

# An Improved Particle Swarm Optimization Approach for Unit Commitment Problem

Yiran Guo<sup>1,2</sup>, Jingrui Zhang<sup>1,\*</sup> and Zheng Fang<sup>1</sup>

<sup>1</sup>Department of Mechanical and Electrical Engineering, Xiamen University, Xiamen, Fujian, 361005, P.R. China

<sup>2</sup>Shenzhen Research Institute of Xiamen University, Shenzhen, Guangdong, 518057, P.R. China

**Abstract:** Unit commitment problem is a large scale nonlinear hybrid integer programming problem. Optimal unit commitment scheduling involves determining on/off states of units and determining generations of units. This paper proposes an improved particle swarm optimization (IPSO) for the solution of optimal unit commitment problem (UCP). In the proposed approach, the on/off states of units are limited into feasible schedules by providing a new method related to a new time order at first. After that, the problem is transformed into a simple economic load dispatch problem. Then this dispatch problem is solved by an improved priority list technique instead of the classical equal lambda-iteration method. All the above improvements are embedding into the framework of particle swarm optimization approach for UCP. It is seen from the results of numerical example that the proposed IPSO approach surely possesses a high quality and a high convergence speed.

**Keywords:** Particle Swarm Optimization, Priority List, Time Order, Unit Commitment Problem.

## 1. INTRODUCTION

The main purpose of optimal unit commitment problem (UCP) for power system is to determining the on/off state of every generating unit and the generation of every committed unit for a given horizon, under various operating constraints, including fuel constraints, multiple emission requirements, ramp rate limits, minimum up and down time limits and proper spinning reserves. Since the optimal commitment scheduling can save huge amount of costs and improve reliability of power system, many methods have been developed to solve the UCP, such as Lagrange Relax (LR) [1], Dynamic Programming (DP) [2], and Genetic Algorithm (GA) [3]. But they all have disadvantages. The main problem of LR is the difficulty encountered in obtaining feasible solutions. DP is flexible but it may lead to "curse of dimensionality". The shortcoming of GA is massive calculations and it is difficult in dealing with nonlinear constraints.

Based on the analogy of swarm of bird and school of fish, particle swarm optimization (PSO) is suggested by Kennedy and Eberhart in 1995 [4]. It searches not only local optimal solution but also global optimal solution and it can easily deal with various difficult nonlinear constraints, with fewer experienced parameters than other methods, so it can give out a relatively optimal solution quickly. In recent years, it has been used in many fields to solve complex optimal problems for its flexibility and efficiency [5-7]. In power systems, PSO has also been greatly used, including solving optimization of reactive power flow [8], state estimation of

distribution systems [9] and optimization of compensation condenser configurations [10], proving that the algorithm is easy to manipulate. Recently, evolution algorithms [11-16] have also been widely used in the solution of optimal unit commitment problem.

In this paper, the authors split up traditional UCP into two stages: optimization of on/off states of generating units and optimization of generations of committed units. The on/off states of generating units are limited into feasible regions at first stage through a novel time order introduced. An improved priority list technique is employed instead of using the equal lambda-iteration technique which only uses the average fuel cost at the full load of units. All the above improvements are embedding into the particle swarm optimization approach for UCP, and the result of the numerical example given by later shows that the new method can really give out good result quickly.

## 2. FORMULATION OF UCP

### 2.1. Objective Function

The object for UCP is to find out the solution that the total production cost can be minimized and satisfy all kinds of constraints. The total production cost  $F$  is the sum of the fuel cost and start-up cost for all units during a time horizon. Thus, the objective function of the UC problem is

$$\min F = \sum_{t=1}^T \sum_{i=1}^N [f_i(P(i,t)) + ST_i^t(1-U(i,t-1))]U(i,t) \quad (1)$$

where  $N$  is the number of generators;  $T$  is the total scheduling periods;  $U(i,t)$  is the on/off status of unit  $i$  at time

