GENERATOR SCHEDULING (A COMBINATORIAL OPTIMIZATION PROBLEM) BY ANNEALING METHOD

A. Y. Saber[1], Wahed Newaz[2] and A. K. M. Zaidi Sattar[3]
[1]The University of Asia Pacific (UAP)
[2]Rajshahi University of Engineering and Technology (RUET)
[3]Daffodil International University (DIU)
E-mail: saber@uap-bd.edu

Abstract: *Generator scheduling is a combinatorial optimization problem and this paper presents a new version of annealing (SA) method to model and solve the scheduling problem. Firstly, solution is decomposed into hourly schedules and each hourly schedule is modified by decomposed-SA using bits flipping. If the generated new hourly schedule is better, by convention it is accepted deterministically. A worse hourly schedule is accepted with temperature dependent SA probability. A new solution consists of these hourly schedules of entire scheduling period after repair as unit-wise constraints may not be fulfilled at the time of individual hourly schedule modification. This helps to direct and modify schedules of appropriate hours. Secondly, this new solution is accepted for the next iteration if its cost is less than that of current solution. A higher cost new solution is accepted with temperature dependent SA probability again. Besides, problem dependent other features are incorporated to save the execution time. The proposed method is tested using the reported problem data sets. Simulation results are compared to previous reported results. Numerical results show an improvement in solution cost and time compared to the results obtained from powerful algorithms.*

Keywords: *Simulated annealing, decomposition, probability distribution, local minima, best heat rate.*

1. Introduction

Generator scheduling, familiar as unit commitment (UC) in power systems, involves to properly schedule the on/off states of all the generators in a system. In addition to fulfill a large number of constraints, the optimal generator scheduling should meet the forecasted load demand, calculated in advance, plus the spinning reserve requirement at every time interval such that the total cost is minimum. The generator scheduling problem is a combinatorial optimization problem with both binary and continuous variables. The number of combinations of 0-1 variables grows exponentially for a large scale problem. Therefore, the UC is one of the most difficult problems in optimization area.

Various numerical optimization techniques have been employed to approach the UC problem since the last 4-decade. Among these methods, the priority list (PL) [1-3] commits in ascending order of units with full-load cost so that the most economic base load units are committed first in order to meet the load demand. PL method is very fast but highly heuristic and gives schedules with relatively higher operation cost. Branch-and-bound (BB) method [4-6] has the danger of a deficiency of storage capacity and increasing the calculation time enormously as being a large scale problem. Lagrangian relaxation [LR] method [7-11] concentrates on finding an appropriate co-ordination technique for generating feasible primal solution, while minimizing the duality gap. The main problem with the LR method is the difficulty encountered in obtaining feasible solutions.

The meta-heuristic methods [12-16] are iterative techniques that can search not only local optimal solutions but also a global optimal solution depending on problem domain and time. In the meta-heuristic methods, the techniques frequently applied to the UC problem are genetic algorithm (GA), tabu search (TS), evolutionary programming (EP), etc. They are general-purpose search techniques based on principles inspired from the genetic and evolution mechanisms observed in natural systems and populations of living beings. These methods have the advantage of searching the solution space more thoroughly, and avoiding premature convergence to local minima. The main difficulty is their sensitivity to the choice of parameters. However, in case of large scale problem they consume a lot of time and space due to their iterative nature.

Owing to having the aft lily to seek for near global optimal solutions, simulated annealing (SA) [17] has been applied to numerous optimization problems. Performance of raw SA is not satisfactory. In the standard simulated annealing algorithm, a large share of the computation lime is spent in randomly generating and evaluating solutions after a single hit flipping that turn out to be infeasible [118-20]. Expected improvement of SA has not been done yet. Researchers always try to merge SA with other methods where SA solves one of the parts of UC problem [21-23].

In this paper, a twofold simulated annealing method has been developed for the solution of UC problem. It improves only those hours' schedule where appropriate. Better hourly schedules have higher probabilities to make a better solution. Though unpromising hourly schedule may be rejected, it wastes less calculation as a hourly schedule cost calculation is easier than that of whole solution of entire scheduling period. Besides, repair a algorithm for constraint handle and continuous, calculation for economic load dispatch (ELD) are incorporated so that robustness, solution quality, and execution time of SA are improved.

