Smart Grid and Renewable Energy, 2017, 8, 178-194
http://www.scirp.org/journal/sgre
ISSN Online: 2151-4844
ISSN Print: 2151-481X

A Hybrid Unit Commitment Approach Incorporating Modified Priority List with Charged System Search Methods

Yuan-Kang Wu[1*], Chih-Cheng Huang[2], Chun-Liang Lin[2], Shih-Ming Chang[1]

[1]Department of Electrical Engineering, National Chung-Cheng University, Taiwan
[2]Department of Electrical Engineering, National Chung-Hsing University, Taiwan

Email: *allenwu@ccu.edu.tw

How to cite this paper: Wu, Y.-K., Huang, C.-C., Lin, C.-L. and Chang, S.-M. (2017) A Hybrid Unit Commitment Approach Incorporating Modified Priority List with Charged System Search Methods. *Smart Grid and Renewable Energy*, **8**, 178-194.
https://doi.org/10.4236/sgre.2017.86012

Received: May 15, 2017
Accepted: June 23, 2017
Published: June 26, 2017

Copyright © 2017 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

This paper presents a new hybrid approach that combines Modified Priority List (MPL) with Charged System Search (CSS), termed MPL-CSS, to solve one of the most crucial power system's operational optimization problems, known as unit commitment (UC) scheduling. The UC scheduling problem is a mixed-integer nonlinear problem, highly-dimensional and extremely constrained. Existing meta-heuristic UC solution methods have the problems of stopping at a local optimum and slow convergence when applied to large-scale, heavily-constrained UC applications. In the first step of the proposed method, initial hourly optimum solutions of UC are obtained by Modified Priority List (MPL); however, the obtained UC solution may still be possible to be further improved. Therefore, in the second step, the CSS is utilized to achieve higher quality solutions. The UC is formulated as mixed integer linear programming to ensure the tractability of the results. The proposed method is successfully applied to a popular test system up to 100 units generators for both 24-hr and 168-hr system. Computational results show that both solution cost and execution time are superior to those of published methods.

Keywords

Modified Priority List, Charged System Search, Unit Commitment

1. Introduction

The immense competition in power industry is forcing the operators to run the system by maximizing the benefits to both supplier and consumer. The unit commitment (UC) problem is a non-linear, non-convex, large-scale, mixed integer problem. It involves the determination of an optimum start-up and shut-down schedule of generating units that minimizes operating cost, while satisfying a set of system constraints over a time period [1] [2]. Numerous efficient and robust UC methods have been developed and can be classified into two main categories [3]: The first category represents numerical optimization techniques such as priority list methods (PL) [4] [5] [6], dynamic programming method (DP) [7] [8], Lagrangian relaxation (LR) [9] [10] [11], and the popular branch-and-bound method (BB) [12] [13] [14] [15]. The PL method is fast but highly heuristic and gives schedules with relatively high operation costs. The DP method was widely used for the UC problem but suffered from the curse of dimensionality [16] when applied to a modern large-scale system with heavy constraints. LR has shown some potential in solving large-scale unit commitment problems by decomposing the primal problem into a set of single unit optimization sub-problems that are easier to solve with dynamic programming. The primary difficulty of this method is that it requires adopting certain measures to convert optimal dual solutions into feasible solutions for the primal problem because of the duality gap. The BB method uses a linear function to represent fuel consumption and time-dependent start-up cost, and obtains the required lower and upper bounds. However, its computational time increases exponentially with a number of dimensions of the UC problem. The second category represents meta-heuristic algorithm such as genetic algorithms (GA) [17] [18] [19], evolutionary programming (EP) [20] [21], simulated annealing (SA) [22] [23], particle swarm optimization (PSO) [24] [25] [26] [27] and others [28]-[37]. In recent years, meta-heuristic algorithms have been widely used to solve some complex optimization problems in power systems. However, the biggest problem that the meta-heuristic algorithm faced is that the optimization space can be extended by penalty function method through the processing of constraints. As a result, the computational efficiency is rather low. Meta-heuristic algorithms require excessive computation time, especially for a large system size due to their random and iterative nature. Additionally, pure meta-heuristic methods commonly get stuck at a local optimum rather than at the global optimum. Even small percentage reduction in fuel costs typically leads to considerable savings for electric utilities. Consequently, a complete and efficient approach for solving the UC problem is urgently required.

In this paper, a hybrid algorithm that combines the Modified Priority List (MPL) and Charged System Search (CSS) methods is proposed for solving the UC problem. The MPL method is utilized to obtain an initial UC solution for a system over a 24-hr and 168-hr period. Next, this paper seeks better UC solutions to reduce total production cost using the CSS method. Charged System Search (CSS) is a population based meta-heuristic algorithm that was proposed recently by Kaveh and Talatahari [38]. In the CSS, each solution candidate is considered as a charged sphere called a Charged Particle (CP). The effectiveness of the proposed methodology was evaluated on several case studies and the results have been presented in this paper.

Therefore, the main contributions of this paper can be summarized as follows:

i) The Modified Priority List (MPL) method is proposed to solve the UC problem. MPL has multifold computational advantages over other UC algorithms. This is validated through large-scale unit commitment problem studies.

ii) In order to further improve the performance of the proposed MPL method, it is combined with Charged System Search algorithm and the obtained algorithm is known as hybrid MPL-CSS method. MPL-CSS manages to produce lower production cost solutions in most test cases.

This remainder of this paper is organized as follows: Section 2 formulates the UC problem. Section 3 introduces the proposed method that combines the MPL with CSS algorithms. Section 4 conducts numerical simulations and compares various UC solving methods. Finally, concluding remarks are discussed in Section 5.

