Multi-Agent Based Differential Evolution Algorithm for Economic Dispatch With Generator Constraints

This paper proposes a multi-agent based differential evolution algorithm for solving economic dispatch problem considering ramp rate limit and prohibited operating zones. In this work, multiagent system (MAS) concept is integrated with differential evolution (DE) to form a multiagent based differential evolution (MADE) algorithm. In MADE, a member in the population is treated as an agent. They all live in a lattice like environment. Neighborhood competition and cooperation among agents and differential evolution approach make this algorithm reach optimum value quickly. The proposed MADE has been applied to three test systems namely 3 unit, 6 unit and 15 unit systems and the results are compared with DE and existing approaches. Simulation results show that MADE gives better value and converges quickly to optimum values than earlier reported methods.

Keywords: Multi-Agent systems, Differential evolution, Economic dispatch and Prohibited operating zones.

1. Introduction

Economic dispatch (ED) problem is a fundamental issue in power system operation. The objective of ED problem is to minimize the total fuel cost so as to meet the required demand while satisfying equality and inequality constraints. Since it is an optimization problem, many mathematical techniques such as lambda iteration method, base point participation factors method, gradient method and Newton’s method [1-5] have been addressed to solve the ED problem.

In these mathematical methods, the main assumption is that the fuel cost curve is considered as monotonically increasing one. Conventionally ED problem is solved by using lambda-iteration method. This method has an oscillatory problem for large scale system and resulting in longer solution time. However in this work, nonlinearities in the fuel cost curve such as ramp rate limit and prohibited operating zones are considered due to practical limitation of the generators. These nonlinearities make the mathematical techniques fail in giving global optimum value because they all stuck at local optima.

Like mathematical techniques, heuristic optimization techniques such as genetic algorithm [6,7] (GA), particle swarm optimization [7] (PSO), evolutionary programming [8] (EP) and efficient evolution strategy [9] have also been reported to solve the ED problem with ramp rate limit and prohibited operating zones constraints. Differential evolution (DE) was first proposed by Price and Storn [10]. It is a simple and efficient algorithm. It has been applied to many power system problems and proved to be the best among the evolutionary techniques [11-12]. DE also has been applied to ED problem with
ramp rate limit and prohibited operating zones and produces better results than existing evolutionary methods [13]. The main drawback of evolutionary approaches is that it needs many trails to reach optimum value because of its stochastic nature.

Recently, agent based computation has been studied in the field of artificial intelligence and has also been used with evolutionary methods to address power system problems [15-16]. This hybrid algorithm gives better results than conventional evolutionary approaches and converges within few number of generations. Similarly many strategies have been adapted with DE to speed up the convergence and avoid trapping in local optima [17-18].

In this work, multiagent system (MAS) is integrated with DE to make multiagent based differential evolution (MADE) algorithm. Every individual in population is treated as an agent. They all live in a lattice like environment. Therefore an agent is a member in the population and a solution vector to the problem. Each agent competes and cooperates with its neighbors. And it also gains knowledge from self learning. Finally, all agents evolve using DE. These all make MADE reach better value than conventional DE quickly. This MADE algorithm is applied to 3 unit, 6 unit and 15 unit test systems. Simulation results reveal MADE outperforms conventional DE and existing approaches.

This paper is organized as follows: Section 2 describes the mathematical modeling of ED problem with generator constraints. Section 3 explains the differential evolution, multiagent system and multiagent based differential evolution algorithm. Section 4 shows that how MADE algorithm is applied to this problem. Section 5 gives the test system used and comparison of the simulation results with existing methods.

2. Problem formulation

The ED problem is a nonlinear optimization problem. The main objective of ED is to minimize the total fuel cost of thermal power plant so as to meet the desired load while meeting power balance constraints, generation limit constraints and practical constraints of generator that include ramp rate limit and prohibited operating zone.

