TAF
Journal of Applied and Physical Sciences
2016, **2**(2):42-47
JAPS

PRIMARY RESEARCH

Binary mean-variance mapping optimization algorithm (BMVMO)

Ali Hakem Al-Saeedi [1], Oğuz Altun [2, *]

[1, 2] Computer Engineering Department, Yildiz Technical University, Istanbul, Turkey

Index Terms
Mean-Variance Mapping Optimization
Binary Meta-Heuristic Optimization
Discrete evolutionary algorithms

Abstract—Mean-Variance Mapping Optimization (MVMO) is the newest class of the modern meta-heuristic algorithms. The original version of this algorithm is suitable for continuous search problems, so can't apply it directly to discrete search problems. In this paper, the binary version of the MVMO (BMVMO) algorithm proposed. The proposed Binary Mean-Variance Mapping Optimization algorithm compare with well-known binary meta-heuristic optimization algorithms such, Binary genetic Algorithm, Binary Particles Swarm Optimization, and Binary Bat Algorithm over fifteen benchmark functions conducted to draw a conclusion. The numeric experiments result proves that BMVMO is better performance

Received: 24 April 2016
Accepted: 23 14 May 2016
Published: 24 June 2016

I. INTRODUCTION

In computer science, the Meta-heuristic optimization is the set of operations and technique models, use randomness to optimization the candidates and find the best solution [1]. Many Meta-heuristic optimization algorithms inspired by nature [2] some of them are, Particle Swarm Optimization (PSO) [3], Genetic Algorithm (GA) [4] Grey Wolf Optimizer Ant Colony Optimization (ACO) [5], Gravitational Search Algorithm (GSA) [6], Bat search Algorithm (BA) [7] and Dolphin Echolocation [8]. The flexibility of deal with different problems and the high performance of these algorithms make them more popular than tradition optimization technique.

The mean-Variance Mapping Optimization (MVMO) One of the algorithms of modern meta-heuristic high-efficiency, flexible to deal with different kinds of problems. The unique features of MVMO algorithm use the special statistical characteristics function for mutation operation named mapping function [9] this function mathematically depend on mean and variance of n- best solutions. And the search range of MVMO algorithm is a continuous value between [0, 1].

The original version of many meta-heuristic algorithms deals with continuous problems. There are different methods to harmonize these algorithms with discrete problems. [10] proposed a probability estimation operator in order to solve discrete problems by DE. But the binary version of BDE different from originated algorithm. The Binary Bat Algorithm (BBA) [11], Binary GSA , Binary PSO (BPSO) [12] use the transfer function for solving binary problems with conserving the Original versions of these algorithms. For that, use the transfer function with the MVMO for binary search to order to preserve standards concepts of MVMO in the search process

In this paper a proposal the Binary version of MVMO algorithm named BMVMO by employing the concept of the transfer function for adapt to binary search problems.

*Corresponding author: Oğuz Altun
E-mail: oguz211@gmail.com

Evaluate the performance of BMVMO and compare with well-known meta-heuristic algorithms, Binary GA (BGA), BBA, and BPSO, by using fifteen functions of CEC 2015 and the result proves the BMVMO is better performance

I. MEAN-VARIANCE MAPPING OPTIMIZATION ALGORITHM

MVNO is the newest class of population-based stochastic optimization technique [13]. The uniformity amongst MVMO and other stochastic optimization technique in basic evolutionary operations characteristic are operations selection such as crossover, and mutation. But the features that distinct the MVMO are the search space and all optimization operations internal of MVMO bounded between [0, 1], and use the unique mutation, by use special mapping function for mutation [14]. The mapping function depends on mean and variance of the n-best solutions, calculated as following:

$$x_i = \frac{1}{n}\sum_{j=1}^{n} x_i(j) \tag{1}$$

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

The new population created by applying the H-function as fellowing:

$$X_i = h_x + (1 - h_1 + h_0).x_i - h_0 \tag{3}$$

The H-function is defined as following:

$$h(x, s_1, s_2, x) = x.(1 - e^{-x.s_1}) + (1 - x).e^{-(1-x).s_2} \tag{4}$$

$$h_x = h(x = x_i), h_0 = h(x = 0), h_1 = h(x = 1)$$

Where j = 1, 2, 3.....n, n = population size, X_i Offspring .x_i = mathematical mean, v_i = variance and s_1, s_2 shape variables.

