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

Solving the IEEE-CEC 2014 Expensive Optimization Test Problems by Using Single-Particle MVMO

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*— Mean-Variance Mapping Optimization (MVMO) constitutes an emerging heuristic optimization algorithm, whose evolutionary mechanism adopts a single parent-offspring pair approach along with a normalized range of the search space for all optimization variables. Besides, MVMO is characterized by an archive of n-best solutions from which the unique mapping function defined by the mean and variance of the optimization variables is derived. The algorithm proceeds by projecting randomly selected variables onto the corresponding mapping function that guides the solution towards the best set achieved so far. Despite the orientation on the best solution the algorithm keeps on searching globally. This paper provides an evaluation of the performance of MVMO when applied for the solution of computationally expensive optimization problems. Experimental tests, conducted on the IEEE-CEC 2014 optimization test bed, highlight the capability of the MVMO to successfully tackle different complex problems within a reduced number of allowed function evaluations.**

Keywords*— *Expensive Optimization; Heuristic optimization; mean-variance mapping optimization; single objective optimization.

I. INTRODUCTION

Many real-world optimization problems exhibit a large-scale, nonlinear, non-convex and mixed integer nature, which do not lend themselves to solution by classical optimization methods [1]. Thus, the development and improvement of heuristic optimization algorithms has received great attention, especially in recent years. These algorithms generally adopt a certain mechanism to repetitively generate new candidate solutions based on preceding solutions, which is commonly evaluated through a fitness measure [2]. Each fitness (function) evaluation usually requires computationally intensive computer simulations, so that the number of fitness evaluations that can be performed is limited by time or some other computing resource constraints. This motivates the exploration and development of new algorithmic procedures, which allow a more efficient and quicker solution of complex and high-dimensional optimization problems.

Mean-variance mapping optimization (MVMO), conceived by I. Erlich and first reported in [3], is an emerging optimization algorithm, whose basic conceptual framework has certain similarities to other state-of-art heuristic approaches. Its novel distinguishing feature, however, is its use of a special mapping function applied for mutating the offspring on the basis of mean and variance of the set comprising of the n-best solutions attained so far and saved in the archive. Based on simple mathematical relationships, the shape of the mapping curve is adjusted according to the progress of the search process. As the mapping function is defined in the interval [0, 1] for all optimization variables the original variables have to be rescaled to within this range. However, the fitness evaluation is performed using the actual values in the problem space (i.e. conversion to the original dimension is embedded in this task). MVMO represents a single particle approach based on random search and its own experiences in the search process.

The MVMO has already been applied successfully for the solution of different power system optimization problems, such as the solution of the optimal reactive power allocation problem [4, 5], (the optimal dispatch of energy and reserve) unit commitment and optimization of control energy [6], (the identification of dynamic equivalents) dynamic equivalencing [7, 8], and the development of optimal control strategies [9]-[11]. These applications have indeed evidenced that thanks to the well-designed balance between search diversification and intensification the MVMO exhibits a fast convergence behavior and can find the optimum solution quickly with minimum risk of premature convergence.

Based on the IEEE-CEC 2014 expensive optimization test bed, this paper aims at examining and discussing the performance of MVMO, as a generic optimization engine, that can be used for solving a variety of problems with different dimensions and mathematical properties within reduced computing time. The outline of the paper is as follows: Section II presents a thorough review of MVMO theory. Section III shows the experimental setup and provides a discussion on numerical results of the study. Finally, conclusions are summarized in Section IV.

II. REVIEW OF MVMO

The flowchart of MVMO is given in Fig. 1. As every population-based stochastic optimization technique, the solution framework of MVMO starts with an initialization stage (i.e. definition of the algorithm's parameter settings and generation of random samples for control variables from the space of possible solutions). This is followed by an iterative loop to perform fitness evaluation (i.e. in terms of fulfillment of preset conditions), to update the solution database (i.e.

inclusion or exclusion of candidate solutions in the archive), to select the global best solution (i.e. parent assignment), and to create new candidate solutions (i.e. mutation by projection of the selected variables onto the mapping function and crossover).

The distinctive features of MVMO can be summarized as follows:

- A novel mapping function that is used for mutating genes in the offspring based on the mean and variance of the solution archive
- A compact and dynamically updated solution archive that serves as the knowledge base for guiding the search direction (i.e. adaptive memory). The n-best individuals that MVMO has found so far are saved in the archive and sorted in a descending order of fitness.
- A single parent-offspring pair concept
- The restriction of the range of the search space for all optimization variables internally to [0, 1]. This is a precondition for using the mapping function; on the other hand, it guarantees that the generated offspring is always within the search boundaries. However, fitness evaluation is carried out in the original physical dimension.