The rest of the paper is organized as follows. In Section 2, problem formulation and constraints of UC are discussed. The proposed method, applied probability distributions and important operation are explained in Section 3 and appendices. Simulation results on two cases are reported in Section 4. Section 5 gives the conclusion.

2. Problem Formulation

A. Nomenclature

V : Binary Solution

I *(t)* : Schedule at hour *t*

$V_i(t)$: i-th unit status at hour *t* (1/0 for on/off)

N : Number of units

H : Scheduling period

$P_i(t)$: Output power of *i*-th unit at hour *t*

$p_i^{\max}$: Maximum output limit of *i*-th unit

$p_i^{\min}$: Minimum output limit of *i*-th unit

$p_i^{\max}(t)$: Maximum output power of *i*-th unit at hour *t* considering ramp rate

$p_i^{\min}(t)$: Minimum output power of *i*-th unit at hour *t* considering ramp down rate

ELD : Economic load dispatch

HR_i : Heat rate of Unit *i*

D(t) : Demand power hour *t*

R(t) : System reserve at hour *t*

MU_i : Minimum up time of unit *i*

MD_i : Minimum down time of unit *i*

$X_i^{on}(t)$: Duration of continuously on of unit *i* at hoar *t*

$X_i^{off}(t)$: Duration of continuously off of unit *i* at hoar *t*

SC_i : Start-up cost of *i*-th unit

FC_i ($P_i(t)$: Fuel cost of unit *i* at hour *t*

TC : Total cost

ΔF : Amount of *TC* improvement/deterioration of new solution

$h\text{-}cost_i/c\text{-}cost_i$: Hot/cold start cost of *i*-th unit

$c\text{-}s\text{-}hour_i$: Cold start hour of *i*-th unit

I : Iteration count

r : Temperature reduction factor

T_I : Temperature at *I*-th iteration

P_r : Probability distribution

RUR_i/RDR_i : Ramp up/down rate of unit *i*

B. Objective Function

The objective of UC problem is the minimization of the total cost.

1. Total cost

The objective function of UC problem consists of fuel cost and start-up cost, which is defined as [13]

$$min\ TC = \sum_{i=1}^{N} \sum_{t=1}^{H} [FC_i(P_i(t)) + SC_i(1-V_i(t-1))]V_i(t). \quad (1)$$

2. Fuel cost

Fuel cost of a thermal unit is expressed as a second order function of each unit output as follow:

$$FC_i(P_i(t) = a_i + biP_i(t) + c_iP^2_i(t) \quad (2)$$

where a_i,*bi and* c_i are positive fuel cost coefficients.

3. Start-up cost

Start-up cost for restarting a decommitted thermal unit, which is related to the temperature of the boiler, is included n the model [3]

$$SC_i(t) = \begin{cases} h-\cos t_i : MD_i \le X_i^{off}(t) \le H^{off}{}_i \\ c-\cos t_i : X^{off}{}_i(t) > H^{off}{}_i \end{cases} \quad (3)$$

$$H_i^{off} = MD_i + c\text{-}s\text{-}hour_i \quad (4)$$

4. Shut-down cost
Shut-down cost is usually a constant value for each unit. In this paper, the shut-down cost has been taken equal to 0 for all units and it is excluded from the objective function.

1) Constraints: The constraints that must be satisfied during the optimization process are as follows:

1. System power balance
The generated power from all the committed units must satisfy the load demand which is defined as

$$D(t) = \sum_{i=1}^{N} P_i(t)) \qquad (5)$$

2. Spinning reserve
To maintain system reliability, adequate spinning reserves are required.

$$\sum_{i=1}^{N} V_i(t)).\ P_i^{max}(t) \leq D(t) + R(t) \qquad (6)$$

3. Generation limits
Each unit has generation range, which is represented as

$$P_i^{max} \leq P_i(t) \leq P_i^{max} \qquad (7)$$

4. Minimum up/down time
Once unit is committed/decommitted, there is a predefined minimum time after it can be decommitted/committed.

$$\left.\begin{aligned} MU_i &\leq X_i^{on}(t) \\ MD_i &\leq X_i^{off}(t) \end{aligned}\right\} \qquad (8)$$

5. Ramp rate
For each unit, output is limited by ramp up/down rate at each hour as follow:

$$P_i^{max} \leq P_i(t) \leq P_i^{max}(t) \qquad (9)$$

where $P_i^{max}(t) = max\ (P_i(t\text{-}1)\text{-}RDR_i,\ P_i^{max})$
and $P_i^{max}(t) = min\ (P_i(t\text{-}1)\text{-}RUR_i,\ P_i^{max})$

6. Initial status
At the beginning of schedule, the unit initial status must be taken into account.

3. Proposed Method

Ann jug, physically, refers to the process of heating up a solid to a high temperature followed by slow cooling achieved by decreasing the temperature of the environment in steps. By making an analogy between the annealing process and the optimization problem, a great class of combinatorial optimization problems can be solved following the same procedure of transition from equilibrium state to another, reaching minimum energy et the system in SA.

