Retraction

Retracted: Dynamic Harmony Search with Polynomial Mutation Algorithm for Valve-Point Economic Load Dispatch

The Scientific World Journal

The Scientific World Journal has retracted the article titled "Dynamic Harmony Search with Polynomial Mutation Algorithm for Valve-Point Economic Load Dispatch" [1]. After conducting a thorough investigation, we have strong reason to believe that the peer review process was compromised.

This article was originally submitted to a Special Issue titled "Recent Advances in Metaheuristics and Its Hybrids." In late 2015, Dr. Xavier Delorme, the lead guest editor on the Special Issue, alerted us that his identity had been compromised. After further investigation, we discovered that several peer review reports in this issue had been submitted from similarly compromised email accounts.

We are retracting the articles in keeping with the "COPE statement on inappropriate manipulation of the peer review process." There is no evidence that any of the authors or editors, including Dr. Delorme, were aware of this misconduct.

References

[1] M. Karthikeyan and T. Sree Ranga Raja, "Dynamic harmony search with polynomial mutation algorithm for valve-point economic load dispatch," *The Scientific World Journal*, vol. 2015, Article ID 147678, 10 pages, 2015.

Research Article

Dynamic Harmony Search with Polynomial Mutation Algorithm for Valve-Point Economic Load Dispatch

M. Karthikeyan[1] and T. Sree Ranga Raja[2]

[1]*Department of Electrical and Electronics Engineering, University College of Engineering Pattukkottai, Rajamadam, Tamilnadu 614701, India*
[2]*Department of Electrical and Electronics Engineering, University College of Engineering, Nagerkovil, Tamilnadu 629001, India*

Correspondence should be addressed to M. Karthikeyan; karthikeyanaut2015@gmail.com

Received 31 January 2015; Revised 1 April 2015; Accepted 4 April 2015

Economic load dispatch (ELD) problem is an important issue in the operation and control of modern control system. The ELD problem is complex and nonlinear with equality and inequality constraints which makes it hard to be efficiently solved. This paper presents a new modification of harmony search (HS) algorithm named as dynamic harmony search with polynomial mutation (DHSPM) algorithm to solve ORPD problem. In DHSPM algorithm the key parameters of HS algorithm like harmony memory considering rate (HMCR) and pitch adjusting rate (PAR) are changed dynamically and there is no need to predefine these parameters. Additionally polynomial mutation is inserted in the updating step of HS algorithm to favor exploration and exploitation of the search space. The DHSPM algorithm is tested with three power system cases consisting of 3, 13, and 40 thermal units. The computational results show that the DHSPM algorithm is more effective in finding better solutions than other computational intelligence based methods.

1. Introduction

Economic load dispatch (ELD) is an important issue in the operation and control of modern control system. The objective of ELD problem can be defined as determining the real power outputs of generators so as to meet the required load demand at minimum operating cost while satisfying system equality and inequality constraints [1]. The objective of ELD is to minimize the total operating cost, but the various types of physical and operational constraints make ELD a highly nonlinear constrained optimization problem. Traditionally different approaches have been suggested to solve ELD, including linear programming [2], dynamic programming [3], and nonlinear programming [4]. The main drawback of these techniques is that they may not be able to give an optimal solution and may get stuck at local optima.

Recently, different heuristic approaches have been used to solve ELD problem with promising performance, such as genetic algorithm (GA) [5], evolutionary programming (EP) [6], differential evolution (DE) [7], and particle swarm optimization (PSO) [8]. In spite of the fact that these heuristic methods do not always guarantee finding global optimal solutions in specified time, they often provide fast and reasonable solution. Although several heuristic methodologies have been developed for the ELD problem, the difficulty of the problem reveals the need for development of efficient algorithms to exactly locate the optimum solution.

Harmony search (HS) is a new metaheuristic algorithm proposed by Geem et al. [9], which is inspired by the natural musical performance process that happens when a musician searches for a better state of harmony. HS algorithm has been successfully applied to a wide range of applications such as structural optimization [10], design optimization of water distribution networks [11], and vehicle routing [12].

Although HS algorithm is good at identifying the solution in the search space within a reasonable time, it is not

efficient in performing local search in numerical optimization applications [13]. To overcome this drawback, Mahdavi et al. [13] proposed an improved HS algorithm denoted as improved harmony search (IHS) by dynamically updating pitch adjustment rate (PAR) and bandwidth (bw). Omran and Mahdavi [14] proposed a global best HS algorithm denoted as global harmony search (GHS) by borrowing the idea from swarm intelligence. Khalili et al. [15] proposed global dynamic harmony search (GDHS) algorithm for solving continuous optimization problem.

In this paper, we present a novel variant of HS algorithm, named dynamic harmony search with polynomial mutation (DHSPM) algorithm in which harmony memory considering rate (HMCR) and pitch adjusting rate (PAR) are dynamically updated. Additionally, polynomial mutation is inserted in the updating step of HS algorithm to favor exploration and exploitation of the search space.

The paper is organized as follows. Section 2 presents the formulation of ELD problem with valve-point. Section 3 contains a brief overview of HS and DHSPM algorithms. Section 4 reports the application of DHSPM to ELD problem with valve-point effect. Section 5 contains the description of the simulations and a discussion of the results. Conclusions are summarized in Section 6.

