 ISSN: 2321-8169
1417 – 1422

Solution to Profit Based Unit Commitment Problem Using Bacterial Foraging Algorithm

P.V. Rama Krishna,
Gitam University,
Hyderabad, India
E-mail: pvrk123@gmail.com

Dr Sukhdeo Sao
Bharat Institute of Engineering and Technology,
Hyderabad, India
E-mail: drssao53@gmail.com

***Abstract*—** the profit based unit commitment problem involves determining the time intervals at which a particular generating is on considering all the unit constraints including minimum up and down time, ramp up and down time and spinning reserve and all other constraints with main objective of maximization of profit for a power producer. The profit based unit commitment problem has been solved in this paper using bacterial foraging algorithm and the results are verified for the standard IEEE 10 generating units for different time intervals.

***Keywords*—** *profit based unit commitment problem, bacterial foraging algorithm, and constraints*.

*****

I. Profit based unit commitment problem

The unit commitment problem under deregulation involves commitment of generating units of an individual power producer for maximization of his profit; this problem is of highly complex in nature as so many constraints are involved for maximization of profit as main objective function. In this paper the problem is attacked using advanced binary genetic algorithm. The proposed algorithm seems to be efficient as compared to normal conventional technique like dynamic programming in terms of constraints satisfaction and convergence time. The bacterial foraging algorithm is found to be simpler in constraints handling and meeting up the critical constraints like ramping up and ramping down and power generation limits.

II. Problem Formulation

The main objective function of the problem is

Profit =Revenue-Total cost (1)

The revenue is obtained by supplying a certain amount of power at spot market price and the total cost is the cost of production as well as starts up and shut down costs included if any. The startup cost can be considered by taking in to account of the number of hours the unit has been off line and unit cooling time as well. This can be interpreted in the following equation

$$SU_i^t = \alpha_i + \beta_i(1-\exp(-X_{off,i}^t / \tau_i)) \quad (2)$$

Where SU_i^t the startup cost of unit i at the interval of time t.

α_i : Combined crew start-up costs and equipment maintenance costs [$];

β_i : Cold start-up cost [$];

$X_{off,i}^t$: Number of hours the unit has been offline [h];

τ_i : Unit-cooling time constant [h].

In addition to startup cost the generating unit must satisfy all the constraints (minimum up time, minimum down time, ramp up and ramp down, minimum power and maximum power generation) as given below.

Loading constraint:

$$P_{load}^t - \sum_{i=1}^{N} P_i^t * U_i^t = 0 \ \forall t = 1 \ldots\ldots T \quad (3)$$

Where P_i^t is the power generation of i[th] unit at hour t and U_i^t is the state of i[th] unit at hour t

Unit limits:

$$U_i^t * P_i^{min} \le P_i^t \le U_i^t * P_i^{min}$$
$$\forall\, i = 1 \ldots . N \text{ and } t = 1 \ldots \ldots T \quad (4)$$

Unit minimum up and down time constraints:

$$\left.\begin{aligned}(u_i(t) - u_i(t-1)(T_{on,t-1} - MUT)) \le 0 \\ (u_i(t) - u_i(t-1)(T_{off,t-1} - MDT)) \ge 0\end{aligned}\right\} \quad (5)$$

MUT = Minimum up time, MDT=Minimum down time, Ton = Generator on time, Toff = Generator off time

Unit ramp rate limits

$$\left.\begin{aligned}\max\,(P_i^{min}, (P_{i,t-1} - DR_i)) \le P_{i,t} \\ \min\,(P_i^{max}, (P_{i,t-1} + UR_i)) \ge P_{i,t}\end{aligned}\right\} \quad (6)$$

DR = Ramp down limit
UR =Ramp up limit

In addition to all the above constraints there are some other constraints like spinning reserve constraints and crew constraints and must run constraints that must be satisfied. From the above equations there are two decision variables P_i^t and U_i^t where P_i^t denotes the amount of power to be generated

by unit i at time t and U_i^t is the control variable whose value is chosen to be "1" if the generating unit i is committed at hour t and "0" otherwise (of course if U_i^t =0, then P_i^t =0).

