The Comparison of Chaotic Optimization Algorithm and Other Evolutionary Techniques for Optimal Unit Commitment of Power System

1R. Jahani and 2H. Chahkandi Nejad

1Islamic Azad University, Birjand Branch, Birjand, Iran.
2Electrical Engineering Department, Islamic Azad University, Birjand Branch, Birjand, Iran.

Abstract: The solution of the unit commitment problem (UCP) is a complex optimization problem. The exact solution of the UCP can be obtained by a complete enumeration of all feasible combinations of generating units, which could be a huge number. The objective is to find the generation scheduling so that the total operating cost can be minimized, when subjected to a variety of constraints. It also means that it is desirable to find the optimal generating unit commitment in the power system for the next hours. This paper presents a Chaotic Optimization Algorithm (COA) to solve optimal Unit Commitment Problem (UCP). The COA is applied to the widely used ten-unit test system and its multiples (10-100). Comparing our results with those of many UC solving methods demonstrate that not only the COA procedure consider is the effective constraints but it also has some advantages such as good convergence, fast calculating speed and high precision.

Key words: Chaotic Optimization Algorithm (COA); Power System; Unit Commitment Problem (UCP); System Constraints.

INTRODUCTION

In all power stations, investment is quite expensive and the resources needed to operate them are increasingly becoming more scarce. As a result, the main concern today is optimizing the operating cost of power stations. In the modern world, meeting the power demand as well as optimizing generation has become a necessity. Unit commitment in power systems refers to the optimization problem for determining the on/off states of generating units that minimizes the operating cost for a given time horizon. The solution of the unit commitment problem is a complex optimization problem. The exact solution of the UCP can be obtained by complete enumeration of all feasible combinations of generating units, which is a huge number. The unit commitment is commonly formulated as a nonlinear, large scale, mixed integer Combinational optimization problem. Several solution techniques have been applied to this problem, either by using deterministic, meta heuristic, and hybrid approaches. Deterministic approaches include priority list (PL) (A. J. Wood., & B. F. Wollenberg. 1996), dynamic programming (DP) (C. K. Pang., & H. C. Chen. 1976), Lagrangian Relaxation (LR) (S. J. Wang, et al. 1995), integer mixed-integer programming (T. S. Dillon., & K. Wedwin. 1999), (J. A. Muckstadt., & R. C. Wilson. 1968), and the branch-and-bound methods The priority list is the simplest and the fastest but results in poor final solution (A. I Cohen., & M. Yoshimura. 1983). Meta-heuristic approach, such as genetic algorithm (GA) (S. A. Kazarlis, et al. 1996), (H. Ma, et al. 1994), evolutionary programming (EP) (K. A. Juste, et al. 1999), simulated annealing (SA) (F. Zhuang., & F. D. Galiana. 1990), tabu search (TS) (A. H. Mantawy, et al. 1999), particle swarm optimization (PSO) (Lee, T., & Chen, C. 2007), greedy random adaptive search procedure (GRASP) (Viana, A, et al. 2003) are also being widely investigated to solve the UC problem. These meta-heuristic methods optimization methods attract much attention, because of their ability to search not only local optimal solution but also global optimal solution and can easily deal with various difficult nonlinear constraints. However, these meta-heuristic methods require a considerable amount of computational time to find the near-global minimum especially for a large-scale UCP. In this paper Chaotic Optimization Algorithm (COA) is proposed to solve the UCP.

During the present paper , an optimization-based tuning algorithm is also proposed to optimal Unit Commitment Problem. This algorithm optimizes the total system performance by means of COA method. Chaos is a kind of characteristic of non-linear systems which is a bounded unstable dynamic behavior that exhibits sensitive dependence on initial conditions and include infinite unstable periodic motions. The COA is based on ergodicity, stochastic properties and regularity of chaos. It is not like some stochastic optimization algorithms that escape from local minima by accepting some bad solutions according to a certain probability but COA searches on the regularity of chaotic motion to escape from local minima (Coelho, L. D. S, 2007).

