Research Article

Dynamic Economic Dispatch Using Hybrid DE-SQP for Generating Units with Valve-Point Effects

A. M. Elaiw,¹,² X. Xia,³ and A. M. Shehata²

¹ Department of Mathematics, Faculty of Science, King Abdulaziz University, P.O. Box 80203, Jeddah 21589, Saudi Arabia
² Department of Mathematics, Faculty of Science, Al-Azhar University, Assiut 71511, Egypt
³ Centre of New Energy Systems, Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa

Correspondence should be addressed to A. M. Elaiw, a_m_elaiw@yahoo.com

Received 10 June 2012; Revised 20 July 2012; Accepted 30 July 2012

Academic Editor: J. J. Judice

Copyright © 2012 A. M. Elaiw et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents hybrid differential evolution (DE) and sequential quadratic programming (SQP) for solving the dynamic economic dispatch (DED) problem for generating units with valve-point effects. DE is used as a global optimizer and SQP is used as a fine tuning to determine the optimal solution at the final. The feasibility of the proposed method is validated with five- and ten-unit test systems. Results obtained by DE-SQP method are compared with other techniques in the literature.

1. Introduction

The primary objective of the static economic dispatch (SED) problem of electric power generation is to determine the optimal schedule of online generating units’ outputs so as to meet the load demand at a certain time at the minimum operating cost under various system and generator operational constraints. Plant operators, to avoid life-shortening of the turbines and boilers, try to keep thermal stress on the equipments within the safe limits. This mechanical constraint is usually transformed into a limit on the rate of change of the electrical output of generators. Such ramp rate constraints link the generator operation in two consecutive time intervals. Optimal dynamic dispatch problem is an extension of SED problem which is used to determine the generation schedule of the committed units so as to meet the predicted load demand over a time horizon at minimum operating cost under ramp rate constraints and other constraints (see [1–31]). Since the ramp rate constraints couple the time intervals, the optimal dynamic dispatch problem is a difficult optimization
problem. If the ramp rate constraints are not included in the optimization problem, the optimal dynamic dispatch problem is reduced to a set of uncoupled SED problems that can be easily solved.

Optimal dynamic dispatch problem was first formulated by Bechert and Kwatny [1] in 1972 and was followed by [2–5]. In these papers, the problem was formulated as an optimal control problem. The optimal control dynamic dispatch formulation models the power system generation by means of state equations where the state variables are the electrical power outputs of the generators and the control inputs are the ramp rates of the generators. In this approach, the optimization is done with respect to the ramp rates and the solution produces an optimal output generator trajectory for a given initial generation. Since the 1980s, the optimal dynamic dispatch problem has been formulated as a minimization problem of the total cost over the dispatch period under some constraints and has been known as the dynamic economic dispatch (DED) problem (see [6–31]). Since the DED problem was introduced, several optimization techniques and procedures have been used for solving the DED problem with complex objective functions or constraints (see the review paper [6]). There were a number of classical methods that have been applied to solve this problem such as the lambda iterative method [16], gradient projection method [25], Lagrange relaxation [26], linear programming [24], and interior point method [11, 13]. Most of these methods are not applicable for nonsmooth or nonconvex cost functions. To overcome this problem, many stochastic optimization methods have been employed to solve the DED problem, such as simulated annealing (SA) [27], genetic algorithms (GA) [28], differential evolution (DE) [18, 19], particle swarm optimization (PSO) [10, 31], and artificial immune system (AIS) [15]. Many of these techniques have proven their effectiveness in solving the DED problem without any or fewer restrictions on the shape of the cost function curves. Hybrid methods which combine two or more optimization methods have been successfully applied to DED problems with valve-point effects such as EP-SQP [9] and PSO-SQP [29, 30].