$t(U(i,t) = 1$  when on and  $U(i,t) = 0$  when off);  $P(i,t)$  is generation of unit  $i$  at time  $t$ ;  $f_i(P(i,t))$  is the fuel cost function of unit  $i$  with corresponding  $P(i,t)$ , and it can be expressed as a second-order polynomial as follows:

$$f(P(i,t)) = a(i) + b(i)P(i,t) + c(i)P^2(i,t) \quad (2)$$

where  $a(i)$ ,  $b(i)$  and  $c(i)$  represent the unit cost coefficients;  $ST_i^t$  is the generator start-up cost for restarting a de-committed generating unit, which is associated with the number of hours during which the unit has been off. In general, the start-up cost can be expressed as follows:

$$ST_i^t = \begin{cases} \text{Hotcost}(i), \text{Mindown}(i) \leq \text{Toff}(i,t) \leq \text{Mindown}(i) + \\ \text{Coldhrs}(i) \\ \text{Cold cost}(i), \text{Toff}(i,t) > \text{Mindown}(i) + \text{Coldhrs}(i) \end{cases}$$

where  $\text{Toff}(i,t)$  is the duration for which unit  $i$  has remained offline at hour  $t$  and  $\text{Mindown}(i)$  is the minimum down-time of the unit  $i$ .

## 2.2. System and Unit Constraint

a) System load constrains:

$$\sum_{i=1}^N P(i,t)U(i,t) = P_D(t) \text{ where } P_D(t) \text{ is the system load}$$

demand at hour  $t$ .

b) Generation power limits:

$P_{\min}(i) \leq P(i,t) \leq P_{\max}(i)$  where  $P_{\min}(i)$  and  $P_{\max}(i)$  are minimum and maximum generation power of unit  $i$ ;

c) Spinning reserve constraints:

$$\sum_{i=1}^N P_{\max}(i)U(i,t) \geq P_D(t) + P_R(t) \text{ where } P_R(t) \text{ is the}$$

spinning reserve requirement, which is assumed to be 10% of the hourly demand  $P_D(t)$  in this article;

d) Minimum up-time/down-time constraints:

$$\begin{aligned} \text{Ton}(i,t) &\geq \text{Minup}(i) \\ \text{Toff}(i,t) &\geq \text{Mindown}(i) \end{aligned} \quad \text{where } \text{Minup}(i) \text{ and}$$

$\text{Mindown}(i)$  is the minimum up-time and down-time of the unit  $i$ ;

e) Initial states constraints:

The initial status at the start of period should also be taken into account.

$$\begin{aligned} \text{Ton}(i,1) &= \text{Initial}(i) + U(i,1) \quad , \text{Initial}(i) > 0 \ \& \ U(i,1) = 1 \\ \text{Toff}(i,1) &= -\text{Initial}(i) + (1 - U(i,1)), \text{Initial}(i) < 0 \ \& \ U(i,1) = 0 \end{aligned}$$

## 3. OVERVIEW OF PSO

A classical PSO usually generates a randomly diversified population with  $M$  particles. The fitness of all the individuals would be calculated through some way with accordance to their position which usually stands for the solution of the optimization problem. The particle with the best fitness val-

ue would be chose as the gbest and the current position of each particle would be set as its own pbest. Then the iterative process of the PSO would be performed until some preset criteria are met. In the iterative process, all the individuals would adjust their position through the velocity and position update formulations firstly. Then each particle would compare its current new fitness with that of its own best one which means the fitness of its pbest and the winner would replace the pbest. Finally, the best one of all the pbest would be chose as the gbest. The velocity and position update formulations for each particle of the swarm are expressed as:

$$\begin{aligned} v_{i,d}^{k+1} &= \omega v_{i,d}^k + c_1 \text{rand}_1(\text{pbest}_{i,d}^k - p_{i,d}^k) + c_2 \text{rand}_2(\text{gbest}_d^k - p_{i,d}^k) \\ p_{i,d}^{k+1} &= p_{i,d}^k + v_{i,d}^{k+1} \end{aligned}$$

where  $\omega$  is the inertia weight factor related to the iterations;  $c_1$  and  $c_2$  are the learning factors;  $\text{rand}_1$  and  $\text{rand}_2$  are the random numbers lie in  $[0, 1]$ ;  $p_{i,d}^k$  and  $v_{i,d}^k$  are the  $d$ th dimension position and velocity of particle  $i$  at iteration  $k$ , respectively;  $\text{gbest}_d^k$  is the  $d$ th dimension best position for all particles and  $\text{pbest}_{i,d}^k$  is the  $d$ th dimension's best position for particle  $i$  at iteration  $k$ . The flow chart for classical PSO is shown in Fig. (1).

