Global Journal of Technology & Optimization

Khoa et al., Global J Technol Optim 2015, 6:3
DOI: 10.4172/2229-8711.1000184

Research Article Open Access

Solving Economic Dispatch By Using Swarm Based Mean-Variance Mapping Optimization (MVMO^S)

Truong Khoa H[1]*, Vasant P[1], Balbir Singh[1] and Vo Dieu N[2]

[1]*Department of Fundamental and Applied Sciences, Universiti Teknologi Petronas, Malaysia*
[2]*Department of Power Systems, HCMC University of Technology, Vietnam*

Abstract

This paper proposes a novel optimization which named as Swarm based Mean-variance mapping optimization (MVMOS) for solving the economic dispatch. The proposed optimization algorithm is the extension of the original single particle mean-variance mapping optimization (MVMO). The novel feature is the special mapping function applied for the mutation base on the mean and variance of n-best population.The MVMOS outperforms the classical MVMO in global search ability due to the improvement of the mapping. The proposed MVMOS is investigated on four test power systems, including 3-, 13- , 20- thermal generating units and large-scale system 140 units with quadratic cost function and the obtained results are compared with many other known methods in the literature. Test results show that the proposed method can efficiently implement for solving economic dispatch.

Keywords: Economic dispatch; Quadraticfuel cost function; MVMO; MVMO^S

Nomenclature

N	Total number of generator
F_T	Total operationcost
F_i	Fuel cost function of generator i
a_i, b_i, c_i	Fuel cost coefficients of generator i
B_{ij},B_{0i},B_{00}	B-matrix coefficients fortransmissionpower 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_L	Total transmission loss
$iter_{max}$	Maximum number of iterations
n_var	Numberof 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
$f_{s\_\text{ini}}^*$	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
indep.runs	m steps independently to collect a set of reliable individual solutions

Introduction

The Economic Dispatch (ED) is an essential optimization task in the power generation system and its objective is to determine the economical real power output of the thermal generating units to supply required power load demand at the minimum fuel cost while satisfying all units and system constrains [1,2]. Since the concept of economic dispatch (ED) started in the 1950's, there are a lot of various methods have been employed for solving ED problems, but in short there are three main categories: Methods based on mathematical programming (Classical calculus-based techniques), methods based on artificial intelligence and hybrid methods.

For mathematical convenience, the objective cost function of ED problem is the quadratic function approximations [3], was solved by methods based on mathematical programming such as lambda iteration method, Newton's method, gradient search, dynamic programming [3], linear programming [4], non-linear programming [5] and quadratic programming [6]. These methods are conventional techniques that were early employed. Over the past years, more advanced methods based on artificial intelligence have been developed and implemented outstandingly to ED problem such as Hopfield Neural Network (HNN) [7,8], Evolutionary Programming (EP) [9], Differential Evolution (DE) [10], Genetic Algorithm (GA) [11], Ant Colony Optimization (ACO) [12], Particle Swarm Optimization (PSO) [13,14] Bacterial Foraging (BF) [15], and Artificial Bee Colony (ABC) algorithm [16]. These methods do not always guarantee to find the global optimal solution in finite computational time but their ability often find near global optimal solution for optimization problems. Besides the single mentioned methods, hybrid methods have been also developed for solving the

***Corresponding author:** Dr. Truong H Khoa, Department of Fundamental and Applied Sciences, Universiti Teknologi Petronas, Malaysia, Tel: +60 5-368 8000; E-mail: trhkhoa89@gmail.com

Received April 11, 2015; **Accepted** June 29, 2015; **Published** July 06, 2015

Citation: Khoa TH, Vasant P, Singh B, Dieu VN (2015) Solving Economic Dispatch By Using Swarm Based Mean-Variance Mapping Optimization (MVMO^S). Global J Technol Optim 6: 184. doi:10.4172/2229-8711.1000184

ED problems such as hybridization of evolutionary programming with Sequential Quadratic Programming (EP-SQP) [17], combining of chaotic differential evolution and quadratic programming (DEC-SQP) [18] hybrid technique integrating the uniform design with the genetic algorithm (UHGA) [19], self-tuning hybrid differential evolution (self-tuning HDE) [20], and fuzzy adaptive particle swarm optimization algorithm with Nelder–Mead simplex search (FAPSO-NM) [21]. These hybrid methods become powerful search methods for obtaining higher solution quality due to using the advantages of each element method to improve their search ability for the complex problems. Nevertheless, they may be slower and more complicated than the element methods because of combination of several operations.

The above artificial intelligence methods are population based meta-heuristic which can deal with multiform optimization problems [22]. Recently, Prof. István Erlich has been conceived and developed a novel optimization technique which is named Mean-variance mapping optimization (MVMO) [23]. This algorithm is so-called "population-based stochastic optimization techniques". MVMO has the capability to find the optimum solution quickly with minimum risk of premature convergence.