The shape variables depends on value of s_i which calculate :

$$s_i = -ln(vi).fs \tag{5}$$

Where f_s is function control on shapes vaFigure1 explain the basic steps of MVMO algorithm.

A. Binary MVMO algorithm

In binary search style, the particles shift inside search space to different positions by flipping a different number of bits can represent as the things are rolling inside hypercube during rotation (Kennedy and Eberhart 1997). The Original version of MVMO the range of search space bounded between [0,1]. The crossover to generate next generations using a multi-parent strategy as fallowing:

$$X = x_k + \beta(x_a - x_b) \tag{6}$$

Where X is offering, x_k ,x_a ,x_b are parents selected randomly. β is real value and calculated as fallowing:

$$\beta = 2(rand - (1 - ((\frac{\text{Current iteration}}{max-iteration})^2\ 0.9))) \tag{7}$$

Therefore; MVMO cannot be directly applied to the research binary or discrete problems. To solve this problem, using a transfer function to harmonize MVMO with the binary research and also to achieve the essential principal of the binary research it's the search value is either 0 or 1, but before using the transfer function there are some issues that need to be taken into consideration [12]:

1- Transfer function work in range [0,1].
2- The high absolute value of the transfer function gives a high probability of changing particle value and vice versa.

The value of mapping function is restricted [0,1] therefore can be employed as input to the transfer function for the mutation the particle as following:

$$V\left(m_i^k(t)\right) = |\frac{2}{\pi}\arctan(\frac{2}{\pi}(m_i^k(t)))| \tag{8}$$

$$X_i^k(t+1) = \begin{cases} X_i^k(t)^{-1} & \text{If } Rand < V(m_i^k(t)) \\ X_i^k(t) & Rand \geq V(m_i^k(t)) \end{cases} \tag{9}$$

Where $V(m_i^k(t))$ is Transfer function, $X_i^k(t)^{-1}$ is Complement $X_i^k(t)$ " 0 →1,1→0", $X_i^k(t)$ is Offspring of i-th child in t-iteration with k-dimensions, $m_i^k(t)$ The value return from mapping function and Rand is continuous value limited between [0,1]. Figure 2 explain the proposed transfer function

The extension in BMVMO for improvement performance updates a value of shape factors s1, s2, control shape factor fs and Variable increment Δd.

A. Update control shape factor f_s

$$f_2 = f_s^* (1 + Rand) \tag{10}$$

$$f_s^* = f_{s\_ini}^* + (\frac{i}{max-iteration})^2 (f_{s_{fin}}^* + f_{s_{ini}}^*) \tag{11}$$

Where: Values of $f_{s\_ini}^*$ and $f_{s_{fin}}^*$ greeter than zero

B. Update shape factors s1, s2

To update the shape factors s1 and s2, we need to give an initial value to the d_i and adopt the update on the s_i value. Then we check if the s_i is bigger than 0 (we check the d_i if it is bigger than s_i then di= di .Δd otherwise $d_i=d_i/$ Δd. Then we choose the random number and check if it is bigger than 0.5 then $s_1=s_i$ and $s_2=d_i$, but if it is smaller than 0.5 then vise verse. If either s_i is smaller than or equal to 0 then $s_1$and s_2 are equal to the s_i

C. Variable increment Δd

$$\Delta d = (1 + 0\ \Delta d_0) + 2\ \Delta d_0(Rand - 0.5) \tag{12}$$

ISSN: 2414-3103
DOI: 10.20474/japs-2.2.3

$$\Delta d_0 = d_{0\_ini} + \left(\frac{i}{max-iteration}\right)^2 (\Delta d_{0\_fin} + \Delta d_{0\_ini}) \quad (13)$$

Where Values of Δd_{0_fin} and Δd_{0_ini} greeter than zero.

The steps of proposed BMVMO algorithm are:

x_i random population with k-dimension and ,i- Population size

set value of d_i , $f_{s\text{-}ini}$,$f_{s\text{-}fin}$, $\Delta d_{0\text{-}ini}$,$\Delta d_{0\text{-}fin}$

While t < max_iteration

Evaluation population

Save n-best solution

Mean = mean(n-best solution) Eq(1)

Variance = variance(n-best solution)Eq(2)

Classification population good & bad

If $x_i \in$ bad

x_i= uniform crossover (select parents randomly)

endif

Update value d_i , Δ d (Eq12)), , f_s (Eq(10)

x_i = mapping function (x_j)Eq(8,9) ,where ($x_j \subset x_i$)

Endwhile.