## 4. IMPROVED PSO APPROACH FOR UCP (IPSO)

As we can see, there are two types of variable quantities for each particle in UCP,  $U(i,t)$  and  $P(i,t)$ , while  $U(i,t)$  is a 0/1 integer variable and  $P(i,t)$  is continuous variable. Though PSO performs perfect in solving continuous optimization problems, it is not an effective algorithm for mixed integer optimization problems, so we have to do some changes for UCP in using PSO.

### 4.1. Initialization for U(i,t)

The first thing for IPSO is to limit all initializations of unit's state into feasible solutions. Traditional methods have fateful disadvantage on dealing with constraints: they usually restart the handling process once there is no proper or feasible solution. In that case, more time would be consumed and the feasible solutions may not be found finally [17, 18]. As for traditional time order (from  $t=1$  to  $t=T$ ), scheduling often gets stuck in the situation that the load demand is fitted for the hour  $t$  but not fitted for hour  $t+1$  if the load of time  $t+1$  is larger than that of time  $t$ , when considering the constrain of minimum up/down times constraint [19, 20].

The new method points out a way to form a new time order in optimizing on/off states. Finding out the time of load demand's maximum pole which is bigger than other load demands, form a new time order that follows the direction of decreasing load demand, keep the load demand in order  $r$  is always bigger than its next order  $r+1$ . Start the optimization of states from the time of pole and then schedule according to the new time order. In this order, the situation referred before can be avoided and the speed can be improved. Like the example given in the article in Section 5, we can see the time of maximum pole for the example is 12, so the new time optimizing order is 12, 13, 11, 14, 10, 15, 9, ... .

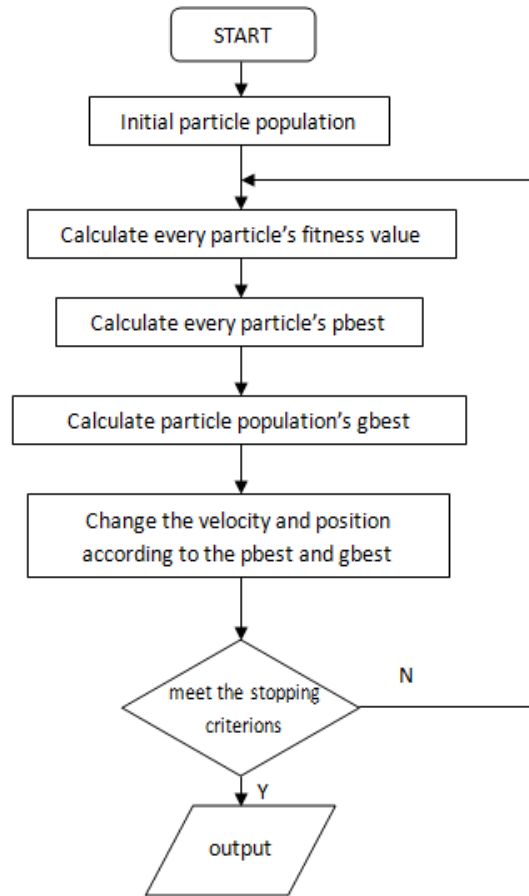


Fig. (1). Flow chart for classical PSO.

The basic steps for optimizing the on/off states are as follows:

Step 1: For  $t=1,2,\dots,T$ ,  $i=1,2,\dots,N$ , create a random integer-matrix  $U$  with dimension  $N \times T$ , like  $U = \begin{bmatrix} U(1,1) & U(1,2) & U(1,3) & \dots & U(1,T) \\ U(2,1) & U(2,2) & U(2,3) & \dots & U(2,T) \\ M & M & M & \dots & M \\ U(N,1) & U(N,2) & U(N,3) & \dots & U(N,T) \end{bmatrix}$ ;

Step 2: According to the new time order, set  $t=r_n(n=1,2,\dots,T)$ . For each unit, considering its minimum up/down time constraint, and if the distance from  $r_n$  to  $r_{n-1}$  is small than its minimum up/down time, only consider the constraint to  $r_{n-1}$ ;