III. Bacterial Foraging Algorithm

Bacteria Foraging Optimization Algorithm (BFOA), proposed by Kevin M. Passino, is a new comer to the family of nature-inspired optimization algorithms. For over the last five decades, optimization algorithms like Genetic Algorithms (GA), Evolutionary Programming (EP), Evolutionary Strategies (ES), which draw their inspiration from evolution and natural genetics, have been dominating the realm of optimization algorithms. Recently natural swarm inspired algorithms like Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) have found their way into this domain and proved their effectiveness. Following the same trend of swarm-based algorithms, Passino proposed the BFOA. Application of group foraging strategy of a swarm of E.coli bacteria in multi-optimal function optimization is the key idea of the new algorithm. Bacteria search for nutrients in a manner to maximize energy obtained per unit time. Individual bacterium also communicates with others by sending signals. A bacterium takes foraging decisions after considering two previous factors. The process, in which a bacterium moves by taking small steps while searching for nutrients, is called chemo taxis and key idea of BFOA is mimicking chemotactic movement of virtual bacteria in the problem search space.
Since its inception, BFOA has drawn the attention of researchers from diverse fields of knowledge especially due to its biological motivation and graceful structure. Researchers are trying to hybridize BFOA with different other algorithms in order to explore its local and global search properties separately. It has already been applied to many real world problems and proved its effectiveness over many variants of GA and PSO. Mathematical modeling, adaptation, and modification of the algorithm might be a major part of the research on BFOA in future

The foraging strategy is governed basically by four processes namely Chemotaxis, Swarming, Reproduction, Elimination and Dispersal.

A) CHEMOTAXIS:

Some bacteria, such as *E. coli*, have several flagella per cell (4–10 typically). These can rotate in two ways:

1. Counter-clockwise rotation aligns the flagella into a single rotating bundle, causing the bacterium to swim in a straight line.
2. Clockwise rotation breaks the flagella bundle apart such that each flagellum points in a different direction, causing the bacterium to tumble in place.

The directions of rotation are given for an observer outside the cell looking down the flagella toward the cell.

1) Behavior:-

The overall movement of a bacterium is the result of alternating tumble and swim phases. If one watches a bacterium swimming in a uniform environment, its movement will look like a random walk with relatively straight swims interrupted by random tumbles that reorient the bacterium. Bacteria such as *E. coli* are unable to choose the direction in which they swim, and are unable to swim in a straight line for more than a few seconds due to rotational diffusion. In other words, bacteria "forget" the direction in which they are going. By repeatedly evaluating their course, and adjusting if they are moving in the wrong direction, bacteria can direct their motion to find favorable locations with high concentrations of attractants (usually food) and avoid repellents (usually poisons).

In the presence of chemical gradient bacteria will chemotax, or direct their overall motion based on the gradient. If the bacterium senses that it is moving in the correct direction (toward attractant/away from repellent), it will keep swimming in a straight line for a longer time before tumbling. If it is moving in the wrong direction, it will tumble sooner and try a new direction at random. In other words, bacteria like *E. coli* use temporal sensing to decide whether their situation is improving or not. In this way, it finds the location with the highest concentration of attractant (usually the source) quite well. Even under very high concentrations, it can still distinguish very small differences in concentration. Fleeing from a repellent works with the same efficiency.

This biased random walk is a result of simply choosing between two methods of random movement; namely tumbling and straight swimming. In fact, chemotactic responses such as *forgetting* direction and *choosing* movements resemble the decision-making abilities of higher life-forms with brains that process sensory data.

The helical nature of the individual flagella filament is critical for this movement to occur. As such, the protein that makes up the flagella filament, flagellin is quite similar among all flagellated bacteria. Vertebrates seem to have taken advantage of this fact by possessing an immune receptor designed to recognize this conserved protein.

