Khoa et al., *Cogent Engineering* (2015), 2: 1076983
http://dx.doi.org/10.1080/23311916.2015.1076983

cogent engineering

ELECTRICAL & ELECTRONIC ENGINEERING | RESEARCH ARTICLE

Solving economic dispatch problem with valve-point effects using swarm-based mean–variance mapping optimization (MVMO^S)

T.H. Khoa[1*], P.M. Vasant[1], M.S. Balbir Singh[1] and V.N. Dieu[2]

Received: 23 April 2015
Accepted: 21 July 2015
Published: 24 August 2015

*Corresponding author: T.H. Khoa, Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, 31750 Tronoh, Perak, Malaysia
E-mail: trhkhoa89@gmail.com

Reviewing editor:
Siew Chong Tan, University of Hong Kong, Hong Kong

Additional information is available at the end of the article

Abstract: Mean–variance mapping optimization (MVMO) is a new population-based metaheuristic technique which is successfully applied for different power system optimization problems. The special feature of MVMO is the mapping function applied for the mutation based on the mean and variance of *n*-best population. Recently, the modified version of MVMO has been developed to become more powerful, named as swarm-based mean–variance mapping optimization (MVMO^S). This paper proposes MVMO^S as a new approach for solving the economic dispatch (ED) problem considering valve-point effects. To validate the performance of the proposed method, the MVMO^S is tested on three systems including 3, 13, and 40 thermal generating units with valve-point effects and the obtained results from MVMO^S are compared to those from other existing methods in the literature. Test results have indicated that the proposed MVMO^S is more robust and produces better solution quality than many other methods. Therefore, the MVMO^S is efficient for solving the ED with valve-point effects.

Subjects: Computer Science; Electrical & Electronic Engineering; Power & Energy

Keywords: mean–variance mapping optimization; economic dispatch; valve-point effects; meta-heuristic; nonconvex objective function; swarm-based mean–variance mapping optimization

ABOUT THE AUTHOR

P.M. Vasant is a senior lecturer at Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS in Malaysia. His research interests include Soft Computing, Hybrid Optimization, Holistic Optimization and Applications. He has co-authored research papers and articles in national journals, international journals, conference proceedings, conference paper presentation, and special issues lead guest editor, lead guest editor for book chapters' project, conference abstracts, edited books and book chapters. In the year 2009, P.M. Vasant was awarded top reviewer for the journal *Applied Soft Computing* (Elsevier).

PUBLIC INTEREST STATEMENT

In recent years, swarm intelligence has been widely applied to a variety of fields in engineering due to its outstanding characteristics for solving optimization problems with complex objective function and constraints. This paper presents a swarm intelligent approach for solving the nonconvex economic dispatch problems which is one of the important tasks in power generation systems. This problem is often related to fuel cost saving. Real-world economic dispatch problems have nonconvex objective functions with complex constraints. This leads to difficulty in finding the global optimal solution. Over the past decades, various optimization techniques have been applied to economic dispatch problems. In general, these techniques can be classified into classical calculus based methods, Artificial Intelligence techniques and hybrid methods. However, nonconvex optimization problems are still a challenge for engineers and decision-makers in the industry. Hence, there is always a need for developing new techniques for solving nonconvex problems.

1. Introduction

In power system, thermal generating units are supplied with multiple fuel sources such as coal, natural gas, and oil. The price of these fuels is highly volatile and faces depletion. Hence, the economical operations of the power systems gained importance. The economic dispatch (ED) is defined as the process of allocating the real power output of generating units to meet required load demand so as their total fuel cost is minimized while satisfying all physical and operational constraints (Dieu, Schegner, & Ongsakul, 2013).

Traditionally, the fuel cost function of each generating unit is presented as the quadratic function approximations and is solved using mathematical programming techniques such as lambda iteration method, Newton's method, gradient search, dynamic programming (Wollenberg & Wood, 1996), and quadratic programming (Fan & Zhang, 1998). However, most of these techniques are not capable of dealing with nonconvex and nonlinear ED problems. The ED problem is more practical when considering the effects of valve-point loadings. The valve-point effects (VPE) can cause the input–output curve of thermal generators to become more complicated. Therefore, the ED should be represented as nonconvex or nonsmooth optimization problem. This leads to difficulty in finding global optimum solution. More advanced optimization methods based on artificial intelligence concepts are implemented effectively to deal with ED problems such as genetic algorithm (GA) (Chiang, 2005), evolutionary programming (EP) (Sinha, Chakrabarti, & Chattopadhyay, 2003), artificial bee colony (ABC) (Hemamalini & Simon, 2010; Le Dinh, Vo Ngoc, & Vasant, 2013; Secui, 2015), ant colony optimization (ACO) (Pothiya, Ngamroo, & Kongprawechnon, 2010), evolutionary strategy optimization (ESO) (Pereira-Neto, Unsihuay, & Saavedra, 2005), and differential evolution (DE) (Noman & Iba, 2008). Recently, particle swarm optimization (PSO) is the most popular method applied for solving the ED problems, especially for nonconvex problems (Lin, Chen, Tsai, Yuan, et al., 2015; Shahinzadeh, Nasr-Azadani, & Jannesari, 2014; Vasant, Ganesan, Elamvazuthi, 2012). Several improvements of PSO method are developed for solving.

ED problem with valve-point loading effects such as modified particle swarm optimization (MPSO) (Park, Lee, Shin, & Lee, 2005), anti-predatory particle swarm optimization (APSO) (Selvakumar & Thanushkodi, 2008), self-organizing hierarchical particle swarm optimization (SOH_PSO) (Chaturvedi, Pandit, & Srivastava, 2008), simulated annealing like particle swarm optimization (SA-PSO) (Kuo, 2008), PSO with recombination and dynamic linkage discovery (PSO-RDL) (Chen, Peng, & Jian, 2007), new PSO with local random search (NPSO-LRS) (Selvakumar & Thanushkodi, 2007), improved coordinated aggregation-based particle swarm optimization (ICA-PSO) (Vlachogiannis & Lee, 2009), quantum-inspired PSO (QPSO) (Meng, Wang, Dong, & Wong, 2010), a modified hybrid PSO and gravitational search algorithm based on fuzzy logic (PSOGSA) (Duman, Yorukeren, & Altas, 2015). These improved PSO methods can obtain high-quality solutions for the problem. The PSO method is continuously improved for dealing with large-scale and complex problems in power systems. In addition, hybrid methods are also developed for solving the nonconvex ED problems by combining advantages of the single methods such as hybrid EP with sequential quadratic programming (EP-SQP) (Attaviriyanupap, Kita, Tanaka, & Hasegawa, 2002), integration particle swarm optimization with sequential quadratic programming (PSO-SQP) (Victoire & Jeyakumar, 2004), hybrid technique integrating the uniform design with the genetic algorithm (UHGA) (He, Wang, & Mao, 2008), self-tuning hybrid differential evolution (self-tuning HDE) (Wang, Chiou, & Liu, 2007), combining of chaotic differential evolution and quadratic programming (DEC-SQP) (Coelho & Mariani, 2006), hybrid GA, pattern search, and sequential quadratic programming (GA-PS-SQP) (Alsumait, Sykulski, & Al-Othman, 2010), hybrid differential evolution with biogeography-based optimization (DE-BBO) (Bhattacharya & Chattopadhyay, 2010), hybrid harmony search with arithmetic crossover operation (ACHS) (Niu, Zhang, Wang, Li, & Irwin, 2014). The hybrid methods have become among the most effective search techniques for obtaining high-quality solutions. However, the hybrid methods may be slower and more algorithmically complicated than conventional methods since they combine several operations into one technique.

