Comparison of Post Outage Bus Voltage Magnitudes Estimated by Harmony Search and Differential Evolution Methods

Oguzhan Ceylan
Informatics Institute
Istanbul Technical University
oguzhan@be.itu.edu.tr

Aydogan Ozdemir
Electrics-Electronics Faculty
Istanbul Technical University
ozdemir@elk.itu.edu.tr

Hasan Dag
Information Technologies Department
Kadir Has University
hasan.dag@khas.edu.tr

***Abstract*—Contingency studies are indispensable tools of both the power system planning and operational studies. Real time implementation of operational problems makes necessary the use of high speed computational methods while requiring reasonable accuracies. On the other hand, accuracy of the results and the speed of calculation depend on branch outage modeling as well as solution algorithm used. This paper presents a comparison of post outage bus voltage magnitudes calculated by two meta-heuristic approaches; namely differential evolution (DE) and harmony search (HS) methods. The methods are tested on IEEE 14, IEEE 30, IEEE 57, and IEEE 118 bus test systems and the results are compared both in terms of accuracy and calculation speed.**

***Index Terms*—Branch outage modeling, post outage state, optimization, differential evolution, harmony search.**

I. INTRODUCTION

Contingency analysis is an important component of the security function, which is considered to be an integral part of the modern energy management system at energy control centers. Online steady-state security analysis of electric power systems requires evaluation of the effects of a large number of generation unit and transmission line contingencies in order to assess the security of the system. Outages of some of these components may cause several local problems such as voltage magnitude limit violation, active power flow limit violation, and complete block out of the system etc. Electric power system operators need to simulate all possible outage scenarios to determine the consequences of such outages and to prepare remedial actions. These simulations aim to estimate post outage voltage magnitudes and power flows.

An exact post-outage state determination requires a full AC load flow, which is a time-consuming process even for a moderate size system and it is not appropriate for real time tasks. Because of the high speed solution required for online processing of a large number of contingencies, fast methods and models providing sufficient accuracies are preferred. DC load flow [1] can be implemented to obtain post outage active power distributions. However, it does not handle reactive power and hence, the bus voltage magnitude calculation is not accurate. There are several methods and approximations that use linearized models [2], [3], [4]. However, almost all of these methods suffer from less accuracy especially for reactive power flows and bus voltage magnitudes.

Recently, the branch outage problem is formulated as a local constrained optimization problem using two fictitious sources [5]. In this model, bus voltage magnitudes determined from the linearized reactive power flow equations are later revised by means of so called local optimization cycle. The method makes use of the base case variables and the linearized MW power flows for a restricted group of network variables. Therefore, it does not require an excessive amount of computational time. However, the optimization cycle provides a non-linear feedback for reactive power mismatches, which in turn minimizes the load bus voltage magnitude errors due to linearized network equations (constraints). The resulting bus voltage magnitudes and reactive line flows are much better than the ones obtained by the traditional approaches.

After having formulated the line outage as a local optimization problem, the second step is the solution of optimization equations. It was first solved by gradient type algorithms [5]. However, analytical methods suffer from some computational problems due to the use of derivatives. Moreover, since most of these methods depend on the iterative improvements for a single solution candidate, they are not appropriate for parallel programming.

Evolutionary algorithms (EAs) are stochastic search methods based on the mechanics of the metaphor of natural biological evolution. They operate on a population of potential solutions applying the principle of survival of the fittest procedure to obtain better and better approximation to a solution. At each generation, a new set of better approximations is created by selecting individuals according to their fitness in the problem domain. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals from those they were created. The basic features of EAs are:

- They work with a coding of solution set, not the solutions themselves,
- They search from a population of solutions instead of a single solution,
- They use fitness function instead of derivatives,

- They do not require any prior knowledge or space limitations and,
- They use probabilistic transition rules instead of deterministic rules.

Although they are slower than the other numerical methods, parallel processing implementation have made them more attractive recently.