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.

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].

B. Solution archive

In the basic MVMO the n best individuals are stored in the solution archive. The archive size is fixed for the entire process and has to be defined by the user beforehand. Experience so far reveals that an archive size of 2-5 is usually sufficient. A larger archive size will result in a rather conservative search with orientation on the saved best populations. The archive is filled up progressively over the iteration steps in a descending order of fitness, so that the first ranked individual is always the best found so far. Once the archive is filled up, an update is performed only if the fitness of the new individual is better than those in the archive. As fitness improves over iterations, the population members keep changing.

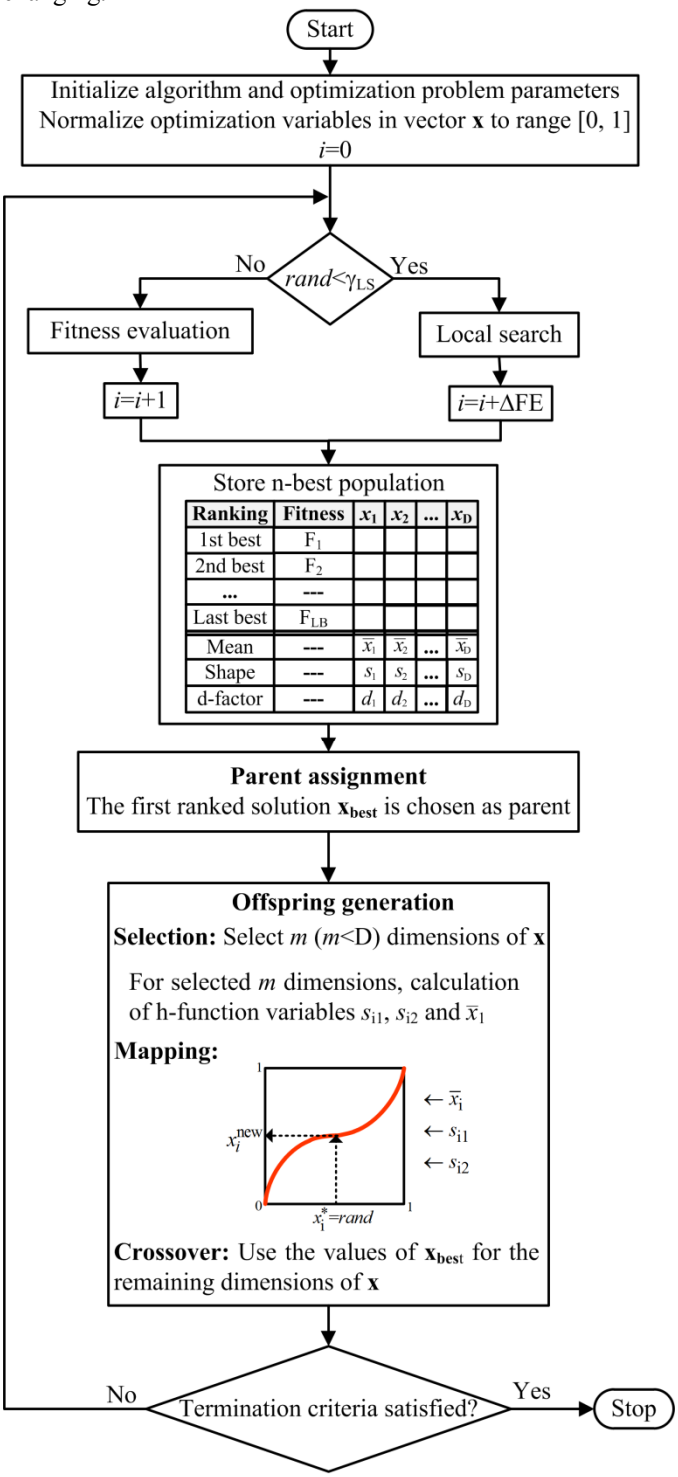

Fig. 1. Flowchart of MVMO. The fitness evaluation counter is denoted by i, whereas ΔFE and *rand* stand for number of fitness evaluations and uniform random number between [0, 1], respectively.

Mean and shape variables are calculated after every update of the archive for each optimization variable x_i (i=1,…,D) using (3) and (4), respectively.