2. Problem Formulation

2.1. Economic Dispatch. The primary objective of the ELD problem is to determine the most economic loading of the generators such that the total demand is met while satisfying equality and inequality constraints. The objective function of ELD is defined as

$$\min F_T = \min \sum_{i=1}^{N} F_i\left(P_i\right). \tag{1}$$

F_T is the objective function describing the total generation cost. $F_i(P_i)$ is the cost function of generator i to generate real power P_i. N is the total number of generators in the power system.

The fuel cost function of unit i is defined by

$$F_i\left(P_i\right) = a_i P_i^2 + b_i P_i + c_i, \tag{2}$$

where a_i, b_i, and c_i are the cost coefficients of unit i.

2.2. Economic Load Dispatch with Valve-Point Loading Effects. Multivalve steam turbines based generating units are characterized by complex nonlinear fuel cost function. This is mostly due to the ripples made by the valve-point loading. To simulate these complex phenomena, a sinusoidal component is added on the quadratic heat rate curve. To take into account this effect, the cost function in (2) is modified as follows:

$$F_i\left(P_i\right) = a_i + b_i P_i + c_i P_i^2 + \left| e_i \sin\left(f_i\left(P_i^{\min} - P_i\right)\right)\right|, \tag{3}$$

where e_i and f_i denote the cost coefficients of ith generator reflecting valve-point loading effect and $P_i^{\min}$ is the minimum output power of ith generator unit.

2.3. Constraints

2.3.1. Real Power Balance Constraint. The total power generated should be equal to the total load demand plus the total transmission losses. The real power balance can be expressed as

$$\sum_{i=1}^{N} P_i = P_D + P_L, \tag{4}$$

where P_D is the total demand and P_L denotes the total transmission losses. In this paper, we disregarded the transmission loss, $P_L = 0$.

2.3.2. Generator Capacity Constraints. Real power output of each generator should be within its minimum and maximum limits. This can defined as follows:

$$P_{i,\min} \leq P_i \leq P_{i,\max}, \tag{5}$$

where $P_{i,\min}$ and $P_{i,\max}$ are the minimum and maximum output power of ith generating unit, respectively.

2.4. Formulation of Fitness Function. In this paper, we use penalty term to transform a constrained optimization problem into an unconstrained one. As a result, the fitness function can be written as

$$\text{Min } F = \sum_{i=1}^{n} F_i\left(P_i\right) + \gamma \left|\sum_{i=1}^{n} P_i - \left(P_D + P_L\right)\right|, \tag{6}$$

where γ is the penalty coefficient. The penalty coefficient should be given large enough to guarantee the system constraints. In this paper, we choose $\gamma = 1000$.

3. Harmony Search Algorithms

3.1. Basic Harmony Search Algorithm. In basic harmony search (HS) algorithm, each solution is called a "harmony" and represented by an n-dimensional real vector. An initial population of harmony vectors is randomly created to form a harmony memory (HM). Then, a new harmony vector is generated by using a memory consideration rule, a pitch adjustment rule, and a random reinitialization. The generated new harmony vector is updated in the HM by comparing the new harmony vector and the worst harmony vector in the HM. The above process is repeated until a certain criterion is met. The steps of HS algorithm are described below in detail.

Step 1 (initialization of problem and algorithm parameters). Consider an optimization problem that is described by

$$\text{Minimize } F\left(x\right) \quad x_i \in X_i, \ i = 1, 2, \ldots, N, \tag{7}$$

where $F(x)$ is the objective function, x is the set of design variables, and X_i is the range set of the possible values for each design variable. The parameters of the HS algorithm are the harmony memory size (HMS), harmony memory considering rate (HMCR), pitch adjusting rate (PAR), number of decision variables (N), and number of improvisations (NI).

Step 2 (harmony memory initialization). The harmony memory (HM) matrix is filled with randomly generated solution vectors for HMS and sorted by the values of objective function $f(x)$ as shown below:

$$\text{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ x_1^{\text{HMS}-1} & x_2^{\text{HMS}-1} & \cdots & x_{N-1}^{\text{HMS}-1} & x_N^{\text{HMS}-1} \\ x_1^{\text{HMS}} & x_2^{\text{HMS}} & \cdots & x_{N-1}^{\text{HMS}} & x_N^{\text{HMS}} \end{bmatrix} \begin{matrix} \longrightarrow f\left(x^{(1)}\right) \\ \longrightarrow f\left(x^{(2)}\right) \\ \vdots \\ \longrightarrow f\left(x^{(\text{HMS}-1)}\right) \\ \longrightarrow f\left(x^{(\text{HMS})}\right). \end{matrix} \tag{8}$$

Step 3 (new harmony improvisations). A new harmony vector X_{new} is created by applying three rules: a memory consideration, a pitch adjustment, and a random selection. A random number r_1 between 0 and 1 is generated. If r_1 is less than HMCR, then X_{new} is generated by the memory consideration; otherwise, X_{new} is obtained by randomly generating a vector between the upper and lower bounds. In the memory consideration, X_{new} is selected from any harmony vector in HM. After memory consideration, X_{new} will undergo a pitch adjustment with a probability of PAR. The pitch adjustment rule is given as follows:

$$X_{\text{new}} = X_{\text{new}} \pm r \times \text{bw}, \tag{9}$$

where r is a random number generated between 0 and 1.

Step 4 (updating harmony memory). If the new harmony vector X_{new} has better fitness function than the worst harmony in the HM, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

Step 5 (checking the stopping criterion). If the stopping criterion, which is based on the maximum number of improvisations, is satisfied, the computation is terminated. Otherwise, Steps 3 and 4 are repeated.