Step 3:  $R(t)$  represents the spinning reserve at hour  $t$  and can be expressed as follows:

$$R(t) = \sum_{i=1}^N P_{\max}(i)U(i,t) - P_D(t) - P_R(t);$$

Step 4: If  $R(t) < 0$ , calculate average full-load cost of each uncommitted unit hour  $t$  and sort the units in the ascending order to obtain a list of  $SS1$  according to Priority List. Set every  $U(i,t)=1$  if not violate its minimum down time constraint in order until  $R(t) + P_{\max(i)} \geq 0$ . If the distance from  $r_n$  to  $r_{n-1}$  is small than its minimum down time, only consider the constraint to  $r_{n-1}$ ;

Step 5: Update the  $R(t)$ ;

Step 6: Calculate the average full-load cost of each committed unit at hour  $t$  and sort the units in the descending order to obtain a list of  $SS2$  according to Priority List. For every  $i$  in  $SS2$ , if  $R(t) \geq P_{\max}(i)$ , then set  $U(i, r_n)=0$  if not violate its minimum up time constraint in order. If the distance from  $r_n$  to  $r_{n-1}$  is small than its minimum up/down time, only consider the constraint to  $r_{n-1}$ ;

Step 7: Consider the initial state constraints. If fitted, set  $t=t+1$  according to the new time order and back to step 2 until the new time order is over.

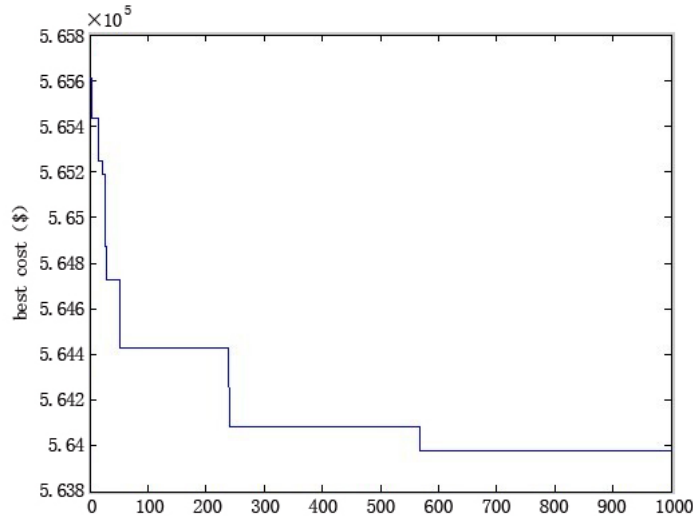
#### 4.2. Optimization for $P(i,t)$ for Certain $U(i,t)$

After  $U(i,t)$  has been determined, the optimization for  $P(i,t)$  becomes a simple economic load dispatch problem, which can be presented as follows:

$$\min F = \sum_{t=1}^T \sum_{i=1}^N [f_i(P(i,t)) + ST'_i(1-U(i,t-1))]U(i,t)$$

s.t.

$$\begin{cases} P_{\min}(i) \leq P(i,t) \leq P_{\max}(i) \\ \sum_{i=1}^N P(i,t)U(i,t) = P_D(t) \end{cases}$$



**Fig. (2).** Convergence characteristics for UCP using IPSO approach.

Within the feasible unit commitment schedule, classical equal lambda-iteration method can be used to solve the problem [11], and classical PSO can be used to solve this problem also. Compared with these methods, Priority list (PL) method is easier and quicker. So PL method can save time if the method is properly used. Traditional PL method is described as follows:

Step 1: The average full-load cost of a unit is defined as the cost per unit of power (\$/MW) when the unit is at its full capacity. This average full-load cost of a unit according to the fuel cost function given by Eq. (2) can be expressed as:

$$\alpha_i = \frac{f_i(P_{\max}(i))}{P_{\max}(i)} = \frac{a_i}{P_{\max}(i)} + b_i + c_i P_{\max}(i)$$

Step 2: The units are ranked by their  $\alpha$  in ascending order. Thus, the traditional priority list of units would be formulated based on the order of  $\alpha_i$  in which a unit with the lowest  $\alpha_i$  would have the highest priority to be dispatched.

