2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Evaluating the Mean-Variance Mapping Optimization on the IEEE-CEC 2014 Test Suite

István Erlich[1], José L. Rueda[2], Sebastian Wildenhues[1], and Fekadu Shewarega[1]
[1]Institute of Electrical Power Systems, University Duisburg-Essen, Duisburg, Germany
[2]Department of Electrical Sustainable Energy, Delft University of Technology, Delft, The Netherlands
{istvan.erlich, sebastian.wildenhues, fekadu.shewarega}@uni-due.de, J.L.RuedaTorres@tudelft.nl

***Abstract*— This paper provides a survey on the performance of the hybrid variant of the Mean-Variance Mapping Optimization (MVMO-SH) when applied for solving the IEEE-CEC 2014 competition test suite on Single Objective Real-Parameter Numerical Optimization. MVMO-SH adopts a swarm intelligence scheme, where each particle is characterized by its own solution archive and mapping function. Besides, multi-parent crossover is incorporated into the offspring creation stage in order to force the particles with worst fitness to explore other sub-regions of the search space. In addition, MVMO-SH can be customized to perform with an embedded local search strategy. Experimental results demonstrate the search ability of MVMO-SH for effectively tackling a variety of problems with different dimensions and mathematical properties.**

Keywords—Heuristic optimization; mean-variance mapping optimization; single objective optimization; swarm intelligence.

I. INTRODUCTION

Despite the recent breakthroughs in heuristic optimization, there is still a high motivation to develop new algorithmic procedures, which allow efficient tackling of complex and high-dimensional optimization problems [1]-[5]. Mean-variance mapping optimization (MVMO) is a recently introduced algorithm, whose evolutionary mechanism adopts a single parent-offspring pair approach along with a normalized range of the search space for all optimization variables within [0,1]. Within this framework, MVMO exploits the statistical attributes of search dynamics by using a special mapping function for mutation operation on the basis of the mean and variance of the n-best solutions attained so far and saved in a continually-updated solution archive [6]. Preliminary applications in power engineering field have shown that MVMO has a promising prospect for solving different types of optimization problems [7]-[11]. Like other heuristic optimization algorithms, MVMO is open for extension with new strategies to improve its search capability.

This paper presents an analysis of the performance of a new variant of MVMO, which henceforth is termed as MVMO-SH, on 30 novel benchmark functions belonging to the IEEE-CEC 2014 competition test suite on Single Objective Real-Parameter Numerical Optimization. MVMO-SH performs the search process by evolving a population of particles (i.e. swarm), each having its own memory, which is represented by the associated knowledge archive and mapping function. Besides, a fitness-based classification is performed to discriminate between good particles (e.g. smaller fitness values) and bad particles (e.g. higher fitness values). For each good particle, the parent assignment is done by considering the first ranked solution in its particular knowledge archive, whereas a multi-parent crossover is used to reorient each bad particle towards different sub-regions of the search space. Additionally, it offers the possibility of occasionally resorting to an interior-point method (IPM) or alternatively to a sequential quadratic programming (SQP) based strategy for local improvement purpose. The statistical performance of MVMO-SH during the optimization repetition is investigated by considering different problem dimensions ranging from 10D up to 100D along with other predefined experimental settings from IEEE-CEC 2014 problem definitions [12].

II. RATIONALE BEHIND MVMO-SH

Fig. 1 shows the algorithmic steps involved in MVMO-SH. The procedure starts with an initialization stage where the algorithm parameter settings are defined and samples of the optimization variables are randomly sampled within their search boundaries for a set of N_P particles (i.e. population/swarm of candidate solutions). Also, the optimization variables are normalized at this stage, that is, the range of the search space for all variables is transformed from [min, max] to [0, 1] range. This is a precondition for the subsequent mutation operation via mapping function; on top of this, it guarantees that the generated offspring will never violate the search boundaries. The optimization variables are de-normalized before performing fitness evaluation or local search. The core of the algorithm is contained in the inner loop of the flowchart, in which, after execution of fitness evaluation or local search for a given particle, updating the solution archive, fitness based classification of particles into good or bad particles, parent selection and offspring generation are performed. The procedure ends once the termination criterion is satisfied.

A. Fitness evaluation and local search

The elements of the candidate solution array (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 also includes a possible penalty value resulting from constraint violation.