3.2. Variants of HS Algorithm. Mahdavi et al. proposed improved harmony search (IHS) algorithm to address the limitations of the basic HS algorithm. IHS algorithm applies the same memory consideration, pitch adjustment, and random selection as the basic HS algorithm, but the author suggests a new formula for PAR and bw which dynamically changes at every iteration [13].

Omran and Mahdavi proposed a global best harmony search (GHS) algorithm which is based on the inspiration by the particle swarm optimization. Unlike the basic HS algorithm, the GHS algorithm generates a new harmony vector by making use of the best harmony vector [14].

Khalili et al. proposed a global dynamic harmony search (GDHS) algorithm by modifying the basic HS algorithm to solve continuous optimization problems [15].

Pan et al. proposed a self-adaptive global best harmony (SGHS) algorithm for solving continuous optimization problem. In SGHS, new improvisation scheme was suggested so that good information obtained in the current global best solution is utilized to generate new harmonies [16].

3.3. Dynamic Harmony Search with Polynomial Mutation (DHSPM) Algorithm. In this paper, a novel HS algorithm, called DHSPM, for solving ORPD problem of power system, is presented. The proposed algorithm is different from the classical HS algorithm in the following two aspects. First, a dynamic parameter adjustment scheme is suggested, which can dynamically update the parameters HMCR and PAR in every improvisation. Second, a polynomial mutation is inserted in the updating step of HS algorithm to favor exploration and exploitation of the search space. The details of the algorithm are given below.

3.3.1. Dynamic Control Parameters. The conventional HS algorithm uses fixed value for both HMCR and PAR. In the HS algorithm, HMCR and PAR are fixed in the initialization step and cannot be changed during the improvisation. The main drawback of this method is that the number of iterations needed to find optimal solution is more [13]. Here, we suggest dynamic formula for HMCR and PAR which change during the improvisation of the optimization. The suggested formulas for HMCR and PAR for the current improvisation t are

$$\text{HMCR} = \begin{cases} C_1 * t + 0.9 & 0 \le t \le \dfrac{\text{NI}}{2} \\ -C_1 * t + 1.1 & \dfrac{\text{NI}}{2} \le t \le \text{NI}, \end{cases}$$

$$\tag{10}$$

$$\text{PAR} = \begin{cases} C_2 * t + 0.3 & 0 \le t \le \dfrac{\text{NI}}{2} \\ -C_2 * t + 1.1 & \dfrac{\text{NI}}{2} \le t \le \text{NI}, \end{cases}$$

where

$$C_1 = \frac{(1 - 0.9)}{\text{NI}/2},$$

$$C_2 = \frac{(0.7 - 0.3)}{\text{NI}/2}. \tag{11}$$

Figures 1 and 2 show the schematic of HMCR and PAR in dynamic mode for the number of improvisations equal to 1000. At initial improvisations, a linear increase of HMCR makes the algorithm generate more new harmony vectors rather than choosing from the harmony memory. At the middle of the improvisations, the HMCR is equal to 1, which consider the harmony vector from the HM itself. At final improvisations, HMCR is linearly decreased, which helps to escape the optimization process from settling in local optima. Similarly, the large value of PAR at the middle of the improvisation enforces the selected harmony vector to have adjustments.

3.3.2. Polynomial Mutation. Mutation is an important operator in genetic algorithms (GAs), as it ensures the maintenance of diversity in the evolving populations of GAs [17]. There are several mutation techniques that can be used in genetic algorithms like random mutation [18], boundary mutation [18], nonuniform random mutation [19], power mutation

TABLE 1: Mean and standard deviation of the benchmark functions.

| Function	HS	IHS	DHSPM
f_1	7.711433 (3.307032)	0.000000 (0.000000)	0.000000 (0.000000)
f_2	0.112437 (0.059248)	0.000009 (0.000001)	0.000003 (0.000001)
f_3	304.359111 (513.738959)	93.636178 (80.553169)	91.3741 (73.55367)
f_4	12.500000 (4.960186)	0.033333 (0.182574)	0.028978 (0.168912)
f_5	4570.725435 (1625.045376)	1841.741864 (711.620590)	1792.6421 (691.28015)
f_6	0 26.074848 (8.945656)	0.000382 (0.000000)	0.000287 (0.000000)
f_7	0.699759 (0.701654)	0.230684 (0.442564)	0.197635 (0.38745)
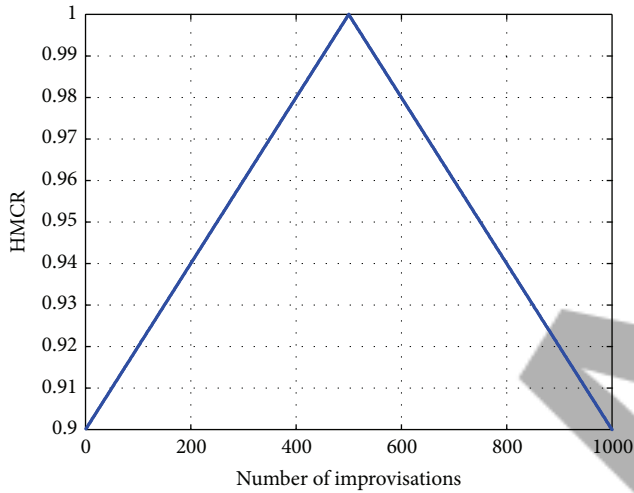
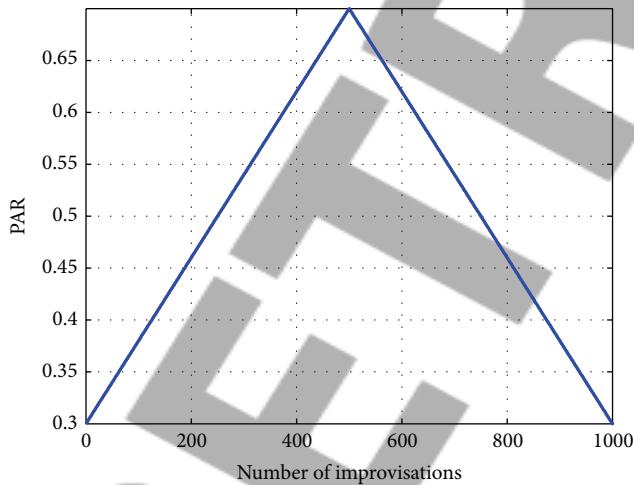FIGURE 1: Variations of HMCR with respect to improvisations.

