CHAPTER 7

PROPOSED HYBRID EIPSO METHOD FOR
SOLVING MULTI AREA UNIT COMMITMENT PROBLEM

7.1 INTRODUCTION

The importance of a more effective optimal solution to the UCP problem is increasing with the regularly varying demand. Hence, we propose a hybrid technique which solves the unit commitment problem subjected to the necessary constraints, and gives the optimal commitment of the units. The possible combination of demand and its corresponding optimal generation schedule can be determined by the PSO algorithm. Being a global optimization technique, evolutionary for solving the unit commitment problem operates on a method which encodes each unit’s operating schedule with respect to the up/down time. When the demand over a time horizon is given as input to the network, it successfully gives the schedule of each unit’s commitment that satisfies the demands of all the periods and results in the minimum total cost. Because of the hybridization dominates this technique for solving the unit commitment problem is more effective.

The proposed hybrid intelligence technique for UCP utilizes the PSO Algorithm and EP. The PSO is used to determine the units and their optimum generation schedule for a particular demand at minimum cost. Evolutionary Programming assisted by the PSO is used to determine the unit commitment that minimizes the cost for different possible demands. Based on the previous period demand, the Evolutionary Programming technique
determines the optimal schedule that satisfies the current period demand. Thus, the problem is divided into two stages; one for determining the unit commitment for a particular demand, and the other for determining the unit commitment for all the periods that results reducing the cost to the minimum. As the demand varies with time, the demand is different for each period and hence, different possible demands need to be optimized, this can be performed by EP.

Evolutional Iteration Particle Swarm Optimization (EIPSO) combines PSO with EP to enhance the computation efficiency of EP and enable the PSO to jump out of the local optimal. In particular PSOs have faster convergence rate than EP early in the run, but often they are outperformed by EP for long simulation runs when the last ones find a better solution. Anyway, the population-based representation of the parameters that characterizes a particular solution is the same for both the algorithms. Therefore, it is possible to implement the hybrid technique, in order to utilize the qualities and uniqueness of the two algorithms. Some attempts have been made in this direction with good results but with the weak integration of the two strategies because one algorithm is used mainly as the pre-optimizer for the initial population of the other one.

The new hybrid technique proposed here, called Evolutionary Iteration Swarm Optimization, consists in a strong co-operation of EP and PSO. Since, it maintains the integration of the two techniques for the entire run. In each iteration, the population is divided into two parts, and it is evolved with the two techniques respectively. The two parts are then recombined in the updated population that is again divided randomly into another two parts in the next iteration for another run of the evolutionary or particle swarm operators.
Initialize EP & PSO Particles

Evaluate the Individuals using Fitness Function

EP
Create the off strings and Discard the parents

PSO
Estimate the velocity and modify each searching points

Perform selection and cross over using EP procedure and list them according to their rank

Update the local and global bests and increment the iteration count

Is the best individual of the present iteration is better than the best individual of previous iteration
Is the Duality Gap

STOP

Figure 7.1 Flow chart of the EIPSO method
The hybrid EIPSO procedure is initialized with a single set of individuals, M (or particles). These individuals will be treated by both the EP and PSO operators independently. The EP and PSO generate 2M individuals, which are done in combination with the selection and competition of the EP procedure, and the individuals are ranked. The procedure will be repeated until a termination criterion is satisfied. The flow chart of proposed EIPSO method is shown in Figure 7.1.

7.2 ALGORITHM OF EIPSO METHOD

The main computational steps of EIPSO algorithms are described as follows.

Step 1 : Get the data for the system.

Step 2 : Randomly initialize a set of individuals and iteration count \( t = 0 \).

Step 3 : Evaluate the fitness function and update the inertia weight.

Step 4 : Create offspring using the EP and modify the search points using the PSO for the same set of individuals.

Step 5 : Perform competition and selection of the EP procedure using the above two sets of individuals.

Step 6 : If the current first-rank individual is better than the first-ranked of the previous iteration.
Step 7: Otherwise, increment \( t = t + 1 \) and modify the velocities and searching points for the PSO.

Step 8: Exit if termination criterion is met; otherwise go to Step 3.

The termination is done when the maximum iteration of \( t_{\text{max}} \) is reached.

### 7.3 HYBRID PROPOSED EIPSO METHOD FOR MAUC PROBLEM

A way of solving the multi-area unit commitment problem is to solve it in two phases. In the first phase, the MAUC problem is formulated taking into some of the constraints and unit commitment is done by simple priority list method. In the next phase, the economic dispatch problem is solved by using proposed EIPSO method. The various steps of the algorithm are given below.