2.1. Practical generator constraints

Generally ED problem is solved by increasing generator output continuously and smoothly in order to reduce the complexity. However in practical, generator constraints such as ramp rate limit and prohibited operating zone force the generators to operate in specific intervals. To achieve true economic operation, generator constraints must be taken into account.

1). Ramp rate limit

According to [7], the inequality constraints due to ramp rate limit is defined as

- if generation increases
  \[ P_i - P_i^0 \leq UR_i \]  \hspace{1cm} (1)
- if generation decreases
  \[ P_i - P_i^0 \leq DR_i \]  \hspace{1cm} (2)

where \( P_i \) and \( P_i^0 \) are current and previous output power (MW) of \( i \)-th unit respectively and \( UR_i \) and \( DR_i \) are up ramp and down ramp limit (MW/time period) of \( i \)-th unit respectively.
2). Prohibited operating zone

According to [4] and [5], the input and output characteristics of thermal unit have many valve points. These valve points generate many prohibited zones. While adjusting generation output of \( P_i \) unit, one must avoid the operation in prohibited zones. Therefore the feasible operating zones of \( i \)-th unit is described as follows

\[
P_i^{\text{min}} \leq P_i \leq P_i^{\text{max}}
\]

\[
P_{i,j-1}^{u} \leq P_i \leq P_{i,j}^{l} \quad j = 2,3\ldots n_i
\]

\[
P_{i,n_i}^{u} \leq P_i \leq P_i^{\text{max}}
\]

where \( j \) is the number of prohibited zones of unit \( i \).

2.2. Objective function

The objective of ED problem is defined as

\[
\min \quad F_T = \sum_{i=1}^{m} \left( a_i P_i^2 + b_i P_i + c_i \right) \$/hr
\]

Constraints

1). Power balance constraint

\[
\sum_{i=1}^{m} \left( P_i - P_D - P_L \right) = 0
\]

2). Generator constraints

\[
\max \left( P_i^{\text{min}}, P_i^{0} - DR_i \right) \leq P_i \leq \min \left( P_i^{\text{max}}, P_i^{0} + UR_i \right)
\]

\[
P_i \in \begin{cases} 
P_i^{\text{min}} \leq P_i \leq P_i^{l}, \\
P_{i,j}^{u} \leq P_i \leq P_{i,j}^{l} \quad j = 2,3\ldots n_i, i = 1\ldots m \\
P_{i,n_i}^{u} \leq P_i \leq P_i^{\text{max}}
\end{cases}
\]

where \( a_i, b_i \) and \( c_i \) are the cost coefficients of \( i \)-th generator, \( m \) is the total number of committed generators in the system, \( P_D \) is the total demand and \( P_L \) is the transmission loss which is a function of unit power output and \( B \) coefficients

\[
P_L = \sum_{i=1}^{m} \sum_{j=1}^{m} P_i B_{ij} P_j + \sum_{i=1}^{m} B_{0i} P_i + B_{00}
\]

3. Multiagent Based Differential Evolution

3.1. Differential Evolution (DE)

DE was first proposed by Price and Storn [10]. It is also a population based algorithm like other evolutionary algorithms. It is a simple and efficient algorithm. The main difference between DE and GA is in mutation process. It also starts with initial set of population and evolves using mutation, crossover and selection. In DE, the mutation is carried out using arithmetical operations on randomly selected vectors. The evolution process of classical DE is given as

1) Initialization: The initial population is generated within the boundary using the following strategy
\[ x_{i,j}^0 = x_{j}^{\text{min}} + \text{rand} \times (x_{j}^{\text{max}} - x_{j}^{\text{min}}) \]  \hspace{1cm} (9)

where \( i=1,2\ldots np \), \( np \) is the population size, \( j=1,2\ldots d \), \( d \) is the variable dimension, \( x_{j}^{\text{min}} \) and \( x_{j}^{\text{max}} \) are the minimum and maximum limit of \( j \)-th dimension respectively and \( \text{rand} \) denotes uniform random number between \([0,1]\).

