International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization on TPE (IOTPE)

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

June 2010	Issue 3	Volume 2	Number 2	Pages 22-27

OPTIMAL UNIT COMMITMENT OF POWER SYSTEM USING FAST MESSY GENETIC ALGORITHM

A. Safari [1] H.A. Shayanfar [1] R. Jahani [2]

1 Center of Excellence for Power Automation and Operation, Electrical Engineering Department,
Iran University of Science and Technology, Tehran, Iran, asafari1650@yahoo.com , hashayanfar@yahoo.com
2 Electrical Engineering Department, Islamic Azad University, South Tehran Branch, Tehran, Iran
rouzbeh_jahani@yahoo.com

Abstract- This paper presents a new approach via improved version of genetic algorithm to solve optimal unit commitment (UC) problem. Fast Messy Genetic Algorithm (FMGA) is applied to the calculation of optimal unit commitment problem. The test results demonstrate that not only the FMGA procedure consider is the constraints very well, but also has some advantages, such as good convergence, fast calculating speed and high precision. A ten unit power system was used as a numerical example to test the new algorithm. The optimal scheduling of on line generation units could be reached in the testing results while satisfying the constraints of the objective function. Numerical results indicate that the performance of the FMGA algorithm outperforms the other algorithms and achieves minimum generation cost.

Keywords: Fast Messy Genetic Algorithm, Power System, Unit Commitment, System Constraints.

I. INTRODUCTION

The objective of the economic scheduling of generators is to guarantee the optimum combination of generators connected to the system to provide the load demand. The economic dispatch problem involves two separate steps namely unit commitment and on-line economic load dispatch. The unit commitment involves the selection of units that will supply the anticipated load of the system at minimum cost over a required period of time as well as providing a specified margin of the operating reserve, known as the spinning reserve.

The on-line economic dispatch distributes the load among those operating units that are paralleled with the system in such a manner so as to minimize the total cost of supplying the minute to minute requirements of the system. It is quite expensive to run too many generating units. A great deal of money can be saved by turning units off (decommiting them) when they are not needed. The generic UC can be formulated as to minimize the operational cost subject to minimum up-time and down-time constraints, crew constraints, ramp constraints, unit capability limits, duration of units, unit status, generation constraints and reserve constraints [1].

The exact solution of the UC can be obtained by complete enumeration, namely, dynamic or integer programming. The drawbacks of these methods are the enormous computational time which increases exponentially with the number of units, and the large memory requirement. Modifying the above-mentioned conventional techniques has helped in improving the accuracy of the solution with respect to the well-timed decision-making [2].

The UC algorithms can be applied to large-scale power systems and have reasonable storage and computation time requirements. A survey of literature on UC methods reveals that various numerical optimization techniques have been employed to address the UC problems. Specifically, there are priority list methods [3, 4], integer programming [5, 6], dynamic programming [7, 8], mixed-integer programming [9], branch-and-bound methods [10], and Lagrangian relaxation methods [11, 12]. Among these methods, the priority list method is simple and fast, but the quality of the final solution is not guaranteed.

Dynamic programming methods, which are based on priority lists, are flexible but are computationally expensive. Branch-and-bound adopts a linear function to represent the fuel consumption and time-dependent start cost and obtains the required lower and upper bounds. The disadvantage of the branch-and-bound method is the exponential growth in the execution time with the size of the UC problem. The integer and mixed-integer methods adopt linear programming technique to solve and check for an integer solution. These methods have only been applied to small UC problems and have required major assumptions that limit the solution space. The Lagrangian relaxation method provides a fast solution, but it may suffer from numerical convergence and solution quality problems. Aside from the above methods, there is another class of numerical techniques applied to the UC problem. Specifically, they are artificial neural networks [13, 14], Simulated Annealing (SA) [15], and Genetic Algorithms (GAs) [16]. These methods can accommodate more complicated constraints and are claimed to have better solution quality.

The SA is a powerful, general purpose stochastic optimization technique, which can theoretically converge asymptotically to a global optimum solution with probability 1. One main drawback, however, of SA is that it takes a large CPU time to find the near-global minimum. GAs are a general-purpose stochastic and parallel search methods based on the mechanics of natural selection and natural genetics. It is a search method and has the potential of obtaining a near-global minimum.

This paper describes the application of the fast messy genetic algorithm for the solution of the unit commitment problem. Solution methodology of the UC is described as gradual consideration constraints and objectives in the total problem formulation.