The extensions of MVMO, which named Swarm based Mean-variance mapping optimization (MVMOS) [24], has been developed to become more effective. In this paper, MVMOS is proposed for solving the economic dispatch problem with quadratic cost function.

Section II presents the formulation of the ED. The review of MVMO, extension of MVMO-MVMOS and implementation of the proposed MVMOS to ED problem are addressed in Section III. The numerical results are showcased in Section IV. The discussion is followed in Section V. After all, the conclusion are given.

Problem Formulation

The power system consists of N thermal generating units. Each unit has a fuel cost function, shown as F_i, togenerates a power out P_i. The total fuel costof the system,F_T,is sum of fuel cost of each unit.

$$F_T = F_1 + F_2 + F_3 + ... + F_N = \sum_{i=1}^{N} F_i(P_i) \quad (1)$$

The optimization problem of the ED is to minimize the total fuel cost F_T, which be written as:

$$\text{Minimize } F_T = \sum_{i=1}^{N} F_i(P_i) \quad i = 1,2,3,...,N \quad (2)$$

Generally, the fuel cost curve of a thermal generating unit is presented as quadratic function as:

$$F_i(P_i) = a_i + b_i P_i + c_i P_i^2 \quad (3)$$

The constraints of the ED problem must be satisfied during the optimization process are peresented as follows:

Real power balance equation

The total active power output of generating units must be equal to total active power load demand plus power loss:

$$\sum_{i=1}^{N} P_i = P_D + P_L \quad (4)$$

The power loss $P_L$is calculated by the below formulation [3]:

$$P_L = \sum_{i=1}^{N}\sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{0i} P_i + B_{00} \quad (5)$$

Generator capacity limits

The active power output of generating units must be within the allowed limits:

$$P_{i,\min} \le P_i \le P_{i,\max} \quad (6)$$

MVMO^S for Eonomic Dispatch

Review of MVMO

Mean-variance mapping optimization (MVMO) is a novel optimization algorithm falls into the category of the so-called "population-based stochastic optimization technique". The similarities between MVMO and the other known stochastic algorithms are three evolutionary operators: selection, mutation and crossover. However, the major differences between MVMO and other existing techniques are as follows:

- The key feature of MVMO is a special mapping function which applied for mutating the offspring based on mean-variance of the solutions stored in the archive.

The mean $\overline{x_i}$ and variance v_i are calculated as follows:

$$\overline{x_i} = \frac{1}{n}\sum_{j=1}^{n} x_i(j) \quad (7)$$

$$v_i = \frac{1}{n}\sum_{j=1}^{n} (x_i(j) - \overline{x_i})^2 \quad (8)$$

where, j= 1,2,..., n (n is population size).

The mapping function is depicted in Figure 1. The transformation mapping function, h, is calculated by the mean $\overline{x}$ and shape variables s_{i1} and s_{i2} as follows:

$$h(\overline{x_i}, s_{i1}, s_{i2}, x) = \overline{x_i}.(1 - e^{-x.s_{i1}}) + (1 - \overline{x_i}).e^{-(1-x).s_{i2}} \quad (9)$$

where,

$$s_i = -\ln(v_i).f_s \quad (10)$$

The scaling factor f_s is a MVMO parameter which allows for controlling the search process during iteration. s_i is the shape variable.

• All variables are initialized within the limit range [0,1]. The output of mapping function is always inside [0,1]. However, the function evaluation is carried out always in the original scales.

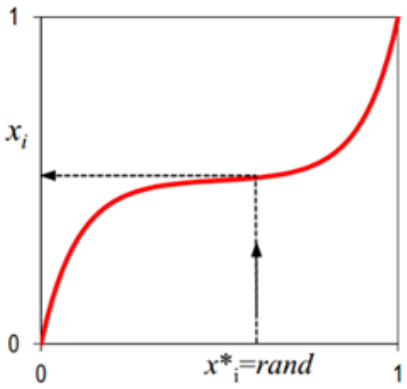

Figure 1: Variable mapping.

• MVMO is a single-agent search algorithm because it uses a single parent-offspring in each iteration. Therefore, the number of fitness evaluations is identical to the number of iterations.

Interested readers can find the basics of algorithm and reference values for the algorithm's settings in [23,24].

Extension of MVMO-MVMO^S

Recently,theswarm version of MVMO has been developed. This version is abbreviated as MVMO^S. The new approach extends the ability of global searching of the classical MVMO by starting the search with a set of particles.

Modified version of MVMO: The MVMO-algorithm extends two important parameters. These two parameters are used for calculation and assignment of s_{i1} and s_{i2} as follows:

Variable FS factor: In (10), the factor f_s allows the modification of the shape factor calculated from the variance.

The extension of f_s factor is 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:

$$f_s = f_s^* . \left(1 + rand()\right) \tag{11}$$

Where

$$f_s^* = f_{s\_\text{ini}}^* + \left(\frac{i}{i_{\text{final}}}\right)^2 \left(f_{s\_\text{final}}^* - f_{s\_\text{ini}}^*\right) \tag{12}$$

r and () is a random number with the bounds [0, 1].

