Energy 35 (2010) 5031–5036

Contents lists available at ScienceDirect

Energy

journal homepage: www.elsevier.com/locate/energy

Hybrid SOA–SQP algorithm for dynamic economic dispatch with valve-point effects

S. Sivasubramani, K.S. Swarup*

Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai 600036, India

ARTICLE INFO

Article history:
Received 9 February 2010
Received in revised form
15 July 2010
Accepted 14 August 2010

Keywords:
Seeker optimization algorithm
Sequential quadratic programming
Dynamic economic dispatch
Valve-point effects

ABSTRACT

This paper proposes a hybrid technique combining a new heuristic algorithm named seeker optimization algorithm (SOA) and sequential quadratic programming (SQP) method for solving dynamic economic dispatch problem with valve-point effects. The SOA is based on the concept of simulating the act of human searching, where the search direction is based on the empirical gradient (EG) by evaluating the response to the position changes and the step length is based on uncertainty reasoning by using a simple fuzzy rule. In this paper, SOA is used as a base level search, which can give a good direction to the optimal global region and SQP as a local search to fine tune the solution obtained from SOA. Thus SQP guides SOA to find optimal or near optimal solution in the complex search space. Two test systems i.e., 5 unit with losses and 10 unit without losses, have been taken to validate the efficiency of the proposed hybrid method. Simulation results clearly show that the proposed method outperforms the existing method in terms of solution quality.

1. Introduction

Dynamic economic dispatch (DED) is one of the important power system optimization problems which is a non-linear and complicated dynamic optimization problem. DED is a method to dispatch the generating units to the predicted load demands over a certain period of time at minimum operating cost while satisfying equality, inequality and ramp-rate limit constraints. Normally, DED is solved by considering the cost function as monotonically increasing one. However, the cost function is non-convex and non-smooth due to the effects of valve-point loading. This will make the problem harder in finding the optimum solution. Many mathematical techniques have been addressed to solve the DED problem with valve-point effects [1–4]. However, none of these methods may be able to provide an optimal solution. They usually get struck at local optimum because of non-linear and non-convex characteristics of the generating units.

Recently, stochastic optimization techniques such as genetic algorithm (GA) [5], evolutionary programming (EP) [6], simulated annealing (SA) [7], particle swarm optimization (PSO) [8] and differential evolution (DE) [9] have been used to solve both static and dynamic economic dispatch problem with valve-point effects. They are found to be effective in solving the problem without any restriction on the shape of the cost curve due to their ability to find global optimal solution. Though, these stochastic methods do not always guarantee the global solution, they generally provide a reasonable solution which is suboptimal. However, the main drawback of the above methods is premature convergence. To overcome the deficiencies in stochastic methods, many strategies have been used such as adaptive particle swarm optimization (APSO) [10], improved particle swarm optimization (IPSO) [11], modified differential evolution (MDE) [12] and hybrid differential evolution (HDE) [13] to address the DED problem with valve-point effects. More precisely, hybrid algorithm combining stochastic and deterministic methods is found to be effective in solving optimization problems with complex, non-linear and non-convex characteristics. Based on this, hybrid algorithm combining evolutionary programming (EP) and sequential quadratic programming (SQP) [14] and PSO with SQP [15] have been reported to address DED problem with valve-point effects.

Seeker optimization algorithm (SOA) proposed by Dai and Chen [16], is a new population based heuristic search algorithm which uses the act of human searching for solving optimization problem. This algorithm has been applied to power system optimization problem such as optimal reactive power dispatch [17,18] which is a mixed integer and highly non-linear problem. The SOA algorithm has been successfully applied and proved to

* Corresponding author. Tel.: +91 44 22574440; fax: +91 44 22574402.
E-mail address: sivasubramani.shanmugavelu@gmail.com (S. Sivasubramani).

0360-5442/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.energy.2010.08.018

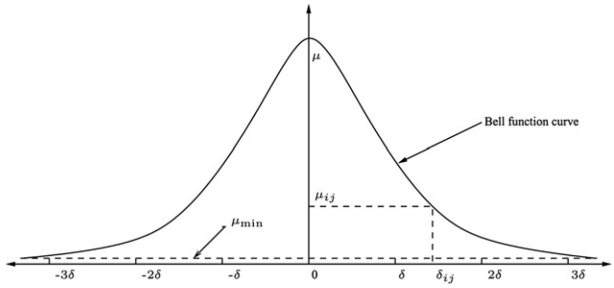

Fig. 1. The act of fuzzy reasoning.

be the best among the existing methods. In this paper SOA and SQP are combined to form a hybrid SOA–SQP for solving dynamic economic dispatch problem with valve-point effects. While combining stochastic and deterministic methods to form a hybrid algorithm, generally stochastic methods are used as a base level search and deterministic methods as a local level search. In this way, SOA is used a base level search and SQP as a local search to fine tune the solution to reach global optimum or near global optimum. Finally, the proposed hybrid method is applied to two test systems namely 5 unit and 10 unit test systems. Simulation results are compared with existing methods reported in literatures and it is shown that proposed hybrid method is giving higher quality solutions.