As in many instances in biology, there are bacteria that do not follow this rule. Many bacteria, such as *Vibrio*, are mono flagellated and have a single flagellum at one pole of the cell. Their method of chemo taxis is different. Others possess a single flagellum that is kept inside the cell wall. These bacteria move by spinning the whole cell, which is shaped like a corkscrew

B) SWARMING:

An interesting group behavior has been observed for several motile species of bacteria including E.coli and S. typhimurium, where intricate and stable spatio-temporal patterns (swarms) are formed in semisolid nutrient medium. A group of E.coli cells arrange themselves in a traveling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemo-effecter. The cells when stimulated by a high level of succinate, release an attractant aspertate, which helps them to aggregate into groups and thus move as concentric patterns of swarms with high bacterial density. Mathematically Swarming behavior can be modeled as:

 ISSN: 2321-8169
1417 – 1422

$$J_{cc}(\theta, p(j,k,l)) = \sum_{i=1}^{s} J_{cc}(\theta, \theta^i(j,k,l))$$
$$= \sum_{i=1}^{s}\left[-d_{attract}\exp\left(-w_{attract}\sum_{i=1}^{n}(\theta_m - \theta_m^i)^2\right)\right] +$$
$$\sum_{i=1}^{s}\left[h_{repellant}\exp\left(-w_{repellant}\sum_{i=1}^{n}(\theta_m - \theta_m^i)^2\right)\right]$$

where J_{cc} $(\theta,P(j,k,l))$ is the objective function value to be added to the actual objective function (to be minimized) to present a time varying objective function, S is the total number of bacteria, p is the number of variables to be optimized, which are present in each bacterium and $\theta = [\theta_1,\theta_2,...,\theta_p]^T$ is a point in the *p*-dimensional search domain. $d_{attractant}$,$w_{attractant}$, $h_{repellant}$,$w_{repellant}$ are different coefficients that should be chosen properly

C) REPRODUCTION:

In this step, population members who have had sufficient nutrients will reproduce and the least healthy bacteria will die. The healthier half of the population replaces with the other half of bacteria which gets eliminated, owing to their poorer foraging abilities. This makes the population of bacteria constant in the evolution process.

D) ELIMINATION AND DISPERSAL:

Gradual or sudden changes in the local environment where a bacterium population lives may occur due to various reasons e.g. a significant local rise of temperature may kill a group of bacteria that are currently in a region with a high concentration of nutrient gradients. Events can take place in such a fashion that all the bacteria in a region are killed or a group is dispersed into a new location. To simulate this phenomenon in BFOA some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space. The pseudo-code as well as flow-chart of the complete algorithm is presented below:

The BFOA Algorithm

Parameters:

[Step 1] Initialize parameters *p, S, Nc, Ns, Nre, Ned, Ped, C(i) (i=1,2...S),i,θ* .

Algorithm:

[Step 2] Elimination-dispersal loop: l=l+1

[Step 3] Reproduction loop: k=k+1

[Step 4] Chemotaxis loop: j=j+1

[a] For i =1,2…S take a chemotactic step for bacterium i as follows.

[b] Compute fitness function, *J (i, j, k, l).* Let, $J(i,j,k,l) = J(i,j,k,l) + J_{cc}(\theta^i(j,k,l), P(j,k,l))$ (i.e. add on the cell-to cell attractant–repellant profile to simulate the swarming behavior)

[c] Let J_{last}=*J (i, j, k, l)* to save this value since we may find a better cost via a run.

[d] Tumble: generate a random vector $\Delta(i) \in \Re^p$ with each element Δ_m *(i), m= 1,2,..., p,* a random number on [-1, 1].

[e] Move: Let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

This results in a step of size C (i) in the direction of the tumble for bacterium i.

[f] Compute *J (i,j+1,k,l)* and let

$$J(i,j+1,k,l) = J(i,j,k,l) + J_{cc}(\theta^i(j+1,k,l), P(j+1,k,l))$$

[g] Swim

i) Let *m=0* (counter for swim length).

ii) While $m < N_s$ (if have not climbed down too long).

• Let *m=m+1.*

• If $J(I, j+1, k,l) < J_{last}$ (if doing better), let $J_{last} = J(I, j+1, k,l)$ and let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

• Else, let $m = N_s$. This is the end of the while statement.

[h] Go to next bacterium *(i+1)* if $i \neq S$ (i.e., go to [b] to process the next bacterium).

[Step 5] If $j<N_c$, go to step 4. In this case continue chemo taxis since the life of the bacteria is not over.