We have already tested several heuristic methods for line outage problem and reported the results of IEEE test system applications [6], [7]. Among them, Differential Evolution (DE) method showed better performance with respect to the computational accuracy and solution speed. This study extends our solution methods by Harmony Search (HS) Algorithm and compare the results with those of the DE method. Post outage voltage magnitude calculations using DE and HS methods are presented. The results obtained for IEEE 14, IEEE 30, IEEE 57, and IEEE 118 test systems are compared both in terms of accuracy and calculation speed.

II. Branch Outage Modeling

A transmission line, connecting two busses, and its associated reactive power flows can be given by using its π equivalent, as shown in Fig. 1. Reactive power flowing through the line ij, transferred reactive power, and reactive power loss are represented by Q_{ij}, Q_{ij}^T, and Q_{Li} respectively. These reactive powers can be expressed in terms of system variables as follows.

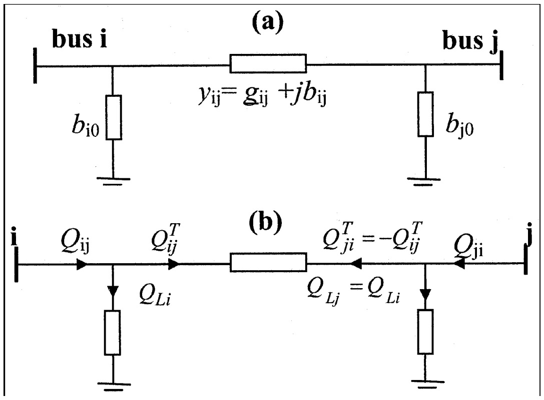

Fig. 1. Transmission line and reactive power flow model. a) π equivalent of a transmission line. b) reactive power flows.

$$Q_{ij} = -[V_i^2 - V_iV_jcos\delta_{ji}]b_{ij} + V_iV_jg_{ij}\sin\delta_{ji} - V_i^2\frac{b_{i0}}{2} \tag{1}$$

$$Q_{ij}^T = -[V_i^2 - V_j^2]\frac{b_{ij}}{2} + V_iV_jg_{ij}sin\delta_{ji} \tag{2}$$

$$Q_{Li} = -[V_i^2 + V_j^2 - 2V_iV_j\cos\delta ji]\frac{b_{ij}}{2} - (V_i^2 + V_j^2)\frac{b_{i0}}{4} \tag{3}$$

Pre-outage and actual outage states of a transmission line are shown in Figures 2.a and 2.b respectively. A line outage is simulated using fictitious sources as shown in Fig. 2.c [5]. The bounded network in which the computation for optimization takes place is shown in Fig. 3. Only load bus voltage magnitudes in this bounded region are taken into consideration during the computation process of the optimization problem.

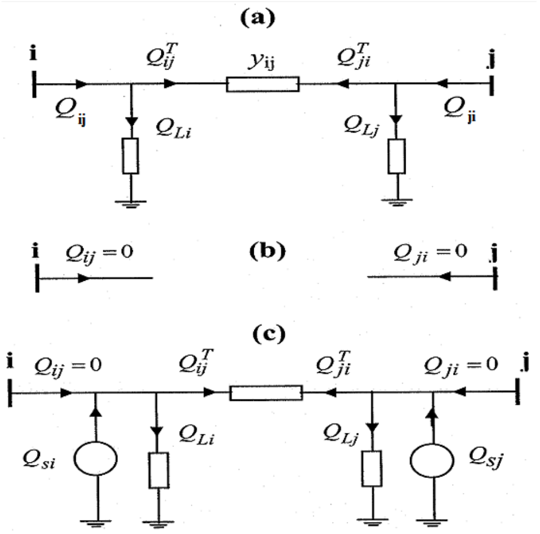

Fig. 2. Line outage modeling. a) pre-outage b) actual outage c) simulated post outage.