#### 7.3.1 Initialization of the Parent Population

Step 1: Read the generating unit data and demand profile.

Step 2: Perform the simple priority list method to get the initial commitment schedule for each area.

The elements in the parent are generating unit active power for all areas. The initial parent population of size \( N_p \) is generated randomly follows:

a) To generate the initial parent population

\[
I_{p1} = [(P_1^{kp} \ldots P_N^{kp})]; k = 1,2,3,4; p = 1,2..N_p
\]  

(7.1)
(b) To calculate the fuel cost for each population

$$F_{C}^{k} = [(a(P_{1}^{k})^{2} - b_{1}(P_{1}^{k}) + c); K = 1, 2...4; P_{1}^{k} = 1, 2...N_{P}]$$

(7.2)

(c) To calculate the start up cost for each population

(d) To calculate the production cost

$$\text{Production Cost} = F_{C}^{k} + S_{C}^{k}$$

(7.3)

(e) To calculate the fitness function for each parent of population

$$F_{p_{1}}^{k} = F_{C}^{k} + S_{C}^{k} + K(\sum_{i=1}^{N_{k}} P_{i}^{k} - D_{j}^{k})$$  

(7.4)

These values of the penalty factor are chosen, such that if there are any constraint violations, then the fitness function value corresponding to that parent will be ineffective.

### 7.3.2 Mutation

(a) To generate an offspring population $I_{o}$ of size $N_{P}$ from each parent $I_{p_{1}}$ are generated as

$$I_{O} = [P_{1}^{k0}, ..., P_{N}^{k0}, k = 1, 2...4; O = 1, 2...N_{P}]$$

(7.5)

$$p_{g_{i}}^{k0} = P_{g_{i}}^{k0} + N(0, \sigma^{2}P_{g_{i}}^{k0})$$

(7.6)

A similarly all $P_{i}$ is generated for all areas subject to

$$P_{g_{i}}^{k0} = (P_{g_{i}}^{0}, P_{g_{i}}^{0}, ...; P_{g_{i}}^{0}, P_{g_{i}}^{0})$$

$$P_{g_{i}}^{0}, P_{g_{i}}^{0}, ..., P_{g_{i}}^{0}$$
\( P_{i,max} ; \text{if} \quad P_{i}^{k_o} < P_{i,max} \) (7.7)

\( N(0, \sigma^2) \) represents a normal random variable with zero mean and standard deviation \( \sigma \). The standard deviation is computed as

\[
\sigma_{P_i} = \beta \left( \frac{F_{P_i}}{F_{max}} \right)^{*} (\sigma_{ij,max} - \sigma_{ij,min})
\] (7.8)

Where \( \beta \) is a scaling factor, \( F_{P_i} \) is the value of the fitness function corresponding to \( I_i \) and \( F_{max} \) is the maximum fitness function value of the parent population.

(b) To compute the fitness function value corresponding to each offspring using Equation (7.4)

### 7.3.3 Initialization of Particle

The elements in the particle are the generating unit active power for all areas. The initial particle of size 2Np is generated randomly as follows:

(a) To generate the initial particle population

\[
I_{P_2} = [(P_1^{k_1}, ..., P_N^{k_2}) : K = 1, 2, 3, 4]
\] (7.9)

(b) To calculate the fuel cost for each particle using the below equation

\[
F_{C_{P_2}}^k = [a(P_1^{k_2})^2 + b(P_1^{k_2}) + c] : K = 1, 2, 3, 4
\] (7.10)

(c) To calculate the start up cost for each particle using equation
(d) To calculate the production cost

\[
\text{Production cost} = mK_P^C + SC_P^K
\]  
(7.11)

(e) To calculate the fitness function for each particle of population

\[
F_{P_i} = F_{C_1}^k + SC_{P_i}^k + K\left( \sum_{j=1}^{N_k} kP_j^2 - D_j^k \right)
\]
(7.12)

The values of the penalty factor are chosen, such that if there are any constraint violations, then the fitness function value corresponding to that parent will be ineffective.

(f) To calculate the Pbest by using the fitness function values, if the current value is better than the pbest, then set the pbest value equal to the current value. The previous Pbest value is replaced by the current Pbest value and the gbest is computed. If the current value is better than the gbest, then reset to the current particles.