$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_i(j) \tag{3}$$

$$s_i = -\ln(v_i)\cdot f_s \tag{4}$$

with the variance

$$v_i = \frac{1}{n}\sum_{j=1}^{n} (x_i(j) - \bar{x}_i)^2 \tag{5}$$

At the beginning, $\bar{x}_i$ corresponds with the initialized value of x_i, and v_i is set to 1. The shape variable used directly for the mapping function is a derivative of the variance. It can be modified by the scaling variable f_s which represents an alternative way to control the form of the mapping function and thus the search process, as will be discussed in the next section.

C. *Offspring generation*

At every iteration the individual with the best fitness so far in the archive (first position) is used to generate a new descendant (i.e. assigned parent). Besides, m out of D dimensions of the optimization problem are selected for mutation operation via mapping function while the remaining dimensions inherit the corresponding values from the parent. Alternative selection methods are described in [3, 4]. The number m of dimensions to be selected for mutation operation is progressively reduced from m_{ini} to m_{final} as follows:

$$m = \text{round}\left(m_{final} + \text{rand}\left(m^* - m_{final}\right)\right) \tag{6}$$

$$m^* = \text{round}\left(m_{ini} - \alpha^2\left(m_{ini} - m_{final}\right)\right) \tag{7}$$

$$\alpha = \frac{i}{i_{max}} \tag{8}$$

where i denotes fitness evaluation number.

The new value of each selected dimension x_i is determined by

$$x_i = h_x + (1 - h_1 + h_0)\cdot x_i^* - h_0 \tag{9}$$

where x_i^* is a variable varied randomly with uniform distribution and the term h refers to transformation mapping function, which is defined as

$$h(\bar{x}_i, s_1, s_2, x) = \bar{x}_i \cdot (1 - e^{-x\cdot s_{i1}}) + (1 - \bar{x}_i)\cdot e^{-(1-x)\cdot s_{i2}} \tag{10}$$

h_x, h_1 and h_0 are the outputs of the mapping function, based on different inputs given by

$$h_x = h(x = x_i^*), \quad h_0 = h(x = 0), \quad h_1 = h(x = 1) \tag{11}$$

Both input and output of the mapping function cover the range [0, 1]. Note that the shape of the mapping function is determined by the mean $\bar{x}_i$ and the shape factors s_{i1} and s_{i2}. The effect of these parameters on the form of the function is illustrated in Fig. 2, where it is inferred that the search diversity can be enhanced through effective assignment of the shape variables. Note also that, with increasing value of the shape parameter, the mapping curve becomes flatter so that the space to be searched focuses on the region near to the mean value. Fig. 3 demonstrates this characteristic for two variables.

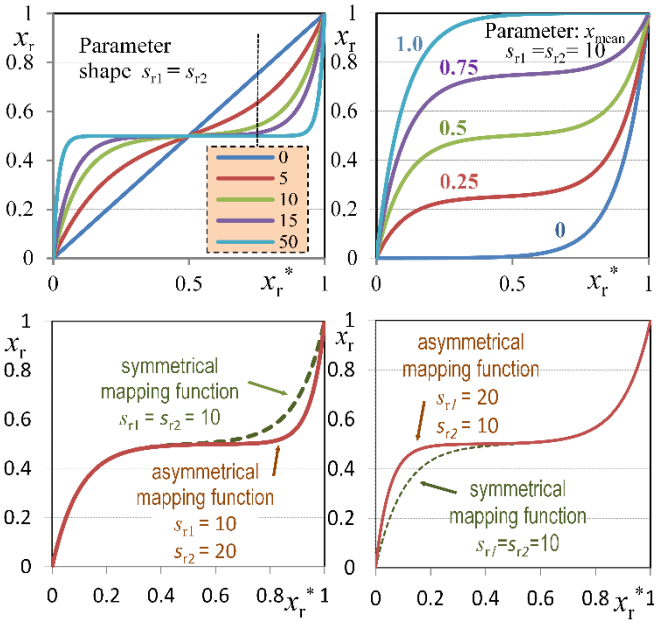

Fig. 2. Change of the mapping function shape for different values of mean and shape factors.

The factor f_s can be used to change the shape of the function (cf. (9)). A small value (e.g. 1.0) allows the slope of the mapping curve to increase and thus enable better exploration, whereas values above 1.0 will result in a rather flat curve and thus lead to improved exploitation. Therefore, f_s is increased as the optimization progresses from a small initial value $f_{s\_ini}^*$ to a higher final value $f_{s\_final}^*$ by using (12) and (13). A considerably high final value of f_s, e.g. $f_{s\_final}^* = 20$, could be used for real-parameter optimization problems with high accuracy concerns.

$$f_s = f_s^*\left(1 + rand\right) \tag{12}$$

$$f_s^* = f_{s\_ini}^* + \alpha^2\left(f_{s\_final}^* - f_{s\_ini}^*\right) \tag{13}$$

where *rand* is a random number with uniform distribution in [0,1].