FIGURE 2: Variations of PAR with respect to improvisations.

TABLE 2: Parameter settings of DHSPM for test systems.

HMS	bw	η_m	NI
5	0.01	10	50000

functions [20]. Deb and Agrawal [18] suggested a polynomial mutation operator with a user-defined index parameter (η_m). They concluded that η_m induces an effect of a perturbation of $O((b - a)/\eta_m)$ in a variable, where a and b are lower and upper bounds. They also found that a value $\eta_m \in [20, 100]$ is adequate in most problems that they tried of the variable. In this paper, we suggest inserting polynomial mutation in the updating step of HS algorithm as follows:

$$
x'_{\text{new}}(j)
= \begin{cases} x_{\text{new}}(j) + \delta_1 \left(x_{\text{new}}(j) - x_{\text{lower},j} \right) & \text{for } r \leq 0.5 \\ x_{\text{new}}(j) + \delta_2 \left(x_{\text{upper},j} - x_{\text{new}}(j) \right) & \text{for } r > 0.5. \end{cases} \quad (12)
$$

Then, either of the two parameters (δ_1 or δ_2) is calculated as follows:

$$
\begin{aligned}
\delta_1 &= (2r)^{1/(1+\eta_m)} - 1 & \text{for } r \leq 0.5, \\
\delta_2 &= 1 - (2(1 - r))^{1/(1+\eta_m)} & \text{for } r > 0.5,
\end{aligned} \quad (13)
$$

where $x_{\text{new}}(j)$ is the new harmony vector selected at the updating step of HS algorithm, $x'_{\text{new}}(j)$ is the mutated new harmony vector, and r is a random number created within $[0, 1]$.

The optimization procedure of DHSPM is as follows.

Step 1. Set the parameters HMS, bw, and NI.

Step 2. Initialize the HM and calculate the objective function of each harmony vector.

[17], polynomial mutation [18], and so forth. The experimental evaluation of the HS algorithm using different mutation methods using a well-known set of test functions shows that using the polynomial mutation improves the performance of the algorithm significantly for a considerable number of

TABLE 3: Units data for 3 thermal units' system.

Generator	P_{min} (MW)	P_{max} (MW)	a	b	c	e	f
1	100	600	0.001562	7.92	561	300	0.0315
2	50	200	0.00482	7.97	78	150	0.063
3	100	400	0.00194	7.85	310	200	0.042

TABLE 4: 3 thermal units' test results.

Generator	Power (MW)
1	300.12
2	149.88
3	400

TABLE 5: Best result comparison with different algorithms for 3 thermal units' system.

Algorithms	3 thermal units ($) Best result	3 thermal units ($) Average result
CEP [21]	8234.07	8235.97
FEP [21]	8234.07	8234.24
MFEP [21]	8234.08	8234.71
IFEP [21]	8234.07	8234.16
EGA [21]	8234.07	8234.41
FIA [21]	8234.07	8234.26
SPSO [21]	8234.07	8234.18
QPSO [21]	8234.07	8234.10
HS	8234.07	8234.15
IHS	8234.07	8234.13
DHSPM	**8234.07**	**8234.09**

Step 3. Determine new harmony vector $x_{new}(j)$ as follows:

$$\text{for } (j = 1 : N), \text{ do}$$

$$\text{HMCR} = \begin{cases} C_1 * t + 0.9 & 0 \le t \le \dfrac{NI}{2} \\ -C_1 * t + 1.1 & \dfrac{NI}{2} \le t \le NI, \end{cases}$$

$$\text{PAR} = \begin{cases} C_2 * t + 0.3 & 0 \le t \le \dfrac{NI}{2} \\ -C_2 * t + 1.1 & \dfrac{NI}{2} \le t \le NI, \end{cases}$$

$$C_1 = \frac{(1 - 0.9)}{NI/2},$$

$$C_2 = \frac{(0.7 - 0.3)}{NI/2}$$

$$\text{if } (r_1 < \text{HMCR}) \text{ then}$$

$$x_{new}(j) = x_a(j) \quad a \in (1, 2, \ldots, \text{HMS}) \tag{15}$$

$$\text{if } (r_2 < \text{PAR}) \text{ then}$$

$$x_{new}(j) = x_{new}(j) \pm \text{bw} \quad r_1, r_2 \in (0, 1) \tag{16}$$

endif

else

$$x_{new}(j) = x_{lower,j} + r_3 \times (x_{upper,j} - x_{lower,j})$$

$$r_3 \in (0, 1) \tag{17}$$

endif

endfor

Step 4 (updating of HM). If the new harmony vector $x_{new}(j)$ is better than the worst harmony in HM, the mutated new harmony vector $x'_{new}(j)$ is created as follows:

$$x'_{new}(j)$$
$$= \begin{cases} x_{new}(j) + \delta_1 (x_{new}(j) - x_{lower,j}) & \text{for } r \le 0.5 \\ x_{new}(j) + \delta_2 (x_{upper,j} - x_{new}(j)) & \text{for } r > 0.5, \end{cases} \tag{18}$$

$$\delta_1 = (2r)^{1/(1+\eta_m)} - 1 \quad \text{for } r \le 0.5,$$

$$\delta_2 = 1 - (2(1 - r))^{1/(1+\eta_m)} \quad \text{for } r > 0.5.$$

The mutated new harmony vector $x'_{new}(j)$ is inserted into the HM, and the worst harmony is removed from the HM.

Step 5. If maximum number of improvisations (NI) is reached, the computation is terminated. Otherwise, Steps 3 and 4 are repeated.

3.3.3. Study on Benchmark Functions. To test the performance of the proposed DHSPM algorithm, an extensive experimental evaluation is provided based on a set of 7 global optimization problems as follows.

Sphere Function (f_1). Sphere function is defined as

$$\min f(x) = \sum_{i=1}^{n} x^2(i), \tag{19}$$

where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-100 \le x(i) \le 100$.

Schwefel's Problem 2.22 (f_2). Schwefel's problem 2.22 is defined as

$$\min f(x) = \sum_{i=1}^{n} |x(i)| + \prod_{i=1}^{n} |x(i)|, \tag{20}$$

TABLE 6: Units data for 13 thermal units' system.

Generator	$P_{\min}$ (MW)	$P_{\max}$ (MW)	a	b	c	e	f
1	0	680	0.00028	8.1	550	300	0.035
2	0	360	0.00056	8.1	309	200	0.042
3	0	360	0.00056	8.1	307	150	0.042
4	60	180	0.00324	7.74	240	150	0.063
5	60	180	0.00324	7.74	240	150	0.063
6	60	180	0.00324	7.74	240	150	0.063
7	60	180	0.00324	7.74	240	150	0.063
8	60	180	0.00324	7.74	240	150	0.063
9	60	180	0.00324	7.74	240	150	0.063
10	40	120	0.00284	8.6	126	100	0.084
11	40	120	0.00284	8.6	126	100	0.084
12	55	120	0.00284	8.6	126	100	0.084
13	55	120	0.00284	8.6	126	100	0.084

TABLE 7: 13 thermal units' test results.

Generator	Power (MW)
1	628.3205
2	149.6024
3	222.7751
4	109.8655
5	109.8620
6	109.8582
7	60.0008
8	109.8614
9	109.8663
10	39.9997
11	39.9877
12	55.0001
13	55.0003

TABLE 8: Best result comparison with different algorithms for 13 thermal units' system.

Algorithms	13 thermal units ($) Best result	13 thermal units ($) Average result
CEP [21]	18048.21	18190.32
FEP [21]	18018.00	18200.79
MFEP [21]	18028.09	18192.00
IFEP [21]	17994.07	18127.06
EGA [21]	18019.15	18144.95
FIA [21]	18014.61	18136.97
SPSO [21]	17988.15	18102.48
QPSO [21]	17969.01	18075.11
DEC-SQP [7]	17963.94	NR
STHDE [22]	17963.89	NR
IGA_MU [23]	17963.98	NR
SDE [24]	17960.71	NR
HS	17963.91	18065.12
IHS	17962.87	18042.14
DHSPM	**17960.54**	**17994.16**

NR: not reported.

where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-10 \leq x(i) \leq 10$.

Rosenbrock Function (f_3). Rosenbrock function is defined as

$$\min f(x) = \sum_{i=1}^{n-1} \left(100 \left(x(i+1) - x^2(i) \right)^2 + (x(i)-1)^2 \right), \tag{21}$$

where global optimum $x^* = (1, 1, \ldots, 1)$ and $f(x^*) = 0$ for $-30 \leq x(i) \leq 30$.

Step Function (f_4). Step function is defined as

$$\min f(x) = \sum_{i=1}^{n} \left(|x(i) + 0.5| \right)^2, \tag{22}$$

where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-100 \leq x(i) \leq 100$.

Rotated Hyperellipsoid Function (f_5). Rotated hyperellipsoid function is defined as

$$\min f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x(j) \right)^2, \tag{23}$$

where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-100 \leq x(i) \leq 100$.

TABLE 9: Units data for 40 thermal units' system.