2) **Mutation:** For each vector \( x_i^G \) in generation \( G \), a mutant vector is generated by the following strategy

\[ v_{i}^{G+1} = x_{r1}^G + F \times (x_{r2}^G - x_{r3}^G) \]  \hspace{1cm} (10)

where \( r1, r2 \) and \( r3 \) are randomly selected mutually different integers and chosen to be different from running index \( i \). \( F \) is a mutation factor whose value in classical DE [12] is a real number between \([0,2]\) and constant during entire optimization process.

3) **Cross over:** A trial vector \( u_{i}^{G+1} \) is generated from \( x_i^G \) and \( v_i^{G+1} \) as

\[ u_{i,j}^{G+1} = \begin{cases} v_{i,j}^{G+1}, & \text{if } \text{rand} \leq CR \text{ and } j = k \\ x_{i,j}^G, & \text{otherwise} \end{cases} \]  \hspace{1cm} (11)

where \( k \in \{1,2,\ldots d\} \) is randomly selected index once for each \( i \). \( CR \) is the cross over rate in the range\([0,1]\) which controls the probability that a trial vector \( u_i^{G+1} \) comes from randomly selected mutated vectors \( v_i^{G+1} \), instead the current vector \( x_i^G \).

4) **Selection:** The selection process for next generation is carried out as

\[ x_{i,j}^{G+1} = \begin{cases} u_{i,j}^{G+1}, & \text{if } f(u_{i,j}^{G+1}) \leq f(x_{i,j}^G) \\ x_{i,j}^G, & \text{otherwise} \end{cases} \]  \hspace{1cm} (12)

3.2. Multiagent system

Multiagent system (MAS) is a computational system in which several agents interact or work together in order to achieve a common goal. Wooldridge [14] defines an agent as a physical or virtual entity which has the following properties.

1) Agents live and act in a given environment
2) Agents are able to sense and interact with other agents in its local environment
3) Agents attempt to achieve particular goals
4) Agents are able to respond to changes in the environment

Recently MAS has been used with evolutionary algorithms for solving power system problems. In this work MAS has been integrated with DE to form MADE algorithm. DE itself is a simple and powerful algorithm for solving continuous time optimization problems. After integrating MAS techniques into DE, further achievement is made in optimum values. While using MAS for solving problem, the following elements need to be defined. They are 1) meaning and purpose of each agent, 2) environment where they live, 3) local environment and 4) set of rules governing interaction between agents in the local environment.
3.3. MADE

MAS and DE are integrated to form MADE algorithm for solving ED problem with generator constraints. In MADE, an agent represents both candidate solution vector to optimization problem and individual in the DE population. Firstly, a lattice like environment is created with each agent fixed on a lattice point. In order to obtain optimal solution quickly and make convergence faster, each agent in the environment competes and cooperates with its neighbors and its knowledge through self learning. DE strategy is also used to speed up the transfer of information between agents and help in making MADE algorithm reach optimum value quickly.

1). Purpose of each agent: An agent in the ED problem is a candidate solution and an individual in DE. Hence, agent $\alpha$ has a fitness value to the optimization problem. The fitness value of an agent in ED problem is total fuel cost, i.e, $F_T$ in (4)

$$f(\alpha) = F_T = \sum_{i=1}^{m} \left( a_i P_i^2 + b_i P_i + c_i \right)$$

The purpose of agent $\alpha$ is to minimize the total fuel cost by meeting power balance and generator constraints. In ED problem, agent $\alpha$ carries control variable i.e, real power generation.

2). Definition of an environment: In MADE, all agents live in a lattice like environment as in Fig. 1. In the environment $L$, each agent is fixed on a lattice point and each circle represents agent. The data in the circle represents its position in the environment $L$. The size of the environment is $L_{\text{size}} \times L_{\text{size}}$ where $L_{\text{size}}$ is an integer. Since each agent represents both candidate solution and a member in the DE population, therefore the number of population and environment size should be same.