II. UC PROBLEM FORMULATION

A. Objective Function

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

$$\min F(U_{it}, P_{it}) = \sum_{t=1}^{24}\sum_{i=1}^{G}[U_{it}F_{it}(P_{it}) + U_{it}(1-U_{it-1})S_i] \tag{1}$$

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

$$F_i(P_{it}) = a_i P_{it}^2 + b_i P_{it} + c_i \tag{2}$$

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

$$S_i = S_{0i} + S_{1i}(1 - e^{-T/\tau_i}) \tag{3}$$

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

B. System Constraints

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

1. Power balance constraint

$$\sum_{i=1}^{G} U_{it}P_{it} = P_{Dt}, \quad t = 1,2,3,... \tag{4}$$

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

$$\frac{dF_{1t}}{dP_{1t}} = \frac{dF_{2t}}{dP_{2t}} = ... = \frac{dF_{it}}{dP_{it}} = \lambda \tag{5}$$

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

2. Spinning Reserve

If spinning reserve needs to be more than 7% of the total load at each time interval, it must met:

$$\sum_{i=1}^{G} U_{it}P_{i\max} \ge 1.07 P_{Dt}, \quad t = 1,2,3,...24 \tag{6}$$

3. Unit Generation Output Limitation

$$P_{i\min} \le P_i \le P_{i\max} \quad t = 1,2,...,24, \quad i = 1,2,...,G \tag{7}$$

4. Start Up- and Down Times Limitation

$$\sum_{t=1}^{24} |U_{it} - U_{it-1}| \le M_i \tag{8}$$

5. Minimum Up and Down-Time Constraints

$$TO_i \ge \underline{TO_i} \tag{9}$$

$$TS_i \ge \underline{TS_i} \tag{10}$$

III. REVIEW OF MGA AND FMGA

A. Messy Genetic Algorithm

Messy Genetic Algorithm differ from normal genetic algorithms in that they allow variable-length strings that may be underspecified or over-specified with respect to the problem being solved. For matching geometric models, this means they can operate over partial matches and so piece together larger and better matches. A Messy GA typically has three phases:

1. Initialization;
2. Primordial phase;
3. Juxtapositional phase.

In the initialization phase some procedure is used to enumerate a set of partial chromosomes. Next, in the primoridial phase, these partial chromosomes are evaluated using a fitness function and some subset of these found to be most fit are used to form the initial population. Finally, the juxtapositional phase is analogous to the normal cycle of selection and recombination used in a traditional GA. For initialization, our Messy GA uses the same set of spatially proximate triples *F* defined above for the Key-Feature Algorithm. This customized initialization phase creates 2n triples to seed the initial population. By the match error rematch and some fraction of the best form the initial population. In the experiments presented here, all the 2n triples are used. During juxtaposition, selection is used together with two operators: cut and splice. Cut 'cuts' the chromosome at random position. Splice 'attaches' two cut chromosomes together. These two operators are the equivalents of crossover in a traditional GA. In our matching problem, a chromosome h is a variable length set of pairs *h, c, S*, representing a match between model and data segments. Both the cut and splice operators pick the positions in the set representations where cutting and splicing take place with uniform probability.

Hence, for example, if a set h contains 6 pairs of model and data segments, then cut will select one of the five possible break points with equal probability. To select parents, the Messy GA uses a variant of the Genitor algorithm which uses fitness ranking to bias selection and a monotonic replacement strategy. Selection based upon rank means that two parents are selected for recombination based upon their ranking in the population rather than upon the absolute difference in fitness between themselves and the others in the population. To make this process more efficient, the population is always maintained in sorted order from lowest to highest E_{match}. The current parameterization of the algorithm makes the most fit individual twice as likely to be selected as a parent as compared to the least fit individual.

Monotonic selection means that each time two parents are selected and used to create a child, that child is inserted back into the original ranked ordered population based upon its newly computed E_{match}. Afterward, the least fit individual is removed from the population. This means that if a child is inferior to all other individuals in the population, then that child is effectively discarded. This also means that the overall quality of the population will increase monotonically, since a worse individual can never displace one which is more fit.

To help drive the Messy Genetic Algorithm to a solution, every three generations the least fit individual in the population is dropped and the population size correspondingly shrinks by one. Every $f=p/2$ generations, an individual is selected from the population and local search is run using the selected match as an initial state. If the result is better than the worst currently in the population, then it is inserted back into the population. This periodic use of local search as part of the genetic search is of great practical benefit, and is consistent with other results suggesting hybridized algorithms of this type often out perform pure Genetic or Local Search. Like Random Starts Local Search, the Messy Genetic Algorithm is a non-deterministic search algorithm. Consequently, the same technique of running multiple independent trials in order to increase the probability of seeing the best match applies. Further, the probability $\hat{p}_s$ that the Messy Genetic Algorithm will succeed on any given trial can again be estimated [17].

