Genetic Algorithm Variant based Effective Solutions for Economic Dispatch Problems

Samin Rastgoufard*, Sumaiya Iqbal†, Md Tamjidul Hoque†, Dimitrios Charalampidis*
Department of Electrical Engineering* and Computer Science†
University of New Orleans
New Orleans, LA, USA
Email: srastgou, siqbal1, thoque, dcharala@uno.edu,

***Abstract*---Economic dispatch (ED) is a nonconvex, nonlinear and nondifferentiable optimization problem in power systems to improve the cost effectiveness of power generation. In order to solve the complex ED problem with prohibited operating zones (POZ), valve point effect (VPE), multi-fuel options (MFO), and transmission line losses, genetic algorithm (GA) variant based methods are proposed. In particular, the proposed GA variants are beeder GA (BGA), fast navigating GA (FNGA), twin removal GA (TRGA), kite GA (KGA), and united GA (UGA). Three benchmark test systems having 6-units, 10-units, and 15-units are used to study the efficacy of the proposed algorithms for solving the ED problem. The simulation results show how each GA method performs for each case study, and which of the variants outperform other algorithms presented in recent literature.**

***Index Terms*---Economic dispatch, variants of GA**

I. Introduction

The massive consumption of fossil fuels has resulted in a dramatic reduction of these resources. The power generation required for the operation of power systems is one of the leading causes of fossil fuel consumption. In order to reduce consumption by optimal usage of fuel resources, power should be generated at the lowest possible cost, while still meeting a known power demand and satisfying various constraints. This can be achieved by solving the economic dispatch (ED) problem which finds the best feasible power generation with minimum fuel cost while satisfying the generation constraints of the power units [1], [2]. Several classical and numerical methods, such as fast lambda iteration, Lagrange relaxation (LR), linear programming, and gradient methods have been traditionally used for solving the ED problem [3], [4], [5]. Often, these methods solve the ED problem by simplifying or ignoring some constraints such as the prohibited operating zones (POZ) of generators, the ramp rate limits [1], the valve-point effect (VPE) [6], and the multi-fuel options (MFO) [7]. Incorporating these real factors increases the complexity of the ED problem, which becomes a non-convex, non-continuous, and non-differentiable constrained optimization problem [2]. In general, traditional methods often fail to successfully solve the non-simplified ED problem.

Recently, several evolutionary and metaheuristic algorithms have been used in the literature to solve the ED problem and to overcome the difficulties of conventional optimization methods. Some of the advanced evolutionary algorithms used include Genetic Algorithm (GA) [6], [8], Tabu Search (TA) [9], Particle Swarm Optimization (PSO) [1], Differential Evolution (DE) [10], Ant Colony Optimization (ACO) [11], Harmony Search [12], Artificial Bee Colony (ABC) [13], and the Social Spider Algorithm (SSA) [14], [15]. Furthermore, hybrid algorithms have also been employed to solve the ED problem [16]--[18].

In this paper, advanced variants of GAs[1] [19] are used to solve the ED problem. The idea of GA was inspired by Darwin's theory of evolution and was first invented by John Holland [20]. GA, in its implementation, starts with a set of individuals (initial population of candidate solutions) that are evolved over consecutive generations (epochs) through selection and variation to solve an optimization problem. In GA, individual problem-solutions, to which the values of the solution variables are encoded, are referred to as chromosomes. GA evolves through the natural adaptation process in which the fitter chromosomes tend to survive, breed, and propagate their genetic information to the future generations. Our proposed GA variants, with integrated advanced and innovative strategies, help produce competitive solutions. Though the GA variants were separately shown, a solution-suite can be easily formulated to have the combined benefits of our proposed GA variants. In this paper, the results obtained by different advanced GA variants are compared for solving the ED problem. It is also shown that for some case studies the kite GA (KGA) and twin removal GA (TRGA) outperform the results obtained by some recently proposed evolutionary and metaheuristic techniques, such as the modified social spider algorithm (MSSA) [15] and the backtracking search algorithm BSA [7].

The rest of the paper is organized as follows. The mathematical formulation of the ED problem is presented in section II, and an overview of various GA algorithms in section III. Section IV presents experimental results. Finally, section V closes with some concluding remarks.