![Fig. 1. Structure of the environment](image)

3). Definition of local environment: Since each agent can sense and interact with neighbors only, the definition of local environment is important in the proposed MADE algorithm.
The agent $\alpha$ located at $(i, j)$ is represented as $\alpha_{i,j}, i, j = 1, 2 \cdots L_{\text{size}}$ in the environment has neighbors $N_{i,j}$ which are defined as follows

$$N_{i,j} = \{\alpha_{i,j}, \alpha_{i,j+1}, \alpha_{i+1,j}, \alpha_{i+1,j+1}\} \quad (14)$$

where

$$i = \begin{cases} \frac{i - 1}{L_{\text{size}}}, & i \neq 1, \\ 1, & i = 1 \end{cases} \quad j = \begin{cases} \frac{j - 1}{L_{\text{size}}}, & j \neq 1, \\ 1, & j = 1 \end{cases}$$

From (14), each agent has four neighbors. They form a local environment and interact with each other.

4). Behavioral rules for agent: In order to achieve optimal solution quickly, some behaviors have been assigned to each agent to pass information to whole environment. Then they use DE evolution mechanism and its knowledge through self learning to reach optimal solution quickly. Based on this, three operators have been defined.

Competition and cooperation operator: This operator is performed on agent $\alpha$ located at $(i, j)$ represented as $\alpha_{i,j} = \{\alpha_1, \alpha_2, \cdots \alpha_n\}$. The $m = \text{Min} N_{i,j} = (m_1, m_2, \cdots m_n)$ is the agent with minimum fitness value among neighbors, i.e., $m \in N_{i,j}$. If agent $\alpha_{i,j}$ satisfies (15), it is a winner; otherwise it is a loser,

$$f(\alpha_{i,j}) \leq f(m) \quad (15)$$

If $\alpha_{i,j}$ is a winner, it can stay in the environment and its location in the search space will not change. If it is a loser, it has to be replaced by new agent $\text{New}_{i,j}$. The new agent $\text{New}_{i,j} = (\alpha'_1, \alpha'_2, \cdots \alpha'_n)$ is determined by the following two strategies in MADE.

Strategy 1: $\text{New}_{i,j}$ is determined by

$$\alpha'_k = m_k + \text{rand} \times (\alpha_k - m_k) \quad k=1,2,\cdots,n \quad (16)$$

where $\text{rand}$ is a uniform random value between $[0,1]$. $\alpha'_k$ is fixed at its limits if it violates the minimum and maximum limit. This strategy is helpful in keeping useful information from loser.

Strategy 2: In this strategy $m_{i,j}$ i.e., $\text{Min} N_{i,j}$ is mapped on $[0, 1]$ according to

$$m'_k = \frac{(m_k - x_{\text{min},k})}{(x_{\text{max},k} - x_{\text{min},k})}, \quad k=1,2,\cdots,n \quad (17)$$

Then $\text{New}_{i,j} = (\beta'_1, \beta'_2, \cdots \beta'_n)$ is determined by

$$\text{New}_{i,j} = (m'_1, m'_2, \cdots, m'_{i_1-1}, m'_{i_1}, m'_{i_2-1}, \cdots, m'_{i_{l+1}}, m'_{i_1}, m'_{i_2+1}, m'_2+2, \cdots, m'_n) \quad (18)$$

where $1 < i_1 < n, 1 < i_2 < n$ and $i_1 < i_2$. Finally $\text{New}_{i,j}$ is obtained by mapping $\text{New}_{i,j}$ back to $[X_{\text{min}}, X_{\text{max}}]$ as
\[ \alpha'_k = x_{\text{min},k} + \beta_k \times (x_{\text{max},k} - x_{\text{min},k}) \quad k=1,2,\ldots,n \]  

(19)

This strategy uses random search for finding \( \text{New}_{i,j} \). Generally, strategy 2 is adopted in first iterative process to make global search and strategy 1 is adopted in next iterative process to use local search.