The MVMO is a novel optimization algorithm which is conceived and developed by István Erlich (Erlich, Venayagamoorthy, & Worawat, 2010). This algorithm also falls into the category of the so-called "population-based stochastic optimization techniques". Recently, the extensions of MVMO has been developed by Rueda and Erlich (2013), which is named MVMO^S. The search process of MVMO^S starts with a set of particles. In addition, two parameters of MVMO including the scaling factor and variable increment parameters are extended to enhance the mapping. Hence, the ability for global search of MVMO^S is more powerful than the single particle version. In this paper, MVMO^S is proposed for solving the ED problem with valve-point effects.

The remaining organization of this paper is as follows. Section 2 presents the formulation of the ED problem with valve-point effects. The review of MVMO, extension of MVMO–MVMO^S, and implementation of the proposed MVMO^S to ED problem are exhibited in Section 3. The numerical test and results discussion are shown in Sections 4 and 5, respectively. The paper is concluded in Section 6.

2. Problem formulation

In this study, the VPE is considered as practical operation of generators. The VPE is a natural characteristic of a thermal turbine. The turbine of generating unit has many admission steam valves. The opening of these steam valves increase the throttling losses rapidly, leading to rise the incremental heat rate suddenly. The VPE is the direct result of the practical operation of thermal generating unit which produces ripples effects on the input–output curve as seen in Figure 1.

The VPE makes the fuel cost function highly nonlinear and nonsmooth containing multiple minima. The fuel cost function is described as the superposition of sinusoidal functions and quadratic functions.

The model of ED problem with VPE is formulated as follows (Dieu et al., 2013):

$$\text{Min } F = \sum_{i=1}^{N} \left(a_i + b_i P_i + c_i P_i^2 + \left| e_i \sin(f_i (P_{i,\,min} - P_i)) \right| \right) \tag{1}$$

subject to

(a) Real power balance constraint: The total active power output of generating units must be equal to total active power load demand plus power loss:

$$\sum_{i=1}^{N} P_i = P_D + P_L \tag{2}$$

where the power loss P_L can be approximately calculated by Kron's formula (Wollenberg & Wood, 1996):

$$P_L = \sum_{i=1}^{N} \sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{0i} P_i + B_{00}. \tag{3}$$

(b) Generator capacity limits: The active power output of generating units must be within the allowed limits:

$$P_{i,\,min} \le P_i \le P_{i,\,max}. \tag{4}$$

3. MVMO^S for ED problem with VPE

3.1. MVMO and MVMO^S

Mean–variance mapping optimization (MVMO) is a new optimization algorithm which falls into the category of the so-called "population-based stochastic optimization technique". The similarities between MVMO and the other known stochastic algorithms are in three evolutionary operators including selection, mutation, and crossover. The major differences between MVMO and other existing techniques are summarized in Erlich et al. (2010) as follows:

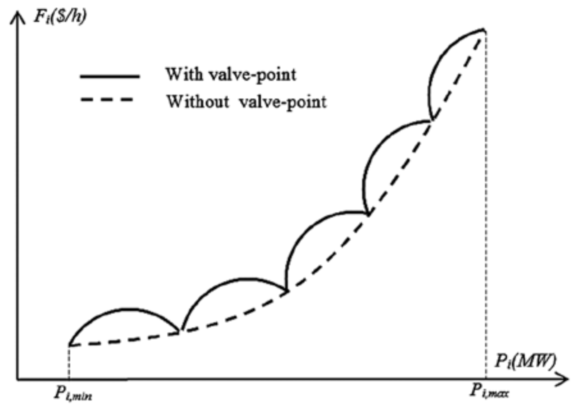


Figure 1. Fuel cost curve of units with valve-point effects.

- The key feature of MVMO is a special mapping function which applied for mutating the offspring. The mapping function is described by the mean and variance of *n* best solutions stored in the archive.
- The total space for searching of all variables is limited within the range from 0 to 1. The output of mapping function is always inside [0, 1]. However, the function evaluation is carried out always in the original scales.
- MVMO is a single-agent search algorithm because only a single offspring is generated in each iteration. Therefore, the number of fitness evaluations is identical to the number of iterations.
- A compact and dynamically updated solution archive serves as the knowledge base for guiding the search direction (i.e. adaptive memory). The normalized *n*-best are filled up in the archive progressively over iterations and sorted in a descending order of fitness.

Swarm-based mean–variance mapping optimization ($MVMO^S$) is an extension of the original version MVMO. The difference between MVMO and $MVMO^S$ is the initial search process with particles. MVMO starts the search with single particle while $MVMO^S$ starts the search with a set of particles. At the beginning of the optimization process of $MVMO^S$, each particle performs *m* steps independently to collect a set of reliable individual solutions. Then, the particles start to communicate and to exchange information. MVMO is extended two parameters including the scaling factor f_s and variable increment Δd parameter to enhance the mapping. Therefore, the search global ability of $MVMO^S$ is strengthened.

3.2. Handing of constraints

Neglecting the transmission power loss, the equality constraint (2) is rewritten by:

$$\sum_{i=1}^{N} P_i = P_D. \tag{5}$$

By using the slack variable method (Kuo, 2008) to guarantee that the equality constraint (5) is always satisfied. The power output of the slack unit is calculated as follows:

$$P_s = P_D - \sum_{\substack{i=1 \\ i \neq s}}^{N} P_i. \tag{6}$$

The fitness function for the proposed $MVMO^S$ will include the objective function (1) and penalty terms for the slack unit if inequality constraint (4) is violated. The fitness function is as follows:

$$F_T = \left(a_i + b_i P_i + c_i P_i^2 + \left|e_i \sin(f_i(P_{i,\min} - P_i))\right|\right) + K.\left[\max(0, P_s - P_{s,\max}) + \max(0, P_{s,\min} - P_s)\right]. \tag{7}$$

The penalty factor K for the slack unit is large enough and set to 1,000 for all systems.

3.3. Implemention of MVMO^S to ED

The flowchart of MVMO^S is depicted in Figure 2:

3.3.1. Initialization of algorithm

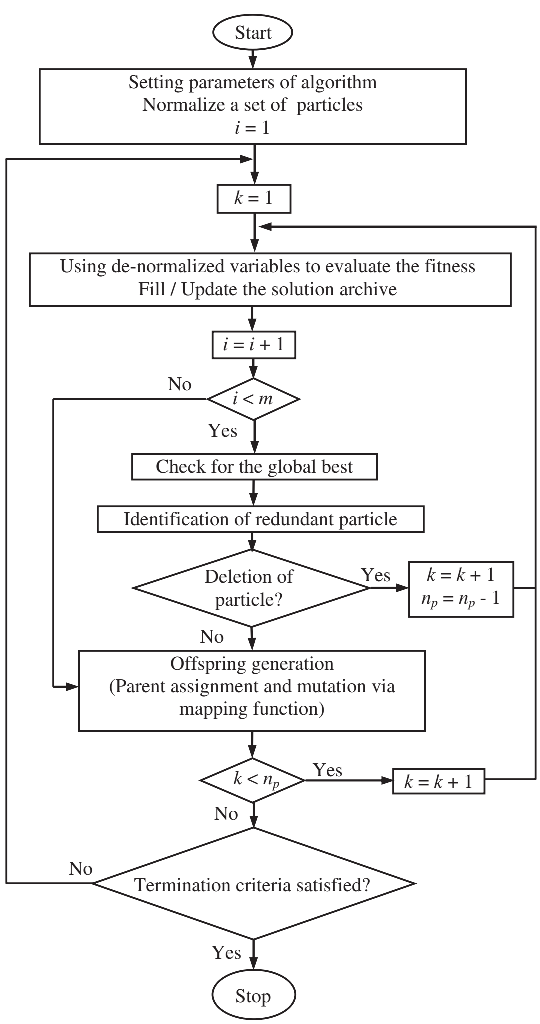
The parameters for MVMO^S have to be initialized including $iter_{max}$, n_var, n_par, $mode$, d_i, Δd_0^{ini}, Δd_0^{final} archive zize, $f^*_{s\_ini}$, $f^*_{s\_final}$, *n_randomly*, *n_randomly_min*, *indep.runs*(m), $D_{min.}$.