2. Problem formulation

The objective of the DED problem is to dispatch the generating units to the predicated load demands over a certain period of time at minimum operating cost while satisfying the various constraints. The problem is formulated as follows

$$\text{Min } F_T = \sum_{t=1}^{T}\sum_{i=1}^{N} F_{i,t}(P_{i,t}) \tag{1}$$

where F is the total operating cost over the whole dispatch period, N is the number of committed generating units, T is the number of the scheduled intervals in the time horizon and $F_{i,t}(P_{i,t})$ is the fuel cost in terms of real power output P_i at time t . The fuel cost function of ith unit with valve-point effects is represented as follows;

$$F_i(P_i) = a_iP_i^2 + b_iP_i + c_i + \left|e_i \times \sin\left(f_i\left(P_{i,\min} - P_i\right)\right)\right| \tag{2}$$

where a_i, b_i, c_i, d_i, e_i and f_i are the fuel cost coefficients of ith unit with valve-point effects and P_i is the power output of ith unit in megawatts.

Subject to the following equality and inequality constraints for tth interval in the time horizon.

Table 1
Data for 5 unit system.

Quantities	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5
a (\$/(MW)^2 h)	0.0080	0.0030	0.0012	0.0010	0.0015
b (\$/(MW) h)	2.0	1.8	2.1	2.0	1.8
c (\$/h)	25	60	100	120	40
e (\$/h)	100	140	160	180	200
f (rad/MW)	0.042	0.040	0.038	0.037	0.035
$P_{\min}$ (MW)	10	20	30	40	50
$P_{\max}$ (MW)	75	125	175	250	300
UR (MW/h)	30	30	40	50	50
DR (MW/h)	30	30	40	50	50

Table 2
Load demand for 24 h for 5 unit system.

Time (h)	Load (MW)	Time (h)	Load (MW)
1	410	13	704
2	435	14	690
3	475	15	654
4	530	16	580
5	558	17	558
6	608	18	608
7	626	19	654
8	654	20	704
9	690	21	680
10	704	22	605
11	720	23	527
12	740	24	463

1. Real power balance constraint

$$\sum_{i=1}^{N}(P_{i,t}) - P_{D,t} - P_{L,t} = 0 \tag{3}$$

where $P_{D,t}$ is the total load demand and $P_{L,t}$ is the loss at tth interval in megawatts.

2. Real power operating limits

$$P_{i\min} \le P_i \le P_{i\max}, \quad i = 1, 2, \cdots, N \tag{4}$$

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

3. Generating unit ramp-rate limits

$$P_{i,t} - P_{i,(t-1)} \le UR_i, i = 1, 2, \cdots, N \tag{5}$$

$$P_{i,(t-1)} - P_{i,t} \le DR_i, i = 1, 2, \cdots, N \tag{6}$$

where UR_i and DR_i are ramp-up and ramp-down rate limits of ith unit, respectively and are expressed in MW/h.

The general form of the loss formula using B coefficients is

$$P_{L,t} = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{it}B_{ij}P_{jt} \tag{7}$$

Table 3
Best solution for 5 unit system.

Hour	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5
1	17.0041	98.5398	30.0000	124.9078	139.7597
2	42.0170	98.5398	30.0000	124.9079	139.7597
3	75.0000	99.9744	30.1290	40.0000	230.1417
4	47.3063	98.5398	30.0000	124.9079	229.5195
5	74.9892	98.5375	30.2424	124.9288	229.5899
6	42.6730	98.5398	112.6734	124.9079	229.5195
7	63.7432	96.2006	112.7710	127.1096	226.4985
8	75.0000	102.7170	122.1910	124.9099	229.5195
9	39.8075	98.5398	112.6734	209.8158	229.5195
10	53.8147	98.5398	112.6734	209.8158	229.5195
11	69.8230	98.5398	112.6734	209.8158	229.5195
12	75.0000	113.3730	112.6734	209.8158	229.5195
13	53.8147	98.5398	112.6734	209.8158	229.5195
14	39.8075	98.5398	112.6734	209.8158	229.5195
15	10.0000	92.3287	112.6734	209.8158	229.5195
16	14.6586	98.5398	112.6734	124.9079	229.5195
17	68.9514	20.0000	30.0001	209.8168	229.5195
18	40.4386	98.5398	30.0000	209.8158	229.5195
19	10.0000	92.3291	112.6734	209.8156	229.5192
20	53.8147	98.5398	112.6734	209.8158	229.5195
21	29.8023	98.5398	112.6734	209.8158	229.5195
22	44.5234	98.5398	112.6734	209.8158	139.7597
23	51.3910	98.5398	112.6734	124.9079	139.7597
24	70.0315	98.5398	30.0000	124.9079	139.7597

