SOLVING ECONOMIC LOAD DISPATCH PROBLEM USING CLUSTERED GRAVITATIONAL SEARCH ALGORITHM

1PRANJAL PRAGYA VERMA, 2PRAVEEN GANESH, 3G.AVINASH, 4K.CHANDRASEKARAN

IEEE, Member
Department of Electrical and Electronics Engineering, National Institute of Technology, Puducherry
Email: 1vpranjal@gmail.com, 2praveenganesh92@gmail.com, 3avi24_92@rediffmail.com, 4chandekaran23@gmail.com

Abstract- Gravitational search algorithm (GSA) is based on the law of gravity and a mass of interaction. In GSA, the searcher agents are a collection of masses which interact with each other based on the Newtonian gravity and the laws of motion. This paper proposes a new clustered gravitational search algorithm (CGSA) to accelerate the performance of the GSA. Here, the whole population is divided into three basic groups: namely the leader, the follower and the freelancer. In this way, the exploration in GSA is kept alive without killing the exploitation. To validate the proposed CGSA, the economic dispatch problem (EDP) is solved using CGSA for IEEE 30 bus system and 40 unit system. The performance of the proposed approach reveals the efficiency and robustness when compared results of other optimization algorithms reported in literature.

Keywords- Clustered gravitational search algorithm, Economic dispatch problem, Valve point effect.

I. INTRODUCTION

The economic load dispatch problem is the problem of varying the generator’s real and reactive power generation so as to meet the demands and losses, with the minimum fuel cost, and the generation being under some constrains. The more efficient generator in the system does not guarantee less cost as it may be present in the system where fuel cost is higher. If the plant is located seeing the cost of fuel to be minimum, then the distance to the load centers would increase and losses increase. The typical approach available in the literature is to augment the constraints into objective function by using lagrangian’s multipliers and then solve it. But as the number of the power plants increase, the search space increases to find the optimum generations, the process becomes long and there is a chance for the process to get stuck at local minima. The regular methods for optimization fail when the power plant number increase as this would increase the search space exponentially.

When dealing with the optimization problems with high-dimensional search space the conventional algorithms like ‘Exhaustive Search’ are not very effective as the search space increases exponentially with the dimensions. In recent years there has been a trend to use nature inspired heuristic algorithms to solve these optimization problems. Many algorithms like Differential Evolution, Differential Search Algorithm, Gravitational Search Algorithm, Bat Algorithm etc, have been proposed. But no single algorithm has been found to solve all the optimization problems efficiently, so new algorithms are always welcome.

There are many algorithms that have been used to solve this Economic load dispatch, like Honey Bee Mating Optimization, objective bacterial colony chemotaxis optimization, Niched pareto genetic algorithm, Fuzzy based bacterial foraging algorithm.

In this paper we propose a new algorithm that merges the clustered behavior in the Gravitational Search Algorithm and use it to solve the Economic Load dispatch problem on a 6 unit system and 40 bus systems. This algorithm is derived from the Gravitational Search algorithm by Rashedi et al and improves its performance by hitting a delicate control between the exploration and exploitation of search space during optimization.

II. PROBLEM FORMULATION OF EDP

The objective of economic dispatch is to minimize total generation cost while satisfying equality and inequality constrains. Mathematically, objective of EDP is represented as

\[ \text{Minimize } F(P_d) = \sum_{i=1}^{n} f_i(P_{iG}) \]  

(1)

where \( f_i(P_{iG}) \) is the fuel cost of \( i^{th} \) generator unit and is approximated as a quadratic function of the power output from the generating units.

\[ f_i(P_{iG}) = a_i (P_{iG})^2 + b_i (P_{iG}) + c_i \]  

(2)

Where, \( a_i, b_i, c_i \) are the cost coefficients of generating unit i. \( N_G \) is the number of generating units and \( P_{iG} \) is the power generation of the \( i^{th} \) generator.

Constraints:

Power Balance constraint: The total power generated must cover the total demand \( P_d \) and the total transmission losses \( P_{Loss} \).

\[ \sum_{i=1}^{n} P_i(P_G) = P_d - P_{Loss} \]  

(3)

Solving Economic Load Dispatch Problem Using Clustered Gravitational Search Algorithm
The transmission losses are calculated using the Korn’s formula.