Since different parameters of the proposed method have effect on the performance of MVMO^S, it is important to determine an optimal set of parameters of the proposed methods for ED problem. For each selection, one parameter is varied from the low value to higher value while the other parameters are fixed. The obtained result after one run is compared with the previous one. Multiple runs are carried out to choose the suitable set of parameters.

3.3.2. Normalization and de-normalization of variables

The search process of the MVMO^S starts with a set of particles. Initial variables is normalized to the range [0, 1] as follows:

Figure 2. The flowchart of MVMO^S.



$$x\_normalized = rand(n\_par, n\_var). \tag{8}$$

However, the function evaluation is carried out always in the original scales of the problem space. The de-normalization of optimization variables is carried using (9):

$$P_i = P_{i,\,min} + \text{Scaling}\; x\_normalized(t, :). \tag{9}$$

where

$$\text{Scaling} = P_{i,\,max} - P_{i,\,min}.$$

After that, the power output for the slack generator is calculated using (6) to evaluate fitness function in (7), store f_{best} and x_{best} in archive.

MVMO^S utilizes swarm implementation to enhance the power of global searching of the classical MVMO by starting the search with a set of n_p particles, each having its own memory and represented by the corresponding archive and mapping function. At the beginning of the optimization process, each particle performs *m* steps independently to collect a set of reliable individual solutions. Then, the particles start to communicate and to exchange information.

It is worthless when particles are very close to each other since this would entail redundancy. To avoid closeness between particles, the normalized distance of each particle's local best solution $x^{lbest,\,i}$ to the global best x^{gbest} is calculated by Rueda and Erlich (2013). The *i*-th particle is discarded from the optimization process if the distance D_i is less than a certain user defined threshold D_{min}.

$$D_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(x_j^{gbest} - x_j^{lbest,\,i})^2}. \tag{10}$$

where *N* denotes the number of optimization variables.

3.3.3. Solution archive

The best *n* individuals are stored in the archive table which is described as Figure 3. The archive size (*n*) is taken to be a minimum of two. If archive size is greater than two, the table of best individuals is filled up progressively over iterations in a descending order of the fitness. When the table is filled with *n* members, an update is performed only if the fitness of the new population is better than those in the table.

Mean $\overline{x_i}$ and variance v_i are computed from the archive where the *n* best populations are stored as follows (Erlich et al., 2010):

$$\bar{x}_i = \frac{1}{n}\sum_{j=1}^{n} x_i(j). \tag{11}$$

$$v_i = \frac{1}{n}\sum_{j=1}^{n}(x_i(j) - \bar{x}_i)^2. \tag{12}$$

Figure 3. The archive is used to store *n*-best population.

#	Fitness	x_1	x_2 ...	x_i ...	x_D
1					
...					
n					
Mean $\overline{x_i}$	---				
Variance v_i	---				

where j goes from 1 to n (archive size). At the beginning $\overline{x_i}$ corresponds with the initialized value of x_i and the variance is set to $v_i = 1.0$.

3.3.4. Parent assignment

The individual with the best fitness f_{best} and its corresponding optimization values, x_{best}, are stored in memory as the parent of the population for that iteration. This parent is used for creation of offspring .

3.3.5. Offspring creation

Creation of an offspring, of N dimensions involves three common evolutionary computation algorithms' operations including selection, mutation and crossover.

3.3.5.1. Selection: Among N variables of the optimization problem, d variables are selected for mutation operation. There are four strategies which are described in Erlich et al. (2010) for selecting the variables.

3.3.5.2. Mutation: For each of the d selected dimension, mutation is used to assign a new value of that variable. Given a uniform random number $x_i^* \in [0, 1]$, the transformation of x_i^* to x_i via mapping function is calculated in (13) and depicted as Figure 4. The transformation mapping function, h, is calculated by the mean $\bar{x}$ and shape variables s_{i1} and s_{i2} as in (15) (Erlich et al., 2010):

$$x_i = h_x + (1 - h_1 + h_o).x_i^* - h_o. \tag{13}$$

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

$$h_x = h(x = x_i^*)h_o = h(x = 0)h_1 = h(x = 1). \tag{14}$$

$$h(\bar{x}_i, s_{i1}, s_{i2}, x) = \bar{x}_i.(1 - e^{-x.s_{i1}}) + (1 - \bar{x}_i).e^{-(1-x).s_{i2}}. \tag{15}$$

where

$$s_i = -\ln(v_i).f_s. \tag{16}$$

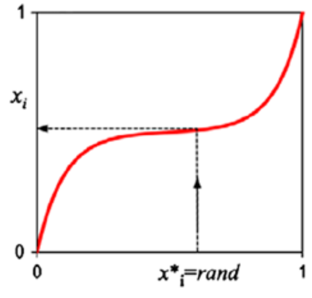
The scaling factor f_s in (16) is a MVMO parameter which can be used to change the shape of the function during iteration. In MVMO^S, this factor is extended for the need of exploring the search space at the beginning more globally, whereas at the end of the iterations, the focus should be on the exploitation. It is determined by (Rueda & Erlich, 2013):

$$f_s = f_s^*.\left(1 + rand()\right). \tag{17}$$

where

$$f_s^* = f_{s\_\text{ini}}^* + \left(\frac{i}{i_{\text{final}}}\right)^2 \left(f_{s\_\text{final}}^* - f_{s\_\text{ini}}^*\right). \tag{18}$$

Figure 4. Variable mapping.



In (18), f_s^* denotes the smallest value of f_s and the variable i represents the iteration number. $f_{s\_ini}^*$ and $f_{s\_final}^*$ are the initial and final values of,f_s^*, respectively. The recommended range of $f_{s\_ini}^*$ is from 0.9 to1.0, and range of $f_{s\_final}^*$ is from 1.0 to 3.0. When, $f_{s\_final}^* = f_{s\_ini}^* = 1$ which means that the option for controlling the f_s factor is not used (Rueda & Erlich, 2013).

```
s_i1 = s_i2 = s_i
if s_i > 0 then
Δd = (1 + Δd_o + 2 . Δd_o(rand() − 0.5)
  if s_i > d_i
    d_i = d_i. Δd
  else
    d_i = d_i/Δd
  end if
  if rand() ≥ 0.5 then
    s_i1 = s_i; s_i2 = d_i
    else
    s_i1 = d_i; s_i2 = s_i
    end if
end if
```

The shape variables s_{i1} and s_{i2} in (15) are determined using the following algorithm (Rueda & Erlich, 2013):

At the start of the algorithm, the initial values of d_i (typically between 1 and 5) are set for all variables. Sometimes, the variance can oscillate over a wide range. Using the factor d_i instead of s_i which is a function of variance a smoothing effect is achieved. At every iteration, if $s_i > d_i$, d_i will be multiplied by Δd leads to increased d_i. In case $s_i < d_i$, the current d_i is divided byΔd which is always greater than 1.0 resulting in reduced value of d_i. Therefore, d_i will always oscillate around the current shape factor s_i. Furthermore, Δd is randomly varied around the value $(1 + \Delta d_0)$ with the amplitude of Δd_0 adjusted in accordance to (19), where Δd_0 can be allowed to decrease from 0.4 to 0.01 (Rueda & Erlich, 2013).

$$\Delta d_0 = \Delta d_0^{\text{ini}} + \left(\frac{i}{i_{\text{final}}}\right)^2 \left(\Delta d_0^{\text{final}} - \Delta d_0^{\text{ini}}\right). \tag{19}$$

3.3.5.3. Crossover: For the remaining un-mutated dimensions, the genes of the parent, x_{best}, are inherited. In other words, the values of these un-mutated dimensions are clones of the parent. Here, crossover is by direct cloning of certain genes. In this way, the offspring is created by combining the vector x_{best}, and vector of m mutated dimensions.