In (12) , the variable *i* represents the iterationnumber.

For the more accuracy of the optimization ,the initial and final values of f_s^* it is recommended that $f_{s\_\text{ini}}^* < 1$ and $f_{s\_\text{final}}^* > 1$. The suggested range of initial values of f_s^* is from 0.9 to 1.0 and forfinal values of f_s^* is from 1.0 to 3.0 .

When $f_{s\_\text{final}}^* = f_{s\_\text{ini}}^* = 1$, which means that the option for controlling the f_s factor is not used.

Variable increment Δd : The MVMO^S algorithm uses the factor Δd as presented below:

s_{i1}=s_{i2}=s_i

if s_i> 0 then

$\Delta d = (1 + \Delta d_0) + 2 . \Delta d_0 (\text{rand}() - 0.5)$

if s_i>d_i

d_i=d_i . Δd

else

d_i=d_i / Δd (13)

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 extension of variable increment Δd is used for the asymmetric characteristic of the mapping function.

At the start ofthe algorithm, the initial values of d_i(typically between 1-5) are set for all variables. 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 by Δd which is always greater than 1.0 and thus resulting in reduced value of *di*. Therefore, d_i will always oscillate around the current shape factor s_i. Furthermore, Δd is varied randomly around the value$(1 + \Delta d_0)$ with the amplitude of Δd_0 adjusted in accordance to (14), where Δd_0 can be allowed to decrease from 0.4 to 0.01.

$$\Delta d_0 = \Delta d_0^{\text{ini}} + \left(\frac{i}{i_{\text{final}}}\right)^2 \left(\Delta d_0^{\text{final}} - \Delta d_0^{\text{ini}}\right) \tag{14}$$

Swarm variant of MVMO: Compared with classical MVMO, the swarm variant explores the solution space more aggressively. The search process is started with a set ofparticles, each having its own memory defined in terms of the corresponding archive and mapping function. Initially, each particle performs *m* steps independently to collect a set of reliable individual solutions. Then, the particles start to communicate and to exchange information.

The scheme of MVMO^S is depicted in Figure 2.

i and *k* donate the function evaluation and particle counters, respectively. Whereas *m* and n_p stand for maximum number of independent runs and total number of particles, respectively.

It is not worth it to follow particles which are very close to each other since this would entail redundancy. To avoid closeness between particles (i.e. redundancy), the normalized distance of each particle's

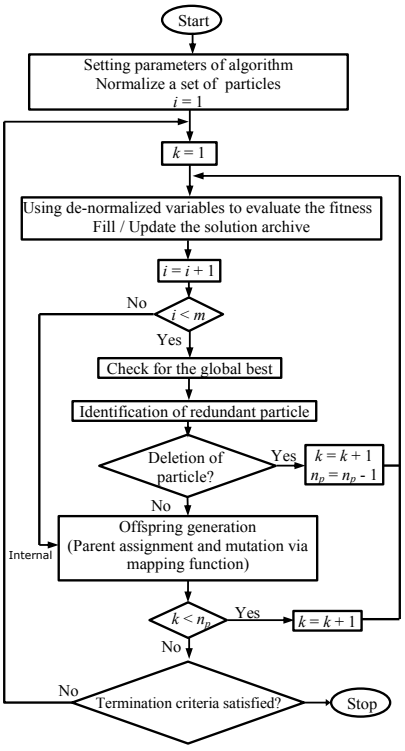

Figure 2: MVMOS flowchart.

local best solution $x^{lbest,i}$ to theglobal best x^{gbest} is calculated by:

$$D_i = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(x_k^{gbest} - x_k^{lbest,i})^2} \quad (15)$$

where, n denotes the number of optimization variables.

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}.

A zero threshold means that all particles are considered throughout the whole process whereas a unit threshold will result in the dropping of all particles except the global best. In this case after $(m^*n_p + n_p)$ fitness evaluations only one particle, the gbest, remains. Intermediate threshold values entail better adaptation to any optimization problem.

After independent evaluation, and if the particle is further considered, the global best solution guides the search by assigning x^{gbest}, instead of $x^{lbest,i}$, as parent for every particle's offspring. The remaining steps are identical with those of the classical MVMO: A subset of dimensions in the parent vector is directly inherited whereas the remaining dimensions are selected and mutated, based on local statistics (mean and variance) of the particle, via mapping function.