It is not an optimistic idea to reach the solution of global minimum cost from huge search space within practical/real time limit. Therefore, our goal is to find relatively better solution than existing solutions. In standard annealing (SA) method, better solutions are accepted deterministically (i.e. probability is 1) and other solutions are accepted with temperature dependent lower probabilities. When there are huge search domain, its performance is not satisfactory due to randomly generating unpromising solutions. It is not easy to find the appropriate positions in an entire solution where improvement is possible by flipping bits. However, it is easy to find a bit ct a si11 i hourly schedule for improvement. Thus, the BC is decomposed into hourly schedule and SA is applied to each schedule individually-called decomposed SA in this paper. This may suffer from coupling effect for unit constraints. A solution consists of the outcomes of hourly SA with proper repair. Besides improvement of hourly schedule does not guarantee better solutions. Hence, after repair second time is applied to the solution for its final acceptance in each iteration to reduce couping effect-called coupling SA. Problem oriented repair function and continuous are incorporated to make the method more powerful, faster and robust. Proposed algorithm is shown below:

Algorithm: twofold-SA for *N*-unit *H*-hour system scheduling

begin
iteration, 1=0 and temperature, $T_r = T_0$;
generate an initial solution (V_o) based on load demand using lookup table or priority list method, and add some excess units to V_o
with probability distribution, $Pr_{excess}i$
both current solution (V_c) and new solution (V_n) are initialized by V_0
cost of V_c, $F(V_c)$ = VERY LARGE VALUE initially without calculating ELD;
repeat

repeat
valid = repair (V_n); // to fulfill constrains
if (valid=O) then go to RETRY;
evaluate F(V_n), intelligently including
modified (continuous λ) ELD;
if (F(V_n,) < F(V_c) then $V_c = V_{ni}$, // Coupling-SA of solution V
else $V_c = V_n$ with lower probability,
$Pr_c = S_c.\exp(F(V_c) - F(V_n))/T$;
RETRY:
t=1;
[$I_c(1)$ $I_c(2)$ ...$I_c(H)$] = V_{ci} // Decompose solution
repeat // Decomposed-SA of hourly schedule 1(t)
generate a new hourly schedule, I*(t)*
in the neighborhood of $I_c(t)$ by flipping
1-bit of $I_c(t)$ randomly;
if (HC(I_n(t)) < HC($I_c(t)$ then $I_n(t) = I_n(t)$; // no action

Table 1: Binary Solution, V

Unit	Hour (1~H)
Unit 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Unit 1	0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
.	....
.	....
Unit *N*	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0
Sign	0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1 0 01

else if
(random[0,1)<$S_{hr}\exp\{(HC(I_n(t)-HC(I_c(t)))/T\}$) then $I_n(t) = I_n(t)_i$
else $I_n(t) = I_c(t)$;
increase, t = t + 1;
until (t ≤ scheduling period H);
$V_n = [I_n(1)\ I_n(2)\ ....\ I_n(H)]$;
until (terminating condition at each temperature);
next iteration, I = I + 1;
new temperature, $T_r = g(T_0,r,I)$;
until (SA stopping criteria);
end.

All the important terms, probability distributions and functions of the proposed algorithm will be described in the following subsections.

3.1 Solution Structure

We have use $(N + 1) \times H$ binary matrix (Table 1) where first $N \times T$ matrix represents the UC solution coding and the last row vector indicates any change of the schedule. The solution is decomposed into I*(1),* I*(2)*, ... I*(H)* hourly schedules (column vector) or [I*(1),* I*(2)*, ... I*(H)*] = V for decomposed SA. Each bit of the last row vector is a sign to indicates whether new or current hourly schedule is accepted after decomposed SA. In this paper, solution indicates $N \times H$ matrix of entire scheduling period and hourly schedule represents a column vector *I(t)* of the solution. Only one extra temporary storage (e.g. $(N + 1)$. H-bit) is used for twofold-SA algorithm. Bitwise XOR operation is performed for SA flipping of hourly schedule. Introduced sign vector will prohibit to recalculate the cost of unchanged hour schedule at the time of coupling SA.