7.3.4 Updating the Velocity

The velocity is updated by considering the current velocity of the particles, and the best fitness function value among the particles in the swarm using the following equation.

\[
V_{i}^{2} = \omega V_{i}^{2} + c_1 \text{Rand}(p_{i}^{2} - p_{i}^{2}) + c_2 \text{Rand}(g_{i}^{2} - p_{i}^{2})
\]
(7.13)

where \( \omega \) is weight factor. The weight W is computed.
7.3.5 Competition and Selection

The $4N_p$ trial solution, $N_p$ solution corresponding to parent population and their offspring population computed survival, the remaining, $N_p$ solution corresponding to particle in the swarm. The weight $W_i$ of each individual in the combined population decides its survival. The weight $W_i$ is computed as

$$W_i = \frac{1}{\sum_{t=1}^{q} w_{t,i}}$$

(7.14)

$$W_i = \begin{cases} 1, & \text{if} \ U_i > (F_{i,t}/F_{1,t} + F_{r,t}), 0, & \text{if otherwise} \end{cases}$$

where the competitor $F_r$ is selected at random from among $2N_p$ trial solutions, $U_i$ is a uniform random number ranging from 0 to 1 and $q$ is the arbitrarily chosen competition number, which is normally taken as $N_p$. The $4N_p$ trial solutions are ranked in descending order of their weights and the first $N_p$ trial solution are taken as the next parent population, and next $N_p$ solutions are taken as the nest particle population. The steps described in the sub sections a and c are repeated, until the maximum generation count is reached.

7.4 RESULTS AND DISCUSSION

The proposed MAUC algorithm has been implemented in a C++ environment and tested extensively. The test results of a multi-area system are presented in this section.

The proposed algorithm has been tested with four areas system and each area has 26 thermal generating units (Ouyang and Shahidpour 1991). The total number of units tested is 104, and their characteristics are presented in Appendix 2. There are identical units in each area. The load demand profile forecasts used in all four areas are presented in Appendix 3. The test are used
to implement the proposed algorithm includes import/export capability and tie line capacity constraints.

**Table 7.1  Operating cost of the proposed EIPSO method for MAUC problem**

<table>
<thead>
<tr>
<th>Hours (24)</th>
<th>Area-1 Cost ($)</th>
<th>Area-2 Cost ($)</th>
<th>Area-3 Cost ($)</th>
<th>Area-4 Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>39587.54</td>
<td>23224.69</td>
<td>43342.30</td>
<td>30231.03</td>
</tr>
<tr>
<td>2</td>
<td>15543.32</td>
<td>41721.69</td>
<td>27163.14</td>
<td>26696.92</td>
</tr>
<tr>
<td>3</td>
<td>25652.26</td>
<td>47172.17</td>
<td>19097.68</td>
<td>12052.87</td>
</tr>
<tr>
<td>4</td>
<td>17209.43</td>
<td>12709.15</td>
<td>30407.70</td>
<td>13687.06</td>
</tr>
<tr>
<td>5</td>
<td>16394.16</td>
<td>21496.50</td>
<td>19808.80</td>
<td>31096.54</td>
</tr>
<tr>
<td>6</td>
<td>25112.67</td>
<td>14911.77</td>
<td>36500.26</td>
<td>32962.37</td>
</tr>
<tr>
<td>7</td>
<td>16412.82</td>
<td>36967.25</td>
<td>37404.76</td>
<td>29148.08</td>
</tr>
<tr>
<td>8</td>
<td>13066.13</td>
<td>10533.84</td>
<td>37428.88</td>
<td>44959.19</td>
</tr>
<tr>
<td>9</td>
<td>14392.29</td>
<td>22571.36</td>
<td>21492.26</td>
<td>23842.93</td>
</tr>
<tr>
<td>10</td>
<td>20612.82</td>
<td>32664.07</td>
<td>23172.62</td>
<td>10744.30</td>
</tr>
<tr>
<td>11</td>
<td>26974.41</td>
<td>22050.36</td>
<td>19175.40</td>
<td>12110.45</td>
</tr>
<tr>
<td>12</td>
<td>27722.14</td>
<td>24870.05</td>
<td>22004.14</td>
<td>25539.92</td>
</tr>
<tr>
<td>13</td>
<td>28045.81</td>
<td>35220.92</td>
<td>20973.68</td>
<td>18409.34</td>
</tr>
<tr>
<td>14</td>
<td>26403.35</td>
<td>20598.11</td>
<td>29093.16</td>
<td>11683.06</td>
</tr>
<tr>
<td>15</td>
<td>52039.25</td>
<td>21319.87</td>
<td>33134.18</td>
<td>13863.81</td>
</tr>
</tbody>
</table>
Table 7.1 (Continued)