[Step 6] Reproduction:

[a] For the given k and l, and for each i = 2,1 ,..., S , let

$$J_{health}^i = \sum_{j=l}^{N_c+1} J(i,j,k,l)$$

Be the health of the bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters C (i) in order of ascending cost health J (higher cost means lower health).

[b] The S_r bacteria with the highest health J values die and the remaining S_r bacteria with the best values split (this process is performed by the copies that are made are placed at the same location as their parent).

[Step 7] If $k<N_{re}$, go to step 3. In this case, we have not reached the number of specified reproduction steps, so we start the next generation of the chemotactic loop.

[Step 8] Elimination-dispersal: For i = 2, 1 ..., S with probability Ped , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain. If $l < N_{ed}$, then go to step 2; otherwise end.

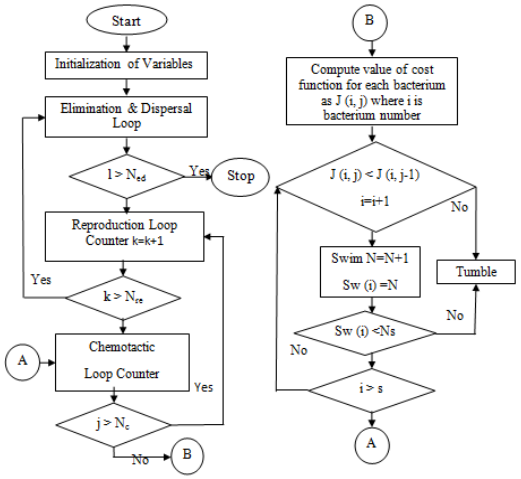

Figure 1

ISSN: 2321-8169
1417 – 1422

Table-1 Unit data (IEEE REFERENCE DATA)

UNIT 1	UNIT 2	UNIT 3	UNIT 4	UNIT 5	UNIT 6
pmin=15.20	pmin=15.20	pmin=15.20	pmin=25.00	pmin=25.00	pmin=25.00
pmax=76.00	pmax=76.00	pmax=76.00	pmax=100.00	pmax=100.00	pmax=100.00
nl1=76.473	nl1=76.558	nl1=76.602	nl1=210.108	nl1=210.685	nl1=211.300
tup=3	tup=3	tup=3	tup=4	tup=4	tup=4
tdown=2	tdown=2	tdown=2	tdown=2	tdown=	tdown=2
x0=-3	x0=-3	x0=-3	x0=-3	x0=-3	x0=-3
alpha=	alpha=50	alpha=50	alpha=70	alpha=70	alpha=70
beta=50	beta=50	beta=50	beta=70	beta=70	beta=70
tao=3	tao=3	tao=3	tao=4	tao=4	tao=4
rup=15	rup=15	rup=20	rup=25	rup=30	rup=30
rdown=15	rdown=20	rdown=20	rdown=25	rdown=30	rdown=30
y0=0	y0=0	y0=0	y0=0	y0=0	y0=0
a=0.00895	a=0.00910	a=0.00932	a=0.00623	a=0.00612	a=0.00598
b=13.3538	b=13.3805	b=13.4073	b=18.0000	b=18.1000	b=18.2000
c=81.2980	c=81.4641	c=81.6259	c=217.8952	c=218.3350	c=218.7752

UNIT 7	UNIT 8	UNIT 9	UNIT 10	Time	Spot market price
pmin=54.25	pmin=54.25	pmin=54.25	pmin=54.25	1	9.00
pmax=155.00	pmax=155.00	pmax=155.00	pmax=155.00	2	9.60
nl1=120.673	nl1=120.491	nl1=120.399	nl1=120.392	3	14.33
tup=5	tup=5	tup=5	tup=5	4	25.49
tdown=3	tdown=3	tdown=3	tdown=3	5	31.80
x0=-5	x0=-5	x0=-5	x0=-5	6	31.00
alpha=150	alpha=150	alpha=150	alpha=150	7	36.28
beta=150	beta=150	beta=150	beta=150	8	42.40
tao=6	tao=6	tao=6	tao=6	9	52.22
rup=100	rup=150	rup=150	rup=150	10	52.20

rdown=100	rdown=150	rdown=150	rdown=150
y0=0	y0=0	y0=0	y0=0
a=0.00463	a=0.00473	a=0.00481	a=0.00487
b=10.6940	b=10.7154	b=10.7367	b=10.7583
c=142.7348	c=143.0288	c=143.3179	c=143.5972

Population size=2000;
Chromosome length=10
Pc=0.6(cross over probability)
Pm=0.6(mutation probability)
dai =200 (maximum number of iterations)

All the parameters are defined in addition to time intervals and corresponding spot market prices are defined when reading unit data.