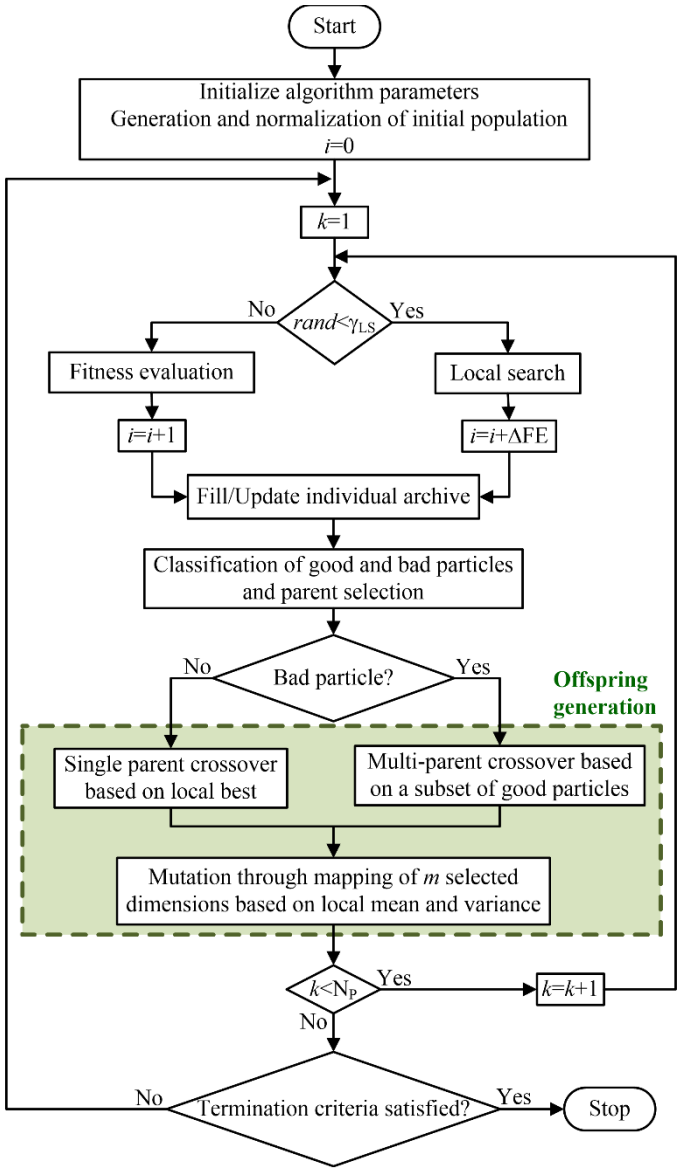

Fig. 1. MVMO-SH overall procedure. The fitness evaluation and particle counters are denoted by i and k, whereas N_P, ΔFE, and *rand* stand for number of particles, number of fitness evaluations, and uniform random number between [0, 1], respectively.

One of the available local search strategies (IPM, SQP) is included in this step as a local improvement option. After a given number of fitness evaluations, local search is performed with a probability γ_{LS} for any child of the population

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

and runs in the range

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

where i denotes fitness evaluation number, and *rand* is a random number with uniform distribution in [0,1].

The execution of local search 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 concerns (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 value of γ_{LS} should be set to zero in order to give preference to the underlying evolutionary mechanism MVMO.

B. Solution archive

As illustrated in Fig. 2, each particle has compact and continually updated solution archive associated to it, which stores its n-best 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 execution of fitness evaluation/local search, an update of its archive takes place only if the new solution is better than those in the archive. Besides, the mean $\bar{x}_i$, shape s_i, and d-factor d_i associated to each optimization variable are recalculated whenever an update of the archive takes place. These parameters influence the change of the shape of the mapping function, which is crucial to arrive at a balance between search exploration and exploitation.

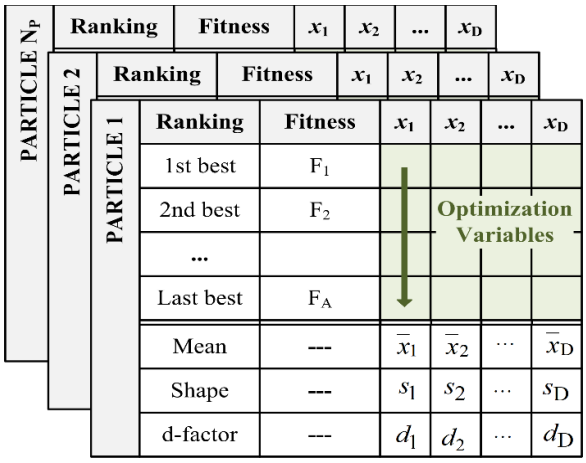

Fig. 2. Layout of the particles'solution archives.

C. Parent selection

In the early stage of the search process, each particle is independently used for a few (at least two) function evaluations, and the solution that produced the individual best fitness achieved so far (i.e. the one corresponding to the first ranked position in its particular solution archive) is chosen as the parent for the next offspring. Thereafter, a special procedure, which is illustrated in Fig. 3, is applied for parent assignment in order to encourage the particles having poor performance (in terms of individual achieved best fitness) to explore different sub-regions of the search space. As illustrated in the figure, the particles are classified into the set of *GP* "good particles", and Np-*GP* "bad particles" based on the individual best fitnesses. Individual best-based parent assignment is adopted for each particle classified as good, whereas for each bad particle $\mathbf{x}_k$, the parent $\mathbf{x}_k^{parent}$ is determined by using the following multi-parent criterion:

$$\mathbf{x}_k^{parent} = \mathbf{x}_{RG}^{best} + \beta\left(\mathbf{x}_{GB}^{best} - \mathbf{x}_{LG}^{best}\right) \tag{3}$$

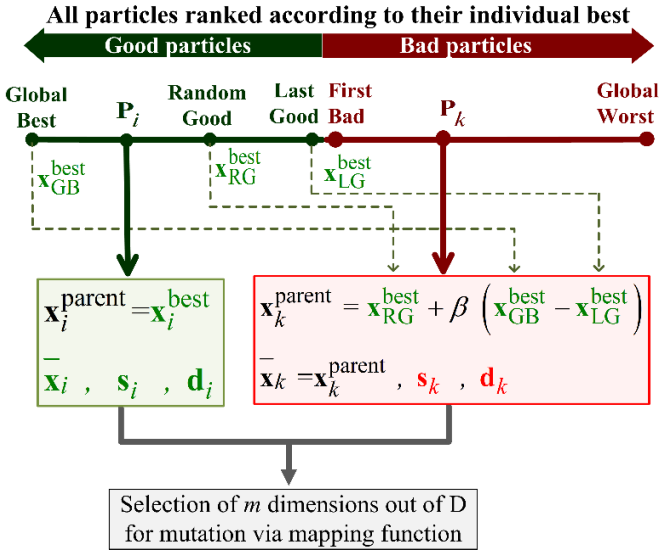

Fig. 3. Parent selection in MVMO-SH.