2. Problem Formulation

2.1. Objective

The total fuel cost of unit k in time period t is usually given as a second order function of p_{kt} as follows:

$$f_k\left(p_{kt}\right) = a_k p_{kt}^2 + b_k p_{kt} + c_k \tag{1}$$

where coefficients a_k, b_k, and c_k are the cost coefficients of unit k. In order to preserve the MILP formulation, the quadratic production cost of thermal generating units (1) is approximated by a piecewise linear function as in [14].The objective function Ψ that is the sum of the fuel and start-up costs for all units is defined as:

$$\Psi\left(p,u\right) = \sum_{t=1}^{T}\sum_{k=1}^{K} f_k\left(p_{kt}\right) + ST_{kt}\left(u\right) \tag{2}$$

The total start-up cost $ST_{kt}\left(u\right)$ can be computed by the equation as below:

$$ST_{kt}\left(u\right) = \begin{cases} ST_{kt}^{hot}, & \text{if } t_{k,off} \le t_k^{dn} + t_k^{cold} \\ ST_{kt}^{cold}, & \text{if } t_{k,off} > t_k^{dn} + t_k^{cold} \end{cases} \tag{3}$$

In general, as the OFF time is increased, then the start-up cost is increased [39]. If unit's OFF time is larger than $t_k^{dn} + t_k^{cold}$, then the start-up cost will be the cold start cost.

2.2. Constraints

2.2.1. System Power Balance Constraints

$$\sum_{k=1}^{K} p_{kt} + \sum_{j=1}^{J} pw_{jt} = D_t \tag{4}$$

D_t means the load demand at time t.

2.2.2. System Spinning Reserve Constraints

Spinning reserve requirements are necessary in the power generation scheduling to prevent a power supply interruption. Spinning reserve requirements can be specified in terms of excess generation output:

$$\sum_{k=1}^{K} u_{kt}\,\overline{p}_{kt} \geq D_t + R_t \tag{5}$$

2.2.3. Generation Limits Constraints

Every online unit has generation limits:

$$\underline{p}_k \leq p_{kt} \leq \overline{p}_k \tag{6}$$

2.2.4. Operation Ramp-Rate Limit Constraints

The operation range of every online unit is also constrained by its up and down ramp rate limits:

$$\left| p_{kt} - p_{kt-1} \right| \leq \delta_k \tag{7}$$

2.2.5. Minimum Up-Time and Down-Time Constraints

A unit must be online for a certain number of time intervals before it can be shut-down:

$$u_{kt} = 1,\ \text{if}\ t_{k,on} < t_k^{up} \tag{8}$$

A unit must be offline for a certain number of time intervals before it can be started-up:

$$u_{kt} = 0,\ \text{if}\ t_{k,off} > t_k^{dn} \tag{9}$$

3. Proposed Hybrid Method

3.1. Modified Priority List (MPL)

In order to improve the efficiency of Meta-heuristic in a large search space, this paper proposed the MPL method to reduce the search space and this will ensure the accuracy of Meta-heuristic algorithm.

3.1.1. Production of All Possible Candidate Units for Each Time Period Based on the Cost

In the beginning, the candidate units are built for each hour to satisfy (4-6), and the minimum online unit at time t is determined by using (10). Next, the cheaper unit will be turn on with high priority. The *heat rate* ($/MW) of each unit is represented as (11). As shown in Figure 1, most of load is supplied by base units and the remaining load is supplied by other units $\overline{p}_{k'}$. Base units and other units form the unit combination, which is represented as (12). In this way, the search space is limited and the computation time is reduced, which is suitable for large-scale unit application. Then, the candidate unit combinations are collected and the final candidate unit combination C_t is obtained by (13).

$$\begin{aligned} X_t = \Big\{ \{(p_t, u_t)\} \Big| & D_t + R_t \leq \sum_{k \in K} u_{kt}\,\overline{p}_k < D_t + R_t + \overline{p}_{k+1}, \\ & \overline{p}_k > \overline{p}_k, \cdots, k \in K, (p_t, u_t)\ \text{satisfies}\ (4\text{ - }6) \Big\} \end{aligned} \tag{10}$$

$$hr_k = \frac{f(\overline{p}_k)}{\overline{p}_k} \tag{11}$$

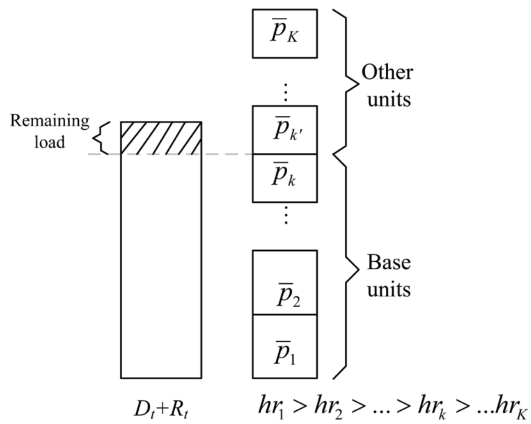

Figure 1. Rearrange units according to the heat rate.

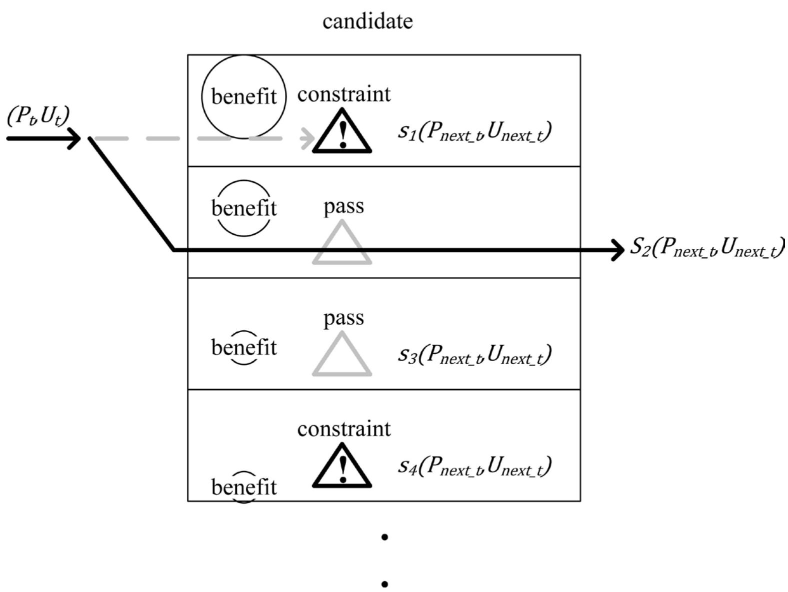

Figure 2. Simulation method.