Implemention of MVMO^S to ED

Handing of constraints: To guarantee that the equality constraint (4) is always satisfied, a slack generating unit is randomlyselected from N generating units and therefore its power output will be dependent on the power outputs of remaining N-1 generating units in the system. The method for calculation of power output for the slack unit is given in [25]. The power output of the slack unit is as follows:

$$P_s = P_D + P_L - \sum_{\substack{i=1 \\ i\neq s}}^{N} P_i \quad (16)$$

where,s is a random unit selected from N units

The power transmission loss in (5) is rewritten by considering P_s as an unknown variable

$$P_L = B_{ss}P_s^2 + \left(2\sum_{\substack{i=1 \\ i\neq s}}^{N} B_{si}P_i + B_{0s}\right)P_s$$

$$+\sum_{\substack{i=1 \\ i\neq s}}^{N}\sum_{\substack{i=1 \\ i\neq s}}^{N} P_iB_{ij}P_j + \sum_{\substack{i=1 \\ i\neq s}}^{N} B_{0i}P_i + B_{00} \quad (17)$$

Substituting (17) into (13), a quadratic equation is abtained as follows:

$$A\times P_s^2 + B\times P_s + C = 0 \quad (18)$$

where A, B and C are given by:

$$A = B_{ss} \quad (19)$$

$$B = 2\sum_{\substack{i=1 \\ i\neq s}}^{N} B_{si}P_i + B_{0s} - 1 \quad (20)$$

$$C = \sum_{\substack{i=1 \\ i\neq s}}^{N}\sum_{\substack{i=1 \\ i\neq s}}^{N} P_iB_{ij}P_j + \sum_{\substack{i=1 \\ i\neq s}}^{N} B_{0i}P_i + B_{00} + P_D - \sum_{\substack{i=1 \\ i\neq s}}^{N} P_i \quad (21)$$

The power output of the slack generator is the posstive root of (18) between the two ones abtained as follows:

$$P_s = \frac{-B \pm \sqrt{B^2 - 4\times A\times C}}{2A} \quad (22)$$

where $B^2 - 4\times A\times C \geq 0$

Based on the slack variable method, the fitness function for the proposed MVMO^S will include the objective function (2) and penalty terms for the slack unit if inequality (6) is violated. The fitness function is as follows:

$$F_T = \sum_{i=1}^{N} F_i(P_i)$$
$$+K\times\left[\left(\max(0, P_s - P_{s,\max})\right)^2 + \left(\max(0, P_{s,\min} - P_s)\right)^2\right] \quad (23)$$

Implemention of MVMO^S to ED: 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} , Δd_0^{final} , archive zize, $f^*_{s\_ini}$, $f^*_{s\_final}$, n_randomly, n_randomly_min, indep.runs(m), D_{min}

Set i=1, i donates 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, kdonate particle counters.

***Step 4*:** Using de-normalized variables to evaluate fitness function, store $f_{best}$and x_{best}in archive

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

***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}.

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 stoping criteria is satisfied, stop. Otherwise, go to step 3.

De-nomalization of optimization variables: The output of mapping function is always inside [0,1]. However, the function evaluation is carried out always in the original scales. De-nomalization of optimization variables is carried by using (24):

P_i=$P_{i,min}$ + *Scaling.x_normalized(i,:)* (24)

where,

Scaling=$P_{i,max}$ -$P_{i,min}$

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

Numerical Analysis

The proposed MVMOS has been applied to the ED problem with

the quadratic cost function.Four test cases including 3, 13, 20 thermal generating units and large-scale systemwith 140 units are carried out. For each case, the algorithm of MVMOS is run 50 independent trialsona Intel Core i5-3470 CPU 3.2 GHz PC, Ram 4GB. The implematation of the proposed MVMOS was done in Matlab R2013a platform.

Selection of parameters

The parameters of MVMOSinclude itermax, n_var, n_par, mode, d_i, Δd_0^{ini}, $\Delta d_0^{\text{final}}$,archive zize, $f_{s\_\text{ini}}^*$, $f_{s\_\text{final}}^*$, n_randomly, indep.runs(m), Dmin. Since different parameters of the proposed method effect on the performance of MVMOS. Hence, it is important to determine an optimal set of parameters of the proposed methods for ED problem. For each problem, selection of parameters is carried out by varying only one parameter at a time and keeping the other. The parameter is first fixed at the low value and then increased. Multiple runs are carried out to choose the suitable set of parameters. The typical parameters are selected as follows:

- itermax : maximum number of iterations depend on the dimension of problems. The maximum number of iterations is selected in the range from 1000 to 50000 iterations for case 1, case 2 and case 3, and 80000 for case 4.

- n_var: number of variable (generators), dimension of problems. n_var is set to 3,13, 20, 140 for case 1, case 2, case 3and case 4, respectively.

- n_par: number of particles is varied from 5, 10, 20, 30, 40 and 50, respectively. By experiments, the good solution is obtained when number of particles isset to 5. Hence, number of particles is set for all cases.

- mode: There are four variable selection strategy for offspring creation [23]. Afer all simulations, stragy 3 (mode=4) is suporior to the other stragy.

Δd_0^{ini}, $\Delta d_0^{\text{final}}$: The range ofin (14) is [0.01-0.4]. By experiments,andis set to 0.4 and 0.02, respectively for all cases.

$f_{s\_\text{ini}}^*$, $f_{s\_\text{final}}^*$: Therange of values ofis from 0.9 to 1.0 and forvalues ofis from 1.0 to 3.0 [24]. For all cases,is set to 0.95 in the range [0.9, 1.0] andis set to 3 in the range 3 in the range [1.0, 3.0].

indep.runs(m) : The maximum number of independent runs can be selected in the range from 100 to 800.