B. Fast Messy Genetic Algorithm

On/Off statue can be easily represented by binary coding: 1 is on statue and 0 is off statue. If the scheduling period is divided into 24 time-steps and there are total G units. Then each unit has 24 bits (Figure1). i.e. 2nd bit of unit 1 represents the on/off statue of unit 1 at 2nd time-step. One binary coding individual can be combined according to the order of units and each individual has total $G \times 24$ bits. Per bit of each individual in one population is produced randomly.

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

$$\min F + \sum_{j=1}^{n_C} u_j R_j \tag{11}$$

where, F is original objective function; n_c is the number of violation constraints; R_j and u_j are the violation value and penalty coefficient of jth constraint , respectively. Equation (11) only includes spinning reserve, start up and down times, minimum up- and down-time constraints.

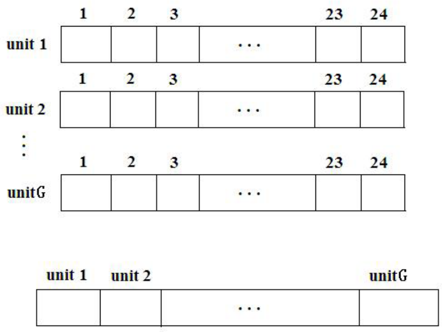

Figure 1. The binary representation of unit commitment

The power balance and unit generation output limitation
is considered in the load dispatch. The fitness function is:

$$TF = \frac{K}{F + \sum_{j=1}^{n_C} u_j R_j} \tag{12}$$

Constant K is proportional coefficient. The value of K and u_j should be selected according to the specific problem. The values should let the fitness value of feasible solution be around 1 to prevent computer treating too large or small value. The FMGA is a special clone of common simple genetic algorithm. This type represents new, more powerful kind in the GA branch. It resists premature local-minimum fall and solves problems in shorter time.

As the initial various individual is produced randomly and hence exist many infeasible solutions that do not satisfy the constraints. Some papers let the fitness value of these infeasible solutions be zeros. This will often result in a few feasible individual fitness values much higher then others, the search range decrease and premature convergence.

A gene is represented by a pair (allele locus, allele value) in messy algorithms, so that for instance chromosome (2,0), (0,1), (1,1) represents 3 bit-long chromosome 110 .Operation cut and splice are used to bread new offspring. Incomplete specification of the chromosome or redundant specification could occur during evolution. For example, ((0,0), (2,1), (1,0), (2,0)) represents chromosome 001. We must complete the chromosome in order to be able to evaluate its fitness function (FF). Hence a template is used as Figure 2.

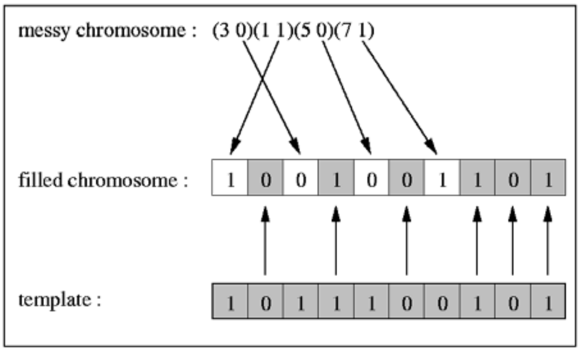

Figure 2. Representation of a solution chromosome

Table 1. Optimal unit commitment result

Unit Number	On/Off Statue of Per Time-Step																						
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
6	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

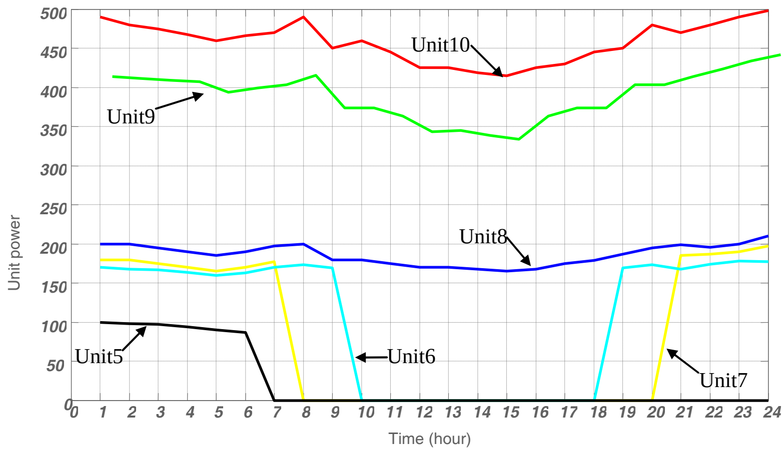

Figure 3. Unit generation output power curves

IV. OPTIMIZATION BY FAST MESSY GENETIC ALGORITHM

The proposed optimization problem can be summarized as:

Step 1) Initialize the parameters of the unit commitment problem.