UC Problem Formulation:

Objective Function:
In this paper, it is assumed that the schedule periods are 24 hours and divide into 24 time-steps. The total cost is the sum of the running cost and start up cost for all units over the whole scheduling periods. Accordingly, overall objective function of the UC problem is:

$$\text{Min}\ F(U_i, P_i) = \sum_{i=1}^{24} \sum_{t=1}^{G} [U_i F_i(P_i) + U_i (1-U_i) S_i]$$

(1)

Generally, the running cost per unit in any given time interval is a function of the generator power output. The cost function is usually in the form of:

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i$$

(2)

The generator start up cost depends on the time the unit has been off prior to the start up. Time-dependent start up cost is represented as follows:

$$S_i = S_{0i} + S_i (1 - e^{-\frac{t_i}{T_i}})$$

(3)

The shut down cost is usually given a constant value for each unit. The shut down cost has been taken equal to 0 for each unit.

### System Constraints:

Many constraints can be applied on the unit commitment problem. Each individual power system, power pool, reliability council, may impose different rules on the scheduling of the units, depending on the generation makeup, load-curve characteristics. Spinning reserve describes the total amount of generation available from all units synchronized on the system, minus the present load supplied and losses being incurred. Spinning reserve must be carried out in such a way that the loss of one or more units does not cause too far a drop in the system frequency. Spinning reserve must obey certain rules which specifies that reserve must be capable of making up the loss of heavily loaded unit in a given period of time. Reserve requirement are also calculated as a function of the probability of not having sufficient generation to meet the load, by making people (T. Senjyu, et al. 2006).

1) Power balance constraint

$$\sum_{i=1}^{G} U_i P_{it} = P_{dt}, t = 1, 2, 3,...$$

(4)

$P_{it}$ is calculated by the running units at time-step $t$ according to equal loss incremental rate principle and met (meet):

$$\frac{dF_{it}}{dP_{it}} = \frac{dF_{it}}{dP_{it}} = \frac{dF_{it}}{dP_{it}} = \lambda$$

$$t = 1, 2,..., 24, i = 1, 2,..., G$$

(5)

2) Spinning Reserve

If spinning reserve is more than 7% of the total load at each time interval, it must meet:

$$\sum_{i=1}^{G} U_i P_{it} \geq 1.10 P_{dt}, t = 1, 2, 3,...24$$

(6)

3) Unit Generation Output Limitation

$$P_{min} \leq P_i \leq P_{max}$$

$$t = 1, 2,..., 24, i = 1, 2,..., G$$

(7)

4) Start Up- and Down Times Limitation

$$\sum_{i=1}^{24} U_{it} - U_{it-1} \leq M_i$$

(8)

5) Minimum Up and Down-Time Constraints

$$T_{Oi} \geq T_{Oj}$$

(9)
Optimization Strategy:

On/Off statue can be easily represented by binary coding: 1 is on statue and 0 is off statue. If the scheduling period is divided into 24 time-steps and there are total G units. Then each unit has 24 bits (Fig. 1). i.e. 2nd bit of unit 1 represents the on/off statue of unit 1 at 2nd time-step. One binary coding individual can be combined according to the order of units and each individual has total G × 24 bits. Per bit of each individual in one population is produced randomly.

This paper transforms the original constrained UC problem into unconstrained one by using penalty function.

\[
\text{Min} F + \sum_{j=1}^{n_c} u_j R_j
\]  \hspace{1cm} (11)

Where, \( F \) is original objective function; \( n_c \) is the number of violation constraints; \( R_j \) and \( u_j \) are the violation value and penalty coefficient of \( j \)th constraint, respectively. Equation (11) only includes spinning reserve, start up and down times, minimum up- and down-time constraints. The power balance and unit generation output limitation is considered in the load dispatch. The fitness function is:

\[
TF = \frac{K}{F + \sum_{j=1}^{n_c} u_j R_j}
\]  \hspace{1cm} (12)

Constant \( K \) is proportional coefficient. The value of \( K \) and \( u_j \) should be selected according to the specific problem. The values should let the fitness value of feasible solution be around 1 to prevent computer treating too large or small value.

Fig. 1: The binary representation of unit commitment.