IV. RESULTS

Table-2 UNIT COMMITMENT SCHEDULE FOR 10 HOURS

Unit	t=1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	0	1	1	1	1	1	1	1	1	1
4	0	0	1	1	1	1	1	1	1	1
5	0	0	1	1	1	1	1	1	1	1
6	0	0	1	1	1	1	1	1	1	1
7	0	0	1	1	1	1	1	1	1	1
8	0	0	1	1	1	1	1	1	1	1
9	0	0	1	1	1	1	1	1	1	1
10	0	0	1	1	1	1	1	1	1	1

Table-3 POWER GENERATION SCHEDULE FOR 10 HOURS

Unit	T=1	2	3	4	5	6
1	15.2	30.2	45.2	60.2	75.2	76.0
2	15.2	30.2	45.2	60.2	75.2	76.0
3	0.0	20.0	40.0	60.0	76.0	76.0
4	0.0	0.0	25.0	50.0	75.0	100.0
5	0.0	0.0	30.0	60.0	90.0	100.0
6	0.0	0.0	30.0	60.0	90.0	100.0
7	0.0	0.0	100.0	155.0	155.0	155.0
8	0.0	0.0	150.0	155.0	155.0	155.0
9	0.0	0.0	150.0	155.0	155.0	155.0
10	0.0	0.0	150.0	155.0	155.0	155.0

7	8	9	10
76.0	76.0	76.0	76.0
76.0	76.0	76.0	76.0
76.0	76.0	76.0	76.0
100.0	100.0	76.0	76.0
100.0	100.0	100.0	100.0
100.0	100.0	100.0	100.0

 ISSN: 2321-8169
1417 – 1422

155.0	155.0	155.0	155.0
155.0	155.0	155.0	155.0
155.0	155.0	155.0	155.0
155.0	155.0	155.0	155.0

Table-4 PROFIT SCHEDULE FOR 10 HOURS

Time	Unit-1	Unit-2	Unit-3	Unit-4	Unit-5
1	-231.149	-231.756	0.000	0.000	0.000
2	-203.532	-204.653	-249.012	0.000	0.000
3	-56.376	-58.071	-60.295	-433.484	-457.500
4	616.235	613.907	609.147	148.817	206.705
5	1255.104	1252.080	1262.387	782.061	964.175
6	1208.118	1205.057	1201.587	1019.805	1010.465
7	1609.398	1606.337	1602.867	1547.805	1538.465
8	2074.518	2071.457	2067.987	2159.805	2150.465
9	2820.838	2817.777	2814.307	3141.805	3132.465
10	2819.318	2816.257	2812.787	3139.805	3130.465

Unit-6	Unit-7	Unit-8	Unit-9	Unit-10
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
-460.800	-73.271	38.770	33.475	28.597
200.685	2039.409	2033.395	2027.883	2022.814
955.890	3017.459	3011.445	3005.933	3000.864
1001.425	2893.459	2887.445	2881.933	2876.864
1529.425	3711.859	3705.845	3700.333	3695.264
2141.425	4660.459	4654.445	4648.933	4643.864
3123.425	6182.559	6176.545	6171.033	6165.964
3121.425	6179.459	6173.445	6167.933	6162.864

V. CONCLUSIONS

It is recognized that the optimal unit commitment of thermal systems results in a great saving for electric utilities. Unit Commitment is the problem of determining the schedule of generating units subject to device and operating constraints. The formulation of profit based unit commitment has been discussed and the solution is obtained by genetic algorithm approach. An algorithm based on genetic algorithm, which is fitness based optimization technique, has been developed to solve the profit based unit commitment problem. The effectiveness of these algorithms has been tested on system comprising 10 units.