3.3.6. Termination criteria

The algorithm of the proposed MVMO^S is terminated when the maximum number of iterations $iter_{\text{max}}$ is reached.

3.4. Overall procedure

The steps of procedure of MVMO^S for the ED problem are described as follows:

Step 1: Setting the parameters for MVMO^S including $iter_{max}$, n_var, n_par, $mode$, d_i, Δd_0^{ini}, $\Delta d_0^{\text{final}}$ archive zize, $f_{s\_ini}^*$, $f_{s\_final}^*$, n_randomly, n_randomly_min, indep.runs(m), D_{min}.

Set $i = 1$, i donotes the function evaluation.

Step 2: Normalize initial variables to the range [0, 1] (i.e. swarm of particles).

x_normalized = *rand*(*n_par*, *n_var*)

Step 3: Set $k = 1$, k denotes particle counters.

Step 4: De-normalized variables using (9), calculate power output for the slack generator using (6) to evaluate fitness function in (7), store f_{best} and x_{best} in archive.

Step 5: Increase $i = i + 1$. If $i < m$ (independent steps), go to step 6. Otherwise, go to step 7.

Step 6: Check the particles for the global best, collect a set of reliable individual solutions. The i-th particle is discarded from the optimization process if the distance D_i is less than a certain user defined threshold D_{min}. If the particle is delected, increase $k = k + 1$, $n_p = n_p - 1$ and go to step 4. Otherwise, go to step 7.

Step 7: Create offspring generation through three evolutionary operators: selection, mutation, and crossover.

Step 8: if $k < n_p$, increase $k = k + 1$ and go to step 4. Otherwise, go to step 9.

Step 9: Check termination criteria. If stopping criteria is satisfied, stop. Otherwise, go to step 3.

4. Numerical results

This part presents results of the implementation of proposed MVMO^S and origional MVMO in solving the ED problem with valve-point effects. The obtained results by the proposed MVMO^S are compared to those from the other optimization methods for three test cases including 3-unit system, 13-unit system, and 40-unit system. For each case, the algorithm of MVMO^S is run 50 independent trials on a core i5 3.4 GHz PC with 4 GB RAM. The implementation of the proposed MVMO^S is coded in the Matlab R2013a platform.

4.1. Case 1: 3-unit system

The data of 3-unit test system with valve-point effects is taken from Sinha et al. (2003). In this case, the power load demand is 850 MW, the transmission power loss is neglected. The obtained results by MVMO and MVMO^S for this case are presented in Table 1. Figure 5 depicts the convergence characteristic of the MVMO and MVMO^S for case 1.

The parameters for MVMO^S for this system are as follows: $iter_{max}$ = 10,000, *n_var* (generators) = 3, n_p = 20, *archive size* = 5, *mode* = 4, *indep.runs* (*m*) = 200, *n_randomly* = 2, *n_randomly_min* = 2, $f^*_{s\_ini} = 0.9$, $f^*_{s\_final} = 3$, $d_i = 1$, $\Delta d_0^{ini} = 0.3$, $\Delta d_0^{final} = 0.01$, $D_{min} = 0$.

The min, average, and max fuel cost and CPU time obtained by the proposed MVMO^S are compared to the results of the other methods in Table 2. The best optimal solution for this case is 8,234.0717 ($/h) (Park et al., 2005). All methods obtain the minimum cost with 8,234.0717 ($/h) in Table 2. However, the proposed MVMO^S achieves the best optimal solution with a high probability (the standard deviation is 0%). For computational time, it may not be directly comparable among the methods because these methods were run and coded on different computers and programming languages. However, a CPU time comparison is used to show the efficiency of the compared methods. The computational time of MVMO^S is faster than EP, EP-PSO, PSO, CEP, FEP, MFEP, and IEEP, and close to PSO-SQP. The PSO, EP-SQP, and PSO-SQP were run on a Pentium II 500 MHz PC. There is no computer processor reported for CEP, FEP, MFEP, and IEEP.

4.2. Case 2: 13-unit system

The data of 13-unit test system with valve-point effects are referred to Sinha et al. (2003). The power load demand is 1,800 and 2,500 MW, respectively. The transmission power loss is also neglected in this case. The obtained results by the MVMO and MVMO^S corresponding to the two load demand are

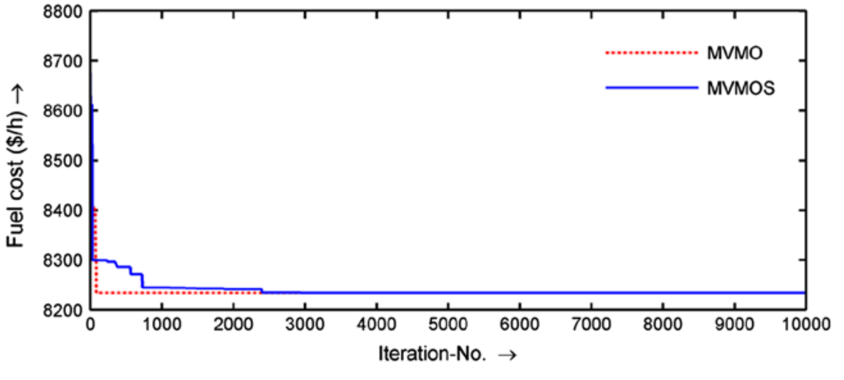
Table 1. Obtained results for 3-unit system by MVMO & MVMO^S

		Power outputs		
		MVMO	MVMO^S	
Unit	$P_{i,\,min}$ (MW)	P_i (MW)	P_i (MW)	$P_{i,\,max}$ (MW)
1	100	300.2669	300.2669	600
2	100	400.0000	400.0000	400
3	50	149.7331	149.7331	200
Total power (MW)		850.0000	850.0000	
Min cost ($/h)		8,234.0717	8,234.0717	
Average cost ($/h)		8,252.8227	8,234.0717	
Max cost ($/h)		8,390.8235	8,234.0717	
Standard deviation ($/h)		39.936	0.0000	
Average CPU time (s)		3.42	3.65	

Table 2. Comparisons of fuel cost for 3-unit system

Method	Min cost ($/h)	Mean cost ($/h)	Max cost ($/h)	CPU (s)
EP (Victoire & Jeyakumar, 2004)	8,234.07	8,234.16	–	6.78
EP-PSO (Victoire & Jeyakumar, 2004)	8,234.07	8,234.09	–	5.12
PSO (Victoire & Jeyakumar, 2004)	8,234.07	8,234.72	–	4.37
PSO-SQP (Victoire & Jeyakumar, 2004)	8,234.07	8,234.07	–	3.37
CEP (Sinha et al., 2003)	8,234.07	8,235.97	8,241.83	20.46
FEP (Sinha et al., 2003)	8,234.07	8,234.24	8,241.78	4.54
MFEP (Sinha et al., 2003)	8,234.08	8,234.71	8,241.80	8.00
IFEP (Sinha et al., 2003)	8,234.07	8,234.16	8,234.54	6.78
MPSO (Park et al., 2005)	8,234.07	–	–	–
MVMO^S	8,234.07	8,234.07	8,234.07	3.65

Figure 5. Convergence property of MVMO and MVMO^S for case 1 (3 units).



shown in Table 3. Figures 6 and 7 show the convergence characteristic of the MVMO and MVMO^S for the case of load demands 1,800 MW and the case of load demands 2,520 MW, respectively.