**Generator Capacity Constraint:**
The real output power of each generator is constrained by upper and lower limits.

\[ P_{G_{min}} \leq P_G \leq P_{G_{max}} \]  

(4)

Where, \( P_{G_{min}} \) and \( P_{G_{max}} \) are the minimum and maximum limit of the \( i^{th} \) generating unit.

### III. GRAVITATIONAL SEARCH ALGORITHM

The algorithm was proposed by Rashidi et al. [3] that uses the Newton’s Gravitational Principle to search the optimum solution. In this algorithm, the coordinates or the agents in the search space are considered as masses. All these masses attract each other according to the laws of gravity and form a direct means of communication through it. The agents or particles in the algorithm follow the following two principles:

1. **Law of Gravity**: Each particle attracts the other particle with a force which is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. In GSA we take the force to be inversely proportional to the distance between them as this has been found to give better results.

2. **Law of Motion**: Current velocity of any particle is equal to the sum of the fraction of its previous velocity and variation in velocity.

Newton’s equations can be expressed as:

\[ F_i = G \frac{M_a M_b}{R^2} \]  

(5)

\[ a_i = \frac{F_i}{M_a} \]  

(6)

Where \( M_a \) is the active mass of particle \( j \), \( M_b \) is the passive mass of the particle \( i \) [6], \( R \) is the distance between the particles, \( a_i \) is the acceleration of the \( i^{th} \) particle.

Consider a system with \( N \) agents and the position of the \( i^{th} \) agent being defined by the coordinates:

\[ X_i = (x_{i1}, x_{i2}, ..., x_{iD}) \]  

for \( i = 1, 2, ..., N \)  

(7)

Where, \( x_{id} \) is the position of the \( i^{th} \) agent in the \( d^{th} \) dimension.

At any time instant the gravitational force acting on the \( i^{th} \) particle due to the effect of \( j^{th} \) particle is given by:

\[ F_{ij}(t) = G(t) \frac{M_a M_b}{R_{ij}^2} (x_{jd}(t) - x_{id}(t)) \]  

(8)

\[ F_{ji}(t) = G(t) \frac{M_a M_b}{R_{ji}^2} (x_{id}(t) - x_{jd}(t)) \]  

(9)

Where, \( G(t) \) is gravitational constant at time \( t \), \( \epsilon \) is a small constant, \( R_{ij}(t) \) is the Euclidian distance between the particles (or agents) \( i \) and \( j \):

\[ R_{ij}(t) = \left| x_i(t) - x_j(t) \right| \]  

(10)

To give a random nature to the search, the total force acting on the system is given by the weighted sum of the forces acting on the particle in the \( d^{th} \) dimension due to all the other particles:

\[ F_{ij}(t) = \sum_{j=1, j \neq i}^{N} a_{ij}(t) \]  

(11)

Where, \( rand_j \) is a random number in the interval [0,1].

Now, the acceleration of this agent in the \( d^{th} \) dimension can be known by the equation:

\[ a_{id}(t) = \frac{F_{id}(t)}{M_i} \]  

(12)

Where, \( M_i \) is the inertial mass of the \( i^{th} \) particle or agent.

The velocity and position of the agents can be determined by the following equations:

\[ v_{id}(t+1) = v_{id}(t) + a_{id}(t) \]  

(13)

\[ x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \]  

(14)

Where, \( rand_i \) is a random number in range [0, 1] to give a random characteristic to the search.

Gravitational Constant will reduce with time to control the speed and accuracy of the search, while its initial value shall be given earlier.

\[ F(t) = G(t_0) \]  

(15)

where, \( G(t) \) is the Gravitational Constant at some time interval \( t \), and \( G(t_0) \) is the Gravitational Constant at the first cosmic interval \( t_0 \).

Where \( G_0 \) is the initial value of Gravitational Constant at the first cosmic time interval.

Fitness evaluation gives the value to the gravitational and inertial masses, a higher fitness shows more efficient agent, resulting in more attraction and slow movement. We calculate the masses using the following equations:

\[ M_i = M_{g0} - M_{g1} = M_{g2} \]  

(17)

\[ m_i(t) = \frac{fit(t) - \text{worst}(t)}{\text{best}(t) - \text{worst}(t)} \]  

(18)

\[ M_{i}(t) = \frac{m_i(t) M_{g1}}{m_1(t)} \]  

(19)

where, \( fit(t) \) is the fitness value and determined by the function being optimized. The worst(t) and best(t) are given by the following equations:

\[ \text{worst}(t) = \left\{ \begin{array}{ll} \min_{t \leq t \leq \text{end}} \text{fit}(t) & \text{for minimization} \\ \max_{t \leq t \leq \text{end}} \text{fit}(t) & \text{for maximization} \end{array} \right. \]  