3.2 Initial Solution

It is difficult to generate feasible solutions when the initial solution is generated it random. So, the generation of initial solution is carried out by focusing on the load, including peak and bottom, of the day. It is an optimistic idea that at peak load all the units will be committed and in the system, there may be no unit which is unused through the entire scheduling period. Units will be turned on in ascending order of cost per to it until (6) is satisfied. All sign bits are set 1for the initial solution as the schedule is new.

$$Pr_{excess} = K \log_{10}(N) \qquad (10)$$

To satisfy (8) and (9) of all the units, some excess units need to be on after satisfying (6). For a large scale problem, condition (8) will be tighter and search space will increase exponentially. So, after satisfying (6), rest of the units will be turned on with a predefined logarithmic probability distribution number of units as in (10). The best parameter value of *K* up to 100-unit systems and 24 hours scheduling period is 0.35 from simulation. Excess units are added so that the optimalsolution is not overlooked during searching.

Minimum up/down time constraint is not considered for V_0. Considering the initial solution as base, new solution V_n us generated and it is repaired in Sections III-C and III-D, respectively in each iteration before ELD calculation.

C. Generating a New Solution

Twofold-SA looks for the appropriate hour to modify and generate new solutions by decomposed SA. So the solution is hourly decomposed first, e.g. $[I_c(1)\ I_c(2)\ .....I_c(H)] =$

V_c. One bit is randomly flipped at each hourly schedule $I_c(t)$. New solution V_n = [$I_n(1)$ $I_n(2)$ .. $I_n(H)$] where I_n *(t)* is randomly one bit inversion of I_c *(t)* i.e. Hamming distance of a I_c *(t)* and is $I_n(t)$is 1. For ELD calculation, continuous is modified according to an included or excluded unit.

Now, the cost of new hourly schedule $I_n(t)$ is calculated by using ELD and equation (2) as follow:

$$HC(I_n(t) = \Sigma\, FC_i(P_i(t).V_i(t). \qquad (11)$$

Decomposed SA selects a schedule by the following rules:

Rule 1 : Being a minimization problem, if HC($I_n(t)$is less than HC($I_c(t)$ then SA algorithm honours to accept HC($I_n(t)$as a new hourly schedule deterministically.

Rule 2: However, if HC($I_n(t)$ is greater than HC($I_c(t)$ then SA algorithm imposes the following probability to accept $I_n(t)$ as new hourly schedule:

$$Pr_{hr} = S_{hr}\,.e^{-\Delta HC/T}{}_{1} \qquad (12)$$
$$\Delta\, HC = HC(I_n(t) - HC(I_c(t))$$
$$T_1 = r^1\ .\ T_0$$

Here problem oriented scaling factors, S_{hr} is introduced as well for more flexibility.

Rule 3 : $I_n(t)$ is dishonoured otherwise and $I_c(t)$ is restored as the new hourly schedule i.e. $I_n(t) = I_c(t)$

Sign vector will be simultaneously changed accordingly. New solution is generated by merging all the new hourly schedules e.g. V_n = [I_n (1) I_n (2) .. I_n(H)]. Owing to the large set of physical and operational constraints inherent in the UC problem, generated new solution may not satisfy all the constraints. Therefore, repair process is applied to satisfy constraints for feasible solutions.

D. Solution Repair

Spinning reserve (6) and minimum up/down time (8) are the vital constraints of UC problem. After random bits flipping, constraints arc frequently violated. In this stage, in intelligent repair can accelerate solution quality and reduce effective execution time. Proposed repair function is shown below:

```
Function repair(schedule Vn)
begin
temporary schedule, Vtemp = Vni
repeat
turn off excess units of Vtemp
using
probability distribution,
Proff;
protect deficiency in units of
Vtemp
if needed;
maintain minimum up/down time
constraint
of Vtemp using bit pattern
splitting
merging and bitwise
operations;
until (minimum up/down time
and spinning
reserve are satisfied) or
(predefined maximum number of
iterations is reached));
if (minimum up/down time and
spinning reserve are
satisfied) then
update Vn = Vtemp and return 1
which indicates successful
repair;
else
return 0 which indicates
unsuccessful
repair;
end
```