VI.REFERENCES

[1] R. H. Kerr, J. L. Scheldt, A. J. Fontana and J. K. Wiley, “Unit Commitment”, IEEE Transactions on Power Apparatus And Systems.vol. PAS-85, No. 5, pp. 417-421. May 1966.

[2] K. Hara, M. Kimura and N. Honda, “A Method for Planning Economic Unit Commitment and Maintenance of Thermal Power Systems”, IEEE Transaction on Power Apparatus And Systems, vol. PAS-85, NO.5, pp.421-436. May 1966

[3] P. G. Lowery, “Generating Unit Commitment by Dynamic Programming”, IEEE Transaction on Power Apparatus and Systems, vol. PAS-85, No. 5, pp. 422-426. May 966.

[4] J. D. Guy. “Security Constrained Unit Commitment”. IEEE Transactions on Power Apparatus and Systems, vol. 90, pp. 1385-1389, May/June 1971.

[5] K. Fang, G. B. Sheble and F. Albu, “Evaluation Of Dynamic Programming Based Methods And Multiple Area Representation For Thermal Unit Commitments”, IEEE Transactions on Power Apparatus and Systems. vol. 100, No. 3, pp. 1212-1218, Mar. 1981.

[6] G. S. Lauer, N. R. Sandell, Jr., Bertsekas and T.A. Posbergh, “Solution Of Large Scale Optimal Unit Commitment Problem”. IEEE Transactions on Power Apparatus and Systems, vol. 101, pp. 79-96, Jan. 1982.

[7] K.D. Lee, R.H. Vierra, G.D. Nagel and R.T. Jenkins, “Problems Associated With Unit Commitment In Uncertainty”, IEEE Transactions on Power Apparatus and Systems, vol. 104, No. 8, pp.2072-2078, Aug.1985.

[8] P.PJ van den Bosch and G. Honderd, “A Solution Of The Unit Commitment Problem Via Decomposition And Dynamic Programming”, IEEE Transactions on Power Apparatus and Systems, vol. 104, pp. 1684-1690, July 1985.

[9]

[10] G. L. Kusic and H. A. Putnam, “Dispatch And Unit Commitment Including Commonly Owned Units”, IEEE Transactions on Power Apparatus and Systems, vol. 04. pp. 2408-2412. Sep. 1985.

[11] Walter L. Snyder. Jr., H. David Powell, Jr. and John C. Rayburn, “Dynamic Programming Approach to Unit Commitment”, IEEE Transaction on Power System. vol. 2, pp. 339-347. May 1987.

[12] R. Nieva, A. lnda and I. Guillen, ”Lagrangian Reduction Of Search-Range For Large-Scale Unit Commitment”, IEEE Transaction on Power System, vol. 2, pp. 465- 473. May 1987.

[13] A .I. Cohen and S.H. Wan,“ A Method For Solving The Fuel Constrained Unit Commitment Problem”, IEEE Transaction on Power System, vol.2, pp. 608-614, Aug. 1987.

ISSN: 2321-8169
1417 – 1422

[14] S. Mukhtari, J. Singh and B. Wollenberg, "A Unit Commitment Expert System", IEEE Transaction on Power System, vol. 3, No. I, pp. 272-277. 1988.

[15] Fred. N. Lee, "Short-Term Thermal Unit Commitment - A New Method", IEEE Transaction on Power System, vol. 3, pp. 421-428, May 1988.

[16] Walter J. Hobbs, Gary Hermon, Stephen Warner and Gerald B. Sheble, "An Enhanced Dynamic Programming Approach for Unit Commitment", IEEE Power Engineering Review, pp.70. Aug. 1988.

[17] K. Aoki, M. Itoh, T. Satoh, K. Nara and M. Kanezashi, "Optimal Long-Term Unit Commitment In Large Scale Systems Including Fuel Constrained Thermal And Pumped Storage Hydro", IEEE Transaction on Power System, vol. 4, pp. 1065-1073, Aug 1989.

[18] S. K. Tong and Shahidehpour, "Combination Of Lagrangian-Relaxation And Linear- Programming Approaches for Fuel-Constrained Unit Commitment Problems". IEE Proc- Gener. Tansnm. Distrib., vol. 136, Pt. C. pp.162-174, May 1989.