Instead of the traditional PL technique using the average full-load cost of a unit to sort the order of units, the improved PL technique adopted in this paper employs the average cost of a unit at different power generation levels, such as the minimum output power, the average output power and even just  $b_i$  without the influence of output changes for simplicity. Through the comparisons among the different orders and their corresponding results obtained by the different power output levels, the best one would be chosen as the final output.

### 4.3. Implementation of the IPSO for Solving UCP

Because the optimization of  $U(i,t)$  is a mixed integer optimization problem, we can change the basic update formulations for the position and velocity of particles into the following expressions [11]:

$$v_{i,d}^{k+1} = \omega_1 \otimes (pbest_{i,d}^k \oplus p_{i,d}^k) + \omega_2 \otimes (gbest_d^k \oplus p_{i,d}^k) \quad (3)$$

$$p_{i,d}^{k+1} = p_{i,d}^k \oplus v_{i,d}^{k+1} \quad (4)$$

where “ $\oplus$ ” denotes “XOR” operator; “ $\otimes$ ” denotes “AND” operator; “+” denotes “OR” operator;  $\omega_1$  and  $\omega_2$  are two random binary integer numbers uniformly distributed in the range of [0, 1]. The detailed implementation with the proposed IPSO approach for UCP is described as follows.

Step 1: Initialization each individual's  $U(i,t)$  and limit them into feasible solutions;

Step 2: For every individual, compare and find out the optimal  $P(i,t)$  for determined  $U(i,t)$  with different PL.

Step 3: Compare each particle's objective function  $F$  to its best one and update the best position (pbest) of the particle and the particle who owns the best objective  $F$  is set as the global best position gbest.

Step 4: Modify the velocity and position of each particle in the swarm using Eqs. (3) and (4).

Step 5: Back to step 2 until the maximum iteration number is reached.

## 5. NUMERICAL RESULTS

The IPSO approach given by the article is applied to UCP for realistic power systems in range of 10 units to 100 units, along with 24 h load demands. The detailed parameters for ten units system come from Ref. [11]. And it can also be proved by larger scale system in which data are obtained by duplicating the ten-unit case and adjusting the load demands in proportion to the size of the system.

The convergence characteristic for UCP using the proposed IPSO approach is demonstrated in Fig. (2). From this figure, it can be obviously found that the IPSO has a quite

Table 1. Result of  $U(i,t)$ .

Time	Unit States									
	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10
1	1	1	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0
3	1	1	0	0	1	0	0	0	0	0
4	1	1	0	0	1	0	0	0	0	0
5	1	1	0	1	1	0	0	0	0	0
6	1	1	1	1	1	0	0	0	0	0
7	1	1	1	1	1	0	0	0	0	0
8	1	1	1	1	1	0	0	0	0	0
9	1	1	1	1	1	1	1	0	0	0
10	1	1	1	1	1	1	1	1	0	0
11	1	1	1	1	1	1	1	1	1	0
12	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	0	0
14	1	1	1	1	1	1	1	0	0	0
15	1	1	1	1	1	0	0	0	0	0
16	1	1	1	1	1	0	0	0	0	0
17	1	1	1	1	1	0	0	0	0	0
18	1	1	1	1	1	0	0	0	0	0
19	1	1	1	1	1	0	0	0	0	0
20	1	1	1	1	1	1	1	1	0	0
21	1	1	1	1	1	1	1	0	0	0
22	1	1	0	0	1	1	1	0	0	0
23	1	1	0	0	0	1	0	0	0	0
24	1	1	0	0	0	0	0	0	0	0

competitive convergence speed. The states of all generating units and the generation for them are shown in Table 1 and Table 2, respectively. It is found from the two figures that the schedule states and the generations satisfy all the constraints of UCP.

The comparison of the IPSO in this paper with other methods is shown in Table 3, including LR [21], GA [3], EP [22], HPSO [23], SA [24] and GAUC [25]. According to the comparison of Table 3, it is clearly seen that the IPSO obtains a better result than other methods considered. Besides, it can be observed that the IPSO has a stable result, which means that IPSO is reliable in solving UCP.

Hence, we can get a conclusion that the proposed approach can get competitive results of UCP at some faster convergence speed.

