An effective GAMS optimization for Dynamic Economic Load Dispatch with Ramp Rate Limit

F. Benhamida, I. Ziane(1), A. Graa
Irecom laboratory, dept. of electrotechnics
UDL university of Sidi Bel Abbes
Sidi Bel Abbes, Algeria
(1) farid.benhamida@yahoo.fr

S. Souag, Y. Salhi(2)
Apelec laboratory, dept. of electrotechnics
UDL university of Sidi Bel Abbes
Sidi Bel Abbes, Algeria
(2) salhiyacine13@yahoo.fr

***Abstract*— This paper presents a comparative analysis study of an efficient general algebraic modeling system (GAMS) to solve dynamic economic load dispatch (DELD) problem with and without Ramp rate limit of thermal power plants. The proposed GAMS take care of different unit and system constraints to find optimal solution. To validate the effectiveness of the proposed GAMS solution, simulations have been performed using three different cases, a 18-unit, 40-unit, and20-unit. Results obtained with GAMS have been compared with other existing relevant approaches available in literatures. Experimental results show a proficiency of GAMS over other existing techniques in terms of robustness.**

***Keywords* – Economic load dispatch; general algebraic modeling system GAMS; Ramp rate limit.**

I. Introduction

Economic load dispatch (ELD) problem concern the determination of the optimal combination of power output for all generating units which will minimize the total fuel cost while satisfying load and operational constraints in a high voltage power system. ELD is a complex problem to solve because of its massive dimension, a non-linear objective function and large number of constraints. Various investigations on the ELD have been undertaken till date. Suitable improvements in the unit output scheduling can contribute to significant cost savings. To improve the quality of solution, lots of researches have been done and various methods have been evolved so far in the field of ELD [1], [2]. Classical optimization techniques, such as the lambda iteration approach, the gradient method, the linear programming method and Newton's method were used to solve the ELD problem [3]. Lambda iteration method is the most common, which has been applied to solve ELD problems. But for effective implementation of this method, the formulation must be continuous. Though fast and reliable, the main drawback of the linear programming methods is that they are associated with the piecewise linear cost approximation [4].

Artificial Neural Network (ANN) techniques such as Hopfield Neural Network (HNN) [4] have been used to solve ELD for units having continuous or piecewise quadratic fuel cost functions and for units having prohibited zone constraints. Hopfield energy function and numerical iterations are applied to minimize the energy function which is mapped to the objective function of the ELD problem. In the conventional Hopfield Neural Network, the input-output relationship for its neurons is described by sigmoid function. Due to the use of the sigmoid function, the Hopfield model suffers from large computational time and curve saturation. To avoid such problem problems, a linear model is also used [5].

Evolutionary programming (EP), genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO) [6], [7] have been also proved to be effective with promising performance etc. Improved fast evolutionary programming algorithm has been successfully applied for solving the ELD problem [1], [5]. Chaotic particle swarm optimization (CPSO) [8], new particle swarm with local random search (NPSO-LRS) [9], Self-Organizing Hierarchical PSO [10], Bacterial foraging optimization [11], improved coordination aggregated based PSO [12], quantum-inspired PSO [13], improved PSO [14], HHS algorithm [15] and HIGA [16] have been successfully applied to solve the ELD problem.

A comparative analysis study of General Algebraic Modeling System (GAMS) approach is proposed to solve ELD problems with and without ramp rate limits. The objective function for the 3 test system used in the simulation is quadratic but the constraints are not linear. GAMS is a high-level model development environment that supports the analysis and solution of mixed integer optimization linear and non linear problems. GAMS is an accurate tool which can be useful easily for large and complex optimization problem. In this paper the effectiveness of the proposed algorithm is demonstrated using 3 test system (i) a 18-unit, (ii) a 40-unit, and (iii) a 20-unit.

II. ELD Problem Formulation

In a power system, the unit commitment problem has various sub-problems varying from linear programming problems to complex non-linear problems. The concerned ELD problem is one of the different non-linear programming sub-problems of unit commitment. The ELD problem is about minimizing the fuel cost of generating units for a specific period of operation so as to accomplish optimal generation dispatch among operating units and in return satisfying the system load demand considering power system operational constraints.

The objective function corresponding to the production cost can be approximated to be a quadratic function of the active power outputs from the generating units. Symbolically, it is represented as

$$\min F_T(P_G) = \sum_{i=1}^{N} F_i(P_{Gi}) \tag{1}$$

where the expression for cost function corresponding to *i*-th generating unit is given by:

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

where a_i , b_i and c_i are the cost coefficients; P_{Gi} is the real power output (MW) of *i*-th generator corresponding to time period *t* and *N* is the number of online generating units to be dispatched.

The objective function is subject to the following constraints:

1) Power Balance Constraints:

The total system generation must be equal to the sum total system loads (P_D) and losses (P_L). That is,

$$P_D + P_L = \sum_{i=1}^{N} P_{Gi} \tag{3}$$

The transmission losses can be expressed using the B-coefficients loss formula

$$P_L\left(\{P_{Gi}\}\right) = \sum_{i}^{N}\sum_{j}^{N} P_{Gi} B_{i,j} P_{n2} + \sum_{i}^{N} B_{i,0}.P_{Gi} + B_{0,0} \tag{4}$$

where the parameters $\{B_{i,j}\}$, $\{B_{i,0}\}$, and $B_{0,0}$ are B-coefficients known for a specific power system.