Small UC problem contains e number of excess units than large scale UC problem to maintain constraints. Excess units, which are tested at each hour if (6) is satisfied by shutting-down one or more units, are decided to turn off according to the normal distribution f number of units in a system, N as

$$Pr_{off} = Pr_0 + S_p \frac{1}{\sqrt{2\pi}} e^{-(N-\mu)^2/2\sigma^2} \qquad (13)$$

The best parameter values of offset Pr_0, scaling factor S_p, standard deviation ó, and mean value μ up to 100-unit systems and 24 hours scheduling period are 0.1, 0.6, 25, and 10, respectively from prior simulation. No excess units are turned off deterministically.

lf (6) is violated it any scheduling period, the system suffers from deficiency in units. Then, decommitted units are forced to turn on randomly until (6) is satisfied.

For each unit, bit patterns of entire scheduling period is a 2-class (0 and 1) system. Patterns of 0's and 1's stand for off and on states of a unit, respectively. If (8) is not satisfied for a pattern, either the pattern vector is inverted and merged to its neighbors or one of the neighbor patterns is split and after inversion near split part is merged to itself. Between the two options, better one is chosen for which less number of bitwise operations a needed Here, neighbor may be left or right and they are equally likely Bit wise operations are shown in Appendix I.

Any modification of the schedule during entire repair process, sign vector will be simultaneously changed accordingly.

For unsuccessful repair of an *N*-unit system, maximum number of trials is [N/3] in this study. It can be seen from prior simulation that there is a little chance to get any feasible solution after first [N/3] trials but significant time is spent.

E. Economic Dispatch

The economic load dispatch (ELD) is a computational part in UC problem to evaluate solutions. Besides, double ELD calculation is needed in each iteration for twofold-SA. System power balance (5), generation limits (7), and ramp rate (9) constraints are fulfilled in this stage. Therefore, to save the computational efforts, λ iteration is performed very carefully and intelligently using the following criteria for ELD:

Criterion 1: ELD is performed if the schedule is able to satisfy (a) minimum up/down time, (b) spinning reserve after process.

Criterion 2 : ELD is performed for those hours when (a) the introduced sign bits are 1's that mean new schedules have accepted due to decomposed SA, (b) unit output range is changed due to ramp rate constraint (if applicable). Always new continuous λ is calculated recursively from its previous value after any change of schedule. It reduces overhead.

Criterion 3 : In iterative method, relaxed ELD is performed at he beginning and exact ELD at near final iterations. In relax ELD, the iteration is continued till error is smaller than a specified (say 0.1% of load demand) accuracy.

Relax ELD does not effect the accuracy of total cost of uric solution as exact ELD is calculated later but it saves time. After ELD, sign vector will be reset (zero). Amount of solution improvement or deterioration, which is calculated after ELD, plays an important role for the acceptance or rejection of a new solution.

F. Acceptance Criteria

A repaired feasible new solution by decomposed SA does not guarantee the improvement of cost due to coupling effect. So another SA, called coupling SA s applied to select a solution finally for the next iteration. For a lower cost feasible solution, couping SA algorithm decides to honour it by accepting deterministically. Other solutions are accepted with a temperature dependent lower probability density function of the exponential distribution, Pr_c, that is

$$\left.\begin{aligned} Pr_c &= S_c.e^{-\Delta F.Ti} \\ \Delta F &= F(V_n) - F(V_c) \\ T_I &= e^I.T_0 \end{aligned}\right\} \qquad (14)$$

F(V) is calculated using (1) and problem oriented scaling factor S_c is multipled for flexibility. In other words, the following function is used for coupling SA :

Solution Coupling-SA (Solution V_c, Solution V_n, Temperature T)

if ($F(V_n)$ < ($F(V_c)$) return V_n;
else if (random[0,1) < S_c . *exp* $(F(V_c) - F(V_n))/T)$ *return* V_n;
else return V_c;
end

As the hourly schedule is modified first by decomposed SA, most of the cases coupling SA selects better solutions V_n for the next iteration. In few cases, higher cost new solution V_n is rejected after all calculations including ELD. It leads to an advantage with respect to single SA.