$$Y_t = \left\{ \left\{ \left(p_t, u_t \right) \right\} \middle| \sum_{k \in K} u_{kt} \overline{p}_k + u_{k't} \overline{p}_{k'} \geq D_t + R_t, \right. \\ \left. \forall k' \in K \setminus \{k\}, \left(p_t, u_t \right) \text{ satisfies } (4\text{-}6) \right\} \tag{12}$$

$$C_t = \{ X_t, Y_t \} \tag{13}$$

3.1.2. Simulation

According to our UC experience, there is a significant rule to determine the unit commitment: the unit will not be off once it turns on before the peak load; the unit will not be on once it turns off after the peak load. Therefore, the unit commitment (p_t, u_t) at time t can be used to choose the next unit commitment and obtain suitable solutions $\{S_l(p_{next\_t}, u_{next\_t})\}$ by using (14). Then the minimum cost of unit commitment can be chosen from these solutions $\{S_l(p_{next\_t}, u_{next\_t})\}$ for next hour by using (15). In **Figure 2**, the reason why Simulation chooses $S_2(p_{next\_t}, u_{next\_t})$ is that S_2 does not violate UC constraints and its benefit is superior to the benefits from the candidate UC combinations after $S_3\left(p_{next\_t}, u_{next\_t}\right)$.

$$\begin{aligned} &\left\{S_l\left(p_{next\_t}, u_{next\_t}\right)\right\} \\ &:=\left\{\left(p_{next\_t}, u_{next\_t}\right) \in C_{next\_t} \middle| u_{kt} - c\left(u_{k,next\_t}\right) \geq 0 \wedge \left|p_{kt} - c\left(p_{k,next\_t}\right)\right| \leq \delta_k, \right. \\ &\qquad \left. k \in K, t \in T, l = 1, 2, \cdots, L, c \in C_{next\_t}, \text{satisfies } (4-6)\right\} \end{aligned} \tag{14}$$

where L is the number of suitable unit candidates in the next hour.

$$\left(p_{next\_t}, u_{next\_t}\right) = \arg\min\left(\left\{\Psi S_l\left(p_{next\_t}, u_{next\_t}\right)\right\}\right) \tag{15}$$

The simulation algorithm is shown in **Algorithm 1**—Simulation.

The **Algorithm 1**—Simulation is suitable for the UC problem in a renewable energy environment. In order to obtain a more accurate solution, the Modified Priority List (MPL) is proposed in this paper. The difference between the MPL method and the Simulation algorithm is that the MPL implemented the simulation for each $\left\{S_l\left(p_{next\_t}, u_{next\_t}\right)\right\}$. By using the MPL algorithm, the solution is more accurate but the computation time is increased. The MPL algorithm is shown in **Algorithm 2**—MPL.

Algorithm 1—Simulation

function Simulation $\left(\left(p_t, u_t\right), t, end\right)$
t = next.
While t non-terminal **do**
Obtained $\left\{S_l\left(p_{next\_t}, u_{next\_t}\right)\right\}$ by (14)
$\left(p_t, u_t\right) = \arg\min\left(\left\{\Psi S_l\left(p_{next\_t}, u_{next\_t}\right)\right\}\right)$ by(15).
t = next.
end
return $\left(p_{t\sim end}, u_{t\sim end}\right)$.
end function

Algorithm 2—MPL

function MPL $\left(\left(p_t, u_t\right), t, end\right)$
t = next, $Ref\left(p_t, u_t\right) = \left(p_t, u_t\right)$.
while t non-terminal **do**
Obtained $\left\{S\left(p_t, u_t\right)\right\}$ by (14).
l = 1. (the l is an index of the set)
best is null.
while l non-terminal **do**
$\left(p_{t\sim end}, u_{t\sim end}\right)$ = Simulation $\left(\left\{S\left(p_t, u_t\right)\right\}, t, end\right)$
if $\Psi\left(p_{t\sim end}, u_{t\sim end}\right) < \Psi best\left(p_{t\sim end}, u_{t\sim end}\right)$
$best\left(p_{t\sim end}, u_{t\sim end}\right) = \left(p_{t\sim end}, u_{t\sim end}\right)$
end
l + 1
end
$Ref\left(p_t, u_t\right) = \left(p_t, u_t\right)$.
t = next
end
return $Ref\left(p_{str\sim end}, u_{str\sim end}\right)$
end function

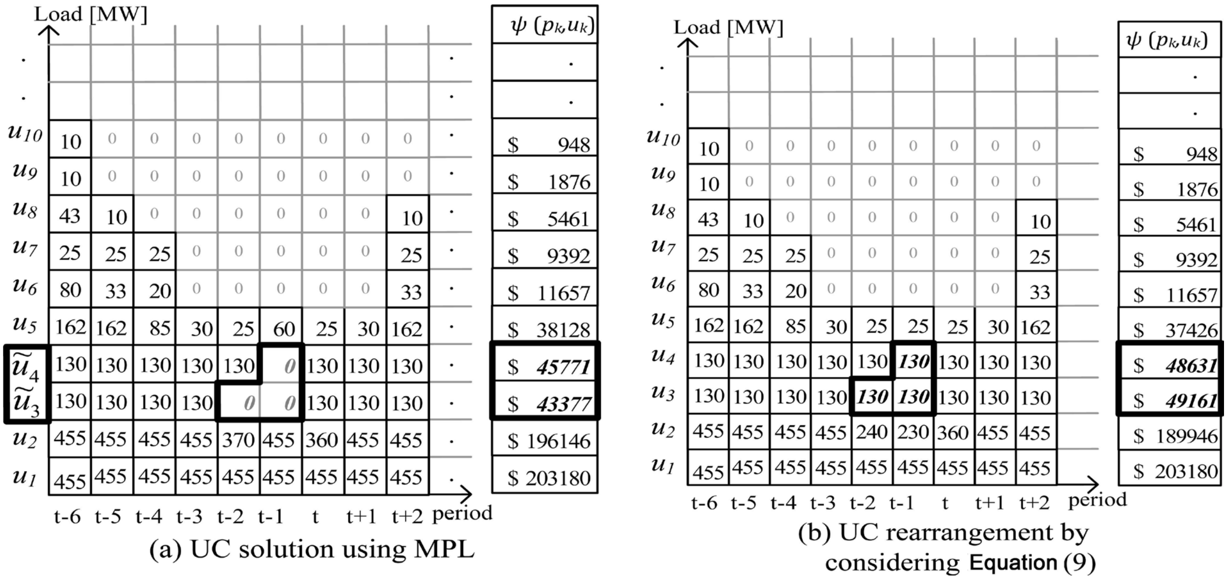

Figure 3. Rearrangement of unit scheduling to satisfy the constraint of minimum down-time.