The shapes factors s_{i1} and s_{i2} of the variable x_i are not calculated directly from (4) but by using the strategy defined in (14).

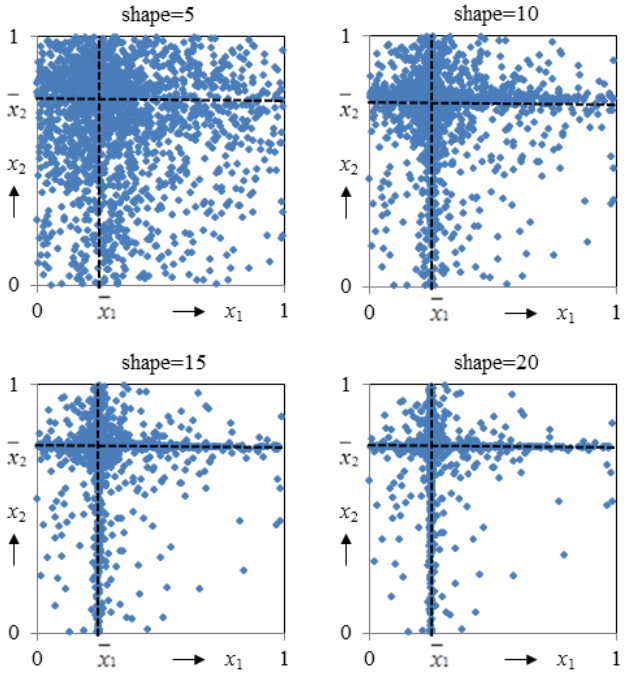

Fig. 3. Search space depending on the shape factor; number of variables=2, $\bar{x}_1$ =0.25, $\bar{x}_2$ =0.75, s_i=s_1 =s_2 for both variables

$$
\begin{aligned}
&s_{i1} = s_{i2} = s_i \\
&\text{if } s_i > 0 \text{ then} \\
&\quad \Delta d = (1 + \Delta d_0) + 2 \cdot \Delta d_0 \cdot (rand - 0.5) \\
&\quad \text{if } s_i > d_i \\
&\quad\quad d_i = d_i \cdot \Delta d \\
&\quad \text{else} \\
&\quad\quad d_i = d_i / \Delta d \\
&\quad \text{end if} \\
&\quad \text{if } rand < 0.5 \text{ then} \\
&\quad\quad s_{i1} = s_i \;\; ; \;\; s_{i2} = d_i \\
&\quad \text{else} \\
&\quad\quad s_{i1} = d_i \; ; \;\; s_{i2} = s_i \\
&\quad \text{end if} \\
&\text{end if}
\end{aligned}
\tag{14}
$$

The initial values of d_i are set to 1 for all variables at the beginning of the optimization. At every iteration, each d_i is scaled up or down with the factor Δd , which is randomly varied between 1 and $1+2\cdot\Delta d_0$. If $d_i > s_i$, the current d_i is divided by Δd which is always larger than 1.0 and thus leads to reduced value of d_i. In case $d_i < s_i$, d_i will be multiplied by Δd resulting in increased d_i. In this way d_i will always oscillate around the current shape factor s_i. The assignment of s_i and d_i to s_{i1} or s_{i2} takes place randomly.

According to our experience, it is recommended to take a non-zero value equal to or smaller than 0.4 for Δd_0. A high value of Δd_0 will indirectly entail wider global search diversification over the entire space whereas a smaller one would lead to concentrated local search aiming at accuracy improvement. By using the described procedure the asymmetric characteristic of the mapping function is also exploited by using different values for s_{i1} and s_{i2} leading to enhanced search performance and zero variance handling. Zero variance can occur when all values of x_i in the archive are identical. In this case the previous non-zero value can be used further. However, this value may result, under circumstances, in stagnating convergence behavior. The procedure overcomes this problem as the mean and variance are calculated only for non-identical values of x_i saved in the archive.

The mean and variance are not calculated before a certain number of solutions are available in the archive. The authors usually start the calculation immediately after two solutions have been obtained. However, it is possible to wait until the archive is filled up completely which will result in more robust initial solutions. In this stage, the search is performed with s_{i1}=s_{i2}=0 which corresponds with a straight line between zero and one as the mapping function. The mean value in this case does not have any effect on the mapping function.