Step 2) Set the chromosomes as the 24-hours bits units.

Step 3) Input the number of units.

Step 4) Regarding to the problem constraints, set the cost function which is to be minimized (Equation (1)).

Step 5) Initialize the messy Genetic Algorithm and determine an appropriate template.

Step 6) Enter the primordial phase of MGA:

a) Produce the initial population.

Step 7) Enter the juxtapositional phase:

b) Apply the operations;

c) Evaluate the chromosomes.

Step 8) Output the statistics.

V. SIMULATION RESULTS

This paper developed the Fast Messy genetic algorithm program using MATLAB. The 10 generation units system [24] is tested to verify the correctness, and the results are compared with other algorithms.

Tables 1 and 2, and Figure 3 show the simulation results of the proposed method. Comparing with other method results, the obtained result in this paper represents a nearer global optimal solution to the problem and verifies the correctness of the proposed algorithm.

Table 2. Comparison results of various algorithms

Algorithms	Total Cost
LR [18]	80766.0
Hopfield-SA [19]	79114.6
Evolutionary Method [20]	79043
AC-PSO[21]	79010.1
HPSO [22]	81118.3
GA[23]	78988.8
GA [24]	79807.0
FMGA	**77069.8**

VI. CONCLUSIONS

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

NOMENCLATURES

i : Index of units
t : Index of time-steps
G : Number of generating units
U_{it} : On/Off status of unit i at time-step t
P_{it} : Generation output of unit i at time-step t
$F_i(P_{it})$: Running cost of the unit i at time-step t
a_i, b_i, c_i : Running cost coefficients of unit i
S_i : Start-up cost of unit i
S_{0i}, S_{1i}, τ_i : Start-up cost coefficients of unit i
P_{Dt} : System load demand at time-step t
λ : Equal loss incremental rate
P_{Rt} : Spinning reserve
$P_{i\max}$: Unit i maximum generation output limit
$P_{i\min}$: Unit i minimum generation output limit
M_i : Start up and down times limitation
$\underline{TO_i}$: Minimum up time of unit ith
$\underline{TS_i}$: Minimum down time of unit ith
TO_i : Duration during which unit is continuously on
TS_i : Duration during which unit is continuously off

REFERENCES

[1] A.F. Wood and B.F. Wollenberg, "Power Generation Operation and Control", 2nd Edition, New York: Wiley, 1966.
[2] R.M. Burns and C.A. Gibson, "Optimization of Priority Lists for a Unit Commitment Program", Proc. IEEE/Power Engineering Society, Summer Meeting, Paper A 75 453-1, 1975.
[3] G.B. Sheble, "Solution of the Unit Commitment Problem by the Method of Unit Periods", IEEE Trans. Power Syst., Vol. 5, No. 1, pp. 257-260, Feb. 1990.
[4] T.S. Dillon, K.W. Edwin, H.D. Kochs and R.J. Taud, "Integer Programming Approach to the Problem of Optimal Unit Commitment with Probabilistic Reserve Determination," IEEE Trans. Power App. Syst., Vol. PAS-97, No. 6, pp. 2154-2166, Dec. 1978.
[5] L.L. Garver, "Power Generation Scheduling by Integer Programming Development of Theory", IEEE Trans. Power App. Syst., Vol. PAS-18, pp. 730-735, Feb. 1963.
[6] W.L. Snyder, H.D. Powell and J.C. Rayburn, "Dynamic Programming Approach to Unit Commitment", IEEE Trans. Power App. Syst., Vol. PAS-2, pp. 339-350, May 1987.
[7] 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, Aug. 1991.
[8] J.A. Muckstadt and R.C. Wilson, "An Application of Mixed-Integer Programming Duality to Scheduling Thermal Generating Systems", IEEE Trans. Power App. Syst., pp. 1968-1978, 1968.
[9] A.I. Cohen and M. Yoshimura, "A Branch-and-Bound Algorithm for Unit Commitment", IEEE Trans. Power App. Syst., Vol. PAS-102, No. 2, pp.444-451, 1983.
[10] A. Merlin and P. Sandrin", A New Method for Unit Commitment at Electricite de France," IEEE Trans. Power App. Syst., Vol. PAS-102, pp.1218-1255, May 1983.
[11] F. Zhuang and F.D. Galiana, "Toward a More Rigorous and Practical Unit Commitment by Lagrangian Relaxation", IEEE Trans. Power Syst., Vol. 3, No. 2, pp. 763-770, May 1988.
[12] Z. Ouyang and S.M. Shahidehpour, "A Hybrid Artificial Neural Network/Dynamic Programming Approach to Unit Commitment", IEEE Trans. Power Syst., Vol. 7, No. 1, pp. 236-242, Feb. 1992.
[13] H. Sasaki, M. Watabable, J. Kubokawa, N. Yorino, and R. Yokoyama, "A Solution Method of Unit Commitment by Artificial Neural Networks", IEEE Trans. Power Syst., Vol. 7, No. 1, pp. 974-985, Feb. 1992.
[14] F. Zhuang and F.D. Galiana, "Unit Commitment by Simulated Annealing", IEEE Trans. Power Syst., Vol. 5, No. 1, pp. 311-317, Feb. 1990.
[15] D. Dasgupta and D.R. McGregor, "Thermal Unit Commitment Using Genetic Algorithms", Proc. Inst. Elect. Eng., Gener., Transm., Distrib., Vol. 141, No. 5, pp. 459-465, Sep. 1994.
[16] S.A. Kazarlis, A.G. Bakirtzis, and V. Petridis, "A Genetic Algorithm Solution to the Unit Commitment Problem", IEEE Trans. Power Syst., Vol. 11, No. 1, pp. 83-92, Feb. 1996.
[17] D.E. Goldberg, B. Korb and K. Deb, "Messy Genetic Algorithms: Motivation, Analysis and First Result", Technical Report, University of Alabama, 1989.