D min is set to 0 for all cases.

Numerical results

Case 1: 3 unit -system: The test system consists of 3 generating units without transmission loss. Here, the system load demand is 450MW and 850MW, respectively. The data of the system is taken from [25]. The power transmission loss is neglected in this case. The obtained results by the MVMOS corresponding to the two load demand are given in Table 1.

The parameters for MVMOS are set as follows: itermax=1000, n_var(generators)=3, np=5, archive size=4, indep.runs (m)=100, n_randomly=2, n_randomly_min=2, $f_{s\_\text{ini}}^* = 0.95$, $f_{s\_\text{final}}^* = 3$, $d_i = 1$, $\Delta d_0^{\text{ini}} = 0.4$, $\Delta d_0^{\text{final}} = 0.02$, Dmin=0

The total cost comparison between MVMOS and the other methods are presented in Table 2. In case ofthe 450MW load demand, the results and computational time of MVMOS are less than PSO and ABC. In case ofthe 850MW load demand, the results ofMVMOS is less than IEP, HS, GA, BGA, and the same as NM, PSO. The proposed MVMOS is faster than HS, GA and BGA. There is no computer processor reported for PSO, ABC, HS, GA and BGA and no computational time for the other methods. Table 1 shows that the power output obtained by the MVMOS is always satisfy the constraints.

Case 2: 13 unit - system: The data of 13 generating unit test system is from [27]. In this case, the power transmission loss is neglected. The obtained results by the MVMOS corresponding to the twoload demand of 1800MW and 2520MW are shown in Table 3.

For the load demand of 1800MW, the parameters for MVMOS are

Unit	Power outputs P_i (MW)	
	P_D=450 MW	P_D=850 MW
1	205.3077	393.1698
2	183.3457	334.6038
3	61.3466	122.2264
Total power(MW)	450	850
Min Cost ($/h)	4652.4735	8194.3561
Average CPU time (s)	0.87	0.88

Table 1: Power output of 3-unit system for load demand of 450 MW and 850 MW by MVMOS.

Method	450 (MW)		850 (MW)	
	Cost ($/h)	CPU (s)	Cost ($/h)	CPU (s)
PSO [26]	4653	7.69	-	-
ABC [26]	4653	3.91	-	-
NM [25]	-	-	8194.3561	-
IEP [25]	-	-	8194.3561	-
PSO [25]	-	-	8194.3561	-
HS [27]	-	-	8194.5	27.62
GA [27]	-	-	8194.3591	10.94
BGA [27]	-	-	8194.357	3.66
MVMO^S	4652.4735	0.87	8194.3561	0.88

Table 2: Comparison of results and CPU time by MVMO and other techniques for 3-unit system.

Unit	Power outputs P_i (MW)	
	P_D=1800 MW	P_D=2520 MW
1	506.9117	679.9970
2	253.4558	359.9957
3	253.4560	360.0000
4	99.3627	155.1418
5	99.3627	155.1929
6	99.3627	155.0396
7	99.3628	156.0892
8	99.3628	154.4661
9	99.3627	154.0705
10	40.0000	40.0015
11	40.0000	40.0038
12	55.0000	55.0022
13	55.0000	55.0000
Total power (MW)	1800.0000	2520.0000
Min Cost ($/h)	17932.4741	24050.1408
Average CPU time (s)	2.97	16.29

Table 3: Power output of each generating unit in 13-unit system for load demand of 1800 MW and 2520 MW by MVMOS.

set as follows: itermax=5000, n_var(generators)=13, npn$_p$=5, archive size=4, indep.runs (m)=300, n_randomly=5, n_randomly_min=4, $f^*_{s\_ini}=0.95$, $f^*_{s\_final}=3$, $d_i=1$, $\Delta d_0^{ini}=0.4$, $\Delta d_0^{final}=0.02$, Dmin=0.

For the load demand of 2520MW, the parameters for MVMOS are set as follows: itermax=30000, n_var(generators)=13, np=5, archive size=5, indep.runs (m)=300, n_randomly=8, n_randomly_min=4 $f^*_{s\_ini}=0.95$, $f^*_{s\_final}=3$, $d_i=1$, $\Delta d_0^{ini}=0.4$, $\Delta d_0^{final}=0.02$, Dmin=0.

The results of MVMOS for 1800 MW and 2520 MW load demands are compared to the other methods as presented in Table 4. In case of the 1800 MW load demand, the total cost obtained by MVMOS is less than HS, GA and BGA. The computational time of MVMOS is less than HS, GA and slower than BGA. There is no computer processor reported for HS, GA and BGA. In case of the 2520 MW load demand, the total cost obtained by MVMOS is less than-Iteration, GA and SQP and same result as ALHN. The computational time of MVMOS is slower than these methods. The computational times for-Iteration, GA, SQP and ALHN methods were from a Petium M 1.5 GHz PC. Table 3 shows that the power output obtained by the MVMOS is always satisfy the constraints

Case 3: 20 unit-system: The test system includes 20 generators with the system load demand of 2500MW. The data of this system is from [28]. The power transmission loss is ignored in this case. The obtained results by the MVMOS is shown in Table 5.