However, in contrast to [14], where all three parents $\mathbf{x}_{RG}^{best}$, $\mathbf{x}_{GB}^{best}$ and $\mathbf{x}_k$ are randomly selected from the group of good particles, for MVMO-SH, $\mathbf{x}_{GB}^{best}$, $\mathbf{x}_{LG}^{best}$ and $\mathbf{x}_{RG}^{best}$ represent the first (global best), the last, and a randomly selected intermediate particle in the group of good particles, respectively. The vector of mean values associated to $\mathbf{x}_k$, which are required later for mutation and mapping by the so called h-function (cf. Subsection II.E) is also set to $\mathbf{x}_k^{parent}$. The factor β is a random number, which is determined from:

$$\beta = 2.5\left(rand + 0.25 \cdot \alpha^2 - 0.5\right) \tag{4}$$

β is re-drawn and (4) is recalculated for any element of $\mathbf{x}_k^{parent}$ going outside the range [0, 1].

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

$$GP = \text{round}\left(\text{N}_\text{P} \cdot g_p^*\right) \tag{5}$$

$$g_p^* = \text{g}_{\text{p\_ini}}^* - \alpha^2\left(\text{g}_{\text{p\_final}}^* - \text{g}_{\text{p\_ini}}^*\right) \tag{6}$$

Equation (6) 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 $\text{g}_{\text{p\_ini}}^*$ to $\text{g}_{\text{p\_final}}^*$.

D. Crossover

For the next generation, a child vector (array) $\mathbf{x}^{new} = [x_1, x_2, x_3, \ldots, x_D]$, where D is the number of problem dimensions, is created for each particle by combining a subset of D-*m* directly inherited dimensions from $\mathbf{x}_p^{parent}$ (i.e. crossover) and m selected dimensions that undergo mutation operation through mapping function based on the actual values of the parameters $\bar{x}_i$, s_i, and d_i associated to each particle. The number m of dimensions to be selected for mutation operation is progressively decreased as follows:

$$m = \text{round}\left(\text{m}_\text{final} + irand\left(m^* - \text{m}_\text{final}\right)\right) \tag{7}$$

$$m^* = \text{round}\left(\text{m}_\text{ini} - \alpha^2\left(\text{m}_\text{ini} - \text{m}_\text{final}\right)\right) \tag{8}$$

where *irand* represents a random integer in the range of zero and the value given in the brackets. The selection of the m variables to be mutated can be done by using any of the strategies given in [13].

E. Mutation through mapping function

The new value of each selected dimension x_r of $\mathbf{x}^{new}$ is determined by

$$x_\text{r} = h_\text{x} + (1 - h_\text{l} + h_0) \cdot x_\text{r}^* - h_0 \tag{9}$$

where x_r^* is a randomly generated number with uniform distribution between [0, 1], and the term h (subscripts specified below) represents the transformation mapping function defined as follows.

$$h(\bar{x}, s_1, s_2, x) = \bar{x} \cdot (1 - e^{-x \cdot s_1}) + (1 - \bar{x}) \cdot e^{-(1-x) \cdot s_2} \tag{10}$$

h_x, h_l and h_0 are the outputs of the mapping function calculated for

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

Thus, x_r is always within the range [0, 1]. s_r is the shape factor calculated as follows:

$$s_\text{r} = -\ln(v_\text{r}) \cdot f_\text{s} \tag{12}$$

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.

In the first evaluation of every particle, the mean $\bar{x}_\text{r}$ corresponds with the initial value of x_r 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 Fig. 4, note that the shape of the mapping function is influenced by the mean $\bar{x}_\text{r}$ and shape factors s_r1 and s_r2. So, the search diversity can be enhanced through proper variation of the shape factors.

The scaling factor f_s can be additionally used to change the shape of the function. For this purpose, f_s is increased as the optimization progresses from a small initial value

(typically $f_{s\_ini}^{*}=1$) up to a higher final value. A considerably higher final value of f_s, e.g. $f_{s\_final}^{*}=20$, could be used for real-parameter optimization problems with high accuracy concerns. Different options are available in MVMO for the calculation of fs. In this study the equations (13), (14)have been used.

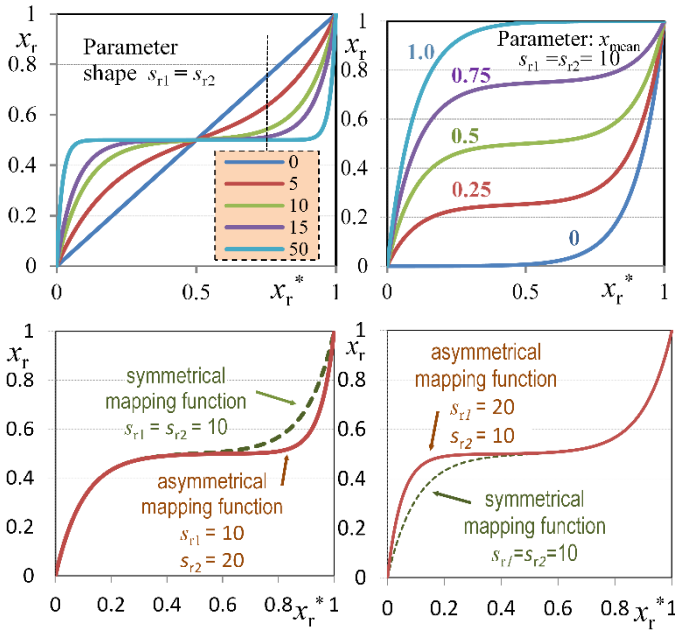

Fig. 4. Change of the mapping function shape for different values of mean and shape factors.