DE which was proposed by Storn and Price [32] is a population-based stochastic parallel search technique. DE uses a rather greedy and less stochastic approach to problem solving compared to other evolutionary algorithms. DE has the ability to handle optimization problems with nonsmooth/nonconvex objective functions [32]. Moreover, it has a simple structure and a good convergence property, and it requires a few robust control parameters [32]. DE has been successfully applied to the DED problem with nonsmooth and nonconvex cost functions (see [18–21]).

DE is one of the good methods which have been used for solving the DED problem with nonsmooth and nonconvex cost functions; however, the obtained solutions are just near global optimum with long computation time. Therefore, hybrid methods such as DE-SQP can be effective in solving the DED problem with valve-point effects. The aim of this paper is to propose hybrid DE-SQP method to solve the DED problem with valve-point effects. DE is used as a base level search for global exploration and SQP is used as a local search to fine-tune the solution obtained from DE. In the DE-SQP techniques, DE will thoroughly search the solution space and stops when the specified maximum iteration count is reached. Thereafter, the SQP technique will be used to fine-tune the final solution obtained by the DE method.

The remainder of this paper is organized as follows: in Section 2, we introduce the DED problem formulation. An overview of the differential evolution and sequential quadratic programming algorithms is presented in Sections 3 and 4. In Section 5, numerical examples and simulation results are presented. Finally, conclusions are drawn in Section 6.
2. Formulation of the DED Problem

The objective of the DED problem is to determine the generation levels for the committed units which minimize the total fuel cost over the dispatch period \([0, T]\),

\[
\min C_T = \sum_{t=1}^{T} \sum_{i=1}^{N} C_i(P_t^i) \tag{2.1}
\]

subject to the following constraints:

(i) power balance constraint:

\[
\sum_{i=1}^{N} P_t^i = D^t + \text{Loss}^t, \quad t = 1, \ldots, T, \tag{2.2}
\]

(ii) generation limits:

\[
P_{t}^{\text{min}} \leq P_t^i \leq P_{t}^{\text{max}}, \quad i = 1, \ldots, N, \quad t = 1, \ldots, T, \tag{2.3}
\]

(iii) generating unit ramp rate limits:

\[-DR_i \leq P_t^i - P_{t-1}^i \leq UR_i, \quad i = 1, \ldots, N, \quad t = 2, \ldots, T, \tag{2.4}
\]

where \(N\) is the number of committed units; \(T\) is the number of intervals in the time horizon; \(P_t^i\) is the generation of unit \(i\) during the \(t\)th time interval \([t-1, t)\); \(D^t\) is the demand at time \(t\) (i.e., the \(t\)-th time interval); \(UR_i\) and \(DR_i\) are the maximum ramp up/down rates for unit \(i\); \(P_i^{\text{min}}\) and \(P_i^{\text{max}}\) are the minimum and maximum capacity of unit \(i\), respectively. The fuel cost of unit \(i\) considering valve-point effects can be expressed as

\[
C_i(P_t^i) = a_i + b_i P_t^i + c_i (P_t^i)^2 + \left| d_i \sin \left( e_i \left( P_{t}^{\text{min}} - P_t^i \right) \right) \right|, \tag{2.5}
\]

where \(a_i, b_i, c_i\) are positive constants, and \(d_i\) and \(e_i\) are the coefficients of unit \(i\) reflecting valve-point effects.

The \(B\)-coefficient method is one of the most commonly used by power utility industry to calculate the network losses. In this method, the network losses are expressed as a quadratic function of the unit’s power outputs that can be approximated by

\[
\text{Loss}^t = \sum_{i=1}^{N} \sum_{j=1}^{N} P_t^i B_{ij} P_t^j, \quad t = 1, \ldots, T, \tag{2.6}
\]

where \(B_{ij}\) is the \(ij\)th element of the loss coefficient square matrix of size \(N\).
3. Overview of Differential Evolution Algorithm