The parameters for MVMO^S are as follows for all the cases of load demands 1,800 and 2,520 MW: $iter_{max}$ = 70,000, *n_var* (generators) = 13, n_p = 20, *archive size* = 5, *mode* = 4, *indep.runs* (*m*) = 2,000, *n_randomly* = 5, *n_randomly_min* = 4, $f^*_{s\_ini} = 0.95$, $f^*_{s\_final} = 3$, $d_i = 1$, $\Delta d_0^{ini} = 0.4$, $\Delta d_0^{final} = 0.02$, $D_{min} = 0$.

In Tables 4 and 5, the fuel cost and CPU time of proposed MVMO^S are compared to those of other optimization methods for two load demands 1,800 and 2,520 MW. For the case of load demands 1,800 MW, the minimum fuel cost obtained by MVMO^S is less than CEP, FEP, MFEP, IEEP, PSO, EP-SQP, PSO-SQP, HDE, and CGA-MU, and close to UHGA, Self-tuning HDE and GA-PS-SQP. It is noted that the mean cost obtained by MVMO^S is less than is less than that of the others, except the UHGA. The computational time of MVMO^S is faster than CEP, FEP, MFEP, IEEP, PSO, and EP-SQP, slower than UHGA, CGA-MU, HDE, self-tuning HDE, and GA-PS-SQP, and close to PSO-SQP. For the case of load demands 2,520 MW, the minimum total cost by MVMO^S is less than that from the other methods in Table 5. The computational time of MVMO^S is slower than ESO . The PSO, EP-SQP, and PSO-SQP were executed on a Pentium II 500 MHz PC. The HDE and self-tuning HDE were run on a Pentium 1.5 GHz with 768 MB of RAM. The computational time for ESO, UHGA, CGA-MU, GA-PS-SQP were from a Pentium IV PC, Pentium IV 2.99 GHz PC, Pentium III—700 PC, and Pentium III—1 GHz—256 MB of RAM, respectively. There is no computer processor reported for CEP, FEP, MFEP, and IEEP and no computational time for the other methods.

4.3. Case 3: 40-unit system

The data of the test system including 40 thermal generating units with VPE are from Sinha et al. (2003). The system load demand for this case is 10,500 MW neglecting transmission power loss. The obtained solutions by MVMO and MVMO^S for this case are given in Table 6. The convergence characteristic of the MVMO and MVMO^S are depicted in Figure 8 for case 3.

The parameters for MVMO^S for this system are as follows: $iter_{max}$ = 150,000, *n_var* (generators) = 40, n_p = 5, *archive size* = 5, *mode* = 4, *indep.runs* (*m*) = 2,000, *n_randomly* = 20, *n_randomly_min* = 10, $f^*_{s\_ini} = 0.9$, $f^*_{s\_final} = 3$, $d_i = 5$, $\Delta d_0^{ini} = 0.4$, $\Delta d_0^{final} = 0.02$, $D_{min} = 0$.

Table 3. Obtained results for 13-unit system for load demands 1,800 and 2,520 MW by MVMO and MVMO^S

Unit	$P_{i,min}$ (MW)	Power outputs				$P_{i,max}$ (MW)
		MVMO	MVMO	MVMO^S	MVMO^S	
		P_i (MW)	P_i (MW)	P_i (MW)	P_i (MW)	
1	0	538.5587	628.3185	628.3185	628.3452	680
2	0	149.7970	299.1741	148.2939	299.1906	360
3	0	224.5880	299.1858	224.2433	299.1924	360
4	60	159.7332	159.7325	60.0000	159.7318	180
5	60	109.8667	159.7314	109.7217	159.7299	180
6	60	109.9145	159.7268	109.8501	159.7328	180
7	60	109.8917	159.7329	109.8602	159.7314	180
8	60	109.8675	159.7317	109.8509	159.7320	180
9	60	109.5884	159.7271	109.8613	159.7077	180
10	40	60.0001	73.7967	40.0000	77.3682	120
11	40	77.7801	76.6576	40.0000	77.3731	120
12	55	55.0004	92.1942	55.0000	92.3625	120
13	55	55.0000	92.2908	55.0000	87.8023	120
Total power (MW)	1,800.0000	2,520.0000	1,800.0000	2,520.0000		
Min cost ($/h)	17,985.4638	24,170.8763	17,964.1226	24,170.0137		
Average cost ($/h)	18,126.9549	24,313.9828	18,011.0370	24,193.4933		
Max cost ($/h)	18,257.2713	24,473.9750	18,070.7615	24,226.8256		
Standard deviation ($/h)	54.7923	70.8093	26.7448	23.6363		
Average CPU time (s)	33.08	33.86	34.02	34.32		

Table 4. Comparisons of fuel cost for 13-unit system with VPE, P_D = 1,800 MW

Method	Min cost ($/h)	Mean cost ($/h)	Max cost ($/h)	CPU (s)
CEP (Sinha et al., 2003)	18,048.21	18,190.32	18,404.04	294.96
FEP (Sinha et al., 2003)	18,018.00	18,200.79	18,453.82	168.11
MFEP (Sinha et al., 2003)	18,028.09	18,192.00	18,416.89	317.12
IFEP (Sinha et al., 2003)	17,994.07	18,127.06	18,267.42	157.43
PSO (Victoire & Jeyakumar, 2004)	18,030.72	18,205.78	–	77.37
EP-SQP (Victoire & Jeyakumar, 2004)	17,991.03	18,106.93	–	121.93
PSO-SQP (Victoire & Jeyakumar, 2004)	17,969.93	18,029.99	–	33.97
UHGA (He et al., 2008)	17,964.81	17,992.92	–	15.33
QPSO (Meng et al., 2010)	17,969.01	18,075.11	–	-
HDE (Wang et al., 2007)	17,975.73	18,134.80	–	1.65
CGA-MU (Chiang, 2005)	17,975.34	–	–	21.91
ST HDE (Wang et al., 2007)	17,963.89	18,046.38	–	1.41
GA-PS-SQP (Alsumait et al., 2010)	17,964.25	18,199		11.06
MVMO^S	17,964.12	18,011.04	18,070.76	33.86

Table 5. Comparisons of fuel cost for 13-unit system with VPE, P_D = 2,520 MW

Method	Min cost ($/h)	Mean cost ($/h)	Max cost ($/h)	CPU (s)
GA (Victoire & Jeyakumar, 2004)	24,398.23	–	–	–
SA (Victoire & Jeyakumar, 2004)	24,970.91	–	–	–
GA-SA (Victoire & Jeyakumar, 2004)	24,275.71	–	–	–
EP-SQP (Victoire & Jeyakumar, 2004)	24,266.44	–	–	–
PSO-SQP (Victoire & Jeyakumar, 2004)	24,261.05	–	–	–
UHGA (He et al., 2008)	24,172.25	–	–	–
ESO (Pereira-Neto et al., 2005)	24,177.78	–	–	1.0
SA-PSO (Kuo, 2008)	24,171.40	–	–	–
MVMO^S	24,170.01	24,193.4933	24,226.8256	34.32