$$f_s = f_s^* \cdot \left(1+(0.9-rand)\cdot 0.25\right) \tag{13}$$

$$f_s^* = f_{s\_ini}^* + \alpha^2\left(f_{s\_final}^* - f_{s\_ini}^*\right) \tag{14}$$

Furthermore, the shape factors s_{r1} and s_{r2} of the variable x_r are assigned by using the procedure given in (15), which is described as follows: The initial values of d_r are set for all variables at the beginning of the optimization. At every evaluation, each d_r is scaled up or down with the factor Δd. If $d_r > s_r$, the current d_r is divided by Δd which is always greater than 1.0 and thus leads to reduced value of d_r. In case $d_r < s_r$, d_r is multiplied by Δd resulting in increased d_r. Consequently, d_r will continuously oscillate around the current shape factor s_r.

The initial values of d_r are set to 1 for all variables at the beginning of the optimization. At every evaluation, each d_r is scaled up or down with the factor Δd, which is randomly varied between 1 and $1+2\cdot\Delta d_0$. If $d_r > s_r$, the current d_r is divided by Δd which is always greater than 1.0 and thus leads to reduced value of d_r. In case $d_r < s_r$, d_r is multiplied by Δd resulting in increased d_r. The subsequent random assignment of d_r and s_r to either s_{r1} or s_{r2} results in a continuous oscillation of these shape factors around the current value of s_r. According to our experience, it is recommended to take a non-zero value equal or smaller than 0.4 for Δd_0.

$$\begin{aligned}
&s_{r1} = s_{r2} = s_r \\
&\text{if } s_r > 0 \text{ then} \\
&\quad \Delta d = \left(1+\Delta d_0\right) + 2\cdot\Delta d_0\cdot\left(rand-0.5\right) \\
&\quad \text{if } s_r > d_r \\
&\quad\quad d_r = d_r \cdot \Delta d \\
&\quad \text{else} \\
&\quad\quad d_r = d_r / \Delta d \\
&\quad \text{end if} \\
&\quad \text{if } rand < 0.5 \text{ then} \\
&\quad\quad s_{r1} = s_r \;;\quad s_{r2} = d_r \\
&\quad \text{else} \\
&\quad\quad s_{r1} = d_r \;;\quad s_{r2} = s_r \\
&\quad \text{end if} \\
&\text{end if}
\end{aligned} \tag{15}$$

The above procedure fully exploits the asymmetric characteristic of the mapping function by using different values for s_{r1} and s_{r2} leading to enhanced searching performance and zero variance handling. Zero variance can occur when all values of x_r in the particle's archive are identical. In this case the previous non-zero value can be used further. However, this value may result, under circumstances, in stagnation of the convergence behavior. The procedure overcomes this problem as the mean and variance are calculated only for non-identical values of x_r saved in the archive.

It is worth pointing out that the mean, variance and shape factors associated to x_r are not calculated before a certain number of solutions are available in the solution archive. Usually, they end up being calculated immediately after two solutions have been stored in the archive. However, one could also decide to do it once the archive is filled up completely. In this stage, the search is performed with s_{r1}=s_{r2}=0 which corresponds with a straight line between zero and one as a shape of the mapping function. The mean value in this case does not have any effect on the mapping function, cf. (10).

III. Numerical Tests

Numerical experiments were performed on a computer with Intel® Core™ i7-47500 HQ CPU, 2.0 GHz and 16 GB RAM, under Windows 8.1 pro, 64 bit OS. The implementation of MVMO-SH was realized in ANSI C using MPI interface for distributed memory parallel computation of optimization trials. Stochastic integrity is guaranteed by performing independent initialization of random number streams on individual processes with respect to time plus the process identifier.

A. Experimental setting

MVMO-SH is used to solve the IEEE-CEC 2014 test suite, which comprises 30 single objective test functions (TFs) grouped into three different categories, namely, unimodal functions (TF1 to TF3), simple multimodal functions (TF4 to

T16), hybrid functions (TF17 to TF22), and composition functions (TF23 to TF30). The details of these functions, which are treated as black-box problems for the competition, are given in [12]. Statistical tests on convergence performance and quality of final solution provided by MVMO-SH were performed under the following considerations:

- Problem dimension D = 10, 30, 50, 100.
- 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.
- The objective function is defined as the error value $OF=TF_i(x)-TF_i(x^*)$, where is the theoretical global optimum of the i-th benchmark function TF given in problem definitions reported in [12]. 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.

MVMO-SH was executed with the following parameters:

N_p=150 for D=100, N_p=100 for D=50 and D=30, N_p=80 for D=10; Archive size=25, $\Delta d_0 = 0.2$ $f^*_{s\_ini} = 1$, $f^*_{s\_final} = 20$, $g^*_{p\_ini} = 0.7$, $g^*_{p\_final} = 0.1$, m_{ini} : D10 = 5, D30 = 15, D50 = 15, D100 = 30, $m_{final} = 1$, $\gamma_{LS} = 0.1$, $\alpha_{LS\_min} = 0.5$, $\alpha_{LS\_max} = 0.9$.

B. Performance statistics and discussion

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 Table ITABLE I. , Table II, and Table IIITABLE III. in the Appendix, for each 10D and 30D and 50D cases. Results for 100D are not provided due to space limitation, but interested readers can obtain them via email from the authors. The measures for computing complexity, which were computed by following the guidelines provided in [12], are given in Table IV. The following remarks can be deduced from these results:

- *Solving unimodal functions:* For 10D case, MVMO-SH 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 the zero value was achieved in most cases before 50% of the predefined maximum number of function evaluations. The proposed algorithm also allows obtaining close to zero error values for 30D, 50D, and 100D cases. Therefore, it constitutes a powerful tool to effectively tackle unimodal problems irrespective of the underlying mathematical features (e.g. asymmetrical, separable/non-separable).