Table 4
Convergence results (50 runs) of 5 unit system.

Method	Min.cost ($)	Mean cost ($)	Max. cost ($)
SOA	42,588.4156	43,273.4473	43,808.0937
SOA−SQP	40,701.4194	412,809.1608	42,133.7055

where P_{it}, P_{jt} are real power of ith and jth unit respectively at time t and B_{ij} are the loss coefficients.

3. Seeker optimization algorithm

Seeker optimization algorithm is a new heuristic optimization algorithm proposed by Dai and Chen [16]. SOA operates on a set of potential solutions called swarm. The individual in the swarm is called as seeker. A neighborhood is defined for each seeker in order to have social component for social sharing of information. In this simulation, the entire swarm is divided into $K = 3$ subpopulations, each has same size. The seekers in the same subpopulation constitute a neighborhood.

3.1. Implementation of seeker optimization algorithm

In SOA, a search direction $d_{ij}(t)$ and a step length $\alpha_{ij}(t)$ are calculated separately for each seeker i on each dimension j for each time step t, where $\alpha_{ij}(t) \geq 1$ and $d_{ij}(t) \in \{1, 0, -1\}$. $d_{ij}(t) = 1$ means ith seeker goes towards positive direction on dimension j, $d_{ij}(t) = -1$ means ith seeker goes towards negative direction on dimension j and $d_{ij}(t) = 0$ means ith seeker stays at the current position. For each seeker i (i = 1, 2,…, S, S is the population size), the position update on dimension j (j = 1, 2,…, D, D is the control variable size) is given by

$$x_{ij}(t+1) = x_{ij}(t) + \alpha_{ij}(t) d_{ij}(t) \tag{8}$$

Since the subpopulations are searching by using their own information, they are easy to converge to a local optimum. In order to avoid this situation, the positions of worst K − 1 seekers of each subpopulation are combined with best one in each of the other K − 1 subpopulations using the following binomial crossover operator.

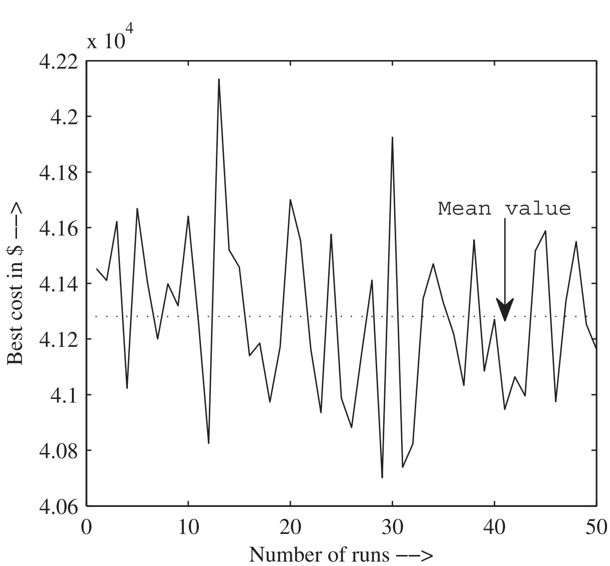

Fig. 2. Distribution of best costs for 5 unit system.

$$x_{k_n j,\text{worst}} = \begin{cases} x_{lj,\text{best}}, & \text{if } R_j \leq 0.5 \\ x_{k_n j,\text{worst}}, & \text{else} \end{cases} \tag{9}$$

where R_j is a uniform random real number between [0,1]. $x_{k_n j,\text{worst}}$ is denoted as nth worst position in j th dimension of kth subpopulation. $x_{lj,\text{best}}$ is the best one in lth subpopulation on dimension j with n, k, l = 1, 2,…, $k - 1$ and $k \neq l$. In this way, good information obtained by each subpopulation is exchanged among the subpopulations and then the diversity of population is increased.