**DE operator:** Since competition and cooperation is performed between agents and its neighbors, information is passed slowly to the whole environment. In order to make it fast, it uses DE mechanism. Hence each agent evolves using mutation, crossover and selection in every generation according to (10), (11) and (12). Thereby the information is diffused quickly to all the agents in the environment.

**Self learning operator:** Each agent is able to learn by using its knowledge in order to further enhance its ability for solving problem. This is performed only on best agent to reduce the computational time. References [15] and [16] use small scale MAGA and MAPSO respectively for self learning. This method uses small scale MADE for self learning.

In self learning operator of agent \( \alpha_{i,j} = (\alpha_1, \alpha_2, \ldots, \alpha_n) \), first of all a lattice like environment, \( sL \) is constructed. The size of \( sL \) is \( sL_{\text{size}} \times sL_{\text{size}} \) where \( sL_{\text{size}} \) is an integer, and all agents \( sa_{i,j}, i', j' = 1, 2, \ldots, sL_{\text{size}} \) are generated as

\[
\begin{align*}
    sa_{i,j} &= \begin{cases} 
    \alpha_{i,j} & \text{if } i' = 1, j' = 1 \\
    \text{New}a_{i',j'} & \text{otherwise}
    \end{cases} \\
\end{align*}
\]

(20)

where \( \text{New}a_{i',j'} = (e_{i',j';1}, e_{i',j';2}, \ldots, e_{i',j';n}) \) is generated according to equation (21).

\[
\begin{align*}
e_{i',j';k} &= \begin{cases} 
    x_{k\text{min}} & \alpha_k \times \text{rand} (1 - sR, 1 + sR) < x_{k\text{min}} \\
    x_{k\text{max}} & \alpha_k \times \text{rand} (1 - sR, 1 + sR) > x_{k\text{max}}, \quad k = 1, 2, \ldots, n \\
    \alpha_k \times \text{rand} (1 - sR, 1 + sR) & \text{otherwise}
    \end{cases}
\end{align*}
\]

(21)

where \( sR \in [0,1] \) represents the search radius. Next, neighborhood competition and cooperation operator with DE strategy is iteratively performed on \( sL \). Finally \( \alpha_{i,j} \) is replaced with agent obtained from self learning process.

**4. Implementation of MADE for Economic Dispatch Problem**

The proposed MADE algorithm can quickly find optimum value for ED problem with generator constraints. The details of the algorithm are given below.

1. Input system parameters and specify the lower and upper bound of control variables. In ED problem with generator constraints, real power output of generators are considered as control variables.
2. Generate a lattice like environment and assign randomly each agent on lattice point. Here each agent carries a solution vector, i.e., control variables. While generating real power output randomly, the generator constraints i.e., equation (6) and (7), must be satisfied.
3. Determine $P_L$ using $B$ coefficients and fitness value using the following equation (22) for each agent.

$$F_T^* = F_T + \lambda \left( \sum_{i=1}^{m} (P_i - P_D - P_L) \right)^2$$

where $\lambda$ is a penalty factor.

4. Perform neighborhood competition and cooperation operator on each agent.

5. Apply DE strategy and further modify each agent in the search space according to equation (10-12)

6. Find the best agent, i.e., agent with minimum fitness value and perform self learning operator.

7. If one of the stopping criteria is met, go to next step else go back to step 3.

8. Output the agent with minimum fitness value in last generation.

5. Test Systems and Results

To verify the efficiency of the proposed MADE algorithm for ED problem with generator constraints, two test systems namely 3unit, 6unit and 15 unit test systems are taken. Before MADE is applied to ED problem, some parameters need to be defined. Since the environment size and number of population are same in MADE, $L_{size}$ is taken as 5. Therefore the number of population is 25. $Gen = 100$, scaling factor $F=0.85$ and cross over rate $CR=0.9$. To reduce the computational time for self learning operator, $s_{L_{size}} = 3$, $s_{Gen} = 5$ and $sR = 0.25$ are taken.