By applying Lagrangian multipliers method and Kuhn tucker conditions the following conditions for optimality can be obtained.

$$2a_i P_i + b_i = \lambda\left(1 - B_{i,0} - 2\sum_{j=1}^{N} B_{i,j} P_i\right)(i = 1, 2, ..., N) \tag{5}$$

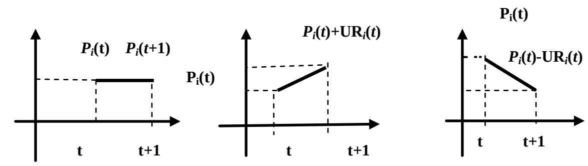
2) The Generator Constraints:

The power generated by each generator should be within its lower limit and upper limit so that

$$P_{Gi}^{\min} \le P_{Gi} \le P_{Gi}^{\max} \tag{6}$$

3) Ramp Rate Limits:

The range of actual operation of online generating unit is restricted by its ramp rate limits. These limits can impact the operation of generating unit. The operational decision at the present hour may affect the operational decision at the later hour due to ramp rate limits.



a) Steady state power demand　b) Increased power demand　c) Decreased power demand

Figure 1. Ramp rate limits of the generating units

The generator constraints due to ramp rate limits of generating units are given as

A) when generation increases

$$P_{it} - P_{i(t-1)} \le UR_i \tag{7}$$

B) when generation decreases

$$P_{i(t-1)} + P_{it} \le DR_i \tag{8}$$

Therefore the generator constraints can be modified as

$$\max(P_{i\min}, P_{i(t-1)} - DR_i) \le P_{it} \le \min(P_{i\max}, P_{i(t-1)} + UR_i) \tag{9}$$

III. General Algebraic Modeling System (GAMS)

GAMS is a high-level model specially designed for modeling linear, nonlinear and mixed integer optimization problems. GAMS can easily handle large and complex problems. It is especially useful for handling large complex problems, which may require much revision to establish an accurate model. Models can be developed, solved and documented simultaneously, maintaining the same GAMS model file. The basic structure of a mathematical model coded in GAMS has the components: sets, data, variable, equation, model and output [18] and the solution procedures are shown below.

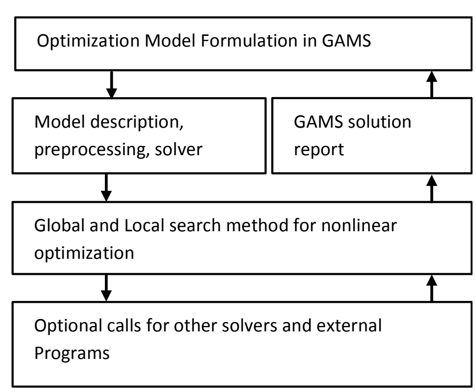


Figure 2. GAMS modeling and solution procedure.

GAMS formulation follows the basic format as given below:

- **Sets**: Declaration, Assignment of members;
- **Data** (parameters, tables, scalars), Declaration, Assignment of values;
- **Variables**: Declaration, Assignment of type, Assignment of bounds and/or initial values (optional);
- **Equations**: Declaration, Definition;
- **Model and solve statements**;
- **Display statements** (optional)

IV. Result And Discussion

GAMS-CONOPT solver has been applied on 3 different standard high voltage power systems where the source code is given in appendix. Test case I is a 18-unit, test case II is a 40-unit, test case III is 20-unit system. The programs were implementation on GAMS with a Pentium 4 processor and 1GB RAM.

A. 18-Unit test sytem

A 18-unit test system having quadratic cost function: The parameters of all thermal units are taken from [18], and given in Table I. The maximum power demand of the system set at P_D = 433.22 MW. The results are compared with λ-iteration and Binary GA [18], RGA [18] and ABC [21] for different demands (95%, 90%, 80% and 70%) without losses. From Table V, we can show that GAMS provides superior result then earlier reported results. The summarized and comparative DELD results of case I are given Table V. The percent changes in results are also given in Table VI and illustrated in Fig. 3.

TABLE I. Parameters Of 18-Unit System

Unit n°	P_i^{min}	P_i^{max}	a_i	b_i	c_i	R^{down}	R^{up}
1	7	15.00	0.602842	22.45526	85.74158	10	10
2	7	45.00	0.602842	22.45526	85.74158	10	10
3	13	25.00	0.214263	22.52789	108.98370	10	10
4	16	25.00	0.077837	26.75263	49.06263	10	10
5	16	25.00	0.077837	26.75263	49.06263	10	10
6	3	14.75	0.734763	80.39345	677.73000	5	5
7	3	14.75	0.734763	80.39345	677.73000	10	10
8	3	12.28	0.514474	13.19474	44.390000	10	10
9	3	12.28	0.514474	13.19474	44.390000	10	10
10	3	12.28	0.514474	13.19474	44.390000	10	10
11	3	12.28	0.514474	13.19474	44.390000	10	10
12	3	24.00	0.657079	56.70947	574.96030	10	10
13	3	16.20	1.236474	84.67579	820.37760	10	10
14	3	36.20	0.394571	59.59026	603.02370	7	7
15	3	45.00	0.420789	56.70947	567.93630	10	10
16	3	37.00	0.420789	55.96500	567.93630	10	10
17	3	45.00	0.420789	55.96500	567.93630	10	10
18	3	16.20	1.236474	84.67579	820.37760	3	3

B. 40-Unit test sytem

A 40-unit with quadratic cost functions where the input data of the entire system are given in [19]. A load demands of 9000 MW and 10500 MW without transmission losses are considered. The results are compared with VSDE [19] and SA [21] methods for this system. The results obtained by GAMS are listed in Table VII. The 10 load demands are posed for study the DELD problem. The results and the percent changes are given in Table VIII and illustrated in Fig. 4.