The MVMOS is run 50 independent trials. The parameters for MVMOS are set as follows: itermax=70000, n_var(generators)=5, np=5, archive size=4, indep. runs(m)=400, n_randomly=7, n_randomly_min=6, $f^*_{s\_ini}=0.95$, $f^*_{s\_final}=3$, $d_i=1$, $\Delta d_0^{ini}=0.4$, $\Delta d_0^{final}=0.02$, Dmin=0.

Method	1800 (MW)		2520 (MW)	
	Cost ($/h)	CPU(s)	Cost ($/h)	CPU(s)
HS [27]	18274.0065	16.135	-	-
GA [27]	18194.9507	5.8	-	-
BGA [27]	17971.5503	1.98	-	-
λ -Iteration [28]	-	-	24058.27	0.85
GA [28]	-	-	25087.45	1.76
SQP [28]	-	-	24058.29	4.57
ALHN [28]	-	-	24050.14	0.044
MVMO^S	17932.4741	2.97	24050.14	16.29

Table 4: Comparison of results and CPU time by MVMO and other techniques for 13-unit system.

Unit	Power outputs P_i(MW)	Unit	Power outputs P_i (MW)
1	600.0002	11	286.9466
2	131.1723	12	432.7209
3	50.0000	13	124.1584
4	50.0000	14	73.3046
5	92.9882	15	94.8873
6	20.0000	16	36.2083
7	125.0000	17	30.0000
8	50.0000	18	37.5181
9	111.7012	19	77.8376
10	45.5563	20	30.0000
Total power (MW)		2500.0000	
Min Cost ($/h)		60152.5509	
Average CPU time (s)		31.45	

Table 5: Power output of 20-unit system for load demand of 2500MW by MVMO^S.

Method	Total CostP_D=2500MW	CPU(s)
λ -Iteration [27]	60245.67	0.32
GA [27]	61107	61.17
SQP [27]	60693.14	1.28
ALHN [27]	60152.55	0.076
MVMO^S	60152.55	31.45

Table 6: Comparison of results and CPU time by MVMO and other techniques for 20-unit system.

Table 6 shows the comparison of results obtained and computational time by MVMOS and the other methods. In this case, the results of MVMOS is less-Iteration, GA and SQP and the same as ALHN. The computation time of MVMOS is less than GA and slower than other methods. The computational times for-Iteration, GA, SQP and ALHN methods were from a Petium M 1.5 GHz PC. Table 5 shows that the power output obtained by the MVMOS is always satisfy the constraints. Although the parameters for two load demands is different, the MVMOS guarantees the convergence to the global solution for the 13-unit test system.

Case 4: large-scale system 140 unit: The Korean power system consists of 140 thermal generating units is the test system for this case. Here, the system load demand is 49342 MW. The data of the system is given in [31]. The power transmission loss is also ignored in this case.

The parameters for MVMOS are set as follows: itermax=80000, n_var(generators)=140, np=5, archive size=4, indep. runs (m)=800, n_randomly=20, n_randomly_min=10, $f^*_{s\_ini}=0.95$, $f^*_{s\_final}=3$, $d_i=1$, $\Delta d_0^{ini}=0.4$, $\Delta d_0^{final}=0.02$, D min=0

The obtained results and computational time by the MVMOS are given in Table 7. As seen in Table 7, the power output obtained by the MVMOS is always satisfy the constraints.

Table 8 shows the comparison of results and computational time obtained by MVMOS and the other methods. In this case, the results of MVMOS is less than CTPSO, CSPSO, COPSO, CCPSO and KVMO. The computation time of MVMOS is slower than these methods. The computational times for CTPSO, CSPSO, COPSO and CCPSO were from Pentium IV 2.0-GHz computer.

Robustness analysis

The convergence of heuristic methods may not obtain exactly same solution because these methods initialize variables randomly at each run. Hence, their performances could not be judged by the results of a single run. Many trials should be carry out to reach a impartial conclusion about the performance of the algorithm. Therefore, in this study, 50 independent trials were carried out. The mean cost, max cost, average cost and standard deviation obtained by the proposed method to evaluate the robustness characteristic of the proposed method for ED problem. The robustness analysis of four cases test are presented in (Tables 9 and 10).

Tables 9 and 10 clearly show that the performance the proposed MVMOS is very robust.

Results and Discussions

A solution of optimization techniques needs concern with two elements:

- Computation time: time to get the best solution is the shortest one.