D. Termination criteria

Like many of the heuristic optimization algorithms, the MVMO search process can be terminated either based on a completion of a specified number of iterations, the attainment of a fitness threshold, or lack of fitness improvement over the last iterations. In any case, the termination criterion is determined by the user for a given optimization problem. It is worth emphasizing that the number of iterations in MVMO is equivalent to the number of offspring fitness evaluations which is in practical applications usually much more time consuming than the optimization algorithm itself.

III. Experimental Results and Discussion

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 done in Matlab® Version R2013b and the functionalities of the Parallel Computing Toolbox are used to set a cluster with 8 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. Experimental setting

MVMO-SH is used to solve the IEEE-CEC 2014 expensive optimization test problems listed in Table I. The details of these problems, 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 were carried out under the following considerations:

- Dimension: D=10 for problems {1,4,7,10,13,16,19,22}; D=20 for problems {2,5,8,11,14,17,20,23}; and D=30 for problems {3,6,9,12,15,18,21,24}.
- 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 $OF=TF_i(x)-TF_i(x^*)$, where $TF_i(x^*)$ is the theoretical global optimum of the i-th benchmark function TF given in the 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.

TABLE I. EXPENSIVE OPTIMIZATION TEST PROBLEMS

No.	Function type	Search range
1-3	Shifted sphere	[-20,20]
4-6	Shifted ellipsoid	[-20,20]
7-9	Shifted and rotated ellipsoid	[-20,20]
10-12	Shifted step	[-20,20]
13-15	Shifted Ackley's	[-32,32]
16-18	Shifted Griewank's	[-600,600]
19-21	Shifted rotated Rosenbrock	[-20,20]
22-24	Shifted rotated Rastrigin	[-20,20]

MVMO was executed for all dimensions D=10, 20, 30 with the following parameters:

N_p=1, $f^*_{s\_ini}=0.1$, $f^*_{s\_final}=20$, $\Delta d_0=0.25$, $m_{ini}=D/6$, $m_{final}=D/2$, $\alpha_{LS\_min}=0.23$, α_{LS_max} is defined in such a way that only one LS run is performed ($\gamma_{LS}=1.0$)

The statistical attributes of the error value OF (i.e. best, worst, mean, median, and standard deviation values) calculated after 20 runs are summarized in Tables II to IV in the Appendix, for each D-dimensional case. The following remarks can be deduced from these results:

- *10D and 20D problems*: MVMO-SH is capable of finding near to zero error values for OF in all runs. Thus, it constitutes an efficient tool to solve different types of optimization problems with 10 and 20- dimensional search space, irrespective of the underlying properties (e.g. multimodal, separable/non-separable, discontinuous).
- *30D problems*: The algorithm also allows obtaining error values that are close to zero for almost all problems in all runs, except for TF9, TF21 and TF24, where errors in the order of 10^2 were encountered in some runs.
- Although not shown in the paper, further tests with different parameter settings revealed that the algorithm would have an enhanced performance for these problems. Nevertheless, the use of the same set of parameters evidences the effectiveness of MVMO, as a generic optimization tool, for successfully tackling optimization tasks within a restricted amount of iterations.

Fig. 4- Fig. 11 demonstrate the convergence behaviour of MVMO by using 1 and 6 particles, respectively. Details on the multi-particle approach of MVMO are given in [14]. Besides the factor $f^*_{s\_final}$ has been varied by setting it either to1 or 5. As can be seen, the fastest convergence occurs when only one particle is used. However, nearly the same final results are achieved by 6 particles too but the initial convergence is slower. Also, increasing $f^*_{s\_final}$ does not contribute to fitness improvement; on the contrary, it results in much slower convergence. However, in our experience, larger $f^*_{s\_final}$ values can contribute to improve the final accuracy of the fitness provided that a large fitness evaluation budget is available. The computational complexity measures associated to all test problems are summarized in Table V in the Appendix.

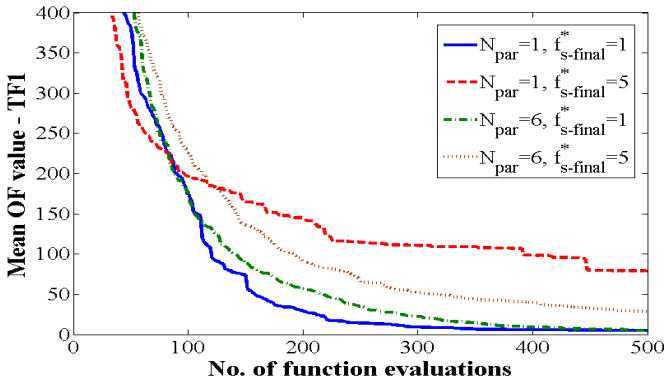

Fig. 4. Average convergence of OF when optimizing TF1.