TABLE II. Parameters Of 40-Unit System

Unit n°	P_i^{min}	P_i^{max}	a_i	b_i	c_i	R^{down}	R^{up}
1	40	80	0.03073	8.3360	170.44	20	20
2	60	120	0.02028	7.0706	309.54	40	40
3	80	190	0.00942	8.1817	369.03	50	50
4	24	42	0.08482	6.9467	135.48	10	10
5	26	42	0.09693	6.5595	135.19	10	10
6	68	140	0.01142	8.0543	222.33	40	40
7	110	300	0.00357	8.0323	287.71	80	100
8	135	300	0.00492	6.9990	391.98	80	100
9	135	300	0.00573	6.6020	455.76	80	100
10	130	300	0.00605	12.908	722.82	80	100
11	94	375	0.00515	12.986	635.20	80	130
12	94	375	0.00569	12.796	654.69	80	130
13	125	500	0.00421	12.501	913.40	80	100
14	125	500	0.00752	8.8412	1760.4	80	100
15	125	500	0.00708	9.1575	1728.3	80	100
16	125	500	0.00708	9.1575	1728.3	80	100
17	125	500	0.00708	9.1575	1728.3	80	100
18	220	500	0.00313	7.9691	647.85	80	100
19	220	500	0.00313	7.9550	949.69	80	100
20	242	550	0.00313	7.9691	947.83	80	100
21	242	550	0.00313	7.9691	647.81	80	100
22	254	550	0.00298	6.6313	785.96	80	100
23	254	550	0.00298	6.6313	785.96	80	100
24	254	550	0.00284	6.6611	794.53	80	100
25	254	550	0.00284	6.6611	794.53	80	100
26	254	550	0.00277	7.1032	801.32	80	100
27	254	550	0.00277	7.1032	801.32	80	100
28	10	150	0.52124	3.3353	1055.1	80	100
29	10	150	0.52124	3.3353	1055.1	80	100
30	10	150	0.52124	3.3353	1055.1	80	100
31	20	70	0.25098	13.052	1207.8	20	40
32	20	70	0.16766	21.887	810.79	20	40
33	20	70	0.26350	10.244	1247.7	20	40
34	20	70	0.30575	8.3707	1219.2	20	20
35	18	60	0.18362	26.258	641.43	15	20
36	18	60	0.32563	9.6956	1112.8	20	40
37	20	60	0.33722	7.1633	1044.4	20	40
38	25	60	0.23915	16.339	832.24	20	40
39	25	60	0.23915	16.339	834.24	20	40
40	25	60	0.23915	16.339	1035.2	20	40

C. 20-Unit test sytem

The system consists of 20 generating units having quadratic cost function taking into account transmission losses. Power demand is set at 2500 MW. The parameters of all thermal units and loss coefficient are taken from [20] and listed in Table III and Table IV, respectively. The results are compared with λ-iteration and Hopfield Model [20], BBO [8] and SA [20] methods for this system. The results obtained by GAMS are listed in Table IX. It can be clearly seen from Table IX the proposed GAMS provides better results as compared to other reported evolutionary algorithm techniques like λ-iteration, Hopfield Model, BBO and SA. The 10 load demands are posed for study the DELD problem. The results and the percent changes are given in Table X and illustrated in Fig. 5.

TABLE III. PARAMETERS OF 20-UNIT SYSTEM

Unit n°	P_i^{min}	P_i^{max}	a_i	b_i	c_i	R^{down}	R^{up}
1	150	600	0.00068	18.19	1000	**70**	**70**
2	50	200	0.00071	19.26	970	**70**	**60**
3	50	200	0.00650	19.80	600	**70**	**60**
4	50	200	0.00500	19.10	700	**70**	**60**
5	50	160	0.00738	18.10	420	**50**	**50**
6	20	100	0.00612	19.26	360	**50**	**50**
7	25	125	0.00790	17.14	490	**50**	**40**
8	50	150	0.00813	18.92	660	**50**	**40**
9	50	200	0.00522	18.27	765	**50**	**40**
10	30	150	0.00573	18.92	770	**50**	**40**
11	100	300	0.00480	16.69	800	**50**	**40**
12	150	500	0.00310	16.76	970	**80**	**70**
13	40	160	0.00850	17.36	900	**50**	**40**
14	20	130	0.00511	18.70	700	**50**	**40**
15	25	185	0.00398	18.70	450	**20**	**20**
16	20	80	0.07120	14.26	370	**20**	**20**
17	30	85	0.00890	19.14	480	**20**	**20**
18	30	120	0.00713	18.92	680	**20**	**20**
19	40	120	0.00622	18.47	700	**20**	**15**
20	30	100	0.00773	19.79	850	**20**	**20**

TABLE IV. $B_{I,J}$ LOSS PARAMETERS (IN $MW^{-1} \times 10^{-5}$) FOR THE 20- UNIT ELD PROBLEM

Unit	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	8.70	0.43	-4.61	0.36	0.32	-0.66	0.96	-1.60	0.80	-0.10	3.60	0.64	0.79	2.10	1.70	0.80	-3.20	0.70	0.48	-0.70
2	0.43	8.30	-0.97	0.22	0.75	-0.28	5.04	1.70	0.54	7.20	-0.28	0.98	-0.46	1.30	0.80	-0.20	0.52	-1.70	0.80	0.20
3	-4.16	-0.97	9.00	-2.00	0.63	3.00	1.70	-4.30	3.10	-2.00	0.70	-0.77	0.93	4.60	-0.30	4.20	0.38	0.70	-2.00	3.60
4	0.36	0.22	-2.00	5.30	0.47	2.62	-1.96	2.10	0.67	1.80	-0.45	0.92	2.40	7.60	-0.20	0.70	-1.00	0.86	1.60	0.87
5	0.32	0.75	0.63	0.47	8.60	-0.80	0.37	0.72	-0.90	0.69	1.80	4.30	-2.80	-0.70	2.30	3.60	0.80	0.20	-3.00	0.50
6	-0.66	-0.28	3.00	2.62	-0.80	11.80	-4.90	0.30	3.00	-3.00	0.40	0.78	6.40	2.60	-0.20	2.10	-0.40	2.30	1.60	-2.10
7	0.96	5.04	1.70	-1.96	0.37	-4.90	8.24	-0.90	5.90	-0.60	8.50	-0.83	7.20	4.80	-0.90	-0.10	1.30	0.76	1.90	1.30
8	-1.60	1.70	-4.30	2.10	0.72	0.30	-0.90	1.20	-0.96	0.56	1.60	0.80	-0.40	0.23	0.75	-0.56	0.80	-0.30	5.30	0.80
9	0.80	0.54	3.10	0.67	-0.90	3.00	5.90	-0.96	0.93	-0.30	6.50	2.30	2.60	0.58	-0.10	0.23	-0.30	1.50	0.74	0.70
10	-0.10	7.20	-2.00	1.80	0.69	-3.00	-0.60	0.56	-0.30	0.99	-6.60	3.90	2.30	-0.30	2.80	-0.80	0.38	1.90	0.47	-0.26
11	3.60	-0.28	0.70	-0.45	1.80	0.40	8.50	1.60	6.50	-6.60	10.70	5.30	-0.60	0.70	1.90	-2.60	0.93	-0.60	3.80	-1.50
12	0.64	0.98	-0.77	0.92	4.30	0.78	-0.83	0.80	2.30	3.90	5.30	8.00	0.90	2.10	-0.70	5.70	5.40	1.50	0.70	0.10
13	0.79	-0.46	0.93	2.40	-2.80	6.40	7.20	-0.40	2.60	2.30	-0.60	0.90	11.00	0.87	-1.00	3.60	0.46	-0.90	0.60	1.50
14	2.10	1.30	4.60	7.60	-0.70	2.60	4.80	0.23	0.58	-0.30	0.70	2.10	0.87	3.80	0.50	-0.70	1.90	2.30	-0.97	0.90
15	1.70	0.80	-0.30	-0.20	2.30	-0.20	-0.90	0.75	-0.10	2.80	1.90	-0.70	-1.00	0.50	11.00	1.90	-0.80	2.60	2.30	-0.10
16	0.80	-0.20	4.20	0.70	3.60	2.10	-0.10	-0.56	0.23	-0.80	-2.60	5.70	3.60	-0.70	1.90	10.80	2.50	-1.80	0.90	-2.60
17	-3.20	0.52	0.38	-1.00	0.80	-0.40	1.30	0.80	-0.30	0.38	0.93	5.40	0.46	1.90	-0.80	2.50	8.70	4.20	-0.30	0.68
18	0.70	-1.70	0.70	0.86	0.20	2.30	0.76	-0.30	1.50	1.90	-0.60	1.50	-0.90	2.30	2.60	-1.80	4.20	2.20	0.16	-0.30
19	0.48	0.80	-2.00	1.60	-3.00	1.60	1.90	5.30	0.74	0.47	3.80	0.70	0.60	-0.97	2.30	0.90	-0.30	0.16	7.60	0.69
20	-0.70	0.20	3.60	0.87	0.50	-2.10	1.30	0.80	0.70	-0.26	-1.50	0.10	1.50	0.90	-0.10	-2.60	0.68	-0.30	0.69	7.00