Table 7 shows the comparison of the fuel cost and CPU time of the MVMO^S and the previously reported methods. As seen in Table 7, the best total cost obtained by the MVMO^S is less than that from the others. The computational time of the MVMO^S is faster than IFEP, ICA-PSO, and UHGA, and slower than the other methods. The computational time from ABC, ACO, DE, IFEP, ESO, NPSO-LRS, ICA-PSO, DEC-SQP, UHGA, self-tuning HDE, GA-PS-SQP, and DE-BBO were from Pentium IV 2.3 GHz with 512-MB of RAM PC, Pentium IV

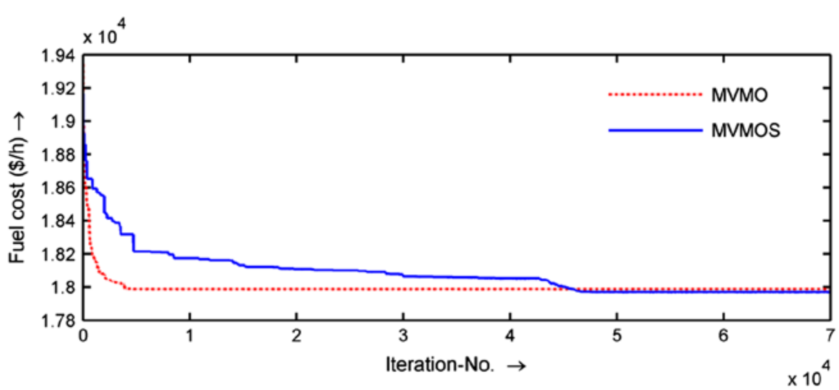


Figure 6. Convergence property of MVMO and MVMO^S for case 2 (13 units with P_D = 1,800 MW).

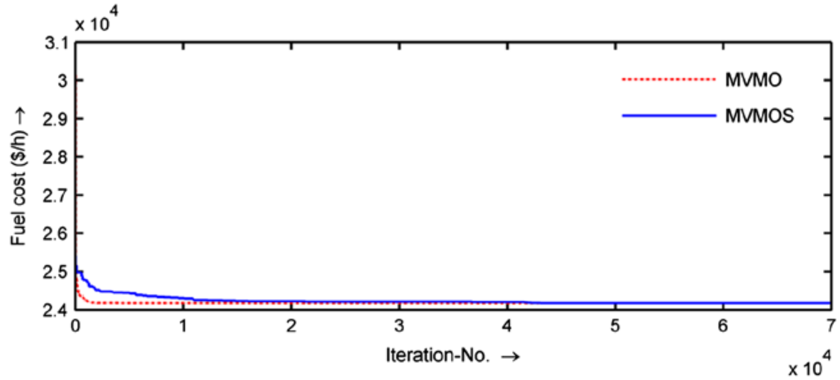


Figure 7. Convergence property of MVMO and MVMO^S for case 2 (13 units with P_D = 2,520 MW).

2.6 GHz with 1 GB of RAM PC, Intel 1.67 GHz with 1 GB of RAM PC, Pentium-II 350 MHz with 128 MB of RAM PC, Pentium IV 1.5 GHz with 128 MB of RAM PC, Pentium IV PC, Pentium IV 1.4-GHz PC, 1.1 AMD Athlon GHz with 112 MB of RAM, Pentium IV 2.99 GHz PC, Pentium 1.5 GHz with 768 MB of RAM, Pentium III 1 GHz with 256 MB of RAM, and Pentium IV 2.3-GHz PC with 512-MB RAM, respectively. There is no computational time or computer processor reported for the other methods.

5. Discussion

5.1. Advantages of MVMO^S

The advantages of MVMO^S are robustness, global solution with high probability, and easy implemention to ED problem. In this study, the MVMO and the MVMO^S are run 50 independent trials. The mean cost, max cost, average cost, and standard deviation obtained by the MVMO and MVMO^S to evaluate the

Table 6. Obtained results for 40-unit system by MVMO and MVMO^S

Unit	$P_{i,min}$ (MW)	Power outputs		$P_{i,max}$ (MW)
		MVMO	MVMO^S	
		P_i (MW)	P_i (MW)	
1	36	110.8011	110.8441	114
2	36	110.8067	110.9734	114
3	60	97.3999	97.4030	120
4	80	179.7331	179.7337	190
5	47	168.7998	88.3994	97
6	68	89.6332	168.8003	140
7	110	140.0000	140.0000	300
8	135	259.5997	259.6099	300
9	135	284.5997	284.6087	300
10	130	284.5997	284.6093	300
11	94	130.0000	130.0000	375
12	94	168.7998	168.8000	375
13	125	214.7598	214.7598	500
14	125	304.5196	394.2795	500
15	125	394.2794	304.5211	500
16	125	394.2794	394.2807	500
17	220	489.2794	489.2802	500
18	220	489.2794	489.2811	500

(Continued)

Table 6. *(Continued)*

		Power outputs		
19	242	511.2794	511.2829	550
20	242	511.2794	511.2806	550
21	254	523.2794	523.2794	550
22	254	523.2794	523.2815	550
23	254	523.2794	523.2802	550
24	254	523.2794	523.2839	550
25	254	523.2794	523.2811	550
26	254	523.2794	523.2827	550
27	10	10.0000	10.0000	150
28	10	10.0000	10.0000	150
29	10	10.0000	10.0000	150
30	47	90.9166	91.8542	97
31	60	190.0000	190.0000	190
32	60	190.0000	190.0000	190
33	60	190.0000	190.0000	190
34	90	164.7999	164.8036	200
35	90	164.7998	164.8088	200
36	90	164.7999	164.8171	200
37	25	110.0000	110.0000	110
38	25	110.0000	110.0000	110
39	25	110.0000	110.0000	110
40	252	511.2794	511.2796	550
Total power (MW)		10,500.0000	10,500.0000	
Min cost ($/h)		121,415.4881	121,415.2346	
Average cost ($/h)		121,675.3501	121,652.7238	
Max cost ($/h)		122,006.5808	121,913.4278	
Standard deviation ($/h)		128.0542	115.3685	
Average CPU time (s)		104.57	107.98	

robustness characteristic of the proposed method for ED problems. As observed from Tables 1, 3, and 6, the power output obtained by MVMO and MVMO^S are always between the minimum and maximum generator capacity limits and the total power output of generating units equals to the power load demand. It is indicated that the equality and inequality constraints always satisfy. The proposed MVMO^S provides not only better solution but also more robust than the MVMO and the difference between the maximum and minimum costs from the proposed MVMO^S is small. Table 8 shows the ratio between the standard deviation and the minimum cost obtained by MVMO^S for all systems. The ratio between the standard deviation and the minimum cost is less than 0.149%. It clearly shows that the performance the proposed MVMO^S is robust. In addition, the comparison of the total cost obtained by MVMO^S and many other methods from Tables 2, 4, 5, and 7 shows that the MVMO^S can obtain better total fuel costs and more robust than most of other reported methods. Consequently, the MVMO^S can obtain near global solution with high probability.