3.1.3. Procedure

After the UC solution is solved by the MPL algorithm, some units $\{u_{\tilde{k}}\}$ may violate the constraint—Equation (9) during the large load change. For example, as shown in **Figure 3(a)**, the minimum down time of $u_{\tilde{3}}$ and $u_{\tilde{4}}$ is 3 hours; therefore, the UC solution in the squared-bold-italic part violates Equation (9). If the UC solution is changed to the squared-bold-italic part in **Figure 3(b)**, the new UC solution can satisfy Equations (4)-(9) but the reserve and total cost increases. However, the reduction on reserve and total cost is significant for the UC problem. Therefore, this paper proposed hybrid MPL-CSS algorithm to solve the UC solution. After the MPL completes the initial unit commitment solutions, the CSS is then used to modify the solution in the repeated search space, which can reduce the total cost.

In the MPL-CSS algorithm, on the basis of $Ref\left(p,u\right)$ for $u_{\tilde{k}}$, the turn-off time of units is extended to satisfy (9) and put them sequentially into the set $U^{\tilde{k}}$ as shown in (16), which would reduce the reserve. The priority is determined by evaluating the contribution on the cost by using Equation (17) for each $u_{\tilde{k}}$. Then a suitable solution is proposed for each $u_{\tilde{k}}$ by considering (18) and (19). The feasible solutions, as shown in (20), are combined to form the search space of CSS for each hour.

$$U_l^{\tilde{k}} := \left\{\left\{p_{\tilde{k}}, u_{\tilde{k}}\right\}\middle| \text{satisfies}\left(4\text{-}9\right)\right|, l = 1,2,\cdots,L, L \text{ is number of } \tilde{k}. \tag{16}$$

$$\Delta^{\tilde{k}} = \Psi\left(p_k, u_k\right) - \Psi\left(p_{\tilde{k}}, u_{\tilde{k}}\right) \tag{17}$$

$$X_t := \left\{ \left\{\left(p_t, u_t\right)\right\} \middle| D_t + R_t \le \sum_{k\in K\setminus\{\tilde{K}\}} u_{kt}\overline{p}_k < D_t + R_t + \overline{p}_{k+1}, \right. \\ \left. \overline{p}_1 > \overline{p}_2, \cdots, k \in K, \left(p_t, u_t\right) \text{ satisfies } \left(4\text{-}6\right) \right\} \tag{18}$$

$$Y_t := \left\{ \{(p_t, u_t)\} \middle| \sum_{k \in K \setminus \{\tilde{K}\}} u_{kt} \overline{p}_k + u_{k't} \overline{p}_{k'} < D_t + R_t, \right.$$
$$\left. \forall k' \in K \setminus \left(\{k\} \cap \{\tilde{k}\} \right), (p_t, u_t) \text{ satisfies } (4\text{-}6) \right\} \tag{19}$$

$$C_t^{\tilde{k}} = \{X_t, Y_t\} \tag{20}$$

In the solution space $C_t^{\tilde{k}}$, it may exist a better solution compared to the last solution of $Ref(p,u)$ Therefore, we use the CSS to search for a more accurate solution in $C_t^{\tilde{k}}$ and update $Ref(p,u)$; then we deal with the next $u_{\tilde{k}}$. Following the above step, the $C_t^{\tilde{k}}$ is updated continuity to optimize the unit commitment $Ref(p,u)$ until the convergence of $Ref(p,u)$ ends. The MPL-CSS algorithm can summarize the characteristic of the UC solution and narrow down the search space. It is very efficient, especially for long-term UC schedule with a large system. The pseudo code of the MPL-CSS algorithm is shown in **Algorithm 3**. It also indicates the clear process about the proposed method.

Algorithm 3—MPL-CSS

```
function MPL-CSS
 Ref(p,u) =schedule all time period by Algorithm 2
  Create U_l^k̃ by (16-20).
  l = 1. (the l is an index of the set)
  non-convergence = 0.
  while l non-terminal and non-convergence < limit do
   create search space C_t^k̃ by U_l^k̃
    X = rand. (where X = {X_1, X_2, ⋯, X_i, ⋯, X_n}, n is number of particles,
     X_i = {(p_1,u_1),(p_2,u_2),⋯,(p_T,u_T)}
 CM = X_best
it = 1. (index of iteration)
while it non-terminal do
 X_new , V_new obtained by (21)-(29)
 X_best = arg min{ΨX_new}
If ΨX_best(p,u) ≤ ΨCM(p,u)
 CM(p,u) = X_best(p,u).
else
 X_worst(p,u) = CM(p,u).
end
If ΨCM(p,u) < ΨRef(p,u)
 Ref(p,u) = CM(p,u).
non-convergence = 0.
break iteration to l + 1.
end
it + 1.
end
l + 1
   If ΨRef(p,u) non-convergence, then non-convergence + 1.
  end
  return Ref(p,u)
end function
```

3.2. Charge System Search Algorithm (CSS)

The Charged System Search (CSS) algorithm is based on the Coulomb and Gauss laws from electrical physics and the governing laws of motion from the Newtonian mechanics [40]. The algorithm can be considered as a multi-agent approach, where each agent is a Charged Particle (CP). Each CP is assigned a random position. The fitness of each CP is calculated first. The magnitude of the charge of each CP is calculated as

$$q_i = \frac{fit_i - fit_{worst}}{fit_{best} - fit_{worst}},\ i = 1, 2, \cdots, n \tag{21}$$

where fit_{best} and fit_{worst} are the so far best and the worst fitness of particles, respectively. fit_i represents the fitness of the i^{th} CP. n is the total number of CPs.

The CSS utilizes a Charged Memory (CM) that saves the best so far CP vectors [40] and their related objective function values. The CM size is equal to a quarter of the number of CPs. At the end of every iteration, the worst particle is replaced by CM.