II. TEST FUNCTIONS

For testing performing of the algorithms (BMVMO, BBA, BGA, and BPSO) use the 15 functions of IEEE-CEC 2015 benchmark functions are single objective optimization (Qu, B. Y., 2014) are divide into 3 groups (f1,f2) unimodal function , (f3,f4,f5) simple multimodal function,(f6,f7,f8) hybrid function , and rest functions are composite functions.

III. NUMERIC AND EXPERIMENT RESULT

The algorithms use in the comparative study with BMVMO are Binary Genetic Algorithm BGA, Binary Particle Swarm Optimization [12], Binary Bat Algorithm [11] because these algorithms popular of binary meta-heuristic fields and succeeded in solving many binary optimization problems. Moreover, the BBA and BPSO deploy transfer function in excellent style without changing the original form of these algorithms. In comparison prefer to use stander version of these algorithms in comparative.

The primary parameters set for BGA crossover percentage and mutation rate 0.3, Roulette Wheel use for parent selection and for crossover used a uniform crossover. While for BBA loudness rate 0.25, plus rate 0.5 , maximum frequency 2 and minimum frequency 0 . While for BPSO inertia weight 1 , maximum inertia weight 1 , minimum inertia weight 0.05 , c_1,c_2 =0.49 , maximum velocity 4 and minimum velocity -4. While for BMVMO size of solution achieve 20 , d_i 1 , $\Delta d_{0\text{-}ini}$ 0.02 , , $\Delta d_{0\text{-}fin}$ 0.05 , $f_{s\text{-}ini}$ 1 and $f_{s\text{-}fin}$ 20. For all algorithms above use 30 dimensions and 100 sizes of the population with 1500 iterations , and repeat every function 30 iteration and use an average of these iterations in the in the comparison, and the stop criteria are the maximum iteration.

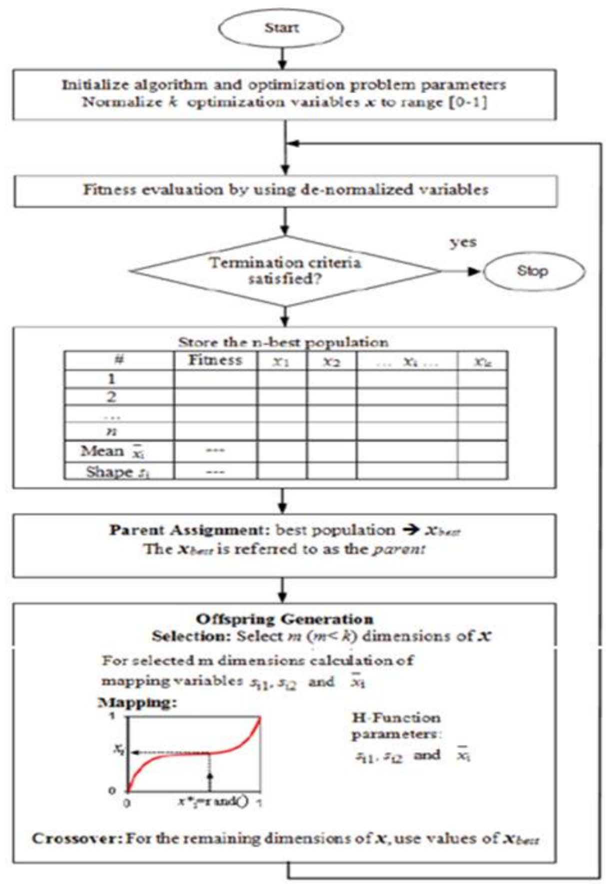


Fig. 1. Basic step of MVMO algorithm

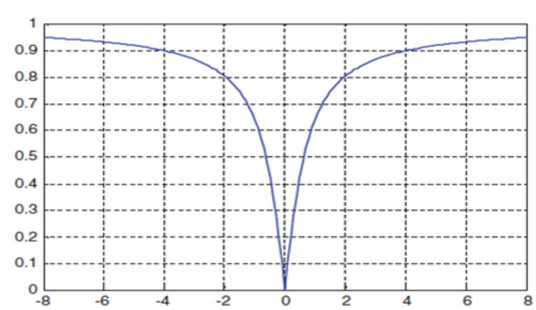


Fig. 2. Proposed transfer function

TABLE 1

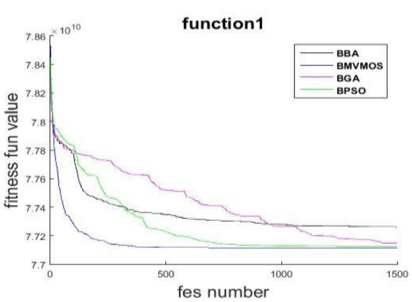
ISSN: 2414-3103
DOI: 10.20474/japs-2.2.3