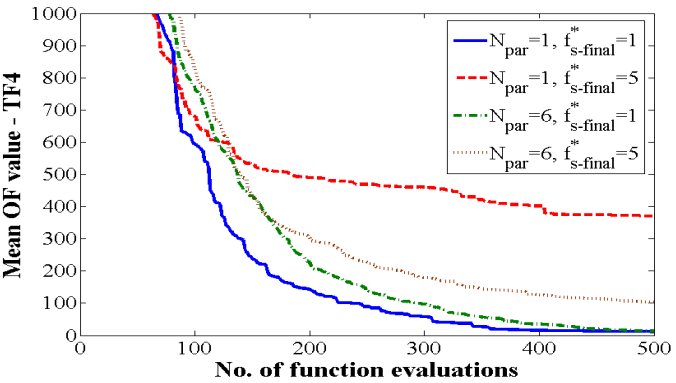

Fig. 5. Average convergence of OF when optimizing TF4.

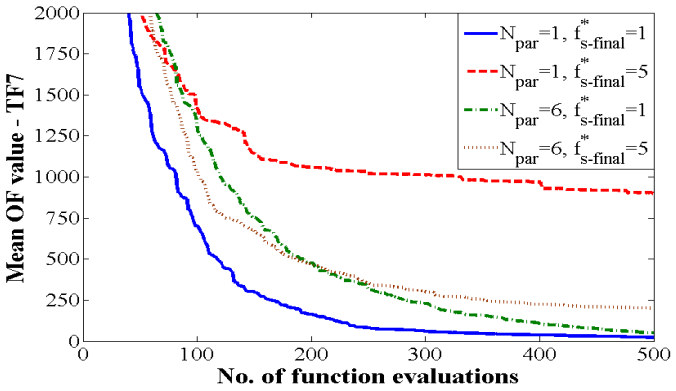

Fig. 6. Average convergence of OF when optimizing TF7.

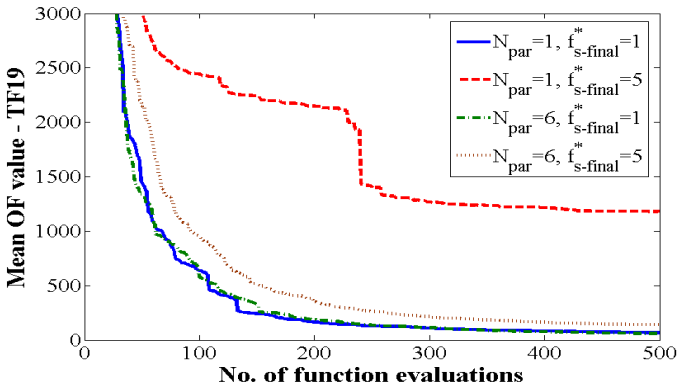

Fig. 10. Average convergence of OF when optimizing TF19.

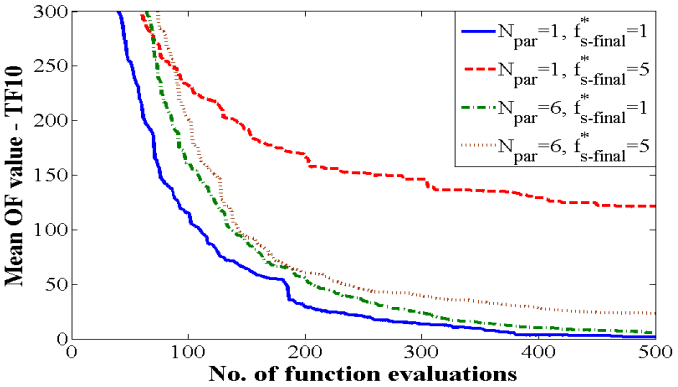

Fig. 7. Average convergence of OF when optimizing TF10.

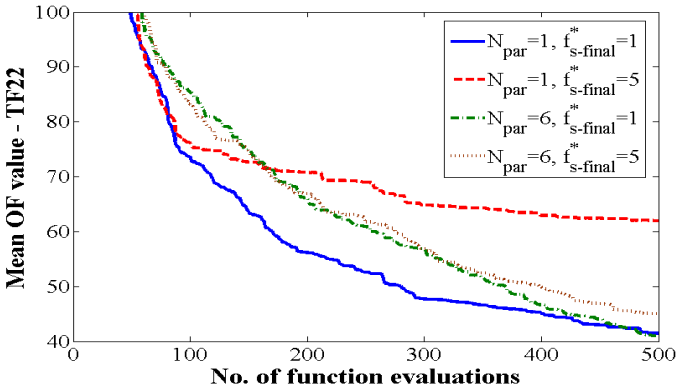

Fig. 11. Average convergence of OF when optimizing TF22.