3.2. Search direction

The search space may be viewed as a gradient field and a so called empirical gradient (EG) can be determined by evaluating the response to the position change especially when the objective function is not in differential form, and the seeker can follow EG to guide his search. Since SOA does not depend on the magnitude of the EG, search direction can be determined only by the signum function of a better position minus a worse position [16].

In SOA, each seeker i finds his search direction using the EGs in egotistic behavior, altruistic behavior and pro-activeness behavior. The search direction for seeker i in egotistic behavior is calculated from the $p_{\text{best}i}$ as

$$\vec{d}_{i,\text{ego}}(t) = sign(\vec{p}_{\text{best } i}(t) - \vec{x}_i(t)) \tag{10}$$

On the other hand, the search direction in altruistic behavior for seeker i is calculated by using neighborhood's historical best $g_{\text{best}}(t)$ and neighborhood's current best $l_{\text{best}}(t)$. Hence, each seeker i is associated with two optional altruistic direction, i.e., $d_{i,\text{alt}_1}(t)$ and $d_{i,\text{alt}_2}(t)$:

$$\vec{d}_{i,\text{alt}_1}(t) = sign(\vec{g}_{\text{best}}(t) - \vec{x}_i(t)) \tag{11}$$

$$\vec{d}_{i,\text{alt}_2}(t) = sign(\vec{l}_{\text{best}}(t) - \vec{x}_i(t)) \tag{12}$$

Finally, the search direction for seeker i in pro-activeness behavior is calculated from his past behavior as

$$\vec{d}_{i,\text{pro}}(t) = sign(\vec{x}_i(t_1) - \vec{x}_i(t_2)) \tag{13}$$

where $t_1, t_2 \in \{t, t-1, t-2\}$, and $\vec{x}_i(t_1)$ is better than $\vec{x}_i(t_2)$. According to, human rational judgment, the actual search direction of the ith seeker $\vec{d}_i(t)$, is based on a compromise among the aforementioned four empirical directions namely $\vec{d}_{i,\text{ego}}(t)$, $\vec{d}_{i,\text{alt}_1}(t)$, $\vec{d}_{i,\text{alt}_2}(t)$ and $\vec{d}_{i,\text{pro}}(t)$. In this study, every dimension j of $\vec{d}_i(t)$ is selected by applying the following proportional selection rule:

$$d_{ij} = \begin{cases} 0 & \text{if } r_j \leq p_j^{(0)} \\ +1 & \text{if } p_j^{(0)} \leq r_j \leq p_j^{(0)} + p_j^{(1)} \\ -1 & \text{if } p_j^{(0)} + p_j^{(1)} \leq r_j \leq 1 \end{cases} \tag{14}$$

where r_j is a uniform random number in [0,1], $p_j^{(m)} (m \in \{0, +1, -1\})$ is the percentage of the number of m from the set $\{d_{ij,\text{ego}}, d_{ij,\text{alt}_1}, d_{ij,\text{alt}_2}, d_{ij,\text{pro}}\}$ on each dimension j of all the four empirical directions, i.e., $p_j^{(m)}$ = the number of $m/4$.

3.3. Step length

In the continuous search space, there often exists a neighborhood region close to an extremum point. In this region, the fitness values of the input variables are proportional to their distances

Table 5
Data for 10 unit system.

Quantities	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10
a (\$/(MW)^2 h)	0.00043	0.00063	0.00039	0.00070	0.00079	0.00056	0.00211	0.00480	0.10908	0.00951
b (\$/(MW) h)	21.60	21.05	20.81	23.90	21.62	17.87	16.51	23.23	19.58	22.54
c (\$/h)	958.20	1313.6	604.97	471.60	480.29	601.75	502.70	639.40	455.60	692.40
e (\$/h)	450	600	320	260	280	310	300	340	270	380
f (rad/MW)	0.041	0.036	0.028	0.052	0.063	0.048	0.086	0.082	0.098	0.094
$P_{\min}$ (MW)	150	135	73	60	73	57	20	47	20	55
$P_{\max}$ (MW)	470	460	340	300	243	160	130	120	80	55
UR (MW/h)	80	80	80	50	50	50	30	30	30	30
DR (MW/h)	80	80	80	50	50	50	30	30	30	30

from the extremum point. It may be assumed that better points are likely to be found in neighborhood of families of good points and search should be intensified in regions containing good solutions through focusing search. Hence, from the standpoint on human searching, one may find the near optimal solutions in the narrow neighborhood of the point with lower fitness value and, contrariwise, wider neighborhood with higher fitness value.