5.1. Test system 1:

This test system consists of three units whose data with $B$ loss coefficients are given in [6]. The load demand $P_D$ is 300 MW. This test system is solved by both DE and the proposed MADE algorithm by meeting all the constraints including ramp rate limits and prohibited operating zones. The best solution achieved by the proposed method is given in table I. Table II gives the comparison of the simulation results obtained by the proposed MADE algorithm with DE and existing method. From the results, it is clear that both DE and MADE give the same results, however, they are better than existing method. It is important to explain that MADE converges quickly to optimum value than DE and is given in Fig. 2.

Table I Best solution of 3 unit system

<table>
<thead>
<tr>
<th>Unit power output</th>
<th>MADE method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$ (MW)</td>
<td>205.589</td>
</tr>
<tr>
<td>$P_2$ (MW)</td>
<td>83.264</td>
</tr>
<tr>
<td>$P_3$ (MW)</td>
<td>15.000</td>
</tr>
<tr>
<td>Total power output (MW)</td>
<td>303.853</td>
</tr>
<tr>
<td>$P_{Loss}$ (MW)</td>
<td>3.853</td>
</tr>
<tr>
<td>Total generation cost ($/hr)</td>
<td>3579.505</td>
</tr>
</tbody>
</table>
Table II Comparison of different methods for 3 unit system

<table>
<thead>
<tr>
<th>S.No</th>
<th>Methods used</th>
<th>Generation cost ($/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MADE</td>
<td>3579.505</td>
</tr>
<tr>
<td>3</td>
<td>Differential evolution</td>
<td>3579.505</td>
</tr>
</tbody>
</table>

Fig. 2. Convergence characteristics of 3 unit system

5.2. Test system 2:
This test system consists of six thermal units, 26 buses and 39 transmission lines. The entire data with B loss coefficients are given in [7]. The load demand $P_L$ is 1263 MW. This problem is solved by both classical DE and MADE algorithm by satisfying all the constraints including ramp rate limits and prohibited operating zones. The best solution after several runs by MADE method was given in table III. The comparison of MADE results with classical DE, IDE and existing approaches are given in table IV. From the simulation results, MADE outperforms both DE and IDE in terms solution and convergence rate. Fig. 3. shows the convergence characteristics of both classical DE and MADE.

Table III Best solution of 6 unit system

<table>
<thead>
<tr>
<th>Unit power output</th>
<th>MADE method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$ (MW)</td>
<td>445.6919</td>
</tr>
<tr>
<td>$P_2$ (MW)</td>
<td>172.4129</td>
</tr>
<tr>
<td>$P_3$ (MW)</td>
<td>263.1046</td>
</tr>
<tr>
<td>$P_4$ (MW)</td>
<td>138.1940</td>
</tr>
<tr>
<td>$P_5$ (MW)</td>
<td>164.7499</td>
</tr>
<tr>
<td>$P_6$ (MW)</td>
<td>84.5283</td>
</tr>
<tr>
<td>Total power output (MW)</td>
<td>1268.68</td>
</tr>
<tr>
<td>$P_{Loss}$ (MW)</td>
<td>5.68</td>
</tr>
<tr>
<td>Total generation cost ($/hr)</td>
<td>15353.22</td>
</tr>
</tbody>
</table>
### Table IV Comparison of different methods for 6 unit system

<table>
<thead>
<tr>
<th>S.No</th>
<th>Methods used</th>
<th>Generation cost ($/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MADE</td>
<td><strong>15353.22</strong></td>
</tr>
<tr>
<td>2</td>
<td>Improved differential evolution [13]</td>
<td>15356.1809</td>
</tr>
<tr>
<td>4</td>
<td>Evolution strategy [9]</td>
<td>15407.527</td>
</tr>
<tr>
<td>5</td>
<td>Particle swarm optimization [7]</td>
<td>15450.00</td>
</tr>
<tr>
<td>6</td>
<td>Genetic algorithm [7]</td>
<td>15459.00</td>
</tr>
</tbody>
</table>