TABLE V. COMPARISION OF ECONOMIC LOAD DISPATCH RESULT OF 18-UNIT

Demand	λ-iteration ($/hr)[17]	Binary GA ($/hr) [17]	Real coded GA ($/hr) [17]	ABC ($/hr) [18]	GAMS ($/hr)
411.559	29731.05	29733.42	29731.05	29730.8	**29731.067**
389.898	27652.47	27681.05	27655.53	27653.3	**27653.750**
346.576	23861.58	23980.24	23861.58	23859.4	**23855.286**
303.254	20393.43	20444.68	20396.39	20391.6	**20386.216**

TABLE VI. COMPARAISON OF DELD RESULTS OF 18-UNIT SYSTEM WITH AND WITHOUT RAMP RATE LIMIT CONSTRAINTS USING GAMS

Time interval	***D***	**GAMS with ramping rate**			**GAMS without ramping rate**			**Percent change (%)**		
		Total cost	**Losses**	**System λ**	**Total cost**	**Losses**	**System λ**	**Total cost**	**Losses**	**System λ**
1	**411.559**	29731.067	0.000	100.535	29731.066	0.000	100.535	0.000000	0.0000	0.0000
2	**389.898**	27654.110	0.000	92.245	27653.750	0.000	92.463	0.001302	0.0000	-0.2363
3	**346.576**	23856.274	0.000	83.817	23855.286	0.000	83.947	0.004141	0.0000	-0.1551
4	**303.254**	20389.390	0.000	75.770	20386.216	0.000	76.267	0.015567	0.0000	-0.6559

TABLE VII. BEST POWER OUTPUT FOR 40-UNIT SYSTEM

	VSHDE [19]	**SA [21]**	**GAMS**	**VCHDE [19]**	**SA [21]**	**GAMS**
Total generation	10500	10500	10500	9000	9000	9000
P_D (MW)	10500	10500	10500	9000	9000	9000
Power Mismatch	0	0	0	0	0	0
Total cost ($/hr)	143943.9	143930.409	143926.424	121253.01	121245.164	**121244.086**

TABLE VIII. COMPARAISON OF DELD RESULTS OF 40-UNIT SYSTEM WITH AND WITHOUT RAMP RATE LIMIT CONSTRAINTS USING GAMS

Time interval	*D*	GAMS with ramping rate			GAMS without ramping rate			Percent change (%)		
		Total cost	Losses	System λ	Total cost	Losses	System λ	Total cost	Losses	System λ
1	**5000**	79233.084	0.000	8.813	79233.084	0.000	8.813	0.0000	0	0.0000
2	**5800**	86515.619	0.000	9.391	86515.540	0.000	9.384	0.0001	0	0.0745
3	**6200**	90313.561	0.000	9.597	90313.561	0.000	9.597	0.0000	0	0.0000
4	**7000**	98152.380	0.000	10.040	98152.380	0.000	10.040	0.0000	0	0.0000
5	**8400**	114722.636	0.000	15.088	113392.395	0.000	12.119	1.1595	0	19.6779
6	**9000**	121361.978	0.000	14.225	121244.086	0.000	13.943	0.0971	0	1.9824
7	**10500**	161177.135	0.000	102.574	143926.424	0.000	16.257	10.7030	0	84.1510
8	**9200**	127032.370	0.000	11.067	124066.733	0.000	14.282	2.3346	0	-29.0503
9	**8700**	117187.861	0.000	13.030	117175.216	0.000	13.100	0.0108	0	-0.5372
10	**8000**	108864.476	0.000	10.870	108761.503	0.000	11.196	0.0946	0	-2.9991

TABLE IX. COMPARISION OF ECONOMIC LOAD DISPATCH RESULT OF 20-UNIT SYSTEM (P_D=2500 MW)

Unit N°	λ-iteration ($/hr)[20]	Hopfield Model[20]	BBO[8]	SA[20]	GAMS
Power loss (MW)	91.967	91.9669	92.1011	91.9662	91.967
Total generation (MW)	2591.967	2591.9669	2591.1011	2591.9662	2591.967
Power Demand (MW)	**2500**	**2500**	**2500**	**2500**	**2500**
Power Mismatch	**0**	**0**	**0**	**0**	0
Total cost ($/hr)	**62456.6391**	**62456.6341**	**62456.7926**	**62456.63309**	**62456.633**