DE is a simple yet powerful heuristic method for solving nonlinear, nondifferentiable, and nonsmooth optimization problems. DE algorithm is a population-based-algorithm using three operators mutation, crossover, and selection to evolve from randomly generated initial population to final individual solution. The key idea behind DE is that it starts with an initial population of feasible target vectors (parents) and new solutions (offsprings) are generated (by mutation, crossover, and selection operations) until the optimal solution is reached. In the mutation operation, three different vectors are selected randomly from the population and a mutant vector is created by perturbing one vector with the difference of the two other vectors. In the crossover operation, a new trial vector (offspring) is created by replacing certain parameters of the target vector by the corresponding parameters of the mutant vector on the bases of a probability distribution. In DE, the competition between the parents and offspring is one to one. The individual with best fitness will remain till the next generation. The iterative process continues until a user-specific stopping criterion is met. DE algorithm has three control parameters, which are differentiation (or mutation) factor $F$, crossover constant $CR$, and size of population $NP$. According to Storn and Price [32], the basic strategy of DE for $m$-dimensional optimization problem can be described as follows.

1. Initialization: generate a population of $NP$ initial feasible target vectors (parents) $X_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}, i = 1, 2, \ldots, NP$ randomly as

\[
x_{ji} = x_{jm}^{\text{min}} + s_1 \cdot (x_{jm}^{\text{max}} - x_{jm}^{\text{min}}), \quad j = 1, 2, \ldots, m, \quad i = 1, 2, \ldots, NP,
\]

where $s_1$ is uniform random number in $[0, 1]$; $x_{jm}^{\text{min}}$ and $x_{jm}^{\text{max}}$ are the lower and upper bounds of the $j$th component of the target vector.

2. Mutation: let $X_i^G = \{x_{i1}^G, x_{i2}^G, \ldots, x_{im}^G\}$ be the individual $i$ at the current generation $G$. A mutant vector $V_i^{G+1} = \{v_{i1}^{G+1}, v_{i2}^{G+1}, \ldots, v_{im}^{G+1}\}$ is generated according to the following:

\[
V_i^{G+1} = X_i^G + F \cdot (X_{jj}^G - X_{ji}^G), \quad r_1 \neq r_2 \neq r_3 \neq i, \quad i = 1, 2, \ldots, NP
\]

with randomly chosen integer indexes $r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$.

3. Crossover: according to the target vector $X_i^G$ and the mutant vector $V_i^{G+1}$, a new trial vector (offspring) $U_i^{G+1} = \{u_{i1}^{G+1}, u_{i2}^{G+1}, \ldots, u_{im}^{G+1}\}$ is created with

\[
u_{ji}^{G+1} = \begin{cases} v_{ji}^{G+1} & \text{if } \text{rand}(j) \leq CR \text{ or } j = rnb(i), \\ x_{ji}^G & \text{otherwise,} \end{cases}
\]

where $j = 1, 2, \ldots, m$, $i = 1, 2, \ldots, NP$ and $\text{rand}(j)$ is the $j$th evaluation of a uniform random number generator between $[0, 1]$. CR is the crossover constant between $[0, 1]$ which has to be determined by the user. $rnb(i)$ is a randomly chosen index from $1, 2, \ldots, m$ which ensures that $U_i^{G+1}$ gets at least one parameter from $V_i^{G+1}$ [32].
(4) Selection: This process determines which of the vectors will be chosen for the next generation by implementing one-to-one competition between the new generated trial vectors and their corresponding parents. The selection operation can be expressed as follows:

\[
X_{i}^{G+1} = \begin{cases} 
U_{i}^{G+1} & \text{if } f(U_{i}^{G+1}) \leq f(X_{i}^{G}), \\
X_{i}^{G} & \text{otherwise}, 
\end{cases}
\]

(3.4)

where \( i = 1, 2, \ldots, N_{P} \) and \( f \) is the objective function to be minimized. The value of \( f \) of each trial vector \( U_{i}^{G+1} \) is compared with that of its parent target vector \( X_{i}^{G} \). If the value of \( f \) of the target vector \( X_{i}^{G} \) is lower than that of the trial vector, the target vector is allowed to advance to the next generation. Otherwise, the target vector is replaced by a trial vector in the next generation. Thus, all the individuals of the next generation are as good as or better than their counterparts in the current generation. The above steps of reproduction and selection are repeated generation after generation until some stopping criteria are satisfied.