- *Solving simple multimodal functions*: For the 10-D case, MVMO-SH is capable of providing near to zero error values for all functions in almost all runs. A similar performance is observed in 30D, 50D, and 100D cases, except for function TF11.

- *Solving hybrid functions*: For the 10-D case, MVMO-SH is also capable of providing near to zero error values for all functions in almost all runs. The errors are in the order of 10^1, 10^{-1} and 10^0 for 30D, 50D, and 100D cases in most optimization repetitions.

- *Solving composition functions*: For the 10-D case, the proposed algorithm is capable of providing near to zero error values for function TF27 in several runs, but it could not provide zero errors for this function in 30-D, 50D, and 100D cases. For the remaining functions the errors are in the order of 10^2.

- Note that the results provided in the paper for all dimensions were obtained by modifying only the number of particles and the number of dimensions to be selected for mutation operation. The performance of MVMO-SH can be further improved by using different parameter settings.

Fig. 5 - Fig. 8 are intended to illustrate the relationship between the convergence progress and the evolution of the shape factors s_1 and s_2, resulting from one run of a 10D case for TF1 and TF30 with and without local search, respectively.

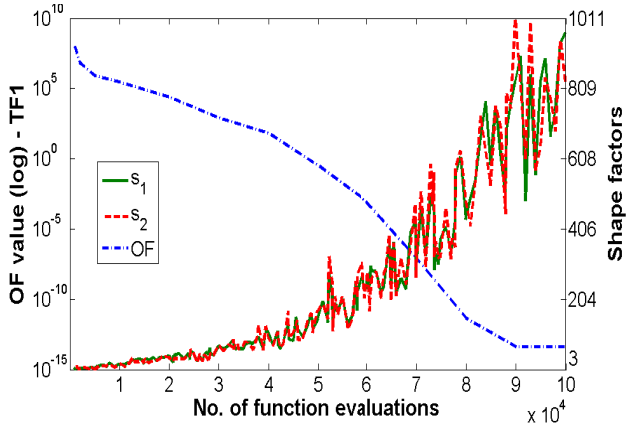

Fig. 5. Evolution of the objective function OF and the shape factors s_1 and s_2 when optimizing TF1 without local search for 10D case.

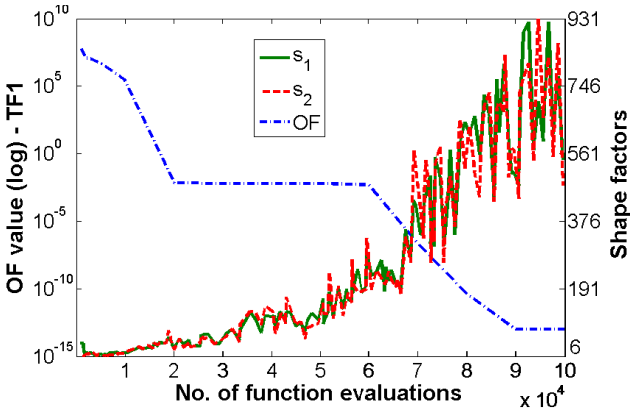

Fig. 6. Evolution of the objective function OF and the shape factors s1 and s2 when optimizing TF1 with local search for 10D case.

It is interesting to note that with local search TF1 can achieve small OF values very soon but is stagnating afterwards until the local search option is switched off at 50% of the total number of iterations. Considerable further improvement is

coming prominently from the MVMO algorithm. For TF30, local search is not able to provide considerable advantages. The figures also illustrate the oscillating behavior of the shape factors s_1 and s_2 that results in a dynamic character of the mapping function. Furthermore, with the iteration number the factors s_1 and s_2 are increasing which leads to improved numerical accuracy.

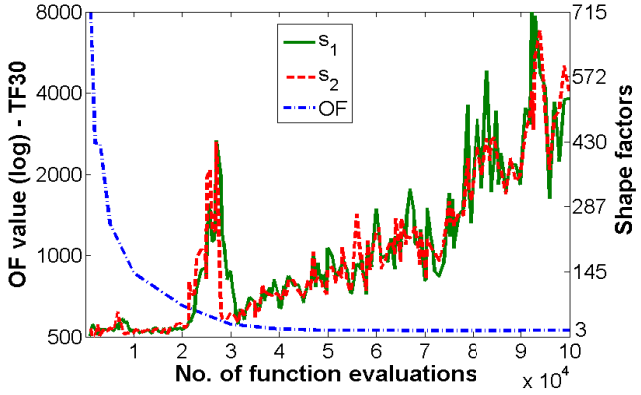

Fig. 7. Evolution of the objective function OF and the shape factors s_1 and s_2 when optimizing TF30 without local search for 10D case.

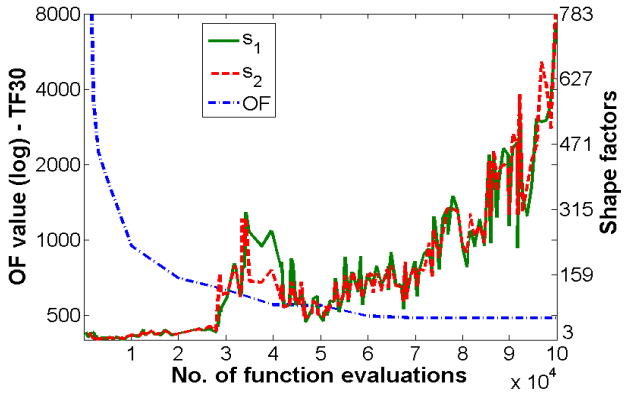

Fig. 8. Evolution of the objective function OF and the shape factors s1 and s2 when optimizing TF30 with local search for 10D case.