G. Cooling Schedule

The reduction of the temperature in successive iteration is governed by the following geometric function.

$$T_I = g(T_0,r,I) = r^I \cdot T_0 \qquad (15)$$

Where T_0 and T_I and the initial temperature and current temperature at the *I*-th or iteration, respectively and r is temperature reduction factor. We have used an initial temperature value of 5,000 and reduction factor value of 0.98 in this study.

H. Stopping Criterion

Simulated annealing loop is stopped running when there is in significant improvement in the solution or the maximum number of iterations is reached. In this paper, stopping criterion is the maximum number of iterations that is 600 (constant) for which the system is cool enough and *[N/5]* number of trials is made at each temperature to reach the terminating condition where *N* is the number of units of the system. Relax and longer stopping criterion and terminating condition produce better solutions with longer execution time.

4. Simulation Results

All calculations have been run on Intel(R) Celeron(TM) CPU(1.2 GHz), 128 MB RAM, Windows 2000 OS and turbo 3++ 3.0 compiler. All results of twofold-SA are collected.

Table-II: Test Results Of The Proposed Two Fold-Sa For Cost And Time

Ramp	Success (%)	Operating Cost				Execution Time		
		Best (M$)	Worst (M$)	Avg. (M$)	Vari. (%)	Max. (sec)	Min (sec	Avg. (sec)
Yes	100	197.11	199.65	198.06	1.29	8.46	5.38	7.69
No	100	195.85	198.12	196.61	1.16	6.87	4.29	5.88

After 600 iterations of which first 300 iterations perform relax ELD where error is less than 0.1% of load demand. No extra temporary storage is needed except one instance of binary solution and some local variables. The standard input data set (e.g. 38-unit system) is used to compare with other popular methods.

In order to perform simulations on the same condition of [4, 8, 19, 24], the spinning reserve re is assumed to be 11% of the load demand, hot and cold start-up costs are assumed to be the same constant value (start-up cost) and total scheduling period is 24 hours.

Test results are shown in Table II. The best, worst, and average findings of produced method are reported together both considering and neglecting ramp rate constraint. It always converges and operating cost variation is tolerable. Average cost and execution time of 10 runs are very near to the middle position between their maximum and minimum values. So, it is clear that solutions are not biased and they are equally distributed between the best and worst solutions. These facts strongly demonstrate the robustness of proposed twofold-SA. Operating cost considering ramp rate is lower than cost neglecting it as ramp rate is lower than cost neglecting it is ramp rate constraint sometimes prohibits to run low (high) cost units at maximum (minimum) generation limit. In case of neglecting ramp rate, time will be saved as extensive ELD calculation will be exempted for Criterion 2(b) of Section III-F.

Table III shows the comparison of the proposed method to the popular methods, e.g. DP reported in [4], LR reported in [8], SA reported in [9], and CLP reported in [24] with respect to total cost and execution time. The proposed method provides the lowest cost schedule with the minimum execution time of all the above methods. Recall that, load dependent initial solution; excess units and repair for constraints handle; twofold-SA etc. help to achieve the above least cost solution and the introduced sign vector as well as continuous λ to reduce complex overhead ELD calculations helps to achieve the above minimum execution time.

Algorithm	DP1	LR2	SA3	CLP	Proposed Method
Cost (M$) with ramp rate	215.2	214.5	215.6	213.9	198.06
Time (sec)	199	29	2589	17	7.69
Cost (M$) without ramp rate	201.5	209.0	207.8	208.1	196.61
Time (sec)	24	7	1690	10	5.88

5. Conclusion

This paper presents a twofold-SA for optimal generator scheduling. In this paper, our contributions are the appropriate use of standard SA for generator scheduling, introduction of appropriate heuristics for constraints and continuous λ for ELD calculation to overcome the drawbacks of existing methods. Solutions selection and constraints management by appropriate probability distributions and overhead protected ELD calculation by introduced sign vector make the method more powerful, faster and robust. Finally, the simulation results show a big improvement of the proposed algorithm, even though powerful methods are set as benchmark. It is, however, easy to implement and is not memory intensive. In future, intelligent bits flipping as well as security and network constraint many be incorporated in parallel twofold-SA for better performance.

References

[1] H. H. Happ, R. C. Johnshon, and W. J, Wright, "Large scale hydro-thermal unit commitment-method and results," IEEE Trans. Power Appa. Syst. vol. PSA-90, no. 3, pp. 1373-1384, 1971.