Unit	Pi (MW)	Unit	Pi (MW)	Unit	Pi (MW)
1	113.6343	47	239.5319	94	984.0000
2	189.0000	48	250.0000	95	978.0000
3	190.0000	49	250.0000	96	682.0000
4	190.0000	50	250.0000	98	720.0000
5	170.4101	51	165.0000	99	718.0000
6	190.0000	52	165.0000	100	720.0000
7	490.0000	53	165.0000	101	964.0000
8	490.0000	54	165.0000	102	958.0000
9	496.0000	55	180.0000	103	1007.0000
10	496.0000	56	180.0000	104	1006.0000
12	496.0000	57	103.0000	105	1013.0000
13	496.0000	58	198.0000	106	1020.0000
14	506.0000	59	312.0000	107	954.0000
15	509.0000	60	279.9162	108	952.0000
16	506.0000	61	163.0000	109	1006.0000
17	505.0000	62	95.0000	110	1013.0000
18	506.0000	63	160.0000	111	1021.0000
19	506.0000	64	160.0000	112	1015.0000
20	505.0000	65	490.0000	113	94.0000
21	505.0000	66	196.0000	114	94.0000
22	505.0000	67	490.0000	115	94.0000
23	505.0000	68	490.0000	116	244.0000
24	505.0000	69	130.0000	117	244.0000
25	505.0000	70	280.6907	118	244.0000
26	537.0000	71	137.0000	119	95.0000
27	537.0000	72	334.031	120	95.0000
28	549.0000	73	195.0000	121	116.0000
29	549.0000	74	175.0000	122	2.0000
30	501.0000	75	175.0000	123	4.0000
31	501.0000	76	175.0000	124	15.0000
32	506.0000	77	175.0000	125	9.0000
33	506.0000	78	330.0000	126	12.0000
34	506.0000	79	531.0000	127	10.0000
35	506.0000	80	531.0000	128	112.0000
36	500.0000	81	350.0494	129	4.0000
37	500.0000	82	56.0000	130	5.0000
38	241.0000	83	115.0000	131	5.0000
39	241.0000	84	115.0000	132	50.0000
40	774.0000	85	115.0000	133	5.0000

Table 7: Power output of 140-unit system for load demand of 49342MW by MVMOS.

Method	Best Total Cost	CPU
	P_D=49342MW	(s)
CTPSO [29]	1655685	50.1
CSPSO[29]	1655685	9.6
COPSO [29]	1655685	76.9
CCPSO [29]	1655685	42.9
KMVO [30]	1577607	-
MVMO^S	1557461	105.39

Table 8: Comparison of results and CPU time by MVMOS and other techniques for 140-unit system.

• Quality of solution: the quality of solution need robustness, near or better the global solutions of the other techniques.

In addition, it takes note the large-scale system problem. The computation time of technique on the large-scale system may be take more time but the quality of solution need optimum.

	Case 1: 3 unit		Case2: 13 unit	
	450(MW)	850(MW)	1800(MW)	2520(MW)
Min Cost ($/h)	4652.4735	8194.3561	117932.4741	24050.1408
Average Cost ($/h)	4652.4735	8194.3561	17932.4741	24050.277
Max Cost ($/h)	4652.4735	8194.3561	17932.4741	24050.189
Standard deviation ($/h)	0	0	0	0.0309

Table 9: Robustness analysis of the proposed MVMOS by 50 independent trials for Case 1 and Case 2.

	Case 3: 20 unit	Case 4 : 140 unit
	2500 (MW)	49342(MW)
Min Cost ($/h)	60152.5509	1557461.803
Average Cost ($/h)	60153.2215	1557481.743
Max Cost ($/h)	60152.7388	1557481.743
Standard deviation($/h)	0.1469	3.0107

Table 10: Robustness analysis of the proposed MVMOS by 50 independent trials for Case 3 and Case 4.

Based on the numerical results and robustness analysis of the proposed method, it indicates that the MVMOS obtained the global solution with high probability, especially for large-scale system due to the global search capability is enhanced. Besides its ability, the proposed MVMOS is also easy to be implemented for ED problem. Unlike other swarm-based optimization techniques, MVMOS does not strictly require many particles to progess. In this study, number of particles is set to 5 for all cases. The MVMOS showed the good performance. However, the computation time is relatively high for large-scale system. Similar to original MVMO, the number of iterations in MVMOS is equivalent to the number of offspring fitness evaluations which is in practical applications usually comsume more time than the optimization algorithm itself.

In future, the MVMOS is proposed for solving the non-convex ED problems with complicated objective function.

Conclusions

In this paper the proposed MVMOS has been tested for the ED problem with quadratic cost function efficiently and effectively.The numerical results show that the MVMOS exhibits a robust performance and also provides the good solutions for all test systems, expecially for lagre-scale system. The proposed method has merits as follows: easy implementation, good solutions, robustness of algorithm; applicable to large-scale system. Therefore, the proposed MVMOS could be favorable for solving other ED problems.

Acknowledgment

This research work is financially supported by Graduate Assistant Scheme (GAS) of Universiti Teknologi PETRONAS and with the help of Department of Fundamental and Applied Sciences, Faculty of Science and Information Technology, UTP.