IV. CONCLUSIONS

Based on the IEEE-CEC 2014 test suite on objective real-parameter numerical optimization, the performance of the hybrid variant of MVMO, called MVMO-SH, is thoroughly evaluated in this paper. Unlike the classical single-particle implementation of MVMO, MVMO-SH adopts the swarm intelligence precept, and incorporates local search and multi-parent crossover strategies to increase the search diversity while pursuing a balance between exploration and exploitation. Statistical performance assessment, performed under the IEEE-CEC 2014 competition criteria, has shown the feasibility and effectiveness of MVMO-SH to tackle a variety of optimization problems with different complexities (e.g. unimodal, simple multimodal, hybrid and composite problems).

Additionally, it is worth highlighting that, unlike other swarm based optimization techniques, MVMO-SH does not strictly require several particles to proceed. But using more than one particle in MVMO-SH extends the global searching capability considerably. For less challenging optimization problems, however, running MVMO-SH with one particle (i.e. the classical single-particle approach) may be sufficient.

Further research effort is being focused on sensitivity of the algorithm's parameters, as well as on the study of other types of local search strategies. Performance comparisons with other state-of-art heuristic optimization algorithms, along with application of MVMO-SH to different large-scale power system optimization problems, are also currently under investigation.

REFERENCES

[1] T. El-Ghazali, Metaheuristics: From design to implementation. Hoboken: John Wiley & Sons, 2009.

[2] P.N. Suganthan, "Testing Evolutionary Algorithms on Real - World Numerical Optimization Problems". [Online]. Available at: http://www.ntu.edu.sg/home/epnsugan/

[3] S. Das, and P.N. Suganthan,"Differential Evolution: A Survey of the State-of-the-Art," IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, pp. 4-31, Feb. 2011.

[4] S.M.Elsayed, R.A.Sarker, and D.L. Essam, "A new genetic algorithm for solving optimization problems," Engineering Applications of Artificial Intelligence, vol. 27, pp. 57–69, Jan. 2014.

[5] D. Simon, Evolutionary optimization algorithms: Biologically inspired and population-based approaches to computer intelligence. Hoboken: John Wiley & Sons, 2013.

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

[7] J.C. Cepeda J.L. Rueda, and I. Erlich, "MVMO^S-based approach for identification of dynamic equivalents from PMU measurements", in Proc. 2013 IEEE Grenoble PowerTech, Grenoble, France, pp. 1-6, June 2013.

[8] J.L. Rueda, and I. Erlich, "Optimal Dispatch of Reactive Power Sources by Using MVMO^S Optimization", in Proc. 2013 IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG), Singapore, pp. 9-36, April 2013.

[9] J.L. Rueda, W.H. Guamán, J.C. Cepeda, I. Erlich, and A. Vargas, "Hybrid Approach for Power System Operational Planning with Smart Grid and Small-Signal Stability Enhancement Considerations," IEEE Transactions on Smart Grid - Special Issue on Computational Intelligence Applications in Smart Grids, vol.4, no.1, pp.530-539, March 2013.

[10] H.V. Pham, J.L. Rueda, and I. Erlich, "Online Optimal Control of Reactive Sources in Wind Power Plants," IEEE Trans. on Sustainable Energy - Special Issue on Real-Time Applications of Intelligent Methods in Sustainable Power and Energy Systems. [Online]. Early view since Aug. 2013 at http://ieeexplore.ieee.org/Xplore/home.jsp.

[11] P. Gavela, J.L. Rueda, A. Vargas and I. Erlich, "Performance comparison of heuristic optimization methods for optimal dynamic transmission expansion planning," International Transactions on Electrical Energy Systems. [Online]. Early view since Aug. 2013 at http://onlinelibrary.wiley.com/.

[12] J.J. Liang1, B.Y. Qu, and P.N. Suganthan, "Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization," Technical Report, Nanyang Technological University (Singapore) and Zhengzhou University (China), Dec. 2013. [Online]. Available at: http://www.ntu.edu.sg/home/epnsugan/

[13] W. Nakawiro, I. Erlich, and J.L. Rueda, "A novel optimization algorithm for optimal reactive power dispatch: A comparative study," 4th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, pp. 1555-1561, July 2011.