Generator	P_{min} (MW)	P_{max} (MW)	a	b	c	e	f
1	36	114	0.0069	6.73	94.705	100	0.084
2	36	114	0.0069	6.73	94.705	100	0.084
3	60	120	0.02028	7.07	309.54	100	0.084
4	80	190	0.00942	8.18	369.03	150	0.063
5	47	97	0.0114	5.35	148.89	120	0.077
6	68	140	0.01142	8.05	222.33	100	0.084
7	110	300	0.00357	8.03	287.71	200	0.042
8	135	300	0.00492	6.99	391.98	200	0.042
9	135	300	0.00573	6.6	455.76	200	0.042
10	130	300	0.00605	12.9	722.82	200	0.042
11	94	375	0.00515	12.9	635.2	200	0.042
12	94	375	0.00569	12.8	654.69	200	0.042
13	125	500	0.00421	12.5	913.4	300	0.035
14	125	500	0.00752	8.84	1760.4	300	0.035
15	125	500	0.00708	9.15	1728.3	300	0.035
16	125	500	0.00708	9.15	1728.3	300	0.035
17	220	500	0.00313	7.97	647.85	300	0.035
18	220	500	0.00313	7.95	649.69	300	0.035
19	242	550	0.00313	7.97	647.83	300	0.035
20	242	550	0.00313	7.97	647.81	300	0.035
21	254	550	0.00298	6.63	785.96	300	0.035
22	254	550	0.00298	6.63	785.96	300	0.035
23	254	550	0.00284	6.66	794.53	300	0.035
24	254	550	0.00284	6.66	794.53	300	0.035
25	254	550	0.00277	7.1	801.32	300	0.035
26	254	550	0.00277	7.1	801.32	300	0.035
27	10	150	0.52124	3.33	1055.1	120	0.077
28	10	150	0.52124	3.33	1055.1	120	0.077
29	10	150	0.52124	3.33	1055.1	120	0.077
30	47	97	0.0114	5.35	148.89	120	0.077
31	60	190	0.0016	6.43	222.92	150	0.063
32	60	190	0.0016	6.43	222.92	150	0.063
33	60	190	0.0016	6.43	222.92	150	0.063
34	90	200	0.0001	8.95	107.87	200	0.042
35	90	200	0.0001	8.62	116.58	200	0.042
36	90	200	0.0001	8.62	116.58	200	0.042
37	25	110	0.0161	5.88	307.45	80	0.098
38	25	110	0.0161	5.88	307.45	80	0.098
39	25	110	0.0161	5.88	307.45	80	0.098
40	242	550	0.00313	7.97	647.83	300	0.035

Schwefel's Problem 2.26 (f_6). Schwefel's problem 2.26 is defined as

$$\min f(x) = 418.9829n - \sum_{i=1}^{n} \left(x(i) \sin \left(\sqrt{|x(i)|} \right) \right), \quad (24)$$

where global optimum $x^* = (420.9687, 420.9687, \ldots, 420.9687)$ and $f(x^*) = 0$ for $-500 \leq x(i) \leq 500$.

Rastrigin Function (f_7). Rastrigin function is defined as

$$\min f(x) = \sum_{i=1}^{n} \left(x^2(i) - 10 \cos(2\pi x(i)) + 10 \right), \quad (25)$$

where global optimum $x^* = 0$ and $f(x^*) = 0$ for $-5.12 \leq x(i) \leq 5.12$.

The parameters of DHSPM are HM = 5, bw = 0.01, and $\eta_m = 10$. Thirty independent replications are carried out for

each function and the number of improvisations (NI) for each run is set to 10000. The average and standard deviations (SD) generated by the three algorithms (HS, IHS, and DHSPM) with dimension 30 are reported in Table 1. It can be seen from Table 1 that DHSPM generates best results for 7 functions compared to HS and IHS algorithms.

4. Application of DHSPM for ELD Problem

In this section, the DHSPM algorithm is applied for economic load dispatch problem with valve-point effect. The main steps symbolizing the search procedure are given below.

Step 1. Specify the generator cost coefficients and valve-point coefficients, choose the number of generator units (N), and specify maximum and minimum real power generation capacity of all generators and load demand P_D. Initialize the parameters of DHSPM.

Step 2. Initialize HM matrix with size HMS × N, where N represents the total number of generators in the system.

Step 3. Calculate the fitness value for each harmony vector in the HM using (6). The penalty factors are chosen high enough to make constraint violations prohibitive in the final solution. Use penalty coefficient $\gamma = 1000$.

Step 4. Calculate HMCR and PAR using (10).

Step 5. Generate new harmony vector using random selection, memory consideration, and pitch adjustment.

Step 6. If the new harmony vector is better than the worst harmony in the HM, then calculate the mutated new harmony using (12) and include it in HM.

Step 7. If the maximum number of improvisations is reached, go to Step 8; otherwise, repeat Steps 4–6.

Step 8. Print the optimal value of real power generation of generators and total cost of generation.

5. Simulation and Results

In this section, the DHSPM algorithm was tested with three standard load dispatch problems (3, 13, and 40 thermal units). The software was written in MATLAB 2009b and applied on a 2.40 GHz Intel Core i5 CPU personal computer with 4 GB RAM. The parameters of DHSPM for all the test systems are given in Table 2

5.1. 3 Thermal Units. A system of three thermal units with the valve-point loading was considered in this test. In this case, the load demand is taken as P_D = 850 MW. The test system comprises three generating units with quadratic cost functions together with the effects of valve-point loadings as given in Table 3. Based on data attained through 100 trials, the comparisons of the three thermal units test by different algorithm are represented in Tables 4 and 5, which show that

TABLE 10: 40 thermal units' test results.

Generator	Power (MW)
1	110.802
2	110.800
3	97.400
4	179.733
5	87.800
6	140.000
7	259.600
8	284.601
9	284.599
10	130.000
11	94.000
12	94.001
13	214.760
14	394.279
15	394.279
16	394.278
17	489.280
18	489.279
19	511.279
20	511.279
21	523.280
22	523.279
23	523.278
24	523.279
25	523.279
26	523.280
27	10.000
28	10.000
29	10.000
30	87.800
31	189.999
32	190.000
33	190.000
34	164.800
35	199.996
36	194.401
37	110.000
38	110.000
39	110.000
40	511.279

the DHSPM also succeeded in finding the satisfactory solution. The fuel cost convergence of harmony search algorithms for 3 thermal units is shown in Figure 3.