## CONCLUSION

The article presents a novel IPSO algorithm for UCPs in power systems. This proposed approach splits up the UCP into two stages. A new time order, which can avoid the terrible situation that traditional methods may meet, is proposed at the first stage for the optimization of on/off state for each unit. At the second stage, which means the optimal

**Table 2. Result of  $P(i,t)$ .**

Time	Unit Generation (MW)									
	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	Unit 6	Unit 7	Unit 8	Unit 9	Unit 10
1	455	245	0	0	0	0	0	0	0	0
2	455	295	0	0	0	0	0	0	0	0
3	455	370	0	0	25	0	0	0	0	0
4	455	455	0	0	40	0	0	0	0	0
5	455	390	0	130	25	0	0	0	0	0
6	455	360	130	130	25	0	0	0	0	0
7	455	410	130	130	25	0	0	0	0	0
8	455	455	130	130	30	0	0	0	0	0
9	455	455	130	130	85	20	25	0	0	0
10	455	455	130	130	162	33	25	10	0	0
11	455	455	130	130	162	73	25	10	10	0
12	455	455	130	130	162	80	25	43	10	10
13	455	455	130	130	162	33	25	10	0	0
14	455	455	130	130	85	20	25	0	0	0
15	455	455	130	130	30	0	0	0	0	0
16	455	310	130	130	25	0	0	0	0	0
17	455	260	130	130	25	0	0	0	0	0
18	455	360	130	130	25	0	0	0	0	0
19	455	455	130	130	30	0	0	0	0	0
20	455	455	130	130	162	33	25	10	0	0
21	455	455	130	130	85	20	25	0	0	0
22	455	455	0	0	145	20	25	0	0	0
23	455	425	0	0	0	20	0	0	0	0
24	455	345	0	0	0	0	0	0	0	0

**Table 3. Comparison of the IPSO with other methods.**

Method	Number of Trials	Population Size	Maximum Generation	Best (\$)	Mean (\$)	Worst (\$)
LR	/	/	/	566,107	/	/
GA	20	50	500	565,825	/	570,032
EP	20	50	500	564,551	565,352	566,231

Table 3. contd..

Method	Number of Trials	Population Size	Maximum Generation	Best (\$)	Mean (\$)	Worst (\$)
HPSO	50	20	1000	563,942	564,772	565,785
SA	10	/	/	565,828	565,988	566,260
UCC-GA	20	20	500	563,977	/	565,606
IPSO	20	20	1000	593,977	564,006	564,018

generations of the committed units, the article proposed a modified priority list technique instead of the classical equal lambda-iteration method. Both of these variations can help to improve the convergence speed and simplify the programs. The simulation results provided by the numerical example clearly reveal that the proposed IPSO algorithm can be used as an excellent optimizer in solving UCP.

### CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

### ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial supports from the National Undergraduate Innovation Training Programs of China (No. DC2013084), the Knowledge Innovation Program of Shenzhen City in China (No. JCYJ201-30327150859765) and the Natural Science Foundation of Fujian Province in China (No. 2012J01413).