In this paper, we define the evaluation function for evaluating the fitness of each individual in the population in DE algorithm as follows:

\[
f = C_{T} + \lambda \sum_{t=1}^{T} \left( \sum_{i=1}^{N} P_{i}^{t} - (D^{t} + \text{Loss}^{t}) \right)^{2},
\]

(3.5)

where \( \lambda \) is a penalty value. Then the objective is to find \( f_{\min} \), the minimum evaluation value of all the individuals in all iterations. The penalty term reflects the violation of the equality constraint. Once the minimum of \( f \) is reached, the equality constraint is satisfied. Also, the generation power output of each unit at time \( t \) should be adjusted to satisfy the following constraints which combine constraints (2.3) and (2.4) as:

\[
P_{i}^{t} = \begin{cases} 
P_{i}^{t,\min} & \text{if } P_{i}^{t} < P_{i}^{t,\min}, \\
P_{i}^{t} & \text{if } P_{i}^{t,\min} \leq P_{i}^{t} \leq P_{i}^{t,\max}, \\
P_{i}^{t,\max} & \text{if } P_{i}^{t} > P_{i}^{t,\max}, 
\end{cases}
\]

(3.6)

where

\[
P_{i}^{t,\min} = \begin{cases} 
P_{i}^{\min} & \text{if } t = 1, \\
\max(P_{i}^{\min}, P_{i}^{t-1} - DR_{i}) & \text{others},
\end{cases}
\]

\[
P_{i}^{t,\max} = \begin{cases} 
P_{i}^{\max} & \text{if } t = 1, \\
\min(P_{i}^{\max}, P_{i}^{t-1} + UR_{i}) & \text{others}.
\end{cases}
\]

\[(3.7)\]

4. Sequential Quadratic Programming

SQP method can be considered as one of the best nonlinear programming methods for constrained optimization problems. It outperforms every other nonlinear 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 of the Hessian of the Lagrangian function is made 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 nonconvex, SQP ensures a local minimum for an initial solution. SQP has been combined with stochastic optimization techniques to constitute hybrid methods for solving the DED problem with nonsmooth cost functions (see [9, 22]). In this paper, DE is used as a global search and finally the best solutions obtained from DE is given as initial condition for SQP method as a local search to fine-tune the solution. SQP simulations are computed by the fmincon code of the MATLAB Optimization Toolbox.

### 5. Simulation Results

In this paper, to assess the efficiency of the proposed DE-SQP method, two case studies (5 units with losses and 10 units without losses, resp.) of DED problems have been considered in which the objective functions are nonsmooth. In each case study, the simulation parameters chosen are population size $N_p = 60$, maximum iteration $G_{\text{max}} = 20000$, mutation factor