5.2. 13 Thermal Units. A system of 13 generating units with the valve-point loadings is given in Table 6. In this case, the load demand is considered as P_D = 1800 MW.
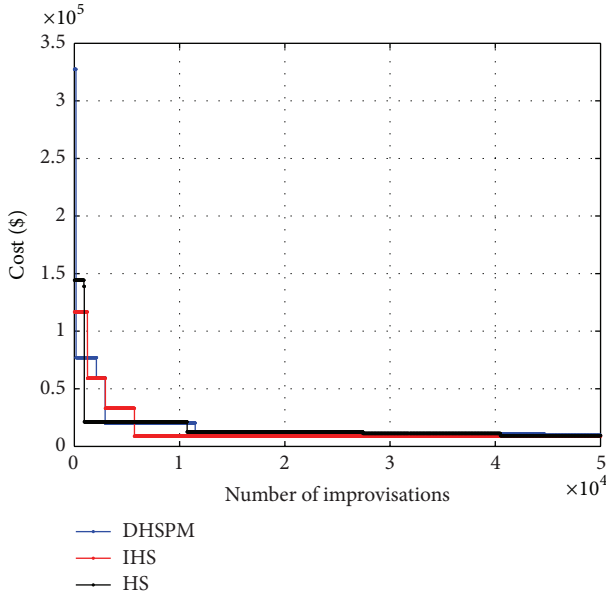
Figure 3: Fuel cost convergence nature of DHSPM, IHS, and HS for 3 thermal units' system.
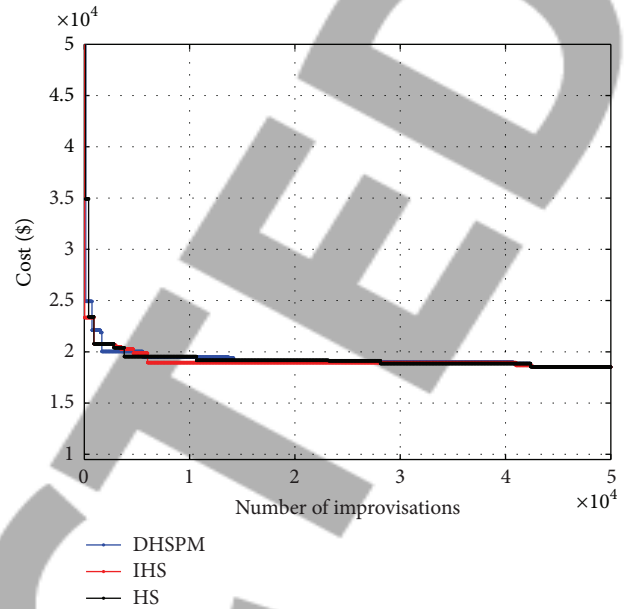

Figure 4: Fuel cost convergence nature of DHSPM, IHS, and HS for 13 thermal units' system.

The optimal real power generation outputs of 13 generators using DHSPM after conducting 100 trails is given in Table 7. The comparisons of the 13 thermal units test by different methods with DHSPM obtained through 100 trials are given in Table 8. From Table 8, it can be observed that the DHSPM algorithm succeeds in finding a satisfactory solution. Table 8 also summarizes the minimum and average cost obtained by all algorithms. From Table 8, the minimum cost obtained by DHSPM is 17960.54 \$/h which is the best cost found so far. The fuel cost convergence of harmony search algorithms for 13 thermal units is shown in Figure 4.

5.3. 40 Thermal Units. In order to test DHSPM algorithm deeper, a system of 40 thermal units with the effects of valve-point loading was considered in this test. The data of 40 thermal units with valve-point loading effect is given in Table 9. In this case, the load demand is equal to $P_D = 10500$ MW. The optimal value of real power generation by DHSPM after conducting 100 trails is depicted in Table 10. The results obtained by applying the DHSPM algorithm and other algorithms are summarized in Table 11. From Table 11, it can be concluded that the DHSPM method succeeds in finding a satisfactory solution. Table 11 also summarizes the minimum and average cost obtained by other settled algorithms. From Table 11, the minimum cost obtained by DHSPM is 121412.66 \$/h which is the best cost when compared with other existing algorithms. The fuel cost convergence of harmony search algorithms for 40 thermal units is shown in Figure 5.

Figure 5: Fuel cost convergence nature of DHSPM, IHS, and HS for 40 thermal units' system.

6. Conclusion

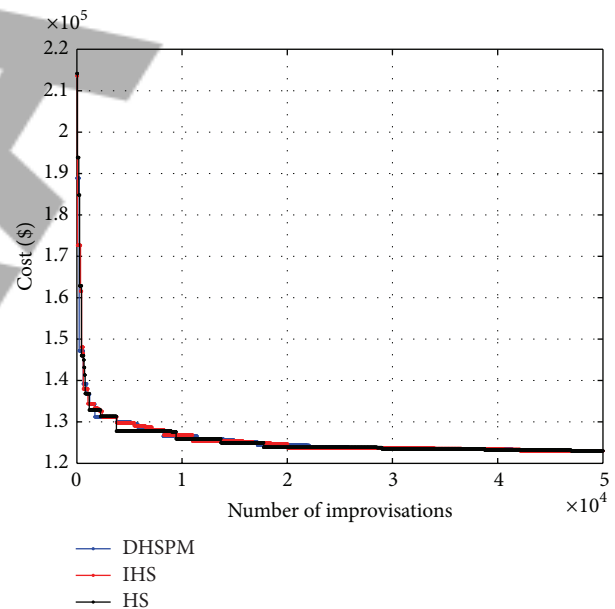In this paper, dynamic harmony search with polynomial mutation algorithm is applied to solve economic load dispatch problem with valve-point loading effect. The feasibility and the effectiveness of DHSPM algorithm have been investigated on three test systems having 3, 13, and 40 units. The DHSPM algorithm achieves the minimum fuel cost for the above unit cases when compared with other optimization methods reported in the literature. The successful optimizing performance on the validation data sets proves the efficiency