The separation distance r_{ij} between two CPs is defined as follows:

$$r_{ij} = \frac{\left\| X_i - X_j \right\|}{\left\| \frac{\left(X_i + X_j \right)}{2} - X_{best} \right\| + \varepsilon} \tag{22}$$

where X_i and X_j are the position of the i^{th} and j^{th} CPs respectively, X_{best} is the position of the best CP, and ε is a small positive number that is taken to prevent singularity. Moving probability of each CP towards the others is determined by using

$$p_{ij} = \begin{cases} 1, \text{ if } \frac{fit_i - fit_{worst}}{fit_{best} - fit_{worst}} > rand \vee \left(fit_j > fit_i \right) \\ 0, \text{ otherwise} \end{cases} \tag{23}$$

where *rand* is uniformly distributed in the range (0,1).

Each CP is considered as a sphere with radius a that is limited to the size of the search space.

$$a = 0.1 * \max\left(\left\{ X_{i,\max} - X_{i,\min} \mid i = 1, 2, \cdots, n \right\} \right) \tag{24}$$

The resultant force acting on the ith CPs is calculated by using (14):

$$F_j = q_i \sum_{i, i \neq j} \left(\frac{q_i}{a^3} r_{ij} i_1 + \frac{q_i}{r_{ij}^2} i_2 \right) p_{ij},\ \begin{cases} i_1 = 1, i_2 = 0, \text{ if } r_{ij} < a \\ i_1 = 0, i_2 = 1, \text{ if } r_{ij} \geq a \end{cases} \tag{25}$$

where F_j is the resultant force acting on the j^{th}.

Each CP moves to the new position and the new velocity is calculated as:

$$X_{i,new} = rand_1 k_a \frac{F_j}{m_i} \Delta t^2 + rand_2 k_v v_{i,old} + X_{i,old} \tag{26}$$

$$V_{i,new} = \frac{X_{i,new} - X_{i,old}}{\Delta t} \tag{27}$$

where $rand_1$ and $rand_2$ are two random numbers that are uniformly distributed in the range of (0,1). Δt is the size of each step. m_i is the mass of the i^{th} CP and is equal to q_i. k_a and k_v represent the acceleration coefficient and velocity coefficient respectively. The parameters k_a and k_v are defined as:

$$k_a = 0.5\left(1+\frac{\text{iteration}}{\text{iteration}_{\max}}\right) \tag{28}$$

$$k_v = 0.5\left(1-\frac{\text{iteration}}{\text{iteration}_{\max}}\right) \tag{29}$$

k_a is regarded as the weight of exploitation; k_v is regarded as the weight of exploration. As the value of iteration increases, then k_a will be increased but k_v will be reduced.

4. Numerical Results

4.1. One-Day Unit Scheduling

The proposed MPL-CSS method is tested on the systems with 10 to 100 units, considering a 24-h scheduling horizon. The detailed data for 10-, 20-, 40-, 60-, 80-, and 100-units, and the corresponding load demands can be found in [17]. The spinning reserve is assumed to be 10% of the demand in our cases. The proposed program that combines MPL with CSS is coded in MATLAB and implemented on a personal computer with an Intel i7-2600 CPU 2.6 GHz and a 4.0 GB RAM. To prevent misleading results obtained from our simulation owing to the stochastic nature of the CSS, in each case, the results of 20 trial runs were averaged.

The result of the generation scheduling of the best solution of MPL and MPL-CSS for 10-unit until 100-unit systems is given in **Table 1**. The UC schedule for 100-unit system is shown in **Table 2**. It is noted that the maximum iteration number is set based on the convergence characteristic of the proposed MPL-CSS method. The MPL-CSS manages to generate a better result as compared to the MPL method. In term of computation time, MPL capable in achieving solutions in short amount of time, while MPL-CSS further improves the solution quality within reasonable time. Additionally, the improved results from CSS are very precise, since all of them have zero standard deviation values.

In **Table 3**, the UC results which obtained using the proposed MPL-CSS are compared with those in previous works. In this paper, the CSS particle number is set at 5 because the optimal solution with a short computation time can be achieved. **Table 3** shows the effectiveness and robustness of the proposed method to solve the UC problem, as the result is comparable to previous works. The best results from **Table 3** is represented by bold numbers, it is obvious that the MPL-CSS method performs superior and can achieves lowest UC cost. The results of MPL-CSS are more accurate and with zero standard deviation, compared with other algorithms. This is significant in power system operation, as this will provide the most reliable info for the system operator in any decision making.

Table 1. UC costs and computation time using the proposed method.

units	MPL result		MPL-CSS result					
	Cost ($)	Time (sec)	Max. iteration of $u_{\bar{k}}$	Best cost ($)	Mean cost ($)	worst cost ($)	Mean Time (sec)	Stddev.
10	563,938	0.2	100	563,938	563,938	563,938	0.2	0
20	1,123,522	0.7	100	1,123,522	1,123,522	1,123,522	0.7	0
40	2,244,800	2.5	100	2,243,545	2,243,545	2,243,545	6	0
60	3,362,714	6.8	100	3,361,407	3,361,407	3,361,407	14	0
80	4,485,174	33.3	100	4,482,807	4,482,807	4,482,807	46	0
100	5,604,574	74.42	100	5,601,253	5,601,253	5,601,253	294	0

Table 2. Best UC schedule of MPL-CSS in large-scale 100-unit case.

Unit	Hour																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1 - 18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19 - 20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
21 - 22	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
23 - 24	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
25 - 30	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
31 - 37	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
38 - 41	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
42 - 47	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
48	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
49	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
50 - 51	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
52 - 56	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	1	1	1	0	0	0
57 - 61	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	0
62 - 64	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
65 - 70	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
71 - 72	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	0
73	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	1	1	0	0
74 - 79	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	0	0
80 - 89	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0
90 - 93	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
94 - 98	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
90 - 100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 3. Comparison of UC costs by using various methods.