[18] X.S. Han and Z. Liu, "Optimal Unit Commitment Considering Unit's Ramp-Rate Limits," Power System Technology, Vol. 18, pp. 11-16, November 1994.
[19] J.H. Wu, Y.W. Wu and X.I. Xiong, "Optimization of Unit Commitment by Improved Hopfield Neural Network Algorithm", Automation of Electric Power System, Vol. 27, pp. 41-44, April 2003.
[20] H.Y. Chen, K.S. Zhang and X.F. Wang, "Evolutionary Optimization Method of Power System Unit Commitment Problem", Proceeding of the CESS, Vol. 19, pp. 9-13, 40, December 1999.
[21] Y. Chen, G.B. Zhao, J.Y. Liu, T.Q. Liu and H.Q. Li, "An Ant Colony Optimization and Particle Swarm Optimization Hybrid Algorithm for Unit Commitment based on Operation Coding", Power System Technology, Vol. 32, pp. 52-56, March 2008.
[22] J.S. Hu, C.X. Guo and Y.J. Cao, "A Hybrid Particle Swarm Optimization Method for Unit Commitment Problem", Proceeding of the CESS, Vol. 24, pp. 24-28, April 2004.
[23] P. Wei and N.H. Li, "Daily Generation Scheduling based on Genetic Algorithm", Automation of Electric Power Systems", Vol. 23, pp. 23-27, May 1999.
[24] C.H. Cai and Y.Y. Cai, "Optimization of Unit Commitment by Genetic Algorithm", Power System Technology, Vol. 21, pp. 44-47, January 1997.

BIOGRAPHIES

Amin Safari received the B.Sc. and M.Sc. degrees in Electrical Engineering in 2007 and 2009, respectively. Currently, he is a Ph.D. student of Power Electrical Engineering, Iran University of Science and Technology, Tehran, Iran. His areas of interest in research are application of artificial intelligence to power system control design, FACTS device and fuzzy logic control.

Rouzbeh Jahani received his B.Sc. degree from Amirkabir University of Technology, Tehran, Iran, in 2009. He is currently a M.Sc. student of Power Electrical Engineering in Islamic Azad University of Tehran, South Branch, Tehran, Iran. His research interests include FACTS devices, fuzzy logic control, unit commitment and power system optimization.

Heidar Ali Shayanfar received the B.Sc. and M.Sc. degrees in Electrical Engineering in 1973 and 1979, respectively. He received his Ph.D. degree in Electrical Engineering from Michigan State University, U.S.A., in 1981. Currently, he is a Full Professor in Electrical Engineering Department of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran. His research interests are in the application of artificial intelligence to power system control design, dynamic load modeling, power system observability studies and voltage collapse. He is a member of Iranian Association of Electrical and Electronic Engineers and IEEE.