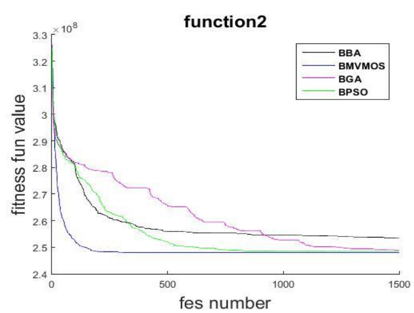
TAF Publishing

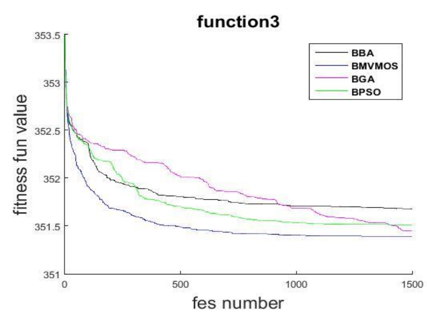
COMPARISON OF BPSO, BGA, BBA AND DMVMO OVER 15 TEST FUNCTIONS OF 30 DIMENSIONS AND 1500 ITERATION

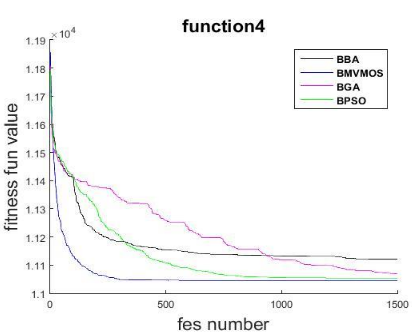
fun	BMVMO		BBA		BPSO		BGA	
	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev	Mean	Std. Dev
f1	**7.72E+10**	**1.32E+08**	7.74E+10	1.61E+08	7.73E+10	2.36E+08	7.74E+10	2.29E+08
f2	**2.49E+08**	**6393937**	2.59E+08	9210521	2.55E+08	11464837	2.62E+08	12287816
f3	**351.5272**	**0.23812**	351.8331	0.217999	351.7235	0.291511	351.9016	0.319559
f4	**11066.79**	**76.5067**	11175.01	93.35784	11133.27	127.1745	11215.55	124.9636
f5	507.9713	0.853272	508.3915	0.995869	**507.9457**	**1.298267**	508.6814	1.231251
f6	**606.6553**	**0.007412**	606.6665	0.008235	606.6607	0.011355	606.6693	0.011671
f7	**844.361**	**0.269909**	844.7394	0.347229	844.5716	0.47364	844.8921	0.465846
f8	**49061590**	**490642.7**	49657187	570070.2	49435109	759183.1	49956352	754423.1
f9	**914.3755**	**0.055842**	914.441	0.055959	914.4059	0.072981	914.4553	0.060275
f10	**1.01E+09**	**8658304**	1.02E+09	10308575	1.01E+09	13055275	1.02E+09	13360288
f11	**2176.334**	**9.445389**	2187.157	10.54527	2183	14.28329	2192.542	14.30543
f12	**1619036**	**36316.39**	1655830	39589.26	1641424	54531.86	1672572	58681.06
f13	**4671.701**	**7.429789**	4682.868	10.32592	4677.858	12.40378	4687.291	11.79262
f14	**2325.193**	**4.067205**	2331.665	4.743113	2328.729	6.928223	2334.058	7.102059
f15	**6693.273**	**18.14667**	6712.949	21.46824	6704.959	26.36587	6723.303	26.11719

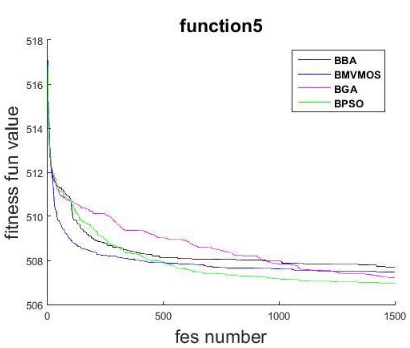
The mean and standard deviation of the results found over the 30 independent runs of each algorithm