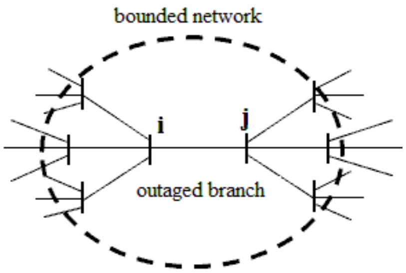

Fig. 3. Outaged line and bounded network.

The procedure for the existing method can be given as follows.

1) Select an outage of a branch, connected between busses i and j, and number it as k.
2) Calculate bus voltage phase angles by using linearized MW flows (see [1] for details).

$$\delta_l = \delta_l - (X_{li} - X_{lj}) \triangle P_k, \quad l = 2, 3, \cdots, \text{NB} \tag{4}$$

$$\triangle P_k = \frac{P_{ij}}{\frac{1-(X_{ii}+X_{jj}-2X_{ij})}{x_k}} \tag{5}$$

where, X represents the inverse of the bus susceptance matrix, P_{ij} is the pre-outage active power flow through the line, and x_k represents the reactance of the line at hand. If the voltage magnitudes are calculated, then the calculation of the busses included in the bounded network would suffice.

3) Calculate the reactive power transfer $\overline{Q}_{ij}^{T}$between the busses. This power includes the increment due to the change in bus voltage phase angles.
4) Minimize the reactive power mismatches of busses i and j. This process is mathematically equivalent to the following constrained optimization problem.

$$\begin{aligned} \min_{wrt\ Q_{si},Q_{sj}} \quad & \parallel Q_i - (\overline{Q}_{ij} + \overline{Q}_{Li}) + Q_{Di} \\ & Q_j - (-\overline{Q}_{ij} + \overline{Q}_{Li}) + Q_{Dj} \parallel \\ \text{subject to} \quad & g_q(V_b) = \triangle Q_b - B_b \triangle V_b = 0 \end{aligned} \tag{6}$$

where, $\parallel \cdot \parallel$ is the Euclidean norm of a vector. Equation (6) is linear reactive power equation for load busses, $\triangle Q$ is reactive power mismatch vector, V is bus voltage magnitude vector and B is bus susceptance matrix. It should be stated that only two elements of $\triangle Q$ vector are nonzero, and they are represented as shown below.

$$[\triangle Q] : [\triangle Q]_i = -[\triangle Q]_j = Q_{si} - Q_{ij}. \tag{7}$$

On the other hand, we use subscript b to denote the bounded region where the optimization process is done.

III. HEURISTIC METHODS

Among several heuristic methods those we have tested so far for the solution of the local optimization problem Differential Evolution method showed the better performance in computational accuracy and solution speed. Therefore, this study is concentrated on two heuristic methods; namely, Differential Evolution algorithm as being the best of the previous studies and the new Harmony Search Algorithm.

A. Differential Evolution Method

Differential evolution, which was firstly introduced by Storn and Price [8], [9], is a stochastic direct search optimization method. Differential evolution is a population-based algorithm and it has similar operators as those of the other evolutionary algorithms, such as crossover, mutation and selection.

Differential evolution has been applied to several power system problems, such as, power system planning [10], transient stability constrained optimal power flow [11], generation expansion planning [12], unit commitment [13], locating the voltage collapse points [14], reactive power optimization [15], economic dispatch [16], [17], etc.

The steps of a general differential evolution algorithm can be summarized as below:

1) Initialize the population: A population consisting of Np random vectors is generated. Np is a user specified value and is generally selected as 10-15 times of the number of unknown variables. Dimension of each vector in a population is equal to the number of unknowns. Representation of a population is shown below.

$$P^{(G)} = [x_1^{(G)}, \cdots, x_{Np}^{(G)}] \tag{8}$$

Random initialization of a vector in a population can be performed as,

$$x_i^{(G)} = x_{i(L)} + rand_i[0,1](x_{i(H)} - x_{i(L)}) \tag{9}$$

where, $x_{i(L)}$ and $x_{i(H)}$ are the lower and upper bounds for the initial vectors to be constructed in the population, respectively.