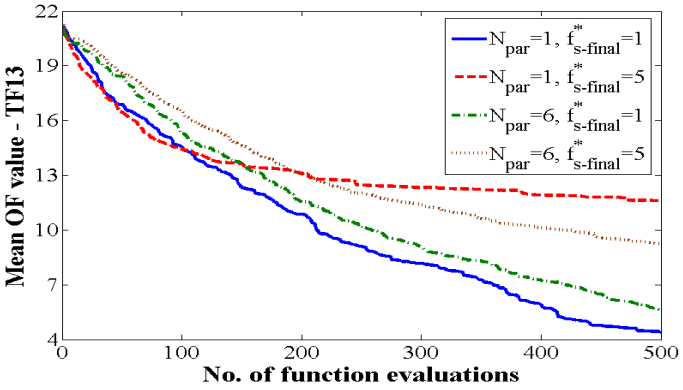

Fig. 8. Average convergence of OF when optimizing TF13.

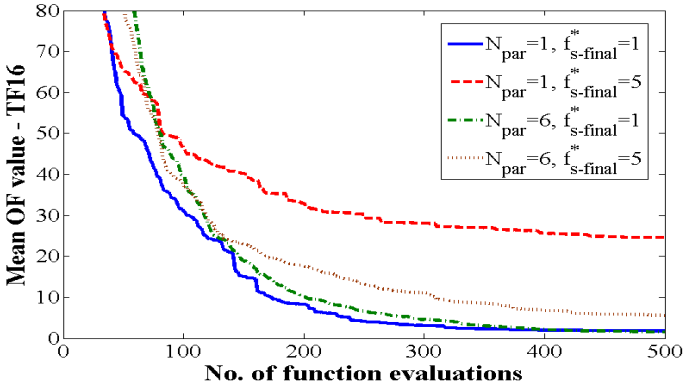

Fig. 9. Average convergence of OF when optimizing TF16.

IV. Conclusions

Despite some similarities with other established heuristic optimization methods, MVMO can be characterized as a novel evolutionary optimization algorithm on account of its unique mapping function and the corresponding search procedure. MVMO underlies a single parent-offspring pair concept. It proceeds based on its own experiences described by a certain number of best solutions saved in the archive. The adaptation of the shape factors throughout the optimization process allows the mapping function to be able to cover the whole search space so that exploration and exploitation is properly balanced. Numerical tests on the IEEE-CEC 2014 expensive optimization test problems show the ability of MVMO to quickly find the global optima of several types of problems with different dimensionality and search landscape even for a limited number of function evaluations, as in the current competition. For this case the single particle approach is the most suitable which represents a unique characteristic of MVMO.

Further research is being carried out towards development of MVMO-based applications for real-time optimization tasks in power system. Possible algorithmic extensions are also being investigated. To make MVMO available to a wider professional community, a dedicated webpage has been set-up [13]. Scientists interested in this topic are encouraged to download the Matlab MVMO source code, test the algorithm and include new ideas and approaches.

REFERENCES

[1] K.Y. Lee and M.A. El-Sharkawi, Modern heuristic optimization techniques. Hoboken: John Wiley & Sons, 2008.

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

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

[4] 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.

[5] I. Erlich, W. Nakawiro, and M. Martinez, "Optimal Dispatch of Reactive Sources in Wind Farms," in *Proc. 2011 IEEE PES General Meeting*, pp. 1-7, Detroit, USA, July 2011.

[6] M. S. Chamba, and O. Añó, "Despacho óptimo de energía y reserva en mercados competitivos empleando algoritmos meta-heurísticos," in *Proc. 2012 IEEE Argencon*, Córdoba, Argentina, June 2012.

[7] I. Erlich, F. Shewarega, C. Feltes, F. Koch and J. Fortmann, "Determination of Dynamic Wind Farm Equivalents using Heuristic Optimization," in *Proc. 2012 IEEE PES General Meeting*, San Diego, USA, July 2012.

[8] J.C. Cepeda, J.L. Rueda, and I. Erlich, "Identification of Dynamic Equivalents based on Heuristic Optimization for Smart Grid Applications" in *Proc. 2012 IEEE world congress on computational intelligence*, Brisbane, Australia, June 2012.