[2] C. J. Baldwin, K. M. Dale, and R. F. Dittrich, "A study of economic shutdown of generating units in daily dispatch," IEEE Trans. Power.Appa.Syst., vol. PAS-78, pp. 1272- 1284, 1960.

[3I T. Senjyu, K Saimabukuro, K. Uezato, and T. Funabashi, "A fast technique for unit commitment problem by extended priority list," IEEE Trans. Power Syst., vol 18, no, 2, pp. 882-888, May 2003.

[4] Z. Ouyang and S. M. Shahidehpour, "An intelligent dynamic programming for unit commitment application," IEEE Trans. Power Syst., vol. 6, no 3, pp. 1203-1209, August 1991.

[5] A. I. Cohen and M. Yoshimura, "A branch-and-bound algorithm for unit commitment," IEEE Trans. Power Appa. Syst., vol. PAS-102, pp. 444-451, Feb. 193.

[6] G. S. Lauer, D. P. Bertsekas, N. R.. Sandell, Jr, and T. A. Posbergh, "Solution of large-scale optimal unit commitment problems," IEEE Trans. Power Appa .Syst. vol. PAS-101, pp. 79-86, Jan.1982.

[7] A. J. Svoboda, C.L. Tseng, C. A. Li, and R. B. Johnson, "Short-term resource scheduling with ramp constraints [power generation scheduling]," IEEE Trans. Power Syst., Vol. 12. no. 1, pp. 77-83, Feb. 1997.

[8] W. L. Peterson and S.R. Brammer, "A capacity based Lagrangian relaxation unit commitment with ramp rate constraints,'' IEEE Trans. Power Syst., Vol. 10, no. 2, pp. 1077-1084, May 1995.

[9] F. No. Lee, "A fuel-constrained unit commitment method," IEEE Trans. Power Syst., vol. 4, no. 3, pp. 1208-1218, August 1989.

[10] S. Virmani, E. C. Adrian, K, Imbof, and S. Mukherjee, "Imp1ement relaxation based unit commitment problem," IEEE Trans. Power Syst., vol. 4, no. 4, pp. 1373-1380, Nov. 1989.

[11] F. Zhuang and F. D. Galiana, 'Towards a more rigorous and practical unit by Lagrangia relaxation," IEEE Trans. Power Syst., vol. 3, no. 2, pp. 763-773, May 1988.

[12] I. G. Damousis, A. G. Bakirtzis, and P. S. Dokopoulos, "A solution to the unit-commitment problem using integer-coded genetic algorithm," IEEE Trans. Power Syst. vol. 19, no. 2, pp. 1165-1172, May 2004.

[13] C. P. Cheng, C. W. Liu, and C.-C. Liu, "Unit commitment by Lagrangian relaxation and genetic algorithms," IEEE Trans. Power Syst. vol. 15, no. 2, pp. 707-714, May 2000.

[14] S. A. Kazarlis, A. G. Bakirtzis, and V. Petridis, "A genetic aIgorithm solution to the unit commitment problem' IEEE Trans. Power Syst., vol. 11, no. 1, pp. 83-92, Feb. 1996.

[15] H. Mori and Matsuzaki, "Application of priority-list-embedded tabu search to unit commitment in power systems," Inst. Elect. Eng. Japan, vol. 121-B, no. 4, pp. 535-541, 2001.

[16] K. A. Juste, H. Kita, E. Tanaka, and J. Hasegawa, "An evolutionary programming solution to the unit commitment problem' IEEE Trans. Power Syst., vol. 14, no. 4, pp. 1452-1459, Nov. 1999.

[17] T. Satoh and K, Nara, "Maintenance scheduling by using simulated annealing method [for power pIants]," IEEE Trans. Power Syst., vol. 6, no 2, pp. 850-857, May 1991.

[18] S. Y. W. Wong, "An enhanced simulated annealing approach to unit commitment," Electric Power Energy Syst., vol. 20, PP. 359-368, May 1998.

[19] F. Zhuang and F. D. Galiana, "Unit commitment by simulated annealing," IEEE Trans. Power Syst., vol. 5, no. 1, pp. 311-318, Feb. 1990.

[20] A. H. Mantawy, Y. L. Abel-Mogid, and S. Z. Selim, 'A simulated annealing algorithm for unit commitment," IEEE Trans. Power Syst., vol. 13. no. 1. pp 197-204, Feb. 1998.