2) Create mutant vectors: Add a weighted difference vector of the two individual vectors to the third one and create Np new mutant vectors. This process performed for all vectors in the population can be mathematically expressed as follows.

$$x_i^{'(G)} = x_{r3}^{(G)} + F(x_{r1}^{(G)} - x_{r2}^{G}) \tag{10}$$

where, $i \neq r_1 \neq r_2 \neq r_3$ and r_1, r_2 and r_3 are randomly selected numbers from 1 to Np, F is a real positive constant scaling factor chosen within the range (0,2] and $x_i^{'(G)}$ represents the created mutant vector [18].

3) Create trial vectors: Trial vectors are randomly selected from the mutant vectors or from the population with respect to the value of a constant number called crossover constant CR or a random parameter q chosen once for each i. Crossover constant is chosen within the range [0,1]. The selection process is defined below.

$$x_i^{\text{trial}(G)} = \begin{cases} x_{ji}^{'(G)} & \text{if rand}(0,1) \leq (CR) \text{ or } j = q, \\ x_{ji}^{(G)} & \text{otherwise.} \end{cases} \tag{11}$$

4) Determine the next generation: In selection step, the algorithm decides whether the trial vector will be part of the next generation. This decision is done by comparing the fitness values of the trial vectors with the fitness values of the associated target vectors. This process can be expressed as follows.

$$x_i^{(G+1)} = \begin{cases} x_i^{\text{trial}(G)} & \text{if} f(x_i^{\text{trial}(G)}) \leq f(x_i^{(G)}), \\ x_i^{(G)} & \text{otherwise.} \end{cases} \tag{12}$$

5) Stop if a predetermined stopping criterion is met, otherwise go to step 2.

B. Application of Differential Evolution Algorithm to Branch Outage Problem

The proposed differential evolution algorithm for solving the local constrained optimization problem formulated for branch outage problem can be outlined as follows:

1) Run a power flow for pre-outage (initial) state and obtain initial load bus voltage magnitudes for the busses included in the bounded region.
2) Create Np different Q_{si} values by using $Q_{ijt} \times$ unifrnd$(0, 5)$ expression where, Q_{ijt} is the initial transferred reactive power between the outaged busses, and unifrnd$(0, 5)$ is a uniform random number between 0 and 5. Solve (13) for bus voltage magnitude increments and update bus voltage magnitudes.

3) Create Np new mutant Q_{si} vectors using (10) for all vectors in the population.
4) Generate Q_{si} trial vectors by using (11).
5) Evaluate the fitness function using (6) and determine the new generation.
6) Stop the calculation if one of the objective functions corresponding to the members of the generation (the best one) is smaller than a predetermined value or the maximum number of iterations is reached, otherwise go to step 2.

C. Harmony Search

Harmony search is a recently developed heuristic method [19], [20]. It has been applied to the economic dispatch problem [21] in electrical power systems so far. Other heuristic methods such as simulated annealing, tabu search, particle swarm optimization are naturally inspired techniques. Harmony search is inspired from the observation that the aim of the music to find the best harmony.

Harmony search algorithm comprises four basic steps:

1) Initialize the harmony memory,
2) Improvise a new harmony by using the harmony memory,
3) Update the harmony memory, and
4) If stopping criterion is not met, go to Step 2.

The first step of the harmony search algorithm is the initialization process. Harmony memory size (HMS), harmony memory considering rate (HMCR), pitch adjusting rate (PAR) parameters are chosen and the calculation is initialized. After parameter initialization, the first harmony is generated by using randomly created values in the specified range of variables. HMS is typically set between 10 and 100 [22]. If HMS is chosen as 10, initial HM will have ($10 \times$ number of variables) elements. Initial HM is sorted with respect to objective function values.

$$HM = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^{HMS} \end{bmatrix} \tag{13}$$

At the second step, a new harmony vector $x^{'} = (x_1^{'}, x_2^{'}, \cdots, x_N^{'})$ is generated with respect to memory considerations, pitch adjustments, and randomization [19]. These steps are explained in the following paragraphs.