Fuzzy system has been used in this study to determine the step length as in [17,18]. The simple control rule used here to calculate the step length is if {*fitness value is small*}(conditional part) then {*step length is short*} (action part). To design a fuzzy system to be applicable to a wide range of optimization problems, the fitness values of all the seekers are sorted in the descending order and turned into sequence numbers from 1 to S as the inputs of fuzzy reasoning. The linear membership function is used in the condition part and the expression is given as

$$\mu_i = \mu_{\max} - \frac{S - I_i}{S - 1}(\mu_{\max} - \mu_{\min}) \tag{15}$$

where I_i is the sequence number of $x_i(t)$ after sorting the fitness values, $\mu_{\max}$ is the maximum membership degree value which is equal to or a little less than 1.0.

In this study, bell membership function is used and shown in Fig. 1. For convenience, one dimension is considered. Since the membership degree values of input variables beyond $[-3\delta, 3\delta]$ are less than 0.011, therefore they are neglected. Hence minimum value $\mu_{\min} = 0.011$ is set. The parameter $\vec{\delta}$ of the bell membership function is determined as

$$\vec{\delta} = \omega \times \mathrm{abs}(\vec{x}_{\mathrm{best}} - \vec{x}_{\mathrm{rand}}) \tag{16}$$

Where abs(.) returns an output vector such that each element of the vector is the absolute value of the corresponding element of the input vector. The parameter ω is linearly decreased from $\omega_{\max}$ to $\omega_{\min}$ during run to reduce the step length with time increasing and hence improving the search precision. The $\vec{x}_{\mathrm{best}}$ and $\vec{x}_{\mathrm{rand}}$ are the best seeker and a randomly selected seeker from the same subpopulation respectively where ith seeker belongs. It is important that $\vec{x}_{\mathrm{rand}}$ is different from $\vec{x}_{\mathrm{best}}$ and $\vec{\delta}$ is shared by all the seekers in the same subpopulation.

To introduce the randomicity on each dimension and improve the local search capability, Eq. (17) is used to change μ_i to vector $\vec{\mu}_i$. Then the action part of the fuzzy reasoning gives every dimension j of step length by Eq. (18)

$$\mu_{ij} = \mathrm{rand}(\mu_i, 1) \tag{17}$$

$$\alpha_{ij} = \delta_j \sqrt{-\ln\left(\mu_{ij}\right)} \tag{18}$$

4. Sequential quadratic programming

The SQP method seems to be the best non-linear programming method for constrained optimization problems. It outperforms every other non-linear programming method in terms of efficiency, accuracy and percentage of successful solutions over a large number of test problems. The method closely resembles Newton's method for constrained optimization, just as is done for unconstrained optimization. At each iteration, an approximation is made of the Hessian of the Lagrangian function using Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton updating method. The result of the approximation is then used to generate a quadratic programming (QP) subproblem whose solution is used to form a search direction for a line search procedure. Since the objective function to be minimized is non-convex, SQP ensures a local minimum for an initial solution.

The SQP used in this paper consists of three main stages, as follows: 1) calculation of an approximation of the Hessian matrix of the Lagrangian function using quasi-Newton method; 2) formulation of the QP problem; 3) line search and merit function calculation. Hence in this paper, first SOA is applied to optimization problem as a global search and finally the best solution obtained from SOA is given as initial condition for SQP method as a local search to fine tune the solution. SQP simulations are done using the Matlab optimization toolbox.

Table 6
Load demand for 24 h for 10 unit system.

Time(h)	Load(MW)	Time(h)	Load(MW)
1	1036	13	2072
2	1110	14	1924
3	1258	15	1776
4	1406	16	1554
5	1480	17	1480
6	1628	18	1628
7	1702	19	1776
8	1776	20	2072
9	1924	21	1924
10	2072	22	1628
11	2146	23	1332
12	2220	24	1184

5. Implementation of hybrid SOA–SQP algorithm to DED problem

The dynamic economic dispatch problem based on hybrid algorithm is described as follows:

1. Input the system parameters consisting of fuel cost coefficients, transmission loss coefficients, lower and upper bound of control variables and predicted load demands for T intervals in the scheduled time horizon. In DED problem, real power generation values of all generating units for T periods are control variables.

Table 7
Best solution for 10 unit system.