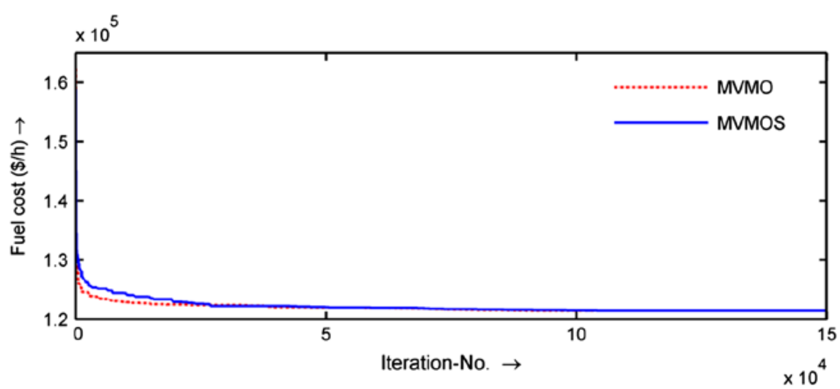
5.2. Disadvantages of MVMO^S

The only disadvantage of MVMO^S is computational time. The computation time of the MVMO^S is relatively high. Similar to the original MVMO, the number of iterations in MVMO^S is equivalent to the number of offspring fitness evaluations which is usually time consuming in practical applications. The computational

Table 7. Comparison of fuel cost and CPU times for 40 unit system with VPE

Method	Min cost ($/h)	Mean cost ($/h)	Max cost ($/h)	CPU (s)
ABC (Hemamalini & Simon, 2010)	121,441.03	121,995.82	122.123.77	32.45
ACO (Pothiya et al., 2010)	121,532.41	121,606.45	121,679.64	52.45
DE (Noman & Iba, 2008)	121,416.29	121,422.72	121,431.47	72.94
IFEP (Sinha et al., 2003)	122,624.35	123,382.00	125,740.63	1,167.35
MPSO (Park et al., 2005)	122,252.27	–	–	–
APSO (Selvakumar & Thanushkodi, 2008)	121,663.52	122,153.67	122,912.40	5.05
ESO (Pereira-Neto et al., 2005)	122,122.16	122,542.07	123,143.07	0.261
SOH_PSO (Chaturvedi et al., 2008)	121,501.14	121,853.57	122,446.3	-
SA-PSO (Kuo, 2008)	121,430.00	121,525	121,645	23.89
PSO-RDL (Chen et al., 2007)	121,468.82	–	–	–
NPSO-LRS (Selvakumar & Thanushkodi, 2007)	121,664.43	122,209.32	122,981.59	20.74
ICA-PSO (Vlachogiannis & Lee, 2009)	121,422.17	121,428.14	121,453.56	139.92
QPSO (Meng et al., 2010)	121,448.21	122,225.047	–	–
DEC-SQP (Coelho & Mariani, 2006)	121,741.98	12,295.1278	–	14.26
UHGA (He et al., 2008)	121,424.48	121,602.81	–	333.68
ST HDE (Wang et al., 2007)	121,698.51	122,304.30	–	6.07
GA-PS-SQP (Alsumait et al., 2010)	121,458.14	122.039	–	46.98
DE-BBO (Bhattacharya & Chattopadhyay, 2010)	121,420.89	121,420.90	121,420.90	1.23
BSA (Modiri-Delshad & Rahim, 2014)	121,415.61	121,474.88	121,524.96	13.12
$MVMO^S$	121,415.23	121,652.72	121,913.43	107.98

Figure 8. Convergence property of MVMO and $MVMO^S$ for case 3 (40 units).



time of the $MVMO^S$ is slower than the classical MVMO. This is because the $MVMO^S$ starts the search with a set of particles while the MVMO starts the search with single particle.

6. Conclusion

This paper has presented an application of new method for solving the ED problem. The proposed $MVMO^S$ has been successfully solved the ED problem with valve-point effects. Three test cases have been carried out to demonstrate its effectiveness and efficiency. The comparisons of numerical results have shown that the proposed $MVMO^S$ has better performance than other optimization techniques exist in the literature. It is also confirmed that the $MVMO^S$ outperformed the classical MVMO in global search for the

Table 8. The ratio between the standard deviation and the minimum cost obtained by MVMO^S for all systems

System	Case 1: 3units	Case 2: 13 units		Case 3: 40 units
P_D (MW)	850	1,800	2,520	2,500
Ratio (%)	0.0	0.149	0.098	0.095

nonconvex problem. Therefore, the proposed MVMO^S could be favorable for solving the ED problem with valve-point effects and other nonconvex ED problems as well. In the future, the MVMO^S will be applied for solving dynamic ED, hydrothermal ED with cascaded hydro plants and emission constrained ED.

Nomenclature

N	total number of generating units, optimization variables
F	total operation cost
a_i, b_i, c_i	fuel cost coefficients of generator *i*
e_i, f_i	fuel cost coefficients of unit *i* reflecting valve-point effects
B_{ij}, B_{0i}, B_{00}	B-matrix coefficients for transmission power loss
P_D	total system load demand
P_i	power output of generator *i*
$P_{i,\max}$	maximum power output of generator *i*
$P_{i,\min}$	minimum power output of generator *i*
P_s	power output of slack unit
$P_{s,\max}$	maximum power output of slack unit
$P_{s,\min}$	minimum power output of slack unit
K	the penalty factor for the slack unit
P_L	total transmission loss
$iter_{max}$	maximum number of iterations
n_var	number of variable (generators)
n_par	number of particles
mode	variable selection strategy for offspring creation
archive zize	*n*-best individuals to be stored in the table
d_i	initial smoothing factor
Δd_0^{ini}	initial smoothing factor increment
$\Delta d_0^{\text{final}}$	final smoothing factor increment
rand ()	a random number in the range [0, 1]
$f_{s\_\text{ini}}^{*}$	initial shape scaling factor
$f_{s\_\text{final}}^{*}$	final shape scaling factor
$D_{\min}$	minimum distance threshold to the global best solution
n_randomly	Initial number of variables selected for mutation*i*
ndep.runs m	steps independently to collect a set of reliable individual solutions

Acknowledgment
The authors would like to sincerely thank Universiti Teknologi PETRONAS for providing the research laboratory facilities under Graduate Assistance Scheme. This research work is sponsored by the Centre of Graduate Studies with the support of the Department of Fundamental & Applied Sciences, Universiti Teknologi Petronas.

Funding
This work was supported by Graduate Assistance Scheme of Universiti Teknologi PETRONAS.

Author details
T.H. Khoa[1]
E-mail: trhkhoa89@gmail.com
ORCID ID: http://orcid.org/0000-0002-1237-0071
P.M. Vasant[1]
E-mail: pvasant@gmail.com
M.S. Balbir Singh[1]
E-mail: balbir@petronas.com.my
V.N. Dieu[2]
E-mail: vndieu@gmail.com

[1] Department of Fundamental and Applied Sciences, Universiti Teknologi PETRONAS, 31750 Tronoh, Perak, Malaysia.
[2] Department of Power Systems, HCMC University of Technology, Ho Chi Minh City, Vietnam.

Citation information
Cite this article as: Solving economic dispatch problem with valve - Point effects by using swarm based mean–variance mapping optimization (MVMO^S^), T.H. Khoa, P.M. Vasant, M.S. Balbir Singh & V.N. Dieu, *Cogent Engineering* (2015), 2: 1076983.