In memory consideration, the value of the first decision variable is randomly selected from the elements at the same column of HM. The other decision values are chosen in the same manner. The usage of the values existing in HM itself, limits the solutions to the past history values of HM. Therefore, HMCR is defined to provide selection of the values from the extended solution space. HMCR varies between 0 and 1. Extension of the solution space by using HMCR can be stated as follows.

$$x_i^{'} \leftarrow \begin{cases} x_i^{'} & \in \{x_i^1, x_i^2, \cdots, x_i^{HMS}\} \text{with probability HMCR}, \\ x_i^{'} & \in X_i \text{with probability (1-HMCR)}. \end{cases} \tag{14}$$

PAR parameter is used to determine whether a component of the new harmony vector $x^{'} = (x_1^{'}, x_2^{'}, \cdots x_N^{'})$ should be pitch-adjusted or not. This process is shown below.

$$\text{Pitch adjust decision for } x_i^{'} \leftarrow \begin{cases} \text{Yes} & \text{with probability PAR}, \\ \text{No} & \text{with probability} \\ & (1 - PAR) \end{cases} \tag{15}$$

The value $(1 - PAR)$ sets the rate of doing nothing. If the pitch adjustment decision for $x_i^{'}$ is Yes, $x_i^{'}$ is changed as follows.

$$x_i^{'} \leftarrow x_i^{'} \pm \text{rand}() * bw \tag{16}$$

where rand() is a random number between 0 and 1, and bw is an arbitrary distance bandwidth.

At the third step, after having applied memory consideration pitch adjustment and random selection to each variable of the new harmony vector, the objective function corresponding to new harmony vector is compared with the objective function of the worst harmony in HM. If this new value is better than the worst one in HM then the new harmony is replaced with the worst harmony in HM.

At the final step, the algorithm terminates if stopping criterion is met or the process is restarted from step 3.

D. Application of Harmony Search Method to Branch Outage Problem

The application of the harmony search algorithm for solving the local constrained optimization problem formulated for branch outage problem can be outlined as follows:

1) Run a power flow for pre-outage (initial) state and obtain initial voltage magnitudes of the load buses included in the bounded region.
2) Create HMS different Q_{si} values using $Q_{ijt} \times \text{unifrnd}(0, 5)$ expression where, Q_{ijt} is the initial transferred reactive power between the outaged busses, and $\text{unifrnd}(0, 5)$ is a uniform random number between 0 and 5. Using (7), form $\triangle Q$ vectors and after solving the equation below, update the voltage magnitudes.
$$(B_b)^{-1} \triangle Q_b = \triangle V_b \tag{17}$$
3) Generate a new harmony vector $Q_{si}^{'}$ by using memory consideration, pitch adjustment and random selection as explained above.
4) Compare the value of obtained new harmony vector's objective function value using (6), with the worst function value in the harmony memory, if this new value is better than the worst one in HM the worst harmony is replaced with the new harmony.
5) Stop if the maximum number of improvisations is reached to the upper limit, otherwise go to step 2.

TABLE I
POST-OUTAGE VOLTAGE MAGNITUDES FOR OUTAGE OF THE BRANCH 7-9 IN IEEE 14 BUS TEST SYSTEM

Bus number	(AC)	(DE)	(HS)
4	1.017	1.017	1.018
7	1.067	1.07	1.07
9	1.029	1.035	1.035
10	1.028	1.034	1.034
14	1.018	1.022	1.022

TABLE II
SIMULATION ERRORS FOR OUTAGE OF THE BRANCH 7-9 IN IEEE 14 BUS TEST SYSTEM

Bus number	Error (HS) %	Error (DE) %	σ (HS)	σ (DE)
4	0.01	0.00	0.000	0.000
7	0.28	0.28	0.001	0.002
9	0.58	0.58	0.003	0.006
10	0.58	0.58	0.002	0.005
14	0.39	0.39	0.002	0.004