[21] H. Mantawy, Y. L. Abel-Mogid, and S. Z. Selim, "Integrating genetic algorithm, tabu search, and simulated annealing for the unit commitment problem," IEEE Trans. Power Syst., vol. 14, no. 3, pp. 829-836, August 1999.

[22] C. P. Cheng, C. –W. Liu, and C.-C. Liu, "Unit commitment by annealing genetic algorithms," Electrical Power and Energy Systems, vol. 24, pp. 149-158, 2000.

[23] G. K. Purushothama and L. Jenkins, "Simulated annealing with local search-A hybrid algorithm for unit commitment." IEEE Trans. Power Syst. vol. 18, no. 1, pp. 273-278, Feb. 2003.

[24] K. Y. Huang, H. T. Yang, and C. L. Yang, "A new thermal unit commitment approach using constraint logic programming," IEEE Trans. Power Syst. vo. 13, no. 3, pp. 936-945, August 1998.

Appendix 1 (Bitwise Operations To Handle Minimum Up/Down Time Constraint)

Let, Schedule (bit patterns) of unit i for H hours is $G_i = P^{m}_{1}P^{n}_{m+1}P^{H}_{n+1}$ and bit pattern P^{n}_{m+1} does not satisfy minimum up/down time constraint.

Option 1 : $G_i = P^{m}_{1}(P^{n}_{m+1})P^{H}_{n+1} \longrightarrow$
$P^{m}_{1}(P^{n}_{m+1})P^{H}_{n+1} \longrightarrow P^{m}_{1}(P^{n}_{m+1})P^{H}_{n+1} \longrightarrow$
P^{H}_{1} (Pattern of 1's or 0's)

Option 2 (left) : $G_i = (P^{m}_{1})P^{n}_{m+1}\ P^{H}_{n+1} \longrightarrow$
$(P^{m}_{1}\ P^{m}_{m1+1}P^{H}_{n+1} \longrightarrow$
$P_1^{m1}\ (P^{m}_{m1+1}\ P^{n}_{m+1})P^{H}_{n+1} \longrightarrow$
$P_1^{m1}\ (P^{n}_{m1+1})P^{H}_{n+1} \longrightarrow$

or (right) : $G_i = P^{m}_{1}P^{n}_{m+1}(P^{H}_{n+1})$
$P^{m}_{1}P^{n}_{m+1}(P^{T1}_{n+1}\ P^{H}_{T1+1}) \longrightarrow$
$P^{m}_{1}(P^{n}_{m+1}P^{T1}_{n+1})P^{H}_{T1+1}) \longrightarrow$
$P^{m}_{1}(P^{T1}_{m+1})P^{H}_{T1+1}) \longrightarrow$

Here, P is the opposite bit pattern of P and $P_m^{\ n}$ indicates the pattern which consists of bit-position m to n. For example, a schedule 1111110011111 is represented as $P^{6}_{1}P^{8}_{7}P^{13}_{9}$. After bit wise operations, length of patterns P^{II}_{1}, (P^{n}_{m1+1}) or (P^{T1}_{m+1}) must be greater than or equal to minimum up/down time constraint of unit i.

Note: This is an initial article of the local university.

Ahmed Yousuf Saber was born at Rajshahi, Bangladesh in 1974. He received the B.Sc. and M.Sc. degrees in computer science and engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 1999 and 2002, respectively. Since 1999, he has been with Department of Computer Science and Engineering, The University of Asia Pacific, where he is an Assistant Professor. Currently, he is pursuing the Ph.D. degree in University of the Ryukyus, Okinawa, Japan. His research interests include continuous and discrete optimization techniques, power systems software development, computer interface and application, and AI applications. He is a Student Member of the IEEE and IEEJ, and Member of The Institution of Engineers, Bangladesh.

A. K. M. Zaidi Satter received his B.Sc. in Electronics & Computer Science from Jahangirnagar University, Savar, Dhaka, Bangladesh in 1998 and pursuing his M. Sc. in Computer Science & Engineering at the same University. Now he is working as a faculty member in the department of CSE/CIS of Daffodil International University.

Wahed Newaz was born in 1978. He received the B.Sc. degrees in electrical engineering from Rajshahi University of Engineering and Technology (RUET), Bangladesh in 2002. He has been with Rural Electrification Board (REB) where he is an Asst. General Manager. His research interest includes power systems.