APPENDIX

TABLE I. RESULTS FOR 10D

Type	Function	Best	Worst	Median	Mean	Std.
Unimodal functions	TF1	4.1538505e-005	3.3624079e-003	2.2436262e-004	4.9540470e-004	7.3489370e-004
	TF2	8.5265128e-014	1.3106106e-007	1.8474111e-010	7.0984178e-009	2.2815024e-008
	TF3	0.0000000e+000	2.4199949e-009	7.9580786e-013	9.8596582e-011	4.3985026e-010
Simple multimodal Functions	TF4	0.0000000e+000	3.4780275e+001	1.1368684e-013	9.5456242e+000	1.4985979e+001
	TF5	4.8459015e-010	1.9998708e+001	1.9969694e+001	1.6580585e+001	7.3335387e+000
	TF6	7.9580786e-013	5.3816735e-002	1.8396041e-004	3.4445483e-003	1.0642149e-002
	TF7	0.0000000e+000	5.6623494e-002	1.7226294e-002	1.8583556e-002	1.4468310e-002
	TF8	0.0000000e+000	1.1368684e-013	0.0000000e+000	6.6874612e-015	2.7016019e-014
	TF9	9.9495906e-001	7.9596674e+000	3.9798362e+000	3.4921111e+000	1.4537646e+000
	TF10	6.2454441e-002	7.0172878e+000	3.7472665e-001	2.1369189e+000	2.3422584e+000
	TF11	3.4149607e+000	2.6033799e+002	1.2204066e+002	9.6276037e+001	7.6480825e+001
	TF12	5.5313035e-006	1.2515415e-001	4.2251393e-002	4.2227834e-002	2.8159324e-002
	TF13	1.3164533e-002	7.2168471e-002	3.4303739e-002	3.5533498e-002	1.4427346e-002
	TF14	3.5090763e-002	1.9656057e-001	8.2094714e-002	8.9059202e-002	3.3743724e-002
	TF15	2.0707686e-001	9.2743499e-001	4.0737798e-001	4.3460098e-001	1.4115855e-001
	TF16	3.8829686e-001	2.1665315e+000	1.4593894e+000	1.4485149e+000	4.0681434e-001
Hybrid functions	TF17	2.0819470e-001	3.9795023e+001	1.1173069e+001	9.3566657e+000	1.0203322e+001
	TF18	7.8135446e-003	3.1372986e+000	1.0082690e+000	7.8259990e-001	7.5051405e-001
	TF19	2.3484966e-002	7.9774510e-001	1.0627733e-001	1.5834023e-001	1.7257304e-001
	TF20	1.0797286e-003	1.7006967e+000	1.5382139e-001	3.1255588e-001	3.9449412e-001
	TF21	1.6782275e-003	1.7177388e+001	3.3086879e-001	1.9347212e+000	4.9969079e+000
	TF22	2.5193267e-002	1.8482503e+000	2.0400219e-001	2.6288606e-001	2.7868574e-001
Composition functions	TF23	3.2945747e+002	3.2945747e+002	3.2945747e+002	3.2945747e+002	2.2963616e-013
	TF24	1.0000000e+002	1.1345264e+002	1.0979901e+002	1.0922582e+002	2.9792711e+000
	TF25	1.0000000e+002	1.3056547e+002	1.1654164e+002	1.1612595e+002	7.3027971e+000
	TF26	1.0001183e+002	1.0005666e+002	1.0003187e+002	1.0003227e+002	1.1270960e-002
	TF27	7.3602613e-001	4.0013955e+002	1.5999972e+000	1.7201876e+001	7.8121553e+001
	TF28	1.0001846e+002	4.7638378e+002	3.5710324e+002	3.6105482e+002	4.4886214e+001
	TF29	1.2763910e+002	2.2569749e+002	1.6006349e+002	1.8140214e+002	3.8833199e+001
	TF30	4.6365669e+002	5.6635445e+002	4.8716618e+002	4.9174528e+002	2.3026773e+001

TABLE II. RESULTS FOR 30D

Type	Function	Best	Worst	Median	Mean	Std.
Unimodal functions	TF1	6.9284155e-004	1.3416580e-003	1.0937358e-003	1.0664527e-003	1.5126562e-004
	TF2	3.9343897e-006	2.3398313e-004	1.1902408e-005	2.3800519e-005	3.6577863e-005
	TF3	2.0615637e-005	2.9302342e-003	1.0312443e-003	1.1057164e-003	6.9600696e-004
Simple multimodal Functions	TF4	2.2737368e-013	8.5265128e-013	3.9790393e-013	4.3802870e-013	1.4167320e-013
	TF5	1.9995461e+001	1.9999967e+001	1.9999802e+001	1.9999556e+001	7.0056880e-004
	TF6	5.0337088e-001	9.9452370e+000	3.0441230e+000	3.6202263e+000	2.1431794e+000
	TF7	4.5474735e-013	5.6564598e-002	1.3642421e-012	2.9900416e-003	9.7666655e-003
	TF8	5.6843419e-013	2.9848772e+000	9.9495909e-001	8.5841154e-001	7.7121192e-001
	TF9	1.0944550e+001	3.7808409e+001	2.3879002e+001	2.5127580e+001	7.6704529e+000
	TF10	2.4224955e+000	1.7305505e+002	9.7602007e+000	1.7863610e+001	3.2232646e+001
	TF11	7.2655596e+002	2.1761397e+003	1.5936079e+003	1.5416901e+003	3.3984542e+002
	TF12	2.1218576e-002	1.8692749e-001	6.2435019e-002	7.2055202e-002	4.1222550e-002
	TF13	9.9612318e-002	2.2697784e-001	1.6201751e-001	1.5729416e-001	3.1523838e-002
	TF14	1.4427060e-001	2.5404204e-001	1.9918166e-001	1.9885619e-001	2.2474439e-002
	TF15	2.0101268e+000	5.3070854e+000	2.6858825e+000	2.8552269e+000	6.5069647e-001
	TF16	8.4915258e+000	1.2904543e+001	9.8367522e+000	1.0209492e+001	1.2670829e+000
Hybrid functions	TF17	1.4295717e+002	1.5310445e+003	1.0256839e+003	9.0081976e+002	3.8675498e+002
	TF18	6.4637000e+000	8.3344861e+001	2.0831026e+001	2.8938239e+001	2.0575334e+001