References

Alsumait, J., Sykulski, J., & Al-Othman, A. (2010). A hybrid GA–PS–SQP method to solve power system valve-point economic dispatch problems. *Applied Energy, 87*, 1773–1781. http://dx.doi.org/10.1016/j.apenergy.2009.10.007

Attaviriyanupap, P., Kita, H., Tanaka, E., & Hasegawa, J. (2002). A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function. *IEEE Transactions on Power Systems, 17*, 411–416. http://dx.doi.org/10.1109/TPWRS.2002.1007911

Bhattacharya, A., & Chattopadhyay, P. K. (2010). Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. *IEEE Transactions on Power Systems, 25*, 1955–1964. http://dx.doi.org/10.1109/TPWRS.2010.2043270

Chaturvedi, K. T., Pandit, M., & Srivastava, L. (2008). Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch. *IEEE Transactions on Power Systems, 23*, 1079–1087. http://dx.doi.org/10.1109/TPWRS.2008.926455

Chen, Y.-P., Peng, W.-C., & Jian, M.-C. (2007). Particle swarm optimization with recombination and dynamic linkage discovery. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 37*, 1460–1470. http://dx.doi.org/10.1109/TSMCB.2007.904019

Chiang, C.-L. (2005). Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. *IEEE Transactions on Power Systems, 20*, 1690–1699. http://dx.doi.org/10.1109/TPWRS.2005.857924

Coelho, L. S., & Mariani, V. C. (2006). Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. *IEEE Transactions on Power Systems, 21*, 989–996.

Dieu, V., Schegner, P., & Ongsakul, W. (2013). Pseudo-gradient based particle swarm optimization method for nonconvex economic dispatch. In I. Zelinka, P. Vasant, & N. Barsoum (Eds.), *Power, control and optimization* (Vol. 239, pp. 1–27). Springer International. doi:10.1007/978-3-319-00206-4_1

Duman, S., Yorukeren, N., & Altas, I. H. (2015). A novel modified hybrid PSOGSA based on fuzzy logic for non-convex economic dispatch problem with valve-point effect. *International Journal of Electrical Power & Energy Systems, 64*, 121–135.

Erlich, I., Venayagamoorthy, G. K., & Worawat, N. (2010). A mean-variance optimization algorithm. In *Evolutionary Computation (CEC), IEEE Congress*. Barcelona. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5586027&tag=1. doi:10.1109/CEC.2010.5586027

Fan, J.-Y., & Zhang, L. (1998). Real-time economic dispatch with line flow and emission constraints using quadratic programming. *IEEE Transactions on Power Systems*, *13*, 320–325. http://dx.doi.org/10.1109/59.667345

He, D.-K., Wang, F.-L., & Mao, Z.-Z. (2008). Hybrid genetic algorithm for economic dispatch with valve-point effect. *Electric Power Systems Research, 78*, 626–633. http://dx.doi.org/10.1016/j.epsr.2007.05.008

Hemamalini, S., & Simon, S. P. (2010). Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions. *Electric Power Components and Systems, 38*, 786–803. http://dx.doi.org/10.1080/15325000903489710

Kuo, C.-C. (2008). A novel coding scheme for practical economic dispatch by modified particle swarm approach. *IEEE Transactions on Power Systems, 23*, 1825–1835.

Le Dinh, L., Vo Ngoc, D., & Vasant, P. (2013). Artificial bee colony algorithm for solving optimal power flow problem. *The Scientific World Journal 2013*, Article ID: 159040, 9 p. doi:10.1155/2013/159040

Lin, J., Chen, C.-L., Tsai, S.-F., & Yuan, C. (2015). New intelligent particle swarm optimization algorithm for solving economic dispatch with valve-point effects. *Journal of Marine Science and Technology, 23*, 44–53.

Meng, K., Wang, H. G., Dong, Z. Y., & Wong, K. P. (2010). Quantum-inspired particle swarm optimization for valve-point economic load dispatch. *IEEE Transactions on Power Systems, 25*, 215–222. http://dx.doi.org/10.1109/TPWRS.2009.2030359

Modiri-Delshad, M., & Rahim, N. A. (2014). Solving non-convex economic dispatch problem via backtracking search algorithm. *Energy, 77*, 372–381.

Niu, Q., Zhang, H., Wang, X., Li, K., & Irwin, G. W. (2014). A hybrid harmony search with arithmetic crossover operation for economic dispatch. *International Journal of Electrical Power & Energy Systems, 62*, 237–257.

Noman, N., & Iba, H. (2008). Differential evolution for economic load dispatch problems. *Electric Power Systems Research, 78*, 1322–1331. http://dx.doi.org/10.1016/j.epsr.2007.11.007

Park, J.-B., Lee, K. S., Shin, J.-R., & Lee, K. Y. (2005). A particle swarm optimization for economic dispatch with nonsmooth cost functions. *IEEE Transactions on Power Systems, 20*, 34–42. http://dx.doi.org/10.1109/TPWRS.2004.831275

Pereira-Neto, A., Unsihuay, C., & Saavedra, O. (2005). Efficient evolutionary strategy optimisation procedure to solve the nonconvex economic dispatch problem with generator constraints. *IEE Proceedings-Generation, Transmission and Distribution, 152*, 653–660. http://dx.doi.org/10.1049/ip-gtd:20045287

Pothiya, S., Ngamroo, I., & Kongprawechnon, W. (2010). Ant colony optimisation for economic dispatch problem with non-smooth cost functions. *International Journal of Electrical Power & Energy Systems, 32*, 478–487.

Rueda, J. L., & Erlich, I. (2013). *Evaluation of the mean-variance mapping optimization for solving multimodal problems*. In *Swarm Intelligence (SIS), 2013 IEEE Symposium*. Singapore. Retrieved from http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6615153. doi:10.1109/SIS.2013.6615153

Secui, D. C. (2015). A new modified artificial bee colony algorithm for the economic dispatch problem. *Energy Conversion and Management, 89*, 43–62. http://dx.doi.org/10.1016/j.enconman.2014.09.034

Selvakumar, A. I., & Thanushkodi, K. (2007). A new particle swarm optimization solution to nonconvex economic dispatch problems. *IEEE Transactions on Power Systems, 22*, 42–51. http://dx.doi.org/10.1109/TPWRS.2006.889132

Selvakumar, A. I., & Thanushkodi, K. (2008). Anti-predatory particle swarm optimization: Solution to nonconvex economic dispatch problems. *Electric Power Systems Research, 78*, 2–10. http://dx.doi.org/10.1016/j.epsr.2006.12.001

Shahinzadeh, H., Nasr-Azadani, S. M., & Jannesari, N. (2014). Applications of particle swarm optimization algorithm to solving the economic load dispatch of units in power systems with valve-point effects. *International Journal of Electrical and Computer Engineering (IJECE), 4*, 858–867.

Sinha, N., Chakrabarti, R., & Chattopadhyay, P. (2003). Evolutionary programming techniques for economic load dispatch. *IEEE Transactions on Evolutionary Computation, 7*, 83–94. http://dx.doi.org/10.1109/TEVC.2002.806788

Vasant, P., Ganesan, T., & Elamvazuthi, I. (2012). An improved PSO approach for solving non-convex optimization problems. In *2011 9th International Conference on ICT and Knowledge Engineering (ICT & Knowledge Engineering)*. Bangkok: IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6152418. doi:10.1109/ICTKE.2012.6152418

Victoire, T. A. A., & Jeyakumar, A. E. (2004). Hybrid PSO–SQP for economic dispatch with valve-point effect. *Electric Power Systems Research, 71*, 51–59. http://dx.doi.org/10.1016/j.epsr.2003.12.017

Vlachogiannis, J. G., & Lee, K. Y. (2009). Economic load dispatch—A comparative study on heuristic optimization techniques with an improved coordinated aggregation-based PSO. *IEEE Transactions on Power Systems, 24*, 991–1001. http://dx.doi.org/10.1109/TPWRS.2009.2016524

Wang, S. K., Chiou, J. P., & Liu, C. W. (2007). Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm. *IET Generation, Transmission & Distribution, 1*, 793–803.

Wollenberg, B., & Wood, A. (1996). *Power generation, operation and control* (pp. 264–327). New York, NY: Wiley.