(20)

Initially all the agents apply force, this is the time exploration is going on. For better results in the final part of the search the exploration fails and exploitation starts where only few, more efficient agents continue to apply force. In order to avoid into local minima we go for exploration initially and...
gradually shift to exploitation where only $K_{best}$ agents apply force. This $K_{best}$ set is decreased gradually to make the change.

$$F^i(x) = \sum_{j \in Leader_{i}} w_{r} F^i_{j}(x) \quad and \quad F_{j}^i(x)$$

Where, $K_{best}$ is the set of more efficient agents with more masses. This means near the end of the iteration nearly 2% of the initial particles shall only apply force.

IV. CLUSTERED APPROACH TO GSA

To keep the exploration in GSA alive without killing the exploitation, in this paper, a group method is proposed. Here, the whole population is divided into three basic groups: namely the Leader, the follower, and the freelancer. The Leaders are the best particles obtained at the end of the first iteration. Each leader particle shall lead a group of optimizers. The Leader and the optimizer group together shall work like a simple GSA population thereafter. In this way, there would be some independent GSA populations led by their leader that will search for the optimum solution. The last group, the freelancers shall be randomly initiated every iteration and in this way, they shall keep the search alive. Each group those led by a leader and the freelancers shall have a best particle. The best out of these bests shall be the final best particle of the iteration. Depending on the requirements of the function, the ratio of the population of Leader, follower and the freelancer can be adjusted. The pseudo-codes are given below:

Step 1. Initialize the random population of agents.
Step 2. Evaluate the population on the given function.
Step 3. Sort the population on the fitness values.
Step 4. The first 10% of the population (size can vary according to function) is called the Leader.
Step 5. The next 80% of the population (size can vary according to function) is called the follower.
Step 6. The last 10% of the population (size can vary according to the function) is called the freelancer.
Step 7. To each of the agent in the leader group a set of agents from follower group is allotted and they together make a single sub-population.

for $i=1$ to max no of iterations
{ for $j=1$ to max no of sub-population
{ Run GSA for each of the sub-group.
}
Evaluate each particle in the freelancer group.

Find the minimum (according to the requirements of the problem) of the best fitness values obtained among all the sub groups and the freelancer group.

Again randomly initialize the entire freelancer population. (But see to it that the best particle is from the freelancer group, then that is not deleted in the next iteration.)

Step 8. The best population is thus the minimum of all the best fitness values at the end of the iterations.

Now there is no form of communication between the subpopulations so they would always be searching independently (and exploring), rather than exploiting at the same place. The problem of exploitation is solved by the subgroups themselves. They exploit their local search spaces to get a minimum out of that region while since they do not communicate the exploration happens.

V. IMPLEMENTATION OF CGSA FOR EDP

In this section, the implementation of CGSA for EDP is given in flowchart as shown in Fig. 1.

Fig. 1 Flowchart for the proposed algorithm
VI. SIMULATION RESULTS

To evaluate the performance of our algorithm, EDP is solved using CGSA for IEEE 30 bus system and 40 unit system. All the programs are coded using MATLAB 7.10 and the system configuration is I7 processor with 2.2GHz speed and 4GB RAM.

4.1 IEEE 30 bus system: 6 units

The IEEE 30 bus system consists of 6 generating unit and unit data is adapted from the ref. [9]. The proper tuning of the CGSA parameter will increase the convergence speed and solution quality. By trial and error method, the CGSA parameters are tuned and the optimal parameter that used to obtain the best solution is given below.

Population size: 140  
No. of iterations: 2000  
G0=100  
Alfa=20  
No of clusters: 8  
No of leaders: 14  
No of followers: 112  
No of free lancers: 14

In IEEE 30 bus system, two different case studies have been carried out. In case 1, the EDP is solved without considering the system losses and in case 2 the EDP is solved considering the system losses.