Method	Best Cost ($)	Mean Cost ($)	Worst Cost ($)	Diff. (%)	Mean Time (s)
		10 Unit system			
IBSFL [30]	**563,938**	-	-	-	-
BF [28]	564,842	-	-	-	84
SA [23]	565,828	565,988	566,260	0.08	3
IQEA [31]	563,977	563,977	563,977	0.0	7
EPSO [32]	**563,938**	564,206	564,266	0.33	7
GA [33]	**563,938**	564,088	564,253	0.06	4.8
MPL	**563,938**	-	-	-	**0.2**
MPL-CSS	**563,938**	**563,938**	**563,938**	**0.00**	**0.2**
		20 Unit system			
BF [28]	1,124,892	-	-	-	269
SA [23]	1,126,251	1,127,955	1,129,112	0.25	17
IQEA [31]	1,123,890	1,124,320	1,124,504	0.05	42
EPSO [32]	1,123,773	1,125,513	1,127,070	0.29	16
GA [33]	1,124,290	1,124,678	1,125,103	0.07	12.3
MPL	**1,123,522**	-	-	-	**0.7**
MPL-CSS	**1,123,522**	**1,123,522**	**1,123,522**	**0.00**	**0.7**
		40 Unit system			
BF [28]	2,246,223	-	-	-	476
SA [23]	2,250,063	2,252,125	2,254,539	0.20	88
IQEA [31]	2,245,151	2,246,026	2,246,707	0.07	132
EPSO [32]	2,244,772	2,248,741	2,251,241	0.29	36
GA [33]	2,246,165	2,246,818	2,247,532	0.06	20
MPL	2,244,800	-	-	-	**3**
MPL-CSS	**2,243,545**	**2,243,545**	**2,243,545**	**0.00**	**6**
		60 Unit system			
BF [28]	3,369,237	-	-	-	1076
SA [23]	-	-	-	-	-
IQEA [31]	3,365,003	3,365,667	3,366,223	0.04	273
EPSO [32]	3,364,250	3,368,686	3,371,783	0.22	54
GA [33]	3,365,431	3,366,178	3,366,995	0.05	60
MPL	3,362,714	-	-	-	**7**
MPL-CSS	**3,361,407**	**3,361,407**	**3,361,407**	**0.00**	**14**
		80 Unit system			
BF [28]	4,491,287	-	-	-	2460
SA [23]	4,498,076	4,501,156	4,503,987	0.13	405
IQEA [31]	4,486,963	4,487,985	4,489,286	0.05	453

Continued

EPSO [32]	4,487,742	4,491,749	4,494,032	0.14	71
GA [33]	4,487,766	4,488,826	4,489,983	0.05	98
MPL	4,485,174	-	-	-	**33**
MPL-CSS	**4,482,807**	**4,482,807**	**4,482,807**	**0.00**	**46**
			100 Unit System		
BF [28]	5,611,514	-	-	-	4460
SA [23]	5,617,876	5,624,301	5,628,506	0.19	696
IQEA [31]	5,606,022	5,507,561	5,608,525	0.04	710
EPSO [32]	5,608,055	5,614,073	5,619,445	0.20	91
GA [33]	5,606,811	5,609,492	5,612,420	0.10	150
MPL	5,604,574	-	-	-	74
MPL-CSS	**5,601,253**	**5,601,253**	**5,601,253**	**0.00**	**294**

The execution time is an important factor, too. It is to be mentioned here that computational time is not a good measure for comparing performances of two algorithms, as the computing machines as well as their technical specifications are usually different. Moreover, the computational time will generally vary, even in the same machine, mainly due to the levels of code optimization and programming skills. Therefore, computational times of the different algorithms have been linearly normalized by frequency proportions of the employed CPUs (scaled for a 2.6 GHz processor) for fairer comparison, and are implicitly reported in Table 3. The bold cells in Table 3 indicate the minimum cost and computation time among various UC methods for 10-units to 100-units scheduling, which indicate that the computation time of the proposed MPL-CSS is the fastest, except the MPL method. Additionally, the computation time of the proposed MPL-CSS is higher than that of the EPSO in the 100-unit scheduling condition; however, the operating cost of the proposed MPL-CSS is lower than that of the EPSO. Therefore, the proposed methods can achieve high-quality UC solutions within a reasonable time.

4.2. Seven-Day Unit Scheduling

In the section, the MPL-CSS is implemented to a seven-day unit scheduling. The load curve in [27] is used, and shown in Table 4. For each hour, UC is carried out considering the corresponding load factor. The seven-day unit scheduling result for MPL-CSS, and BF algorithms, which includes operating cost and computation time of 10- to 100-unit systems, is shown in Table 5. It is shown that MPL-CSS results in lower cost as compared to other algorithms. Seldom publications showed the result of a seven-day UC problem. Therefore, Table 5 only give a comparison between the proposed algorithm and BF method.

5. Conclusion

This paper proposes a hybrid algorithm for solving the UC problem. The algo-

Table 4. Load factor of each day.

Day	1	2	3	4	5	6	7
Load Factor	1	0.95	0.9	0.9	0.92	0.85	0.8

Table 5. Comparison of a seven-days UC problem.

Units	Total cost ($)		Execution time (sec)	
	BF [27]	MPL-CSS	BF [27]	MPL-CSS
10	3,529,021	**3,503,776**	-	2.1
20	7,034,913	**6,990,389**	-	12.5
40	14,038,265	**13,955,288**	-	74.9
60	21,098,765	**20,941,214**	-	234.2
80	28,167,453	**27,907,032**	-	608.7
100	35,257,271	**34,901,485**	-	1392.5

rithm combines the MPL algorithm with the CSS method; the MPL method is utilized to obtain an initial UC solution over a 24-hr period, and the CSS method is then applied to a limited period to achieve a better UC solution. The algorithm allows users to choose between two alternatives—a fast engine using the MPL only and an accurate engine with the scarification of computational time using the MPL-CSS method. The efficiency of the proposed method is proved by a typical system with 10 to 100 units. The UC results by using the proposed algorithm are compared with those obtained by using previously developed methods. The numerical results reveal that the cost of generation using the MPL-CSS is consistently less than those by using other algorithms in most cases. Additionally, the computation time is also superior as compared to other heuristic algorithms.

Acknowledgements

This work was financially supported by the Ministry of Science and Technology in Taiwan under the project of This work was financially supported by the Ministry of Science and Technology in Taiwan under the project of Enhancement of Power Quality for Transmission/Distribution Network and Development of Wheeling Technologies and Operations Planning (Project number: 106-3113-E-194-001-).