TABLE III
POST-OUTAGE VOLTAGE MAGNITUDES FOR OUTAGE OF THE BRANCH 4-6 IN IEEE 30 BUS TEST SYSTEM

Bus number	(AC)	(DE)	(HS)
3	0.994	0.993	0.994
4	0.994	0.993	0.994
6	0.960	0.959	0.960
7	0.957	0.957	0.957
8	0.947	0.946	0.947
9	0.974	0.973	0.974
10	0.982	0.982	0.982
12	0.989	0.988	0.988
28	0.963	0.963	0.963

TABLE IV
SIMULATION ERRORS FOR OUTAGE OF THE BRANCH 4-6 IN IEEE 30 BUS TEST SYSTEM

Bus number	Error (HS) %	Error (DE)
3	0.02	0.05
4	0.02	0.05
6	0.00	0.03
7	0.00	0.03
8	0.00	0.04
9	0.00	0.03
10	0.03	0.03
12	0.12	0.13
28	0.00	0.03

TABLE V
POST-OUTAGE VOLTAGE MAGNITUDES FOR OUTAGE OF THE BRANCH 12-13 IN IEEE 57 BUS TEST SYSTEM

Bus number	AC	DE	HS
10	0.984	0.984	0.984
11	0.961	0.960	0.961
12	1.015	1.015	1.015
13	0.955	0.954	0.954
14	0.953	0.952	0.952
15	0.978	0.977	0.978
16	1.013	1.013	1.013
17	1.017	1.017	1.017
49	1.019	1.017	1.018

IV. TESTS AND THE RESULTS

HS and DE algorithms developed for post outage voltage magnitude calculations are tested on IEEE 14, IEEE 30, IEEE 57, and IEEE 118 bus test systems. Open source electrical power system package Matpower [23] and Matlab are used as tools. All simulations are run on a laptop, that has a 2.20 GHz Core Duo CPU, and 2.00 GB Memory.

Due to the limited space, only critical outage results are illustrated here. Heavily loaded branch outages are preferred for IEEE 14, IEEE 30, and IEEE 57 bus test systems whereas an arbitrary branch outage is simulated for IEEE 118 test system. Post outage voltage magnitudes are determined by three methods; namely: by full AC load flow as a reference simulation, by using DE algorithm and by using HS algorithm. As being the most critical ones only the voltage magnitudes of the busses included in the bounded region are illustrated in Tables, I, III, V, VII. The remaining busses showing smaller errors, when compared with those of the busses in the bounded region are not included in the tables.

Post-outage voltage magnitudes for the outage of the branch connected between busses 7 and 9 in IEEE 14 bus test system are given in Table I. Post-outage voltage magnitudes for the outage of the branch connected between busses 4 and 6 for IEEE 30 bus test system are given in Table III. Post-outage voltage magnitudes for the outage of the branch connected between busses 12 and 13 in IEEE 57 bus test system are given in Table V. Finally, post-outage voltage magnitudes for the outage of the branch connected between busses 70 and 75 in IEEE 118 bus test system are given in Table VII.

Tables II, IV, VI, VIII illustrate absolute percentage errors for bus voltage magnitude simulations for the above simulations. Taking AC load flow simulations as reference, calculation method for absolute percentage errors is shown below.

$$\%\text{Error} = 100 \times \frac{\text{abs}(V_{AC} - V_{Method})}{V_{AC}} \tag{18}$$

In addition, since heuristic algorithms depend on the stochastic principles, standard deviation is used as an indicator of several performance critera. Therefore, standard deviations are illustrated in Table II for branch 7-9 outage in IEEE 14 bus test system. Standard deviations are less than 10^{-3} for the other outages and are not therefore given in the tables.