Hour	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10
1	226.6239	135.0000	73.0000	60.0000	167.3568	122.4285	129.5906	47.0000	20.0000	55
2	226.6242	135.0000	91.7358	60.0000	222.5996	122.4498	129.5904	47.0000	20.0000	55
3	150.0000	135.0000	316.2641	60.0000	222.6108	122.5034	129.6216	47.0000	20.0000	55
4	226.6242	222.2665	202.5037	120.4152	222.5996	159.9999	129.5904	47.0000	20.0000	55
5	303.2513	222.2978	198.7553	120.4412	222.5959	122.7536	129.5880	85.3165	20.0000	55
6	303.2685	396.9875	194.1240	60.0000	222.6767	126.3892	129.5539	120.0000	20.0000	55
7	150.0000	460.0000	299.0324	120.4549	222.6020	160.0000	129.5956	85.3149	20.0000	55
8	379.8282	396.7764	295.2306	112.1503	222.5824	122.4319	93.0475	47.0000	51.9523	55
9	379.8726	309.5422	298.7844	230.8514	222.6155	159.9993	129.9633	85.3138	52.0571	55
10	456.4968	396.7993	310.4488	241.2457	222.5996	122.4498	129.5904	85.3121	52.0570	55
11	456.4968	396.7993	326.4566	300.0000	222.5996	160.0000	129.5904	47.0000	52.0570	55
12	456.6067	460.0000	298.8276	300.0000	222.6011	160.0000	129.5950	85.3123	52.0571	55
13	379.8726	396.7993	322.8257	300.0000	222.5996	160.0000	129.5904	85.3121	20.0000	55
14	378.5956	396.0497	279.6300	300.0000	122.5917	122.6721	129.4607	120.0000	20.0000	55
15	150.0000	395.9671	295.6221	249.9998	220.9067	138.4098	129.5392	87.0737	53.4813	55
16	150.0000	309.5296	328.5209	180.8352	172.7232	122.4731	129.6094	85.3082	20.0000	55
17	150.0000	222.2664	307.1271	180.8315	172.7331	122.4509	129.5907	120.0000	20.0000	55
18	226.6171	309.4914	279.2813	179.5827	222.5513	122.4298	93.0461	120.0000	20.0000	55
19	303.2496	309.5337	312.7390	180.8329	222.5991	122.4528	129.5927	120.0000	20.0000	55
20	456.4968	396.7994	304.9558	241.2457	222.5996	160.0000	129.5904	85.3121	20.0000	55
21	379.4145	397.4845	296.0587	181.2572	222.5308	122.4351	129.8188	120.0000	20.0000	55
22	226.6241	309.5329	326.6888	180.8305	172.7330	160.0000	129.5904	47.0000	20.0000	55
23	150.0000	135.0000	306.5035	60.0000	222.5274	133.3785	129.5904	120.0000	20.0000	55
24	150.0000	135.0000	185.5949	125.9676	173.1554	124.3571	129.6015	85.3191	20.0042	55

2. Create an initial population for N generating units and T intervals and initialize the position of each seeker in the search space uniformly and randomly. While generating an initial population, all the control variables should satisfy the inequality constraint (4).

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1T} \\ P_{21} & P_{22} & \cdots & P_{2T} \\ \vdots & \vdots & \ddots & \vdots \\ P_{N1} & P_{N2} & \cdots & P_{NT} \end{bmatrix} \tag{19}$$

It is noted that the real power balance equality constraints (3) should be satisfied when loss is neglected. To satisfy the equality constraints, the following procedure is used. Step 1) Set $t = 1$

Step 2) Set $l = 1$

Step 3) The dependant power generation P_{lt} is calculated as

$$P_{lt} = P_{Dt} \sum_{\substack{i=1 \\ i \neq l}}^{N} (P_{it}), \; t = 1, 2, \cdots, T \tag{20}$$

Step 4) If the dependent power generation P_{lt} satisfies the inequality constraint (4), go to Step 5; otherwise set $l = l + 1$, go to Step 3.

Step 5) $t = t+1$, if $t < T$, go to Step 2; otherwise, stop the procedure.

3. Start the iteration counter.
4. Determine P_L using B loss coefficients and evaluate the fitness value of each seeker using the following Eq. (21)

$$F^{*} = \sum_{t=1}^{T}\sum_{i=1}^{N} F_{it}(P_{it}) + \lambda \left(\sum_{t=1}^{T} \left(\sum_{i=1}^{N} P_{it} \right) - P_{Lt} - P_{Dt} \right)^{2} \tag{21}$$

where F^{*} is the augmented total production cost for whole dispatch period and λ is the penalty value. The initial historical best position among the population is achieved. The current best position is set to the personal historic best position.