References

[1] Shahidehpour, M., Yamin, H. and Li, Z. (2002) Market Operations in Electric Power Systems. Wiley, New York. https://doi.org/10.1002/047122412X

[2] Padhy, N.P. (2004) Unit Commitment—A Bibliographical Survey. *IEEE Transactions on Power Systems*, **19**, 1196-1205. https://doi.org/10.1109/TPWRS.2003.821611

[3] Jeong, Y.W., Park, J.B., Jang, S.H. and Lee, K.Y. (2010) A New Quantum-Inspired Binary PSO: Application to Unit Commitment Problems for Power Systems. *IEEE Transactions on Power Systems*, **25**, 1486-1495.

https://doi.org/10.1109/TPWRS.2010.2042472

[4] Delarue, E., Cattrysse, D. and D'haeseleer, W. (2013) Enhanced Priority List Unit Commitment Method for Power Systems with a High Share of Renewable. *Electric Power Systems Research*, **10,** 115-123. https://doi.org/10.1016/j.epsr.2013.07.014

[5] Sheble, G.B. (1990) Solution of the Unit Commitment Problem by the Method of Unit Periods. *IEEE Transactions on Power Systems*, **5**, 257-260. https://doi.org/10.1109/59.49114

[6] Senjyu, T., Shimabukuro, K., Uezato, K. and Funabashi, Y.A. (2003) Fast Technique for Unit Commitment Problem by Extended Priority List. *IEEE Transactions on Power Systems*, **18**, 882-888. https://doi.org/10.1109/TPWRS.2003.811000

[7] Kumar, S.S. and Palanisamy, V. (2007) A Dynamic Programming Based Fast Computation Hopfield Neural Network for Unit Commitment and Economic Dispatch. *Electric Power Systems Research*, **77**, 917-925. https://doi.org/10.1016/j.epsr.2006.08.005

[8] Pang, C.K., Sheble, G.B. and Albuyeh, F. (1981) Evaluation of Dynamic Programming Based Methods and Multiple Area Representation for Thermal Unit Commitments. *IEEE Transactions on Power Apparatus and Systems*, **PAS-100**, 1212-1218. https://doi.org/10.1109/TPAS.1981.316592

[9] Benhamida, F. and Abdelbar, B. *(*2010) Enhanced Lagrangian Relaxation Solution to the Generation Scheduling Problem. *International Journal of Electrical Power & Energy Systems*, **32**, 1099-1105. https://doi.org/10.1016/j.ijepes.2010.06.007

[10] Ongsakul, W. and Petcharaks, N. (2004) Unit Commitment by Enhanced Adaptive Lagrangian Relaxation. *IEEE Transactions on Power Systems*, **19**, 620-628. https://doi.org/10.1109/TPWRS.2003.820707

[11] Virmani, S., Adrian, E.C., Imhof, K. and Muhherjee, S. (1989) Implementation of a Lagrangian Based Unit Commitment Problem. *IEEE Transactions on Power Systems*, **4**, 1373-1380. https://doi.org/10.1109/59.41687

[12] Cohen, A.I. amd Yoshimura, M. (1983) A Branch-and-Bound Algorithm for Unit Commitment. *IEEE Transactions on Power Apparatus and Systems*, **PAS-102**, 444-451.https://doi.org/10.1109/TPAS.1983.317714

[13] Ohuch, A. and Kaji, I. (1975) A Branch-and-Bound Algorithm for Start-Up and Shutdown Problem of Thermal Generating Units. *Institute of Electrical Engineering*, **Jpn. 95-B**, 461-468.

[14] Carrión, M. and Arroyo, J.M. (2006) A Computationally Efficient Mixed-Integer Linear Formulation for the Thermal Unit Commitment Problem. *IEEE Transactions on Power Systems*, **21**, 1371-1378. https://doi.org/10.1109/TPWRS.2006.876672

[15] Norouzi, M.R., Ahmadi, A., Nezhad, A.E. and Amir, G. (2014) Mixed Integer Programming of Multi-Objective Security-Constrained Hydro/Thermal Unit Commitment. *Renewable and Sustainable Energy Reviews*, **33**, 585-593.

[16] Appala, V.S. and Erlich, I. (2008) A New Approach for Solving the Unit Commitment Problem by Adaptive Particle Swarm Optimization. *Proceedings of the 2008 IEEE Power & Energy Society General Meeting*, Pittsburgh, 20-24 July 2008, 1-6. https://doi.org/10.1109/PES.2008.4596390

[17] Kazarlis, S.A., Bakirtzis, A.G. and Petridis, V.A (1996) Genetic Algorithm Solution to the Unit Commitment Problem. *IEEE Transactions on Power Systems*, **11**, 83-92. https://doi.org/10.1109/59.485989

[18] Swarup, K.S. and Yamashiro, S. (2002) Unit Commitment Solution Methodology

Using Genetic Algorithm. *IEEE Transactions on Power Systems*, **17**, 87-91. https://doi.org/10.1109/59.982197

[19] Senjyu, T., Yamashiro, H., Shimabukuro, K., Uezato, K. and Funabashi, T. (2003) Fast Solution Technique for Large-Scale Unit Commitment Problem Using Genetic Algorithm. *IEE Proceedings—Generation, Transmission and Distribution*, **150**, 753-760. https://doi.org/10.1049/ip-gtd:20030939

[20] Christober, A.R.C. and Mohan, M.R. (2007) An Evolutionary Programming Based Simulated Annealing Method for Solving the Unit Commitment Problem. *International Journal of Electrical Power & Energy Systems*, **29**, 540-550. https://doi.org/10.1016/j.ijepes.2006.12.001

[21] Venkatesh, P., Gnanadass, R. and Padhy, N.P. (2003) Comparision and Application of Evolutionary Programming Techniques to Combined Economic Emission Dispatch with Line Flow Constraints. *IEEE Transactions on Power Systems*, **18**, 688-692. https://doi.org/10.1109/TPWRS.2003.811008

[22] Dudek, G. (2010) Adaptive Simulated Annealing Schedule to the Unit Commitment Problem. *Electric Power Systems Research*, **80**, 465-472. https://doi.org/10.1016/j.epsr.2009.10.019

