-02	2.51007E-03
Simple multimodal Functions	TF3	8.25651E+00	1.49345E+01	1.14211E+01	1.11539E+01	1.60788E+00
	TF4	2.59481E+02	1.23340E+03	6.63274E+02	7.72082E+02	3.33769E+02
	TF5	2.70970E-01	3.35751E+00	1.24475E+00	1.39259E+00	7.95275E-01
	TF6	1.41483E-01	9.58650E-01	3.18079E-01	3.57934E-01	1.99266E-01
	TF7	1.95121E-01	1.07092E+00	4.36860E-01	5.10845E-01	2.51496E-01
	TF8	1.52242E+00	7.88627E+02	7.08500E+01	1.84558E+02	2.41708E+02
	TF9	3.31284E+00	4.47835E+00	4.10544E+00	4.07662E+00	2.97936E-01
Hybrid functions	TF10	1.18268E+02	8.82968E+02	4.71818E+02	4.75114E+02	1.82483E+02
	TF11	7.02795E+00	4.46712E+01	1.61211E+01	1.98122E+01	1.17808E+01
	TF12	1.48219E+02	6.57597E+02	3.21122E+02	3.55939E+02	1.37501E+02
Composition functions	TF13	3.14939E+02	3.20785E+02	3.15664E+02	3.15872E+02	1.42599E+00
	TF14	1.92443E+02	2.24854E+02	2.00112E+02	2.03034E+02	8.21744E+00
	TF15	1.59235E+01	7.07747E+02	5.27061E+02	5.16030E+02	1.46239E+02

TABLE III. RESULTS FOR 30D – MVMO-SHM

Type	Function	Best	Worst	Median	Mean	Std.
Unimodal functions	TF1	2.87213E-04	6.98154E-02	2.47551E-02	2.93593E-02	2.33688E-02
	TF2	3.46866E-03	7.28638E-03	6.40834E-03	6.08591E-03	9.47794E-04
Simple multimodal Functions	TF3	3.28250E+01	4.67295E+01	3.85620E+01	3.91330E+01	3.54745E+00
	TF4	2.42808E+03	4.49875E+03	3.33807E+03	3.43435E+03	6.72010E+02
	TF5	7.23655E-01	3.78745E+00	1.86301E+00	2.00662E+00	1.02938E+00
	TF6	2.62380E+00	5.70072E+00	4.69906E+00	4.63497E+00	6.92699E-01
	TF7	3.99214E-01	5.11117E-01	4.82845E-01	4.72709E-01	3.08405E-02
	TF8	1.86999E+02	2.72956E+03	7.82813E+02	8.88833E+02	5.86437E+02
	TF9	1.19680E+01	1.40679E+01	1.34707E+01	1.34360E+01	4.96090E-01
Hybrid functions	TF10	3.75475E+02	2.20727E+03	1.59640E+03	1.51601E+03	4.53166E+02
	TF11	8.50074E+01	3.26769E+02	1.78887E+02	1.82562E+02	7.37962E+01
	TF12	5.86577E+02	1.19907E+03	8.38226E+02	8.28670E+02	1.59927E+02
Composition functions	TF13	3.29392E+02	3.68503E+02	3.46724E+02	3.46677E+02	1.18188E+01
	TF14	2.19918E+02	3.82868E+02	2.61874E+02	2.67966E+02	4.13940E+01
	TF15	9.37273E+02	1.59405E+03	1.40400E+03	1.39244E+03	1.60798E+02

TABLE IV. COMPARISON OF SCORES MEASURES

Run	D=10		D=30	
	Score 1	**Score 2**	**Score 1**	**Score 2**
1	5.54994E+03	6.62802E+03	1.83422E+04	5.84547E+04
2	5.50220E+03	6.10154E+03	1.77233E+04	4.07810E+04
3	5.73708E+03	7.04628E+03	1.74526E+04	5.09296E+04
4	5.47975E+03	5.87058E+03	1.80263E+04	5.06534E+04
5	5.82854E+03	6.20555E+03	1.70378E+04	5.67913E+04
6	5.92119E+03	6.57144E+03	1.85146E+04	5.93868E+04
7	5.87085E+03	6.17880E+03	1.81458E+04	6.33531E+04
8	5.81196E+03	6.90656E+03	1.82905E+04	4.92681E+04
9	5.67373E+03	6.06553E+03	1.74854E+04	5.05804E+04
10	6.06963E+03	6.76064E+03	1.73548E+04	7.89782E+04
Mean	5.74449E+03	6.43349E+03	1.78373E+04	5.59176E+04

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

Function	**D=10**	**D=30**
TF1	2.6740000e+01	1.1068333e+02
TF2	3.4050000e+01	8.5933333e+01
TF3	2.3230000e+01	1.0093333e+02
TF4	2.7150000e+01	5.8416667e+01
TF5	2.2180000e+01	5.9450000e+01
TF6	1.7850000e+01	1.8624167e+02
TF7	2.2350000e+01	6.3091667e+01
TF8	1.6670000e+01	1.2024167e+02
TF9	3.0020000e+01	4.1625000e+01
TF10	2.5710000e+01	2.2971667e+02
TF11	3.0440000e+01	6.8983333e+01
TF12	2.4560000e+01	5.5741667e+01
TF13	1.8660000e+01	6.4058333e+01
TF14	2.4270000e+01	4.1641667e+01
TF15	2.1000000e+01	2.5445833e+02