5. Divide the entire population into K subpopulation and select the neighbors of each seeker.
6. Determine the search direction and step length for each seeker and update his current position.
7. Using subpopulation learn, the worst $K - 1$ seeker positions of each subpopulation are replaced with best one in other $K - 1$ subpopulation.
8. Determine PL and fitness value of new position, and update the historical best position among the population and the historical best position of each seeker.
9. Repeat the above procedure till the stopping condition is achieved or the maximum number of iteration is reached.
10. Input the best solution obtained from the above SOA steps as an initial condition to SQP to fine tune the solution.
11. Output the solution.

Table 8
Convergence results (50 runs) of 10 unit system.

Method	Min. cost ($)	Mean cost ($)	Max. cost ($)
SOA	1,023,945.6329	1,026,288.5263	1,029,212.9082
SOA–SQP	1,021,460.0101	1,023,840.6543	1,026,852.4248

6. Test systems and simulation results

In order to validate the effectiveness of the proposed hybrid method, two test systems (5 unit with losses and 10 unit without losses) of DED problems have been considered in which the objective functions are non-smooth because the valve-point effects

Table 9
Comparison of best costs for two test systems.

Optimization technique	5 unit system with losses	10 unit system without losses
SA [7]	47,356	–
EP [14]	–	1,048,638
Hybrid EP–SQP [14]	–	1,035,748
PSO [11]	–	1,036,506
APSO [10]	44678	–
PSO–SQP [15]	–	1,031,371
DE [13]	–	1,033,958
Hybrid DE [13]	–	1,031,077
IPSO [11]	–	1,023,807
SOA	42,588.41	1,023,945.63
Proposed method	40,701.4194	1,021,460.01

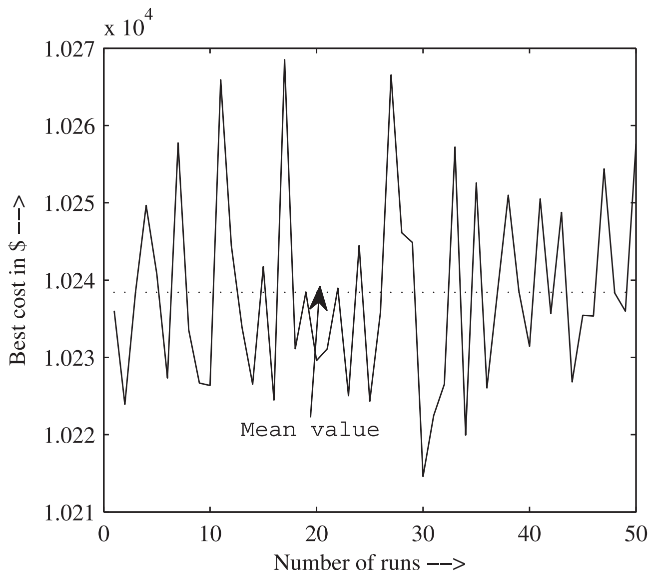

Fig. 3. Distribution of best costs for 10 unit system.

were taken into account. The optimization method SQP is done using Matlab Optimization Toolbox. The parameters used in the proposed SOA algorithm are: the population size $S = 60$, the number of generation $Gen = 500$, the number of subpopulations $K = 3$, the limits of membership degree value for Fuzzy reasoning, i.e., $\mu_{\max} = 0.95$ and $\mu_{\min} = 0.0111$ and the limits of ω, i.e., $\omega_{max} = 0.9$ and $\omega_{\min} = 0.1$.

6.1. Test system 1

This system consists of five thermal units [7]. The system data and load demands for 24 h are given in Tables 1 and 2. The *B* loss coefficient matrix is given in Appendix. This system has been solved by the proposed hybrid method and the best solution is given in Table 3. 50 independent runs were made to identify the best, mean and worst cost and are given in Table 4 for both SOA and hybrid SOA–SQP method. The best value achieved by SOA method is 42588.41$ and by the proposed hybrid method 40701.41$. The comparison of best costs of the proposed method with the methods reported in literature is given in Table 9. From the simulation results, the proposed hybrid method is giving better quality solution. The spread of best costs for 50 runs against its mean value is shown in Fig. 2

6.2. Test system 2

This system consists of ten thermal generators. The system data and load pattern for 24 h are given in Tables 5 and 6 and also available in [14]. In this test system, transmission losses are neglected. This test system is solved by the proposed hybrid method and the optimal results are given in Table 7. The best, mean and worst costs achieved by both SOA and the proposed hybrid method after 50 trials are listed in Table 8. The comparison of best costs of different methods with proposed method is given in Table 9. It is clear from the results, the proposed hybrid method is producing higher quality solution. The distribution of best costs for 50 runs against its mean value is shown in Fig. 3.