#### Table 1: Hourly generation (MW) schedule of 5-unit system with losses (MW) using DE-SQP.

<table>
<thead>
<tr>
<th>Hour</th>
<th>$P_1$</th>
<th>$P_2$</th>
<th>$P_3$</th>
<th>$P_4$</th>
<th>$P_5$</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>21.6845</td>
<td>99.3693</td>
<td>113.2191</td>
<td>40.0047</td>
<td>139.3571</td>
<td>3.6348</td>
</tr>
<tr>
<td>2</td>
<td>10.0000</td>
<td>98.4091</td>
<td>112.8296</td>
<td>78.0092</td>
<td>139.7746</td>
<td>4.0225</td>
</tr>
<tr>
<td>3</td>
<td>10.0000</td>
<td>93.0102</td>
<td>112.6376</td>
<td>124.5983</td>
<td>139.5085</td>
<td>4.7547</td>
</tr>
<tr>
<td>4</td>
<td>10.0208</td>
<td>98.9394</td>
<td>112.7739</td>
<td>174.5821</td>
<td>139.6980</td>
<td>6.0141</td>
</tr>
<tr>
<td>5</td>
<td>10.0000</td>
<td>94.6503</td>
<td>111.1824</td>
<td>209.3376</td>
<td>139.5944</td>
<td>6.7647</td>
</tr>
<tr>
<td>6</td>
<td>39.9323</td>
<td>98.7250</td>
<td>112.7829</td>
<td>210.1512</td>
<td>154.2991</td>
<td>7.8906</td>
</tr>
<tr>
<td>7</td>
<td>10.0000</td>
<td>98.0631</td>
<td>112.7434</td>
<td>209.7011</td>
<td>203.9654</td>
<td>8.4729</td>
</tr>
<tr>
<td>8</td>
<td>12.5522</td>
<td>98.9663</td>
<td>112.9829</td>
<td>209.5933</td>
<td>229.1619</td>
<td>9.2567</td>
</tr>
<tr>
<td>9</td>
<td>42.5522</td>
<td>101.9286</td>
<td>114.7389</td>
<td>210.6817</td>
<td>230.2829</td>
<td>10.1843</td>
</tr>
<tr>
<td>10</td>
<td>64.8347</td>
<td>98.4322</td>
<td>112.3767</td>
<td>209.6753</td>
<td>229.2401</td>
<td>10.5591</td>
</tr>
<tr>
<td>11</td>
<td>75.0000</td>
<td>100.3215</td>
<td>114.8471</td>
<td>211.1004</td>
<td>229.7591</td>
<td>11.0283</td>
</tr>
<tr>
<td>12</td>
<td>75.0000</td>
<td>98.9111</td>
<td>112.9582</td>
<td>234.6857</td>
<td>230.1685</td>
<td>11.7235</td>
</tr>
<tr>
<td>13</td>
<td>64.2203</td>
<td>97.9704</td>
<td>112.5101</td>
<td>210.1351</td>
<td>229.7242</td>
<td>10.5602</td>
</tr>
<tr>
<td>14</td>
<td>50.0381</td>
<td>98.4511</td>
<td>112.5499</td>
<td>209.7742</td>
<td>229.3542</td>
<td>10.1675</td>
</tr>
<tr>
<td>15</td>
<td>35.4162</td>
<td>98.8278</td>
<td>112.6282</td>
<td>186.4410</td>
<td>229.8143</td>
<td>9.1277</td>
</tr>
<tr>
<td>16</td>
<td>10.0000</td>
<td>98.4782</td>
<td>112.6986</td>
<td>136.4410</td>
<td>229.6153</td>
<td>7.2332</td>
</tr>
<tr>
<td>17</td>
<td>10.0000</td>
<td>88.2951</td>
<td>112.6200</td>
<td>124.6074</td>
<td>229.1609</td>
<td>6.6833</td>
</tr>
<tr>
<td>18</td>
<td>35.3553</td>
<td>98.7119</td>
<td>112.7691</td>
<td>139.3772</td>
<td>229.6497</td>
<td>7.8633</td>
</tr>
<tr>
<td>19</td>
<td>33.6668</td>
<td>98.5493</td>
<td>111.9794</td>
<td>189.3772</td>
<td>229.5668</td>
<td>9.1396</td>
</tr>
<tr>
<td>20</td>
<td>63.6668</td>
<td>98.5990</td>
<td>112.5890</td>
<td>210.0303</td>
<td>229.6762</td>
<td>10.5612</td>
</tr>
<tr>
<td>21</td>
<td>39.2846</td>
<td>98.5966</td>
<td>112.7120</td>
<td>209.8748</td>
<td>229.4336</td>
<td>9.9017</td>
</tr>
<tr>
<td>22</td>
<td>10.0437</td>
<td>98.6538</td>
<td>112.9970</td>
<td>161.5680</td>
<td>229.6066</td>
<td>7.8692</td>
</tr>
<tr>
<td>23</td>
<td>10.0000</td>
<td>98.8754</td>
<td>112.6148</td>
<td>124.7293</td>
<td>186.6872</td>
<td>5.9067</td>
</tr>
<tr>
<td>24</td>
<td>10.0000</td>
<td>81.0109</td>
<td>112.1181</td>
<td>124.8490</td>
<td>139.5118</td>
<td>4.4899</td>
</tr>
</tbody>
</table>
$F = 0.423$, and crossover factor $CR = 0.885$ and the results represent the average of 30 runs of the proposed method. All computations are carried out by MATLAB program.