Chaotic Optimization Algorithm:

Chaos which is a kind of highly unstable motion of deterministic systems in finite phase space exists in nonlinear systems. An essential feature of chaotic systems is that small changes in the parameters or the starting values for the data lead to great different future behaviors, such as stable fixed points, periodic oscillations, bifurcations, and periodicity. This sensitive dependence on initial conditions is generally exhibited by systems containing multiple elements with nonlinear interactions, particularly when the system is forced and dissipative. Sensitive dependence on initial conditions is not only observed in complex systems, but it is also observable in the simplest logistic equation (Yan, X. F, et al. 2003). The application of chaotic sequences may be an interesting alternative to provide the search diversity in an optimization procedure. Due to the non-repetition of chaos, it can carry out overall searches at higher speeds than stochastic argotic searches that depend on probabilities (Liu, B. F, et al. 2005). The design of approaches to improve the convergence of chaotic optimization is a challenging issue. The simple philosophy of the COA includes two main steps: first mapping from the chaotic space to the solution space, and secondly searching optimal regions using chaotic dynamics instead of random search (Yan, X. F, et al. 2003). This chaotic map also involves non-differentiable functions which difficult the modeling of the associate time series. The Lozi map is given by (Coelho, L. D. S, 2007):

\[
y_i(k) = 1 - a \times |y_i(k - 1)| + y(k - 1)
\]  \hspace{1cm} (13)
\[
y(k) = b \times y_i(k - 1)
\]  \hspace{1cm} (14)
\[ z(k) = \frac{y(k) - \alpha}{\beta - \alpha} \]  

(15)

Where, \( k \) is the iteration number. In this work, the values of \( y \) are normalized in the range \([0,1]\) to each decision variable in \( n \)-dimensional space of optimization problem. Therefore, \( y_1 \in [-0.6418, 0.6716]\) and \([\alpha, \beta]\) = \((-0.6418, 0.6716)\). The parameters used in this study are \(a = 1.7\) and \(b = 0.5\), these values suggested by (Coelho, L. D. S, 2007). Many unconstrained optimization problems with continuous variables can be formulated as the following functional optimization problem. Find \( X \) to minimize \( f(X) \), \( X = [x_1, x_2, ..., x_n] \). Where, \( f \) is the objective function, and \( X \) is the decision solution vector consisting of the \( n \) variables, \( x_j \) bounded by lower \( (L_i) \) and upper limits \( (U_i) \). Figure 1 shows the flowchart of the proposed chaotic search procedure based on the Lozi map.

Where \( f \) is the objective function, and \( X \) is the decision solution vector consisting of \( n \) variables, \( x_j \), bounded by lower \( (L_i) \) and upper limits \( (U_i) \). \( M_g \) and \( M_l \) are maximum number of iterations of chaotic global search and maximum number of iterations of chaotic Local search, respectively. In this paper \( \lambda \) is step size in chaotic local search and linearly decreases from 0.1 to 0.01. Also, \( \bar{f} \) and \( \bar{x} \) are best objective function and best solution from current run of chaotic search, respectively.

Fig. 2: Flowchart of the proposed COA.

V. Numerical Results:

This paper (proposed) developed the Chaotic Optimization Algorithm program using visual C++. The 10 generation units system and its multiples (10-100), is tested to verify the correctness, and the results are compared with other algorithms. Test results compared with other method results, are shown in Table 1 and Table 2, respectively. The CPU execution times of the COA and other methods in the literatures are shown in Table 3. The obtained results in this paper represents a nearer global optimal solution to the problem and verifies the correctness of the proposed algorithm.

Table 1: Total costs of the COA method for test systems.

<table>
<thead>
<tr>
<th>No. of units</th>
<th>Best cost ($)</th>
<th>Average cost ($)</th>
<th>Worst cost ($)</th>
</tr>
</thead>
</table>