TABLE X. COMPARAISON OF DELD RESULTS OF 20-UNIT SYSTEM WITH AND WITHOUT RAMP RATE LIMIT CONSTRAINTS USING GAMS

Time interval	*D*	GAMS with ramping rate			GAMS without ramping rate			Percent change (%)		
		Total cost	Losses	System λ	Total cost	Losses	System λ	Total cost	Losses	System λ
1	**1500**	41997.611	40.873	19.910	41997.611	40.873	19.910	0,000000	0	0
2	**1700**	46006.646	50.420	20.167	46006.646	50.420	20.167	0,000000	0	0
3	**1800**	48028.339	54.934	20.267	48028.339	54.934	20.267	0,000000	0	0
4	**2100**	54152.971	69.611	20.563	54152.971	69.611	20.563	0,000000	0	0
5	**2400**	60365.872	86.045	20.858	60365.872	86.045	20.858	0,000000	0	0
6	**2500**	62456.633	91.967	20.958	62456.633	91.967	20.958	0,000000	0	0
7	**2000**	52105.605	64.974	20.396	52101.582	64.575	20.465	0.007721	0.614091791	-0.3383016
8	**1990**	51896.976	64.084	20.456	51896.976	64.084	20.456	0,000000	0	0
9	**1850**	49042.908	57.269	20.316	49042.908	57.269	20.316	0,000000	0	0
10	**1700**	46006.646	50.420	20.167	46006.646	50.420	20.167	0,000000	0	0

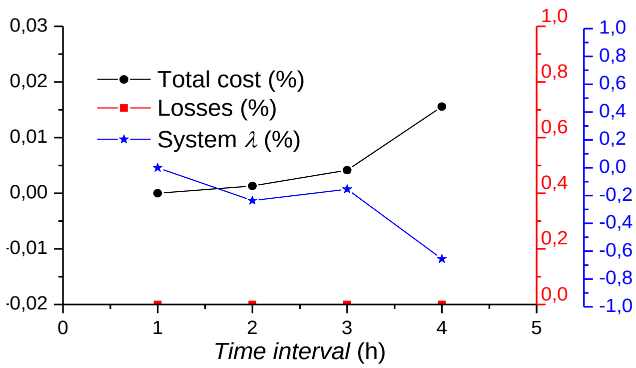


Figure 3. Percent variation in dynamic economic dispatch solution of the 18-generator problem for the 4 hour time interval using GAMS with ramping rate constraint comparing to GAMS optimal solution without ramping rate constraints.

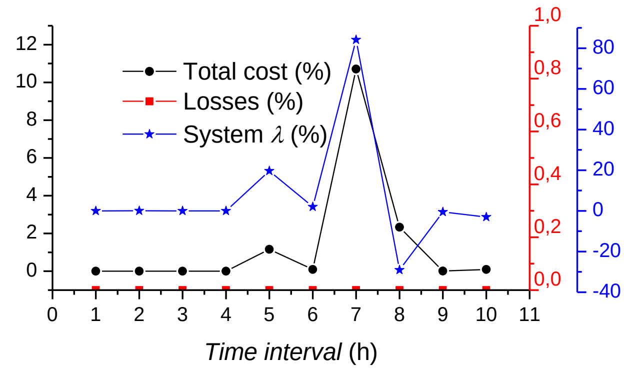


Figure 4. Percent variation in dynamic economic dispatch solution of the 40-generator problem for the 10 hour time interval using GAMS with ramping rate constraint comparing to GAMS optimal solution without ramping rate constraints.

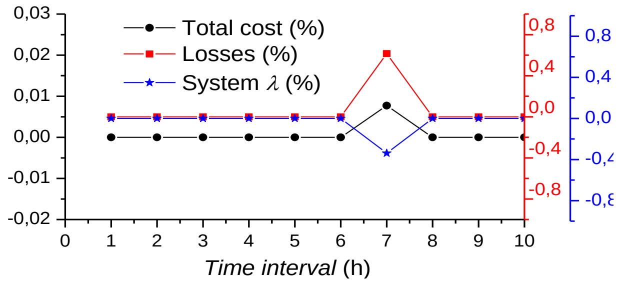


Figure 5. Percent variation in dynamic economic dispatch solution of the 20-generator problem for the 4 hour time interval using GAMS with ramping rate constraint comparing to GAMS optimal solution without ramping rate constraints.

V. CONCLUSION

GAMS has been used for solving 3 tests of high voltage power systems. Case I is 18-unit with quadratic cost characteristics without transmission loss, which is investigated by change in percentage of maximum demand (95%, 90%, 80% and 70%) and comparison is made with λ-iteration, Binary GA, RGA and ABC. Based on the simulated results, GAMS provides superior result than previously reported methods. Case II (40-unit) is investigated through two load demand levels and comparison is made with VSDE and SA. SGA and Hybrid GA reported in literature, the result shows that GAMS performs are better than above mentioned methods. In case III (20-unit test system) including losses, the obtained results are compared with λ-iteration, Hopfield Model, BBO algorithms. In this case also GAMS provides superior. GAMS algorithm has superior features, including quality of solution and good computational efficiency. The results show that GAMS is a promising technique for solving complicated problems in power system. The results show that the ramp rate limits change the solution of DELD.

REFERENCES

[1] B. H. Choudhary and S. Rahman, A review of recent advances in economic dispatch, IEEE Trans Power Sys., Vol. 5, No. 4, pp. 1248-1259, 1990.

[2] H. H. Happ, Optimal power dispatches – a comprehensive survey, IEEE Trans. Power Apparatus Syst., PAS-96, pp. 841-854, 1971.

[3] A. J. Wood and B. F. Wollenberg, Power Generation, Operation and Control, Wiley., New York 2nd ed, 1996.

[4] J. H. Park, Y. S. Kim, I. K. Eom and K. Y. Lee, Economic Load Dispatch for pricewise Quadratic Cost Function Using Hopfield Neural, IEEE Trans. on Power Systems, Vol. 8, No. 3, pp. 1030-1038, 1993.

[5] F. Benhamida et al, "Generation allocation problem using a Hopfield bisection approach including transmission losses," Elect. Power and Energ. Syst., vol. 33, no. 5, pp. 1165-1171, 2011.