### 5.1. Five-Unit System

This example presents an application of the DE-SQP method to the DED problem consisting of five units with valve point effects and transmission line losses. The technical data of the units are taken from [17]. The optimal solution of the DED problem among 30 runs is over, for example, 24 h ($T = 24$), and is given in Table 1.

### 5.2. Ten-Unit System

This example presents an application of the DE-SQP method to the DED problem consisting of ten units without losses. The data of the ten-unit system are taken from [9]. The optimal solution of the DED problem is over, for example, 24 h ($T = 24$), and is given in Table 2. Comparisons between our proposed method (DE-SQP) and other methods for both examples (five units with losses and ten units without losses) are given in Table 3. It is observed that the proposed method reduces the total generation cost better than the other methods reported in the literature. These methods can be classified into (1) heuristic methods such as pattern search [23], particle swarm optimization [17], differential evolution [18],
### Table 3: Comparison of the results with other methods.

<table>
<thead>
<tr>
<th>Optimization technique</th>
<th>5-unit system with losses</th>
<th>10-unit system without losses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cost ($), Total losses (MW)</td>
<td>Cost ($)</td>
</tr>
<tr>
<td>Pattern search [23]</td>
<td>46530, 192.21</td>
<td>—</td>
</tr>
<tr>
<td>Particle swarm optimization [17]</td>
<td>47852, —</td>
<td>—</td>
</tr>
<tr>
<td>Differential evolution [18]</td>
<td>45800, 194.35</td>
<td>—</td>
</tr>
<tr>
<td>Sequential quadratic programming [9]</td>
<td>—, —</td>
<td>1051163</td>
</tr>
<tr>
<td>Evolutionary programming [9]</td>
<td>—, —</td>
<td>1048638</td>
</tr>
<tr>
<td>Hybrid evolutionary programming and sequential quadratic programming [9]</td>
<td>—, —</td>
<td>1031746</td>
</tr>
<tr>
<td>Modified differential evolution [20]</td>
<td>—, —</td>
<td>1031612</td>
</tr>
<tr>
<td>Hybrid particle swarm optimization and sequential quadratic programming [30]</td>
<td>—, —</td>
<td>1030773</td>
</tr>
<tr>
<td>Proposed hybrid differential evolution and sequential quadratic programming</td>
<td>43231, 193.81</td>
<td>1030500</td>
</tr>
</tbody>
</table>

Moreover, it is observed that the transmission line losses calculated by our method are smaller than those of other methods. For more details about these methods and their way of working we refer the reader to the review paper [6].

### 6. Conclusion

This paper presents hybrid method, combining differential evolution (DE), and sequential quadratic programming (SQP) for solving the DED problem with valve-point effects. At first we, applied DE to find the best solution, then this best solution is given to SQP as an initial condition to fine-tune the optimal solution at the final. The feasibility and efficiency of the DE-SQP method are illustrated by conducting two examples consisting of five and ten units with valve-point effects, respectively. Our results are compared with other methods. It has been shown that our proposed methods give less cost than other methods reported in the literature.

### Acknowledgments

This paper was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, Saudi Arabia. The first author, therefore, acknowledges with thanks DSR technical and financial support. The authors are grateful to Professor J. J. Judice and the anonymous reviewers for constructive suggestions and valuable comments, which improve the quality of the paper.

### References