### REFERENCES

- [1] A. Merlin, and P. Sandrin, "A new method for unit commitment at Electricite de France", *IEEE Transactions on Power Applied Systems*, vol. PAS-102, pp. 1218-1255, 1983.
- [2] W. L. Snyder, Jr., H. D. Powell, Jr., and J. C. Rayburn, "Dynamic programming approach to unit commitment", *IEEE Transactions on Power Systems*, vol. 2, no. 2, pp. 339-350, 1987.
- [3] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, "A genetic algorithm solution to the unit commitment problem", *IEEE Transactions on Power Systems*, vol. 11, no. 1, pp. 83-92, 1996.
- [4] J. Kennedy and R. C. Eberhart, "Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, Piscataway, NJ, pp. 1942-1948, 1995.
- [5] T. Niknam, and B. Amiri, "An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis", *Applied Soft Computing*, vol. 10, no. 1, pp. 183-197, 2010.
- [6] Y. Zhang, and L. Wu, "A hybrid TS-PSO optimization algorithm", *Journal of Convergence Information Technology*, vol. 6, no. 5, pp. 169-174, 2011.
- [7] M. Miyatake, M. Veerachary, F. Toriumi, S. Univ, M. Veerachary, F. Toriumi, and N. Fujii, "Maximum power point tracking of multiple photovoltaic arrays: A PSO approach", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 367-380, 2011.
- [8] H. Yoshida, K. Kawata, Y. Fukuyama, and S. Takayama, "A particle swarm optimization for reactive power and voltage control considering voltage security assessment", *IEEE Transactions on Power Systems*, vol. 15, no. 4, pp. 1232-1239, 2000.
- [9] S. Naka, T. Genji, T. Yura, and Y. Fukuyama, "A hybrid particle swarm optimization for distribution state estimation", *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 60-68, 2003.
- [10] Y. Xinmei, L. Yan, X. Xinyin, and W. Yao-wu, "Optimal shunt capacitor placement using particle swarm optimization algorithm with harmonic distortion consideration", *Proceedings of the CSEE*, vol. 23, no. 2, pp. 26-30, 2003.
- [11] X. Yuan, H. Nie, A. Su, L. Wang, and Y. Yuan, "An improved binary particle swarm optimization for unit commitment problem", *Expert Systems with Applications*, vol. 36, no. 4, pp. 8049-8055, 2009.
- [12] Á. López, C. L. Meza, G. Moya, and R. N. Gómezb, "A MIQCP formulation to solve the unit commitment problem for large-scale power systems", *International Journal of Electrical Power & Energy Systems*, vol. 36, no. 1, pp. 68-75, 2012.
- [13] X. Yuan, A. Sua, H. Y. Yuanb, and L. Wanga, "Application of enhanced discrete differential evolution approach to unit commitment problem", *Energy Conversion and Management*, vol. 50, no. 9, pp. 2449-2456, 2009.
- [14] Y. Jeong, J. B. Park, J. R. Shin, and Y. Kwang, "A thermal unit commitment approach using an improved quantum evolutionary algorithm", *Electric Power Components and Systems*, vol. 37, no. 7, pp. 770-786, 2009.
- [15] J. Ebrahimi, S. H. Hosseinia, and G. B. Gharehpetian, "Unit commitment problem solution using shuffled frog leaping algorithm", *IEEE Transactions on Power Systems*, vol. 26, no. 2, pp. 573-581, 2011.
- [16] J. Ostrowski, M. F. Anjos, and A. Vannelli, "Tight mixed integer linear programming formulations for the unit commitment problem", *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 39-46, 2012.
- [17] M. Eslamian, S. H. Hosseinian, and B. Vahidi, "Bacterial foraging-based solution to the unit-commitment problem", *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1478-1488, 2009.
- [18] A. Frangioni, C. Gentile, and F. Lacalandra, "Tighter approximated MILP formulations for unit commitment problems", *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 105-113, 2009.
- [19] J. Raglend, C. Raghuvver, G. R. Avinash, N. P. Padhy, and D. P. Kothari, "Solution to profit based unit commitment problem using particle swarm optimization", *Applied Soft Computing*, vol. 10, pp. 1247-1256, 2010.
- [20] T. W. Lau, C. Y. Chung, K. P. Wong, and T. S. Chung, "Quantum-inspired evolutionary algorithm approach for unit commitment", *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1503-1512, 2009.
- [21] H. H. Balci, and J. F. Valenzuela, "Scheduling electric power generators using particle swarm optimization combined with the Lagrangian relaxation method", *International Journal of Applied Mathematics and Computer Science*, vol. 14, no. 3, pp. 411-421, 2004.
- [22] A. Juste, H. Kita, E. Tanaka and J. Hasegawa, "An evolutionary programming solution to the unit commitment problem", *IEEE Transactions on Power Systems*, vol. 14, no. 4, pp. 1452-1459, 1999.
- [23] T. O. Ting, M. V. C. Rao, and C. K. Loo, "A novel approach for unit commitment problem via an effective hybrid particle swarm optimization", *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 411-418, 2006.

- [24] D. N. Simopoulos, S. D. Kavatza, and C. D. Vournas, "Unit commitment by an enhanced simulated annealing algorithm", *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 193-201, 2006.
- [25] T. Senjyu, H. Yamashiro, K. Uezato, and T. Funabashi, "A unit commitment problem by using genetic algorithm based on unit characteristic classification", In: *Proceedings of IEEE Power Engineering Society Winter Meeting*, vol. 1, pp. 58-63, 2002.

---

Received: November 28, 2014

Revised: January 01, 2015

Accepted: January 19, 2015

© Guo et al.; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.