[6] L. S. Coelho and V. C. Mariani, Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect, IEEE Trans. Power Syst., Vol. 21, No.2, pp. 989-996, 2006.

[7] J. B. Park, K. S. Lee, J. R. Shin and K.Y. Lee, A particle swarm optimization for economic dispatch with nonsmooth cost functions, IEEE Trans. Power Syst., Vol. 20, No. 1, pp. 34-42, 2005.

[8] A. Bhattacharya and P. K. Chattopadhyay, Biogeography-Based optimization for different

[9] K. T. Chaturvedi, M. Pandit and L. Srivastava, Self-Organizing Hierarchical Particle Swarm Optimization for Non-Convex Economic Dispatch, IEEE Trans. Power Syst., Vol. 23, No. 3, pp. 1079-1087,2008.

[10] B. K. Panigrahi and V. R. Pandi, Bacterial foraging optimization nelder mead hybrid algorithm for economic load dispatch, IET Gener. Transm. Distrib., Vol. 2, No. 4, pp. 556-565, 2008.

[11] G. John Vlachogiannis and K. Y. Lee, Economic load dispatch – a comparative study on heuristic optimization techniques with an improved coordinated aggregation-based PSO, IEEE Trans. Power Syst., Vol. 24 , No. 2, pp. 991-1001, 2009.

[12] Ke Meng, H. G. Wang and Z. Y. Dong, Quantum-inspired particle swarm optimization for valve-point economic load dispatch, IEEE Trans. Power Syst., Vol. 25, No. 1, pp. 215-222, 2010.

[13] J. B. Park, Y. W. Jeong, J. R. Sin and K. Y. Lee, An Improved particle swarm optimization for Non-convex Economic Load Dispatch Problems, IEEE Trans. Power Syst., Vol. 25, No. 1, pp. 156-166, 2010.

[14] V. R. Pandi, B. K. Panigrahi, R. C. Bansal, S. Das and A. Mohapatra, Economic load dispatch using hybrid swarm intelligence based harmonics search algorithm, Electric Power Comp. and Systems, Vol. 39 pp. 751-767, 2011.

[15] M. M. Hosseini, H. Ghorbani, A. Rabii and Sh. Anvari, A novel heuristic algorithm for solving Non-convex economic load dispatch problem with non smooth cast function, J. Basic Appl. Sci. Res., 2(2)1130-1135, 2012.

[16] Sichard E. Rosenthal, GAMS, A User's Guide, Tutorial GAMS Development Corporation, Washington, 2010.

[17] G. Ioannis, Damousis, G. Anastasios, G. Bakirtzis and S. Dokopoulos Petros, Network-Constrained Economic Dispatch Using Real-Coded Genetic Algorithm, IEEE Trans on power system, Vol. 18, No. 1, pp. 198-204, 2003.

[18] G. P. Dixit, H. M. Dubey, M. Pandit and B. K. Panigrahi, Economic Load Dispatch using Artificial Bee Colony Optimization, International Journal of Advances in Electronics Engineering, pp. 129-124, 2011.

[19] Ji-Pyng Chiou, Variable Scaling hybrid differential evolution for large scale economic dispatch problem, Electrical power systems Reserch, Vol. 77, pp. 212-218, 2007.

[20] Leandro dos Santos Coelho and Chu-Sheng Lee, Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches, Electrical Power and Energy Systems Vol. 30, pp.297–307, 2008.

[21] M. S. Kaurav, H. M. Dubey, M. Pandit and B. K. Panigrahi, Simulated Annealing Algorithm for Combined Economic and Emission Dispatch, International Conference, ICACCN pp. 631-636, 07-09 Oct 2011.

APPENDIX

GAMS SOURCE TEXT FOR THE ELD PROBLEM USING GAMS-CONOPT SOLVER

A. 18-Unit test sytem DELD source code

```
sets
    i  generators / p1*p18 /
    genchar generator characteristics / a,b,c,upplim,lowlim /
    cg(genchar) cost categories / a,b,c /
alias (i,j) ;
```

```
table gendata(i,genchar) generator cost characteristics and limits
       a              b              c          upplim    lowlim
p1  85.74158        22.45526       0.602842       15.00      7
p2  85.74158        22.45526       0.602842       45.00      7
p3 108.98370        22.52789       0.214263       25.00     13
p4  49.06263        26.75263       0.077837       25.00     16
p5  49.06263        26.75263       0.077837       25.00     16
p6  677.7300        80.39345       0.734763       14.75      3
p7  677.7300        80.39345       0.734763       14.75      3
p8   44.3900        13.19474       0.514474       12.28      3
p9   44.3900        13.19474       0.514474       12.28      3
p10  44.3900        13.19474       0.514474       12.28      3
p11  44.3900        13.19474       0.514474       12.28      3
p12 574.9603        56.70947       0.657079       24.00      3
p13 820.3776        84.67579       1.236474       16.20      3
p14 603.0237        59.59026       0.394571       36.20      3
p15 567.9363        56.70947       0.420789       45.00      3
p16 567.9363        55.96500       0.420789       37.00      3
p17 567.9363        55.96500       0.420789       45.00      3
p18 820.3776        84.67579       1.236474       16.20      3;
parameter pexp(cg) exponent for cost function / a 0, b 1, c 2 /;
scalar demand total power demand in MW /303.254/ ;
variables
p(i) power generation level in MW
loss total transmission loss in MW
cost total generation cost - the objective function
positive variables p;
p.up(i) = gendata(i,"upplim") ;
p.lo(i) = gendata(i,"lowlim") ;
equations
costfn total cost calculation
lossfn total loss calculation
demcons total generation must equal demand and loss ;
costfn.. cost =e= sum((i,cg), gendata(i,cg)*power(p(i),pexp(cg)));
lossfn.. loss =e= 0;
demcons.. sum(i, p(i)) =g= demand ;
model edc /all/ ;
solve edc minimizing cost using nlp ;
set s trade-off points / min-loss, s1*s4, min-cost /
    st(s) in between points / s1*s4 /
parameter trace trace report ;
trace('cost','min-cost') = cost.l;
trace('loss','min-cost') = loss.l;
option limrow=0,limcol=0;
solve edc minimizing loss using nlp ;
trace('cost','min-loss') = cost.l;
trace('loss','min-loss') = loss.l;
loop(st, loss.fx = trace('loss','min-loss') + ord(st)/(card(st)+1)*(trace('loss','min-cost') -
trace('loss','min-loss'));
solve edc minimizing cost using nlp ;
trace('cost',st) = cost.l;
trace('loss',st) = loss.l);
display trace;
```