<table>
<thead>
<tr>
<th></th>
<th>Area-1 Cost ($)</th>
<th>Area-2 Cost ($)</th>
<th>Area-3 Cost ($)</th>
<th>Area-4 Cost ($)</th>
<th>Total Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>13703.08</td>
<td>38961.41</td>
<td>13219.10</td>
<td>17421.47</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>10915.20</td>
<td>29403.63</td>
<td>18594.76</td>
<td>30299.99</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>31951.65</td>
<td>7286.829</td>
<td>10773.60</td>
<td>40318.92</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>35390.50</td>
<td>31797.29</td>
<td>28512.94</td>
<td>21198.55</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>37502.71</td>
<td>22593.08</td>
<td>18282.54</td>
<td>25809.97</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>21101.01</td>
<td>13531.79</td>
<td>24003.42</td>
<td>17305.37</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>26568.05</td>
<td>10370.32</td>
<td>12354.80</td>
<td>14573.65</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>35449.25</td>
<td>24415.81</td>
<td>24258.02</td>
<td>31230.04</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>10530.06</td>
<td>25744.66</td>
<td>15829.42</td>
<td>42541.14</td>
<td></td>
</tr>
<tr>
<td>Total cost ($)</td>
<td>588279.90</td>
<td>592136.60</td>
<td>586027.60</td>
<td>577727.00</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.2  Comparison Result of DP, EP, PSO and EIPSO method

<table>
<thead>
<tr>
<th>Method</th>
<th>Area- 1 Cost ($)</th>
<th>Area- 2 Cost ($)</th>
<th>Area- 3 Cost ($)</th>
<th>Area- 4 Cost ($)</th>
<th>Total Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DP</td>
<td>590383.69</td>
<td>574583.80</td>
<td>560621.10</td>
<td>627072.10</td>
<td>2352660.80</td>
</tr>
<tr>
<td>EP</td>
<td>585740.10</td>
<td>588939.60</td>
<td>602163.30</td>
<td>572357.50</td>
<td>2349201.00</td>
</tr>
<tr>
<td>PSO</td>
<td>582367.90</td>
<td>586392.40</td>
<td>605628.80</td>
<td>572396.00</td>
<td>2346785.00</td>
</tr>
<tr>
<td>EIPSO</td>
<td>588279.90</td>
<td>592136.60</td>
<td>586027.60</td>
<td>577727.00</td>
<td>2344171.00</td>
</tr>
</tbody>
</table>
Figure 7.2 Convergence characteristics of EIPSO method

A comparative study is also carried out to illustrate the different solutions obtained based on different problem formulations. The simulation results are obtained for operating cost of the four areas using Evolutionary Iteration Particle Swarm Optimization methods are indicated in Table 7.1. Table 7.2 shows comparison result of conventional and Evolutionary programming and PSO method and EIPSO method. The proposed method would save 0.36% when compared to DP method and would save 0.21% compared to EP method. When it compares to PSO method it would save 0.11%. Figure 7.2 shows the convergence characteristics for multi-area obtained using proposed methodology.
7.5 SUMMARY

The new hybrid technique proposed here, called Evolutionary Swarm Optimization, consist in a strong co-operation of EP and PSO. Since, it maintains the integration of the two techniques for the entire run. In each iteration, the population is divided into two parts and it is evolved with the two techniques respectively. The two parts are then recombined in the updated population that is again divided randomly into another two parts in the next iteration, for another run of the evolutionary or particle swarm operators.

The proposed hybrid technique using EI and PSO has performed well in solving the UCP by recognizing the optimal generation schedule. The technique has been tested for the system with the consideration of load balance and spinning reserve constraints which are the most significant constraints. For the test demand set which consists of the demand for 24 periods, the hybrid approach effectively yields optimal generation schedule for the periods. Moreover, the test results were compared for the proposed hybrid approach with the discrete performance of the PSO. The hybridization of the EI with the PSO makes the solution for UCP more efficient. Hence, the technique gives the optimal commitment of units for any demand, so that the schedule of commitment satisfies the defined constraints as well as the demand at the minimum cost.