Type	Function	Best	Worst	Median	Mean	Std.
	TF19	2.0805409e+000	7.3274197e+000	3.0246453e+000	3.0799649e+000	7.4693924e-001
	TF20	9.6456491e+000	4.5813273e+002	5.6901662e+001	1.0915718e+002	1.0924229e+002
	TF21	3.5559068e+001	8.7034775e+002	4.8856072e+002	4.6661696e+002	2.3476450e+002
	TF22	6.4369174e+000	3.3656003e+002	1.4590697e+002	1.4457451e+002	8.1369703e+001
Composition functions	TF23	3.1524410e+002	3.1524410e+002	3.1524410e+002	3.1524410e+002	5.7409039e-014
	TF24	2.2193350e+002	2.2898638e+002	2.2467779e+002	2.2475384e+002	1.0561426e+000
	TF25	2.0259082e+002	2.0583540e+002	2.0322925e+002	2.0329514e+002	5.4842879e-001
	TF26	1.0009903e+002	1.0025226e+002	1.0015760e+002	1.0016223e+002	3.3392707e-002
	TF27	4.0061039e+002	4.0211259e+002	4.0107149e+002	4.0113199e+002	3.6783055e-001
	TF28	7.4880562e+002	9.5279994e+002	8.7765601e+002	8.7677005e+002	3.7514823e+001
	TF29	4.7833087e+002	7.9946761e+002	7.4225190e+002	7.3608474e+002	4.6567941e+001
	TF30	8.3367481e+002	3.3641456e+003	2.0773313e+003	2.0011468e+003	6.3275860e+002

TABLE III. RESULTS FOR 50D

Type	Function	Best	Worst	Median	Mean	Std.
Unimodal functions	TF1	4.2266187e-003	1.1173537e-002	7.3924406e-003	7.5393374e-003	1.6192262e-003
	TF2	1.0424369e-005	4.1093265e-005	1.2963966e-005	1.5210404e-005	5.9390136e-006
	TF3	2.8473395e-003	5.4004028e-003	4.7466100e-003	4.5419901e-003	6.4862612e-004
Simple multimodal Functions	TF4	9.0949470e-013	9.8103112e+001	3.4674486e-012	5.7707713e+000	2.3312773e+001
	TF5	1.9999716e+001	1.9999994e+001	1.9999962e+001	1.9999951e+001	5.2143249e-005
	TF6	5.3564486e+000	2.3550846e+001	9.6660581e+000	1.0513585e+001	3.8081250e+000
	TF7	3.6379788e-012	1.7226294e-002	7.3960403e-003	6.6684322e-003	5.4482413e-003
	TF8	1.0515237e-008	5.9697552e+000	1.9899220e+000	2.2825747e+000	1.5831273e+000
	TF9	3.0843716e+001	1.1243008e+002	8.0591512e+001	7.6931185e+001	1.8507705e+001
	TF10	6.7266472e+000	3.6891721e+002	1.7455123e+001	8.0771513e+001	8.9427244e+001
	TF11	2.3292001e+003	4.3961818e+003	3.6649329e+003	3.5713142e+003	5.5153949e+002
	TF12	1.3209227e-002	2.2067519e-001	3.2788930e-002	4.9288508e-002	4.1326812e-002
	TF13	2.0975908e-001	3.7414928e-001	2.7868431e-001	2.8442643e-001	3.8756850e-002
	TF14	1.7226555e-001	3.1155106e-001	2.3536972e-001	2.3750314e-001	2.8825988e-002
	TF15	2.8945375e+000	8.2415885e+000	5.0008624e+000	4.9989099e+000	1.0526557e+000
	TF16	1.7192708e+001	1.9639758e+001	1.8905269e+001	1.8860060e+001	4.8006770e-001
Hybrid functions	TF17	4.4531697e+001	1.3079483e+003	6.9129152e+002	6.6477133e+002	3.2215270e+002
	TF18	1.3336089e+001	2.3449333e+002	2.9205293e+001	3.4011400e+001	2.9658419e+001
	TF19	5.1735646e+000	1.0442244e+001	6.4318328e+000	6.7580720e+000	1.1645738e+000
	TF20	8.9056013e+000	4.2452261e+001	2.1094352e+001	2.2867161e+001	7.1048284e+000
	TF21	2.4678505e+002	1.0144064e+003	4.9962669e+002	5.3882652e+002	2.0070027e+002
	TF22	5.1836912e+001	7.1691031e+002	4.9214601e+002	4.5841322e+002	1.4391196e+002
Composition functions	TF23	3.4400450e+002	3.4400450e+002	3.4400450e+002	3.4400450e+002	4.5927231e-013
	TF24	2.5478846e+002	2.7110735e+002	2.5668673e+002	2.5788394e+002	3.6553819e+000
	TF25	2.0490574e+002	2.1165716e+002	2.0552516e+002	2.0616942e+002	1.6048113e+000
	TF26	1.0014930e+002	2.0017382e+002	1.0023351e+002	1.0414867e+002	1.9586903e+001
	TF27	3.7308822e+002	7.5844507e+002	5.0499714e+002	5.1075826e+002	7.7736033e+001
	TF28	1.1764687e+003	1.4408022e+003	1.3510220e+003	1.3267138e+003	6.2978029e+001
	TF29	5.0088281e+002	1.5527609e+003	1.1249473e+003	1.1056057e+003	2.3284504e+002
	TF30	8.5320111e+003	1.0216084e+004	9.4339035e+003	9.4212668e+003	3.5441959e+002

TABLE IV. COMPUTATIONAL COMPLEXITY

Dimension	T_0	T_1	$\hat{T}_2$	$\left(\hat{T}_2 - T_1\right)/T_0$
D=10	1.4062500e-001	1.2082375e+004	1.2108575e+004	1.8630818e+002
D=30	1.8750000e-001	5.6994922e+004	5.7937346e+004	5.0262628e+003
D=50	1.4062500e-001	7.4776328e+004	7.4295874e+004	-3.4165588e+003