[9] J.L. Rueda, J.C. Cepeda, and I. Erlich, "Estimation of Location and Coordinated Tuning of PSS based on Mean-Variance Mapping Optimization," in *Proc. 2012 IEEE PES General Meeting*, San Diego, USA, July 2012.

[10] P. Chakravarty and G.K.Venayagamoorthy, "Development of optimal controllers for a DFIG based wind farm in a smart grid under variable wind speed conditions," in *Proc. 2011 IEEE International Electric Machines & Drives Conference*, pp. 723-728, Niagara Falls, Canada, May 2011.

[11] 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.

[12] B. Liu, Q. Chen, Q. Zhang, J.J. Liang, P. N. Suganthan, and B.Y. Qu, "Problem Definitions and Evaluation Criteria for Computational Expensive Optimization," Technical Report, Dec. 2013. [Online]. Available at: http://www.ntu.edu.sg/home/epnsugan/

[13] MVMO Webpage: http://www.uni-due.de/mvmo/

[14] I. Erlich, J.L. Rueda, and S. Wildenhues, " Evaluating the Mean-Variance Mapping Optimization on the IEEE-CEC 2014 Test Suite," submitted to *2014 IEEE World Congress on Computational Intelligence*, Beijing, China, July 2014.

APPENDIX

TABLE II. RESULTS FOR 10D

Function	Best	Worst	Median	Mean	Std.
TF1	0.0000000e-000	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000
TF4	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000
TF7	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000
TF10	0.0000000e+000	7.0000000e+000	2.0000000e+000	2.6500000e+000	1.8144160e+000
TF13	1.2468125e-007	8.3263338e+000	5.1898291e+000	4.9190611e+000	2.4288541e+000
TF16	0.0000000e+000	2.4393548e+000	8.8035763e-002	4.3974484e-001	8.0298629e-001
TF19	0.0000000e+000	4.9205325e+000	2.1478707e-001	1.0744827e+000	1.6241784e+000
TF22	9.9495855e+000	4.9747634e+001	2.4873898e+001	2.6171493e+001	1.1377795e+001

TABLE III. RESULTS FOR 20D

Function	Best	Worst	Median	Mean	Std.
TF2	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000
TF5	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000
TF8	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000
TF11	3.0000000e+000	1.6000000e+001	6.0000000e+000	6.5500000e+000	3.3946552e+000
TF14	5.4411689e+000	1.0203427e+001	7.9987297e+000	7.9352529e+000	1.4153127e+000
TF17	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000
TF20	1.2592298e+000	1.3439683e+001	1.1758541e+001	1.0727035e+001	2.9437431e+000
TF23	1.3929417e+001	7.6611498e+001	4.0793231e+001	4.2534391e+001	1.6766867e+001

TABLE IV. RESULTS FOR 30D

Function	Best	Worst	Median	Mean	Std.
TF3	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000
TF6	3.5262512e-005	1.8856361e-001	1.0588107e-002	2.6469859e-002	4.2612482e-002
TF9	2.1086442e-004	4.6591473e+000	2.2189939e-001	8.8494586e-001	1.3624276e+000
TF12	6.0000000e+000	2.7000000e+001	1.2000000e+001	1.2700000e+001	4.9107830e+000
TF15	3.5187283e+000	1.0254779e+001	7.8756990e+000	7.9180254e+000	1.5723034e+000
TF18	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000	0.0000000e+000
TF21	2.1169883e+001	8.0979721e+001	2.4897872e+001	3.4402636e+001	2.1365974e+001
TF24	5.6712515e+001	1.3829882e+002	8.4073811e+001	8.4925808e+001	2.0512060e+001

TABLE V. COMPUTATIONAL COMPLEXITY

Function	$\hat{T}_1/\mathrm{T}_0$	Function	$\hat{T}_1/\mathrm{T}_0$	Function	$\hat{T}_1/\mathrm{T}_0$
TF1	3.7205556e+001	TF2	3.9785000e+001	TF3	4.7105000e+001
TF4	4.4750000e+001	TF5	5.3330000e+001	TF6	6.8845000e+001
TF7	5.3011111e+001	TF8	6.6170000e+001	TF9	9.0460000e+001
TF10	6.0572222e+001	TF11	8.1975000e+001	TF12	1.1618500e+002
TF13	6.8533333e+001	TF14	9.6250000e+001	TF15	1.4070500e+002
TF16	7.6088889e+001	TF17	1.0863000e+002	TF18	1.6124000e+002
TF19	8.4311111e+001	TF20	1.2277000e+002	TF21	1.8323000e+002
TF22	9.2000000e+001	TF23	1.3605000e+002	TF24	2.0491500e+002