4.1.1 Case 1

Out of 10 trials, the best solution obtained for IEEE 30 bus system using CGSA is given in Table X. The comparison of result is given in Table 2. The proposed method gives better results than the other method GSA, NSGA [9], NPGA [10], SPEA [11] and MBFA [12]. The convergence graph for GSA and CGSA are shown in Figures 2 and 3. From the graph it is clear that the convergence speed of the CGSA is faster than GSA.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Power Generated (MW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PG1</td>
<td>10.9735</td>
</tr>
<tr>
<td>PG2</td>
<td>29.980</td>
</tr>
<tr>
<td>PG3</td>
<td>52.4160</td>
</tr>
<tr>
<td>PG4</td>
<td>101.6320</td>
</tr>
<tr>
<td>PG5</td>
<td>52.4234</td>
</tr>
<tr>
<td>PG6</td>
<td>35.9750</td>
</tr>
</tbody>
</table>

4.1.2 Case 2

In this section the EDP is solved by considering the system losses. Here, the loss of the 30 bus system is calculated using the Korn’s formula. The B coefficients are adapted from the ref. [9]. The comparison of results is given in Table 3. The proposed method shows the better results than the other methods available in the literature. This shows the superiority of the proposed algorithm. The convergence graph for GSA and CGSA are shown in Figures 4 and 5. From the graph it is clear that the convergence speed of the CGSA is faster than GSA.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PG1</td>
<td>10.9735</td>
<td>39.37 09</td>
<td>15.67</td>
<td>10.80</td>
<td>10.62</td>
<td>11.33</td>
</tr>
<tr>
<td>PG2</td>
<td>29.980</td>
<td>14.06</td>
<td>28.70</td>
<td>32.84</td>
<td>28.97</td>
<td>30.05</td>
</tr>
</tbody>
</table>
Table 3 Comparison of results - case 2

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PG1, MW</td>
<td>7.7216</td>
<td>15.8401</td>
<td>14.47</td>
<td>14.25</td>
<td>12.79</td>
<td>10.38</td>
</tr>
<tr>
<td>PG2, MW</td>
<td>25.6274</td>
<td>48.0028</td>
<td>30.66</td>
<td>26.93</td>
<td>31.63</td>
<td>27.43</td>
</tr>
<tr>
<td>PG3, MW</td>
<td>60.6940</td>
<td>76.7784</td>
<td>54.93</td>
<td>59.08</td>
<td>58.03</td>
<td>57.33</td>
</tr>
<tr>
<td>PG4, MW</td>
<td>96.3281</td>
<td>75.01</td>
<td>98.94</td>
<td>99.44</td>
<td>95.80</td>
<td>94.26</td>
</tr>
<tr>
<td>PG5, MW</td>
<td>53.7299</td>
<td>14.4034</td>
<td>52.44</td>
<td>53.15</td>
<td>52.58</td>
<td>53.84</td>
</tr>
<tr>
<td>PG6, MW</td>
<td>41.5531</td>
<td>56.008</td>
<td>35.42</td>
<td>33.92</td>
<td>35.89</td>
<td>42.75</td>
</tr>
<tr>
<td>Fuel cost $/hr</td>
<td>606.20</td>
<td>628.57</td>
<td>607.98</td>
<td>608.06</td>
<td>607.86</td>
<td>606.82</td>
</tr>
</tbody>
</table>

Fig. 4. Convergence characteristic of GSA-Case 2

4.2 40 units system

System consists of 40 generating units and the unit data is adapted from ref. [13]. For 40 unit system, the best tuned CGSA parameters are given below.

Population size: 200
No. of iterations: 2000
G0=100
Alpha=20
No of clusters: 8
No of leaders: 20

Here, EDP is solved for 40 unit system without considering the system loss which is the largest test available in the literature. Out of 10 trials the best result obtained using CGSA algorithm is given in Table 4. The comparison of results is given in Table 5. From the Table 5, it is clear that the proposed method has the capable of producing the better results than the other method available in the literature.