7. Conclusion

This paper presents a hybrid method combining SOA and SQP for solving DED problem with valve-point effects including generator ramp-rate limits. In this algorithm, SOA is used as a base level search and SQP as local level search. Hence SOA is first applied to DED problem to find the best solution. This best solution is given to SQP as an initial condition to fine tune the solution. To verify the effectiveness, two test cases one with losses and another one without losses were considered. From the simulation results of both the test cases, the proposed hybrid SOA–SQP method is giving higher quality solutions than the reported methods for DED problem with valve-point effects. Thus it provides a new effective method to solve DED problem with valve-point effects. In future, this algorithm will be applied to other power system optimization problem having non-linear and non-convex characteristics. This hybrid method can also be successfully applied to other engineering optimization problem.

Appendix

The transmission loss coefficients per MW for 5 unit system are given as follows:

$$B = \begin{bmatrix} 0.000049 & 0.000014 & 0.000015 & 0.000015 & 0.000020 \\ 0.000014 & 0.000045 & 0.000016 & 0.000020 & 0.000018 \\ 0.000015 & 0.000016 & 0.000039 & 0.000010 & 0.000012 \\ 0.000015 & 0.000020 & 0.000010 & 0.000040 & 0.000014 \\ 0.000020 & 0.000018 & 0.000012 & 0.000014 & 0.000035 \end{bmatrix}$$

References

[1] Song Y, Yu I. Dynamic load dispatch with voltage security and environmental constraints. Electr Power Syst Res 1997;43(1):53–60.
[2] Li F, Morgan R, Williams D. Hybrid genetic approaches to ramping rate constrained dynamic economic dispatch. Electr Power Syst Res 1997;43 (2):97–103.
[3] Han X, Gooi H, Kirschen D. Dynamic economic dispatch: feasible and optimal solutions. IEEE Trans Power Syst 2001;16(1):22–8.
[4] Jabr R, Coonick A, Cory B. A study of the homogeneous algorithm for dynamic economic dispatch with network constraints and transmission losses. IEEE Trans Power Syst 2000;15(2):605–11.
[5] Walters D, Sheble G, Co E, Kingsport T. Genetic algorithm solution of economic dispatch with valve point loading. IEEE Trans Power Syst 1993;8(3):1325–32.
[6] Yang H, Yang P, Huang C. Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions. IEEE Trans Power Syst 1996;11 (1):112–8.
[7] Panigrahi C, Chattopadhyay P, Chakrabarti R, Basu M. Simulated annealing technique for dynamic economic dispatch. Electr Power Compon Syst 2006;34 (5):577–86.
[8] Park J, Lee K, Shin J, Lee K. A particle swarm optimization for economic dispatch with nonsmooth cost functions. IEEE Trans Power Syst 2005;20 (1):34–42.
[9] Noman N, Iba H. Differential evolution for economic load dispatch problems. Electr Power Syst Res 2008;78(8):1322–31.
[10] Panigrahi B, Ravikumar Pandi V, Das S. Adaptive particle swarm optimization approach for static and dynamic economic load dispatch. Energy Convers Manage 2008;49(6):1407–15.
[11] Yuan X, Su A, Yuan Y, Nie H, Wang L. An improved PSO for dynamic load dispatch of generators with valve-point effects. Energy 2009;34(1):67–74.
[12] Yuan X, Wang L, Yuan Y, Zhang Y, Cao B, Yang B. A modified differential evolution approach for dynamic economic dispatch with valve-point effects. Energy Convers Manage 2008;49(12):3447–53.
[13] Yuan X, Wang L, Zhang Y, Yuan Y. A hybrid differential evolution method for dynamic economic dispatch with valve-point effects. Expert Syst Appl 2009;36(2P2):4042–8.
[14] Attaviriyanupap P, Kita H, Tanaka E, Hasegawa J. A hybrid EP and SQP for dynamic economic dispatch with nonsmooth fuel cost function. IEEE Trans Power Syst 2002;17(2):411–6.
[15] Victoire T, Jeyakumar A. Reserve constrained dynamic dispatch of units with valve-point effects. IEEE Trans Power Syst 2005;20(3):1273–82.
[16] Dai C, Zhu Y, Chen W. Seeker optimization algorithm. Lect Notes Comput Sci 2007;4456:167–76.
[17] Dai C, Chen W, Zhu Y, Zhang X. Seeker optimization algorithm for optimal reactive power dispatch. IEEE Trans Power Syst 2009;24(3):1218–31.
[18] Dai C, Chen W, Zhu Y, Zhang X. Reactive power dispatch considering voltage stability with seeker optimization algorithm. Electr Power Syst Res 2009;79 (10):1462–71.