Mean cpu time values for the outages in test systems are illustrated in Table IX. Note that the values given in the table are the mean values for 18 single branch outages in IEEE 14 bus test system, 40 single branch outages in IEEE 30 bus test system, 75 single branch outages in IEEE 57 bus test system and 131 single branch outages in IEEE 118 bust test system. The biggest percentage error for the test systems is system for both HS and DE method. On the other hand, HS is faster than DE method as shown in table IX. It is also faster than the existing AC load flow method used in Matpower.

V. CONCLUSION

This paper extends our previous studies by introducing Harmony Search Algorithm for branch outage problem solution.

TABLE VI

SIMULATION ERRORS FOR OUTAGE OF THE BRANCH 12-13 IN IEEE 57 BUS TEST SYSTEM

Bus number	Error (HS) %	Error (DE) %
10	0.02	0.02
11	0.03	0.05
12	0.01	0.01
13	0.08	0.12
14	0.07	0.10
15	0.05	0.06
16	0.00	0.01
17	0.00	0.01
49	0.13	0.16

TABLE VII

POST-OUTAGE VOLTAGE MAGNITUDES FOR OUTAGE OF THE BRANCH 70-75 IN IEEE 118 BUS TEST SYSTEM

Bus number	AC	DE	HS
70	0.984	0.984	0.984
71	0.987	0.987	0.987
75	0.965	0.964	0.964
118	0.948	0.948	0.948

TABLE VIII

SIMULATION ERRORS FOR OUTAGE OF THE BRANCH 70-75 IN IEEE 118 BUS TEST SYSTEM

Bus number	Error (HS) %	Error (DE) %
70	0.00	0.00
71	0.00	0.00
75	0.34	0.33
118	0.19	0.18

TABLE IX

COMPARISON OF CALCULATION METHODS

Test System	IEEE-14	IEEE-30	IEEE-57	IEEE-118
Outage #	18	40	75	131
(DE)	0.087 cpu/s	0.061 cpu/s	0.152 cpu/s	0.090 cpu/s
(HS)	0.014 cpu/s	0.011 cpu/s	0.016 cpu/s	0.016 cpu/s
(AC)	0.043 cpu/s	0.051 cpu/s	0.059 cpu/s	0.069 cpu/s

The method is compared with Differential Evolution Algorithm as being the best of the previous ones. The comparison is done from the point of post outage voltage magnitude calculation effectiveness. Post outage voltage magnitudes determined by DE and HS methods are compared with each other both from the point of computational accuracy and calculation speed. Comparisons are done for several branch outages in well-known IEEE 14, IEEE 30, IEEE 57, and IEEE 118 test systems.

The results have shown that harmony search optimization method provides fast and accurate solutions. In addition, increasing the solution speed by parallel processing, harmony search seems to be an effective solution algorithms for branch outage problem.

REFERENCES

[1] A. J. Wood and B. F. Wollenberg, *Power Generation Operation and Control*, 2nd ed., New York, USA: Wiley, 1998.

[2] C. Y. Lee, and N. Chen, "Distribution factors of reactive power flow in transmission line and transformer outage studies," *IEEE Trans. Power Syst.*, vol. 17, pp. 194–200, Feb. 1992.

[3] M. Ilic, and A. Phadke, "Redistribution of reactive power flow in contingency analysis," *IEEE Trans. Power Syst.*, vol. 1, pp. 266–275, Aug. 1986.

[4] D. G. Taylor, and L. J. Maahs, "A reactive contingency analysis algorithm using MW and MWAR distribution factors", *IEEE Trans. Power Syst.*, vol. 6, pp. 349–355, Feb. 1991.

[5] A. Ozdemir, Y. J. Lim, and C. Singh, "Branch outage simulation for MWAR flows: bounded network solution", *IEEE Trans. Power Syst.*, vol. 18, pp. 1523–1528, Nov. 2003.

[6] O. Ceylan, A. Ozdemir, and H. Dag, "Branch outage solution using particle swarm optimization", *AUPEC 2008, Australasian Universities Power Engineering Conference*, Sydney, Australia, Dec. 2008.