926
<table>
<thead>
<tr>
<th></th>
<th>SPL</th>
<th>EP</th>
<th>LR</th>
<th>GA</th>
<th>PSO</th>
<th>MA</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NO. of Units</strong></td>
<td><strong>SPL</strong></td>
<td><strong>EP</strong></td>
<td><strong>LR</strong></td>
<td><strong>GA</strong></td>
<td><strong>PSO</strong></td>
<td><strong>MA</strong></td>
</tr>
<tr>
<td>10</td>
<td>564,950</td>
<td>565,352</td>
<td>566,869</td>
<td>565,825</td>
<td>574,153</td>
<td>565,827</td>
</tr>
<tr>
<td>20</td>
<td>1,123,938</td>
<td>1,127,256</td>
<td>1,128,362</td>
<td>1,126,243</td>
<td>1,125,983</td>
<td>1,128,192</td>
</tr>
<tr>
<td>40</td>
<td>2,248,645</td>
<td>2,252,612</td>
<td>2,250,223</td>
<td>2,251,911</td>
<td>2,250,012</td>
<td>2,249,589</td>
</tr>
<tr>
<td>60</td>
<td>3,371,178</td>
<td>3,376,255</td>
<td>3,374,994</td>
<td>3,376,625</td>
<td>3,374,174</td>
<td>3,370,820</td>
</tr>
<tr>
<td>80</td>
<td>4,492,909</td>
<td>4,505,536</td>
<td>4,496,729</td>
<td>4,504,933</td>
<td>4,501,538</td>
<td>4,494,214</td>
</tr>
<tr>
<td>100</td>
<td>5,615,530</td>
<td>5,633,800</td>
<td>5,620,305</td>
<td>5,627,437</td>
<td>5,625,376</td>
<td>5,616,314</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>DP</th>
<th>LRG A</th>
<th>MILP</th>
<th>ALR</th>
<th>ACSA</th>
<th>COA in this paper</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NO. of Units</strong></td>
<td><strong>DP</strong></td>
<td><strong>LRGA</strong></td>
<td><strong>MILP</strong></td>
<td><strong>ALR</strong></td>
<td><strong>ACSA</strong></td>
<td>COA in this paper</td>
</tr>
<tr>
<td>10</td>
<td>565,825</td>
<td>564,800</td>
<td>-</td>
<td>565,508</td>
<td>564,049</td>
<td>563,404</td>
</tr>
<tr>
<td>20</td>
<td>-</td>
<td>1,122,622</td>
<td>-</td>
<td>1,126,720</td>
<td>1,123,99</td>
<td>-</td>
</tr>
<tr>
<td>40</td>
<td>-</td>
<td>2,242,178</td>
<td>-</td>
<td>2,249,790</td>
<td>-</td>
<td>2,247,372</td>
</tr>
<tr>
<td>60</td>
<td>-</td>
<td>3,371,079</td>
<td>-</td>
<td>3,371,188</td>
<td>-</td>
<td>3,370,828</td>
</tr>
<tr>
<td>80</td>
<td>-</td>
<td>4,501,844</td>
<td>-</td>
<td>4,494,487</td>
<td>-</td>
<td>4,486,233</td>
</tr>
<tr>
<td>100</td>
<td>-</td>
<td>5,613,127</td>
<td>5,605,189</td>
<td>5,615,893</td>
<td>-</td>
<td>5,605,373</td>
</tr>
</tbody>
</table>

**Table II:** Total Cost Comparison Of Several Methods.

**Table III:** Comparison Of CPU Times.

<table>
<thead>
<tr>
<th>TECHNIQUES</th>
<th>COMPUTER USED</th>
<th>NUMBER OF UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>EP (Juste., 1999)</td>
<td>HP Workstation, 160 MHz</td>
<td>100 340 1176 2267 3584 6120</td>
</tr>
<tr>
<td>GA (Kazarlis, 1996)</td>
<td>HP Apollo 720, 50 MHz</td>
<td>221 733 2697 5840 10036 15733</td>
</tr>
<tr>
<td>LR (Kazarlis, 1996)</td>
<td>Dell Dim 4100, 1 GHz</td>
<td>257 514 1066 1594 2122 2978</td>
</tr>
<tr>
<td>MA (Jorge, 2002)</td>
<td>Sun Ultra 2 Dual 200 MHz</td>
<td>84 287 1063 2772 5145 10463</td>
</tr>
<tr>
<td>COA</td>
<td>Pentium IV-1.5 GHz PC</td>
<td>33 78 139 212 391 585</td>
</tr>
</tbody>
</table>

**VI. Conclusion:**

In this paper, the proposed COA is efficiently and effectively implemented to solve the UC problem. COA total production costs over the scheduled time horizon are less expensive than other methods on the large number of generating units. The proposed algorithm considered various constraints successfully and the genetic operations are improved based on the characteristic of power system. The test results demonstrate the effectiveness of the COA in searching global or near global optimal solution to the UC problem. Also the results show a better convergence and higher precision.

**REFERENCES**