[23] Simopoulos, D.N., Kavatza, S.D. and Vournas, C.D. (2006) Unit Commitment by an Enhanced Simulated Annealing Algorithm. *IEEE Transactions on Power Systems*, **21**, 68-76. https://doi.org/10.1109/TPWRS.2005.860922

[24] Zhao, B., Guo, C.X., Bai, B.R. and Cao, Y.J. (2006) An Improved Particle Swarm Optimization Algorithm for Unit Commitment. *International Journal of Electrical Power & Energy Systems*, **28**, 482-490. https://doi.org/10.1016/j.ijepes.2006.02.011

[25] Pappala, V.S. and Erlich, I.A. (2008) New Approach for Solving the Unit Commitment Problem by Adaptive Particle Swarm Optimization. *Proceedings of IEEE Power & Energy Society General Meeting*, Pittsburgh, 20-24 July 2008, 1-6. https://doi.org/10.1109/pes.2008.4596390

[26] Ting, T.O., Rao, M.V.C. and Loo, C.K. (2006) A Novel Approach for Unit Commitment Problem via an Effective Hybrid Particle Swarm Optimization. *IEEE Transactions on Power Systems*, **21**, 411-418. https://doi.org/10.1109/TPWRS.2005.860907

[27] Shukla, A. and Singh, S.N. (2016) Advanced Three-Stage Pseudo-Inspired Weight-Improved Crazy Particle Swarm Optimization for Unit Commitment Problem. *Energy*, **96**, 23-36. https://doi.org/10.1016/j.energy.2015.12.046

[28] Eslamian, M., Hosseinian, S.H. and Vahidi, B. (2009) Bacterial Foraging-Based Solution to the Unit-Commitment Problem. *IEEE Transactions on Power Systems*, **24**, 1478-1488. https://doi.org/10.1109/TPWRS.2009.2021216

[29] Ebrahimi, J., Hosseinian, S.H. and Gharehpetian, G.B. (2011) Unit Commitment Problem Solution Using Shuffled Frog Leaping Algorithm. *IEEE Transactions on Power Systems*, **26**, 573-581. https://doi.org/10.1109/TPWRS.2010.2052639

[30] Barati, M. and Farsangi, M.M. (2014) Solving Unit Commitment Problem by a Binary Shuffled Frog Leaping Algorithm. *IET Generation, Transmission & Distribution*, **8**, 1050-1060. https://doi.org/10.1049/iet-gtd.2013.0436

[31] Jeong, Y.W., Park, J.B., Shin, J.R. and Lee, K.Y. (2009) A Thermal Unit Commitment Approach Using an Improved Quantum Evolutionary Algorithm. *Electric Power Components and Systems*, **37**, 770-786. https://doi.org/10.1080/15325000902762331

[32] Xiaohui, Y., Anjun, S., Hao, N., Yanbin, Y. and Liang, W. (2011) Unit Commitment Problem Using Enhanced Particle Swarm Optimization Algorithm. *Soft Compu-*

ting, **15**, 139-148. https://doi.org/10.1007/s00500-010-0541-y

[33] Datta, D. (2013) Unit Commitment Problem with Ramp Rate Constraint Using a Binary-Real-Coded Genetic Algorithm. *Applied Soft Computing*, **13**, 3873-3883. https://doi.org/10.1016/j.asoc.2013.05.002

[34] Singhal, P.K., Naresh, R. and Sharma, V. (2013) Binary Fish Swarm Algorithm for Profit-Based Unit Commitment Problem in Competitive Electricity Market with Ramp Rate Constraints. *IET Generation*, *Transmission & Distribution*, **7**, 298-308.

[35] Wu, Z. and Chow, T.W.S. (2015) Binary Neighbourhood Field Optimisation for Unit Commitment Problems. *IET Generation*, *Transmission & Distribution*, **9**, 1697-1707.

[36] Kamboj, V.K., Bath, S.K. and Dhillon, J.S. (2016) Implementation of Hybrid Harmony Search/Random Search Algorithm for Single Area Unit Commitment Problem. *International Journal of Electrical Power & Energy Systems*, **82**, 228-249. https://doi.org/10.1016/j.ijepes.2015.11.045

[37] Reddy, G.V.S., Ganesh, V. and Rao, C.S. (2016) Implementation of Clustering Based Unit Commitment Employing Imperialistic Competition Algorithm. *International Journal of Electrical Power & Energy Systems*, **82**, 621-628. https://doi.org/10.1016/j.ijepes.2016.04.043

[38] Kaveh, A. and Talatahari, S. (2010) A Novel Heuristic Optimization Method: Charged System Search. *Acta Mechanica*, **213**, 267-289. https://doi.org/10.1007/s00707-009-0270-4

[39] Sishaj, P.S., Narayana, P.P. and Anand, R.S. (2006) An Ant Colony System Approach for Unit Commitment Problem. *International Journal of Electrical Power & Energy Systems*, **28**, 315-323. https://doi.org/10.1016/j.ijepes.2005.12.004

[40] Chu, C.C. and Tsai, M.S. (2013) Application of Novel Charged System Search with Real Number String for Distribution System Loss Minimization. *IEEE Transactions on Power Systems*, **28**, 3600-3609. https://doi.org/10.1109/TPWRS.2013.2264836

Nomenclature

D_t	Load demand for time period *t*.
R_t	Spinning reserve requirement.
ST_k^{hot}	Hot start cost of unit *k*.
ST_k^{cold}	Cold start cost of unit *k*.
$\overline{p}_k$	Maximum generation of unit *k*.
$\underline{p}_k$	Minimum generation of unit *k*.
δ_k	Maximum ramp-rate of unit *k*.
K	Set of units.
J	Set of wind units.

Scientific Research Publishing

Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.
A wide selection of journals (inclusive of 9 subjects, more than 200 journals)
Providing 24-hour high-quality service
User-friendly online submission system
Fair and swift peer-review system
Efficient typesetting and proofreading procedure
Display of the result of downloads and visits, as well as the number of cited articles
Maximum dissemination of your research work

Submit your manuscript at: http://papersubmission.scirp.org/
Or contact sgre@scirp.org