[7] O. Ceylan, A. Ozdemir, and H. Dag, "Application of differential evolution method to branch outage problem", *GEM 2009, The 2009 International Conference on Genetic and Evolutionary Methods* , Las Vegas, USA, July 2009.

[8] R. Storn, and K. Price, "Differential Evolution- a simple and efficient adaptive scheme for global optimization over continuous spaces", Technical Report TR-95-012, ICSI, http://http.icsi.berkeley.edu/ storn/litera.html.

[9] R. Storn, and K. Price, "Minimizing the real functions of the ICEC 96 contest by differential evolution", *Int. Conf. on Evolutionary Computation*, Nagoya, Japan, May 1996.

[10] Y. G. Yang, Z. Y. Dong, and K. P. Wong, "A modified differential evolution algorithm with fitness sharing for power system planning", *IEEE Trans. Power Syst.*, vol. 23, pp. 514–522, May 2008.

[11] H. R. Cai, Y. C. Chung, and K. P. Wong, "Application of differential evolution algorithm for transient stability constrained optimal power flow", *IEEE Trans. Power Syst.*, vol. 23, pp. 719–728, May 2008.

[12] S. Kannan, S. M. R. Slochanal, S. Baskar, and P. Murugan, "Application and comparison of meta-heuristic techniques to generation expansion planning in the partially deregulated environment", *IET Generation, Transmission and Distribution*, vol. 1, pp. 111–118, Jan. 2007.

[13] S. Patra, S. K. Gowami, and B. Goswami, "A binary differential evolution algorithm for transmission and voltage constrained unit commitment", *Joint International Conference on Power System Technology and IEEE Power India Conference POWERCON 2008*, pp. 1–8, Oct. 2008.

[14] S. H. Goh, Z. Y. Dong, and T. K. Sana, "Locating voltage collapse points using evolutionary computation techniques", *IEEE Congress on Evolutionary Computation 2007. CEC 2007*, pp. 2923–2930, Sept. 2007.

[15] X. Zhang, W. Chen, C. Dai, and A. Guo, "Self adaptive differential evolution algorithm for reactive power optimization", *Fourth International Conference on Natural Computation 2008. ICNC '08*, vol.6, pp. 560–564, Oct. 2008.

[16] R. E. Perez-Guerrero,and J. R. Cedeno-Maldonado, "Economic power dispatch with non-smooth functions using differential evolution", *Proceedings of the 37th Annual North American Power Symposium 2005*, pp. 183–190, Oct. 2005.

[17] P. S. Manoharan, P. S. Kannan, S. Baskar, and M. W. Iruthayarajan, "Penalty parameter-less constraint handling scheme based evolutionary algorithm solutions to economic dispatch" *IET Generation, Transmission, and Distribution*, vol. 4., pp. 478–490, July 2008.

[18] K. V. Price, "Differential evolution: a fast and simple numerical optimizer", *Biennial Conference of the North American Fuzzy Information Processing Society*, pp. 524–527, June 1996.

[19] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search", Simulation, vol. 76-2, pp. 60–68, Feb. 2001.

[20] K. S. Lee, and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice", Comput. Methods. Appl. Mech. Engrg, vol. 194, pp. 3902–3933, Sept. 2005.

[21] A. Vasebi, M. Fesanghary, and S. M. T. Bathaee, "Combined heat and power economic dispatch by harmony search algorithm", International Journal of Electrical Power and Energy Systems, vol. 29-10, pp. 713–719, Dec. 2007.

[22] X. Z. Gao, X. Wang, and S. J. Osaka, "Harmony search methods for multi-modal and constrained optimization", Z. W. Geem(Ed.): Music-Inspired Harmony Search Algorithm, SCI vol. 191, pp. 39–51, Berlin Heidelberg. 2009.

[23] R. Zimmermann, and D. Gan, "Matpower Manual", USA: PSERC Cornell Univ. 1997.