5.3. Test system 3:

This test system contains 15 thermal units whose data with $B$ loss coefficients are available in [7]. This system was also solved by both classical DE and MADE algorithm. The best solution for this system by MADE algorithm after several trials were given in table V. Table VI gives the comparison of overall cost obtained by MADE with DE, IDE and existing evolutionary approaches. Simulation results reveal that proposed MADE algorithm outperforms DE, IDE and existing evolutionary methods. The convergence characteristics for both DE and MADE are given in Fig. 4. It is clear from the characteristics, MADE converges to optimum value with lesser number of generations.
### Table III Best solution of 15 unit system

<table>
<thead>
<tr>
<th>Unit power output</th>
<th>MADE method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P_1$ (MW)</td>
<td>416.78</td>
</tr>
<tr>
<td>$P_2$ (MW)</td>
<td>378.48</td>
</tr>
<tr>
<td>$P_3$ (MW)</td>
<td>130.00</td>
</tr>
<tr>
<td>$P_4$ (MW)</td>
<td>130.00</td>
</tr>
<tr>
<td>$P_5$ (MW)</td>
<td>470.00</td>
</tr>
<tr>
<td>$P_6$ (MW)</td>
<td>395.00</td>
</tr>
<tr>
<td>$P_7$ (MW)</td>
<td>465.00</td>
</tr>
<tr>
<td>$P_8$ (MW)</td>
<td>60.00</td>
</tr>
<tr>
<td>$P_9$ (MW)</td>
<td>25.00</td>
</tr>
<tr>
<td>$P_{10}$ (MW)</td>
<td>25.00</td>
</tr>
<tr>
<td>$P_{11}$ (MW)</td>
<td>36.11</td>
</tr>
<tr>
<td>$P_{12}$ (MW)</td>
<td>55.00</td>
</tr>
<tr>
<td>$P_{13}$ (MW)</td>
<td>25.00</td>
</tr>
<tr>
<td>$P_{14}$ (MW)</td>
<td>15.00</td>
</tr>
<tr>
<td>$P_{15}$ (MW)</td>
<td>15.00</td>
</tr>
<tr>
<td>Total power output (MW)</td>
<td>2641.38</td>
</tr>
<tr>
<td>$P_{Loss}$ (MW)</td>
<td>11.38</td>
</tr>
<tr>
<td>Total generation cost ($/hr)</td>
<td>32412.626</td>
</tr>
</tbody>
</table>

### Table IV Comparison of different methods for 15 unit system

<table>
<thead>
<tr>
<th>S.No</th>
<th>Methods used</th>
<th>Generation cost ($/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MADE</td>
<td><strong>32412.626</strong></td>
</tr>
<tr>
<td>2</td>
<td>Improved differential evolution [13]</td>
<td>32418.79</td>
</tr>
<tr>
<td>4</td>
<td>Evolution strategy [9]</td>
<td>32568.54</td>
</tr>
<tr>
<td>5</td>
<td>Particle swarm optimization [7]</td>
<td>32858.00</td>
</tr>
<tr>
<td>6</td>
<td>Genetic algorithm [7]</td>
<td>33113.00</td>
</tr>
</tbody>
</table>
6. Conclusion

This paper proposes a new multi agent based differential evolution algorithm for ED problem with generator constraints. This method benefits mainly from environment where all the agents live and their behaviors. In MADE, each agent competes and cooperates with its neighbors also uses DE strategy to adjust its position. Thus the information is quickly transferred to whole environment finally all agents share useful information. This algorithm was tested on three test systems, i.e., 3 unit, 6 unit and 15 unit systems. Simulation results show that MADE algorithm outperforms DE, IDE and existing evolutionary approaches. While comparing the proposed MADE with DE algorithm, the proposed method is giving better solutions for 6 unit and 15 unit test systems in terms of solution accuracy and convergence rate. Whereas for 3 unit systems, both the proposed MADE and DE give same results, however, MADE converges very quickly to optimum values. It should be noted that this algorithm is general and can also be applied to other power system optimization problems.

References