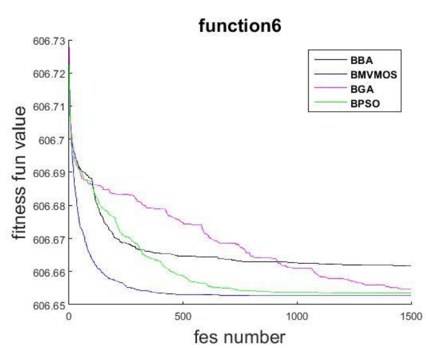












ISSN: 2414-3103
DOI: 10.20474/japs-2.2.3

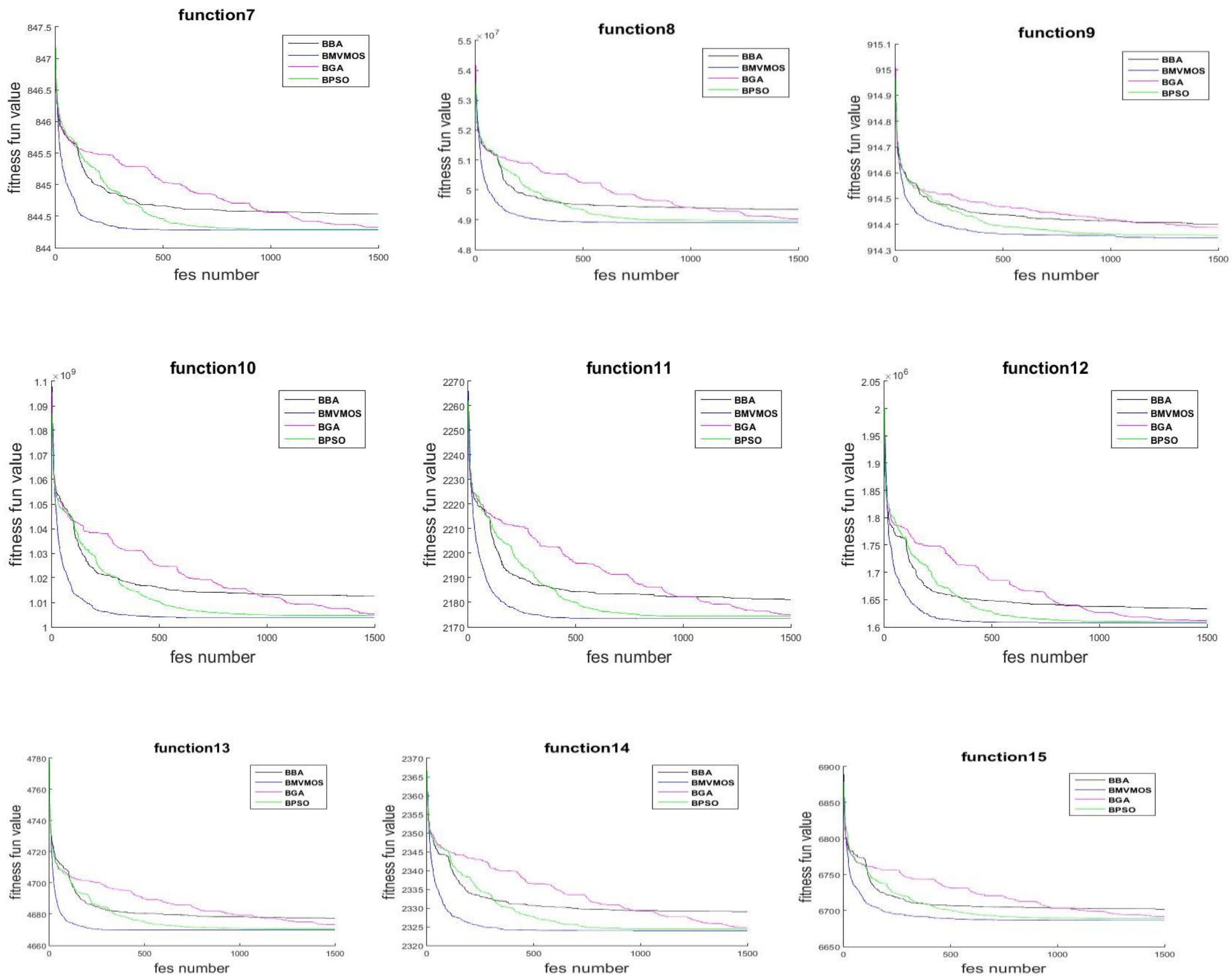


Fig. 3. Comparison between BMVMO, BBA, BPSO, and B

The table 1 shows the statistical results mean and stander division of the comparative algorithms. And figure 3 illustrate the behavior of BMVMO, BGA, BBA, and BPSO in 15 evaluation functions.