Table 11: Best result comparison with different algorithms for 40 thermal units' system.

Algorithms	40 thermal units ($) Best result	40 thermal units ($) Average result
CEP [21]	123488.29	124793.48
FEP [21]	122679.71	124119.37
MFEP [21]	122647.57	123489.74
IFEP [21]	122624.35	123382.00
EGA [21]	122022.96	122942.66
FIA [21]	121823.80	122662.48
SPSO [21]	121787.39	122474.40
QPSO [21]	121448.21	122225.07
DEC-SQP [7]	121741.97	122295.12
STHDE [22]	121698.51	122304.30
SDE [23]	121412.78	121412.54
HS	121438.24	122221.14
IHS	121435.16	122219.28
DHSPM	**121412.66**	**121423.57**

of the DHSPM algorithm and shows that it can be used as a reliable tool for economic load dispatch problem.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] M. E. El-Hawary and G. S. Christensen, *Optimal Economic Operation of Electric Power System*, vol. 142, Academic Press, New York, NY, USA, 1979.

[2] S. Granville, "Optimal reactive dispatch through interior point methods," *IEEE Transactions on Power Systems*, vol. 9, no. 1, pp. 136–146, 1994.

[3] Z.-X. Liang and J. D. Glover, "A zoom feature for a dynamic programming solution to economic dispatch including transmission losses," *IEEE Transactions on Power Systems*, vol. 7, no. 2, pp. 544–550, 1992.

[4] C.-L. Chen and S.-C. Wang, "Branch-and-bound scheduling for thermal generating units," *IEEE Transactions on Energy Conversion*, vol. 8, no. 2, pp. 184–189, 1993.

[5] D. C. Walters and G. B. Sheble, "Genetic algorithm solution of economic dispatch with value point loading," *IEEE Transactions on Power Systems*, vol. 8, no. 3, pp. 1325–1332, 1993.

[6] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 1, pp. 83–94, 2003.

[7] L. dos Santos Coelho and V. C. Mariani, "Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect," *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 989–996, 2006.

[8] J.-B. Park, K.-S. Lee, J.-R. Shin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with non-smooth cost functions," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 34–42, 2005.

[9] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.

[10] S. O. Degertekin, "Optimum design of steel frames using harmony search algorithm," *Structural and Multidisciplinary Optimization*, vol. 36, no. 4, pp. 393–401, 2008.

[11] Z. W. Geem, "Optimal cost design of water distribution networks using harmony search," *Engineering Optimization*, vol. 38, no. 3, pp. 259–280, 2006.

[12] Z. W. Geem, K. S. Lee, and Y. Park, "Application of harmony search to vehicle routing," *American Journal of Applied Sciences*, vol. 2, no. 12, pp. 1552–1557, 2005.

[13] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.

[14] M. G. H. Omran and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643–656, 2008.

[15] M. Khalili, R. Kharrat, K. Salahshoor, and M. H. Sefat, "Global dynamic harmony search algorithm: GDHS," *Applied Mathematics and Computation*, vol. 228, pp. 195–219, 2014.

[16] Q.-K. Pan, P. N. Suganthan, M. F. Tasgetiren, and J. J. Liang, "A self-adaptive global best harmony search algorithm for continuous optimization problems," *Applied Mathematics and Computation*, vol. 216, no. 3, pp. 830–848, 2010.

[17] K. Deep and M. Thakur, "A new mutation operator for real coded genetic algorithms," *Applied Mathematics and Computation*, vol. 193, no. 1, pp. 211–230, 2007.

[18] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.

[19] Z. Michalewicz, T. Logan, and S. Swaminathan, "Evolutionary operators for continuous convex parameter space," in *Proceedings of the 3rd Annual Conference on Evolutionary Programming*, San Diego, Calif, USA, 1994.

[20] B. H. F. Hasan, I. A. Doush, E. Al Maghayreh, F. Alkhateeb, and M. Hamdan, "Hybridizing harmony search algorithm with different mutation operators for continuous problems," *Applied Mathematics and Computation*, vol. 232, pp. 1166–1182, 2014.

[21] K. Meng, H. G. Wang, Z. Y. Dong, and K. P. Wong, "Quantum-inspired particle swarm optimization for valve-point economic load dispatch," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 215–222, 2010.

[22] S.-K. Wang, J.-P. Chiou, and C.-W. Liu, "Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm," *IET Generation, Transmission and Distribution*, vol. 1, no. 5, pp. 793–803, 2007.

[23] C.-L. Chiang, "Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels," *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1690–1699, 2005.

[24] A. Srinivasa Reddy and K. Vaisakh, "Shuffled differential evolution for large scale economic dispatch," *Electric Power Systems Research*, vol. 96, pp. 237–245, 2013.