Table 4 Optimum dispatch

<table>
<thead>
<tr>
<th>Unit no.</th>
<th>CGSA</th>
<th>Unit no.</th>
<th>CGSA</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>111.0060</td>
<td>P21</td>
<td>523.2794</td>
</tr>
<tr>
<td>P2</td>
<td>110.8076</td>
<td>P22</td>
<td>523.2794</td>
</tr>
<tr>
<td>P3</td>
<td>97.4006</td>
<td>P23</td>
<td>523.2828</td>
</tr>
<tr>
<td>P4</td>
<td>179.7325</td>
<td>P24</td>
<td>523.2794</td>
</tr>
<tr>
<td>P5</td>
<td>87.8518</td>
<td>P25</td>
<td>523.2794</td>
</tr>
<tr>
<td>P6</td>
<td>140.0000</td>
<td>P26</td>
<td>523.2794</td>
</tr>
<tr>
<td>P7</td>
<td>259.6006</td>
<td>P27</td>
<td>10.0005</td>
</tr>
<tr>
<td>P8</td>
<td>284.6053</td>
<td>P28</td>
<td>10.0000</td>
</tr>
<tr>
<td>P9</td>
<td>284.6013</td>
<td>P29</td>
<td>10.0000</td>
</tr>
<tr>
<td>P10</td>
<td>130.0000</td>
<td>P30</td>
<td>92.4639</td>
</tr>
<tr>
<td>P11</td>
<td>168.8002</td>
<td>P31</td>
<td>190.0000</td>
</tr>
<tr>
<td>P12</td>
<td>168.7992</td>
<td>P32</td>
<td>190.0000</td>
</tr>
<tr>
<td>P13</td>
<td>214.7597</td>
<td>P33</td>
<td>190.0000</td>
</tr>
<tr>
<td>P14</td>
<td>304.5200</td>
<td>P34</td>
<td>164.8026</td>
</tr>
<tr>
<td>P15</td>
<td>394.2789</td>
<td>P35</td>
<td>164.8017</td>
</tr>
<tr>
<td>P16</td>
<td>394.2783</td>
<td>P36</td>
<td>164.8123</td>
</tr>
<tr>
<td>P17</td>
<td>489.2797</td>
<td>P37</td>
<td>110.0000</td>
</tr>
<tr>
<td>P18</td>
<td>489.2794</td>
<td>P38</td>
<td>110.0000</td>
</tr>
<tr>
<td>P19</td>
<td>511.2798</td>
<td>P39</td>
<td>109.9999</td>
</tr>
<tr>
<td>P20</td>
<td>511.2792</td>
<td>P40</td>
<td>511.2787</td>
</tr>
</tbody>
</table>

Table 5 Comparison of results - Forty unit system

<table>
<thead>
<tr>
<th>Optimization technique</th>
<th>Minimum cost $</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO[14]</td>
<td>123930.45</td>
</tr>
<tr>
<td>CEP[13]</td>
<td>123488.29</td>
</tr>
<tr>
<td>FEP[13]</td>
<td>122679.71</td>
</tr>
<tr>
<td>MFEP[13]</td>
<td>122647.57</td>
</tr>
<tr>
<td>IFEP[13]</td>
<td>122624.35</td>
</tr>
<tr>
<td>TM[14]</td>
<td>122477.78</td>
</tr>
<tr>
<td>MSL[14]</td>
<td>122406.10</td>
</tr>
<tr>
<td>MPSO[14]</td>
<td>122252.27</td>
</tr>
<tr>
<td>ESO[14]</td>
<td>122122.16</td>
</tr>
<tr>
<td>PSO-SQP[14]</td>
<td>122094.67</td>
</tr>
<tr>
<td>AIS binary[15]</td>
<td>121760.58</td>
</tr>
<tr>
<td>DEC(2)-SQP(1)[14]</td>
<td>121741.98</td>
</tr>
<tr>
<td>NPSO-LRS[14]</td>
<td>121664.43</td>
</tr>
<tr>
<td>CSO[15]</td>
<td>121461.67</td>
</tr>
<tr>
<td>Continuous GA[15]</td>
<td>121523.10</td>
</tr>
<tr>
<td>AIS real[15]</td>
<td>121458.18</td>
</tr>
<tr>
<td>ABC[15]</td>
<td>121441.03</td>
</tr>
<tr>
<td>CGSA</td>
<td>121414.90</td>
</tr>
</tbody>
</table>
CONCLUSION

The group method was implemented on the GSA and the results were compared with the traditional GSA. It was noticed that as the iterations go on, all the agents accumulate near the Best particle and in this way we have a loss of exploration. Though this increases exploitation and works very well for low modality functions but this comes out to be a disadvantage in case of multi dimensional functions. Well it is a matter of fact that some functions demand more exploration and some more exploitation. Hence, in this paper, a clustered based gravitational search algorithm is proposed. This algorithm maintains the right balance between the exploration and exploitation. The proposed algorithm is tested on non linear and non smooth economic dispatch problem of power system. The simulation results demonstrate the effectiveness and robustness of the proposed algorithm to solve ED problems of the test power systems. Moreover, the results of the proposed algorithm have been compared to those surfaced in the recent literature. The comparison results confirm the effectiveness and the superiority of the proposed clustered approach.

REFERENCE