B. 40-Unit test sytem DELD source code

```
             sets
    i  generators / p1*p40 /
    genchar generator characteristics / a,b,c,upplim,lowlim /
    cg(genchar) cost categories / a,b,c /
alias (i,j) ;
table gendata(i,genchar) generator cost characteristics and limits
     a             b             c          upplim    lowlim
p1  170.44        8.3360        0.03073        80        40
p2  309.54        7.0706        0.02028       120        60
p3  369.03        8.1817        0.00942       190        80
p4  135.48        6.9467        0.08482        42        24
p5  135.19        6.5595        0.09693        42        26
p6  222.33        8.0543        0.01142       140        68
p7  287.71        8.0323        0.00357       300       110
p8  391.98        6.9990        0.00492       300       135
p9  455.76        6.6020        0.00573       300       135
p10 722.82        12.908        0.00605       300       130
p11 635.20        12.986        0.00515       375        94
p12 654.69        12.796        0.00569       375        94
p13 913.40        12.501        0.00421       500       125
p14 1760.4        8.8412        0.00725       500       125
p15 1728.3        9.1575        0.00708       500       125
p16 1728.3        9.1575        0.00708       500       125
p17 1728.3        9.1575        0.00708       500       125
p18 647.85        7.9691        0.00313       500       220
p19 949.69        7.9550        0.00313       500       220
p20 947.83        7.9691        0.00313       550       242
```

```
p21 647.81        7.9691        0.00313        550      242
p22 785.96        6.6313        0.00298        550      254
p23 785.96        6.6313        0.00298        550      254
p24 794.53        6.6611        0.00284        550      254
p25 794.53        6.6611        0.00284        550      254
p26 801.32        7.1032        0.00277        550      254
p27 801.32        7.1032        0.00277        550      254
p28 1055.1        3.3353        0.52124        150       10
p29 1055.1        3.3353        0.52124        150       10
p30 1055.1        3.3353        0.52124        150       10
p31 1207.8        13.052        0.25098         70       20
p32 810.79        21.887        0.16766         70       20
p33 1247.7        10.244        0.26350         70       20
p34 1219.2        8.3707        0.30575         70       20
p35 641.43        26.258        0.18362         60       18
p36 1112.8        9.6956        0.32563         60       18
p37 1044.4        7.1633        0.33722         60       20
p38 832.24        16.339        0.23915         60       25
p39 834.24        16.339        0.23915         60       25
p40 1035.2        16.339        0.23915         60       25;
parameter pexp(cg) exponent for cost function / a 0, b 1, c 2 /;
scalar demand total power demand in MW /9000/ ;
variables
p(i) power generation level in MW
loss total transmission loss in MW
cost total generation cost - the objective function
positive variables p;
p.up(i) = gendata(i,"upplim") ;
p.lo(i) = gendata(i,"lowlim") ;
equations
costfn total cost calculation
lossfn total loss calculation
demcons total generation must equal demand and loss ;
costfn.. cost =e= sum((i,cg), gendata(i,cg)*power(p(i),pexp(cg)));
*lossfn.. loss =e= b00 + sum(i, b0(i)*p(i))/100 + sum((i,j), p(i)*b(i,j)*p(j))/100;
 lossfn.. loss =e= 0;
demcons.. sum(i, p(i)) =g= demand ;
model edc /all/ ;
solve edc minimizing cost using nlp ;
set s trade-off points / min-loss, s1*s4, min-cost /
    st(s) in between points / s1*s4 /
parameter trace trace report ;
trace('cost','min-cost') = cost.l;
trace('loss','min-cost') = loss.l;
option limrow=0,limcol=0;
solve edc minimizing loss using nlp ;
trace('cost','min-loss') = cost.l;
trace('loss','min-loss') = loss.l;
loop(st, loss.fx = trace('loss','min-loss')+ord(st)/(card(st)+1)*(trace('loss','min-cost') -
trace('loss','min-loss'));
solve edc minimizing cost using nlp ;
trace('cost',st) = cost.l;
trace('loss',st) = loss.l);
display trace;
```