References

1. Dieu VN, Schegner P, Ongsakul W (2013) Pseudo-gradient based particle swarm optimization method for nonconvex economic dispatch. Power, Control and Optimization 239: 1-27.
2. Xia X, Elaiw A (2010) Optimal dynamic economic dispatch of generation: a review. Electric Power Systems Research 80: 975-986.
3. Wallenberg BF, Wood AJ (1996) Power generation, operation and control. John Wiley and Sons Inc 264-327.

4. Parikh J, Chattopadhyay D (1996) A multi-area linear programming approach for analysis of economic operation of the Indian power system. Power Systems IEEE Transactions 11: 52-58.

5. Nanda J, Hair L, Kothari M (1994) Economic emission load dispatch with line flow constraints using a classical technique. IEE Proceedings-Generation, Transmission and Distribution 141: 1-10.

6. Fan JY, Zhang L (1998) Real-time economic dispatch with line flow and emission constraints using quadratic programming. Power Systems IEEE Transactions 13: 320-325.

7. Vasant P, Ganesan T, Elamvazuthi I (2012) Solving deterministic non-linear programming problem using Hopfield artificial neural network and genetic programming techniques. AIP Conference Proceedings-American Institute of Physics.

8. Lee KY, Sode-Yome A, Park JH (1998) Adaptive Hopfield neural networks for economic load dispatch. Power Systems IEEE Transactions 13: 519-526.

9. Sinha N, Chakrabarti R, Chattopadhyay P (2003) Evntolutionary programming techniques for economic load dispatch. Evolutionary Computation IEEE Transactions 7: 83-94.

10. Noman N, Iba H (2008) Differential evolution for economic load dispatch problems. Electric Power Systems Research 78: 1322-1331.

11. Chiang CL (2005) Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. Power Systems IEEE Transactions on 20: 1690-1699.

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

13. Coelho LDS, Lee CS (2008) Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches. International Journal of Electrical Power and Energy Systems 30: 297-307.

14. Vasant P, Ganesan T, Elamvazuthi I (2012) An improved PSO approach for solving non-convex optimization problems. ICT and Knowledge Engineering (ICT and Knowledge Engineering).

15. Padmanabhan B, Sivakumar R, Jasper J, Victoire TAA (2011) Bacterial foraging approach to economic load dispatch problem with non-convex cost function. Swarm, Evolutionary, and Memetic Computing 7076: 577-584.

16. Dinh LL, Ngoc DV, Vasant P (2013) Artificial bee colony algorithm for solving optimal power flow problem. Scientific World Journal

17. Attaviriyanupap P, Kita H, Tanaka E, Hasegawa J (2002) A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function. Power Systems IEEE Transactions 17: 411-416.

18. Coelho LS, Mariani VC (2006) Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. Power Systems IEEE Transactions 21: 989-996.

19. He DK, Wang FL, Mao ZZ (2008) Hybrid genetic algorithm for economic dispatch with valve-point effect. Electric Power Systems Research 78: 626-633.

20. Wang SK, Chiou JP, Liu CW (2007) Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm. Generation, Transmission and Distribution IET 1: 793-803.

21. Niknam T (2010) A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem. Applied Energy 87: 327-339.

22. Vasant P, Global I (2013) Meta-heuristics Optimization Algorithms in Engineering, Business, Economics, and Finance. Information Science Reference

23. Erlich I, Venayagamoorthy GK, Nakawiro W (2010) A mean-variance optimization algorithm. Proc. IEEE Congress on Evolutionary.

24. Rueda JL, Erlich I (2013) Evaluation of the mean-variance mapping optimization for solving multimodal problems. Swarm Intelligence (SIS) 2013: 7-14.

25. Polprasert J, Ongsaku W (2007) Augmented Lagrange Hopfield Network For Economic Dispatch. Power Engineering Conference, Australasian Universities, Australia.

26. Kumar S (2012) Economic Load Dispatch Problem Based Biogeography Algorithm." Undergraduate Academic Research Journal (UARJ).

27. Park JB, Lee KS, Shin JR, Lee Y (2003) Economic load dispatch for nonsmooth cost functions using particle swarm optimization. Power Engineering Society General Meeting, IEEE.

28. Baijal A, Chauhan VS, Jayabarathi T (2011) Application of PSO, Artificial bee colony and bacterial foraging optimization algorithms to economic load dispatch: An analysis.International Journal of Computer Science Issues 8: 467-470.

29. Kuo CC (2008) A novel coding scheme for practical economic dispatch by modified particle swarm approach. Power Systems, IEEE Transactions 23: 1825-1835.

30. Park JB, Jeong YW, Shin JR, Lee KY (2010) An improved particle swarm optimization for nonconvex economic dispatch problems. Power Systems, IEEE Transactions 25: 156-166.

31. Kim MJ, Song HY, Park JB, Roh JH, Lee SU, et al. (2013) An improved mean-variance optimization for nonconvex economic dispatch problems. Journal of Electrical Engineering and Technology 8: 80-89.