The summary of result proven the BMVMO have a good performance at most benchmark functions among binary meta-heuristic optimization algorithms (BGA, BBA, and BPSO).

The performance of a very close between algorithms at f6, while at f5 the BPSO have better performance than BMVMO. According to statistical study in table 1. We can say the BMVMO proven worthiness among the binary meta-heuristic optimization algorithms.

IV. CONCLUSION AND FUTURE WORK

MVMO is newest class of meta-heuristic algorithm, search within the continuous range [0, 1] therefore can't apply directly to the binary problem. Use transfer function for adapting MVMO to binary search without change original form of an algorithm. Comparison the BMVMO performance with BGA, BBA, and BPSO by use 15 benchmark functions of CEC 15. The statistical study proved the BMVMO worthiness among binary meta-heuristic optimization algorithms. For the future work, study the effect change dimensional of the problem and use the different type of the transfer function on the performance of the BMMO algorithm and apply BMVMO in different application such as feature selection.

REFERENCES

ISSN: 2414-3103
DOI: 10.20474/japs-2.2.3

[1] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity.* New York, NY: Courier Dover Publications, 1998.

[3] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *IEEE International Conference on Computational Cybernetics and Simulation,* 1997, pp 4104-4108. **DOI:** 10.1109/icsmc.1997.637339

[4] J. Holland, *Adaptation in Natural and Srtificial Systems*," Ann Arbor, MI: The University of Michigan Press, 1975.

[5] M. Dorigo, V. Maniezzo and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29-41, 1996. **DOI:** 10.1109/3477.484436

[6] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, "BGSA: binary gravitational search algorithm," *Natural Computing*, vol. 9(3), pp. 727-745, 2010. **DOI:** 10.1007/s11047-009-9175-3

[7] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in J. R. Gonzalez, D.A. Pelta, C. Cruz, G. Terrazas, N. Krosnogoret eds. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010).* Berlin, Germany: Springer, Berlin, 2010, pp 65-74. **DOI:** 10.1007/978-3-642-12538-6_6

[8] A. Kaveh and N. Farhoudi, "A new optimization method: Dolphin echolocation," *Advances in Engineering Software*, *59*, 53-70, 2013. **DOI:** 10.1016/j.advengsoft.2013.03.004

[9] I. Erlich, J. L. Rueda, S. Wildenhues and F. Shewarega, "Evaluating the mean-variance mapping optimization on the IEEE-CEC 2014 test suite," in *IEEE Congress on Evolutionary Computation (CEC),* 2014, pp. 1625-1632. **DOI:** 10.1109/cec.2014.6900516

[10] L. Wang, X, Fu, M. I. Menhas and M. Fei, "*A modified binary differential evolution algorithm*," in K. Li, M. Fei and Jia L, eds. *Life System Modeling and Intelligent Computing, Lecture Notes in Computer Science.* Berlin, Germany: Springer, 2010, pp 49-57. **DOI:** 10.1007/978-3-642-15597-0_6

[11] S. Mirjalili, S. M. Mirjalili and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46-61, 2014. **DOI:** 10.1016/j.advengsoft.2013.12.007

[12] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1-14, 2013. **DOI:** 10.1016/j.swevo.2012.09.002

[13] Vasant, P., and Balbir Singh. "Solving Economic Dispatch By Using Swarm Based Mean-Variance Mapping Optimization (MVMOS)," *Global Journal of Technology and Optimization* vol. 6, no. 3, 1-8.

[14] I. Erlich, G. K. Venayagamoorthy and W. Nakawiro, "A mean-variance optimization algorithm," in *IEEE Congress on Evolutionary Computation,* 2010, pp.1-6. **DOI:** 10.1109/CEC.2010.5586027

— This article does not have any appendix. —