C. 20-Unit test sytem DELD source code

```
               sets
    i  generators / p1*p20 /
    genchar generator characteristics / a,b,c,upplim,lowlim /
    cg(genchar) cost categories / a,b,c /
alias (i,j) ;
table gendata(i,genchar) generator cost characteristics and limits
    a            b            c          upplim    lowlim
p1 1000       18.19       0.00680       600       150
p2  970       19.26       0.00071       200        50
p3  600       19.80       0.00650       200        50
p4  700       19.10       0.00500       200        50
p5  420       18.10       0.00738       150        50
p6  360       19.26       0.00612       100        20
p7  490       17.14       0.00790       125        25
p8  660       18.92       0.00813       150        50
p9  765       18.27       0.00522       200        50
p10 770       18.92       0.00573       150        30
p11 800       16.69       0.00480       300       100
p12 970       16.76       0.00310       500       150
p13 900       17.36       0.00850       160        40
p14 700       18.70       0.00511       130        20
p15 450       18.70       0.00398       185        25
p16 370       14.26       0.71200        80        20
p17 480       19.14       0.00890        85        30
p18 680       18.92       0.00713       120        30
p19 700       18.47       0.00622       120        40
```

```
p20 850         19.79         0.00773         100         30;
parameter pexp(cg) exponent for cost function / a 0, b 1, c 2 /;
table b(i,j) the b-matrix loss coefficients - squared components
      p1    p2     p3    p4    p5    p6    p7    p8    p9   p10   p11   p12   p13   p14   p15   p16   p17   p18   p19   p20
p1   8.70  0.43 -4.61  0.36  0.32 -0.66  0.96 -1.60  0.80 -0.10  3.60  0.64  0.79  2.10  1.70  0.80 -3.20  0.70  0.48 -0.70
p2   0.43  8.30 -0.97  0.22  0.75 -0.28  5.04  1.70  0.54  7.20 -0.28  0.98 -0.46  1.30  0.80 -0.20  0.52 -1.70  0.80  0.20
p3  -4.16 -0.97  9.00 -2.00  0.63  3.00  1.70 -4.30  3.10 -2.00  0.70 -0.77  0.93  4.60 -0.30  4.20  0.38  0.70 -2.00  3.60
p4   0.36  0.22 -2.00  5.30  0.47  2.62 -1.96  2.10  0.67  1.80 -0.45  0.92  2.40  7.60 -0.20  0.70 -1.00  0.86  1.60  0.87
p5   0.32  0.75  0.63  0.47  8.60 -0.80  0.37  0.72 -0.90  0.69  1.80  4.30 -2.80 -0.70  2.30  3.60  0.80  0.20 -3.00  0.50
p6  -0.66 -0.28  3.00  2.62 -0.80 11.80 -4.90  0.30  3.00 -3.00  0.40  0.78  6.40  2.60 -0.20  2.10 -0.40  2.30  1.60 -2.10
p7   0.96  5.04  1.70 -1.96  0.37 -4.90  8.24 -0.90  5.90 -0.60  8.50 -0.83  7.20  4.80 -0.90 -0.10  1.30  0.76  1.90  1.30
p8  -1.60  1.70 -4.30  2.10  0.72  0.30 -0.90  1.20 -0.96  0.56  1.60  0.80 -0.40  0.23  0.75 -0.56  0.80 -0.30  5.30  0.80
p9   0.80  0.54  3.10  0.67 -0.90  3.00  5.90 -0.96  0.93 -0.30  6.50  2.30  2.60  0.58 -0.10  0.23 -0.30  1.50  0.74  0.70
p10 -0.10  7.20 -2.00  1.80  0.69 -3.00 -0.60  0.56 -0.30  0.99 -6.60  3.90  2.30 -0.30  2.80 -0.80  0.38  1.90  0.47 -0.26
p11  3.60 -0.28  0.70 -0.45  1.80  0.40  8.50  1.60  6.50 -6.60 10.70  5.30 -0.60  0.70  1.90 -2.60  0.93 -0.60  3.80 -1.50
p12  0.64  0.98 -0.77  0.92  4.30  0.78 -0.83  0.80  2.30  3.90  5.30  8.00  0.90  2.10 -0.70  5.70  5.40  1.50  0.70  0.10
p13  0.79 -0.46  0.93  2.40 -2.80  6.40  7.20 -0.40  2.60  2.30 -0.60  0.90 11.00  0.87 -1.00  3.60  0.46 -0.90  0.60  1.50
p14  2.10  1.30  4.60  7.60 -0.70  2.60  4.80  0.23  0.58 -0.30  0.70  2.10  0.87  3.80  0.50 -0.70  1.90  2.30 -0.97  0.90
p15  1.70  0.80 -0.30 -0.20  2.30 -0.20 -0.90  0.75 -0.10  2.80  1.90 -0.70 -1.00  0.50 11.00  1.90 -0.80  2.60  2.30 -0.10
p16  0.80 -0.20  4.20  0.70  3.60  2.10 -0.10 -0.56  0.23 -0.80 -2.60  5.70  3.60 -0.70  1.90 10.80  2.50 -1.80  0.90 -2.60
p17 -3.20  0.52  0.38 -1.00  0.80 -0.40  1.30  0.80 -0.30  0.38  0.93  5.40  0.46  1.90 -0.80  2.50  8.70  4.20 -0.30  0.68
p18  0.70 -1.70  0.70  0.86  0.20  2.30  0.76 -0.30  1.50  1.90 -0.60  1.50 -0.90  2.30  2.60 -1.80  4.20  2.20  0.16 -0.30
p19  0.48  0.80 -2.00  1.60 -3.00  1.60  1.90  5.30  0.74  0.47  3.80  0.70  0.60 -0.97  2.30  0.90 -0.30  0.16  7.60  0.69
p20 -0.70  0.20  3.60  0.87  0.50 -2.10  1.30  0.80  0.70 -0.26 -1.50  0.10  1.50  0.90 -0.10 -2.60  0.68 -0.30  0.69  7.00;
parameter b0(i) linear loss coefficients /
p1        0
p2        0
p3        0
p4        0
p5        0
p6        0
p7        0
p8        0
p9        0
p10       0
p11       0
p12       0
p13       0
p14       0
p15       0
p16       0
p17       0
p18       0
p19       0
p20       0/;
scalar b00 loss equation constant / 0 / ;
scalar demand total power demand in MW / 2500/ ;
variables
p(i) power generation level in MW
loss total transmission loss in MW
cost total generation cost - the objective function
positive variables p;
p.up(i) = gendata(i,"upplim") ;
p.lo(i) = gendata(i,"lowlim") ;
equations
costfn total cost calculation
lossfn total loss calculation
demcons total generation must equal demand and loss ;
costfn.. cost =e= sum((i,cg), gendata(i,cg)*power(p(i),pexp(cg)));
lossfn.. loss =e= b00 + sum(i, b0(i)*p(i))/100000 + sum((i,j), p(i)*b(i,j)*p(j))/100000;
demcons.. sum(i, p(i)) =g= demand + loss ;
model edc /all/ ;
solve edc minimizing cost using nlp ;
set s trade-off points / min-loss, s1*s4, min-cost /
    st(s) in between points / s1*s4 /
parameter trace trace report ;
trace('cost','min-cost') = cost.l;
trace('loss','min-cost') = loss.l;
option limrow=0,limcol=0;
solve edc minimizing loss using nlp ;
trace('cost','min-loss') = cost.l;
trace('loss','min-loss') = loss.l;
loop(st, loss.fx = trace('loss','min-loss') + ord(st)/(card(st)+1)*(trace('loss','min-cost') -
trace('loss','min-loss'));
solve edc minimizing cost using nlp ;
trace('cost',st) = cost.l;
trace('loss',st) = loss.l);
display trace;
```