II. Mathematical Formulation of the Economic Dispatch Problem

ED is a sub-problem of unit commitment (UC) as well as a nonlinear optimization problem with various constraints. The

[1] GA variant code and data are here: http://cs.uno.edu/~tamjid/Software/GAVariant/Code_Data.zip

objective in solving the ED problem is to find the optimum power generated by all generators in a power system in order to minimize the total fuel cost of the system. At the same time, a number of constraints such as load demand, spinning reverse capacity, ramp rate limits, and generator prohibited operating zones need to be satisfied.

A. Objective Function and Constraints

The ED objective function which needs to be minimized can be expressed as follows:

$$\sum_{j=1}^{n} F_j(P_j) = \sum_{j=1}^{n} a_j + b_j P_j + c_j P_j^2 \tag{1}$$

where, $F_j(P_j)$ is the fuel cost function of the j^{th} power unit, P_j, n is the total number of power units, and a, b, c represent constant coefficients associated with the fuel cost.

In practice, the valve-point effect (VPE) of steam-power plants exhibits ripples which can be modeled as a recurring rectified sinusoid. Therefore, the VPE effect can be incorporated in the ED objective function by modifying $F_j(P_j)$ as follows:

$$F_j(P_j) = a_j + b_j P_j + c_j P_j^2 + |e_j sin(f_j(P_j^{min}) - P_j| \tag{2}$$

where P_j^{min} is the minimum power generated by the j^{th} power unit. Moreover, e_j and f_j are constant coefficients describing the VPE [6]. Furthermore, some power units can operate using multiple fuels with different associated costs [14]. Based on the power generation requirements, the fuel with minimum cost should be selected for each unit. Consequently, the fuel cost function of the j^{th} power unit can be modified to incorporate the multi-fuel option as follows:

$$\begin{aligned} F_j(P_j) = \min_{k=1,\ldots,m} (a_{j,k} + b_{j,k} P_j + c_{j,k} P_j^2 \\ + |e_{j,k} sin(f_{j,k}(P_j^{min}) - P_j|) \end{aligned} \tag{3}$$

where, m is the total number of fuel options while $a_{j,k}, b_{j,k}, c_{j,k}, e_{j,k}$ and $f_{j,k}$ are the fuel cost coefficients of the j^{th} power unit's k^{th} fuel option [7].

In addition to defining the ED objective function, certain constraints should be considered. First, the total generated power has to be equal to the power demand plus any power losses. The requirement to maintain the active power balance of the system can be expressed as follows:

$$\sum_{j=1}^{n} P_j = P_D + P_L \tag{4}$$

where P_D is the load demand and P_L is the line loss. In particular, P_L can be calculated by:

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_j B_{ij} P_j + \sum_{i=1}^{n} B_{i0} P_i + B_{00} \tag{5}$$

where B_{ij} are the line loss coefficients [1]. Each power unit may be restricted by additional constraints depending on the problem at hand. In particular, the power generated by each power unit may not drop below a minimum value, P_j^{min}, or exceed a maximum value, P_j^{max}. Moreover, the ramp rate limits may restrict the power generated by the j^{th} unit as follows:

$$max(P_j^{min}, P_{j0} - LR_j) \leq P_j \leq min(P_j^{max}, P_{j0} + UR_j) \tag{6}$$

where P_{j0} is the previous interval output power, while LR_j and UR_j are the lower and upper ramp rate limits of the j^{th} unit, respectively.

Another constraint considered in this work is the generator prohibited operating zone (POZ). The POZ represents an infeasible or forbidden operation range of a power unit. Eq. 7 introduces this additional constraint considering z_j POZs for the j^{th} power unit.

$$P_j \in [P_j^{min}, P_j^{l,1}] \bigcup [P_j^{u,1}, P_j^{l,2}] \bigcup \ldots \bigcup [P_j^{u,z_j}, P_j^{max}] \tag{7}$$

where $P_j^{l,k}$ and $P_j^{u,k}$ are, respectively, the lower and upper bounds of the j^{th} power unit's k^{th} POZ and z is the total number of POZs [1], [14].

III. Variants of Genetic Algorithms

In this section, we outline five different genetic algorithm (GA) variants that are implemented under this work to solve the economic dispatch (ED) problem. We used real-value encoding for this study. Thus, each chromosome is composed of n number (power units) of real values. During evolution, we quantify the fitness values of the chromosomes by evaluating the objective function under consideration. Subsequently, we select candidate chromosomes with relatively higher fitness using the roulette wheel algorithm. The candidate chromosomes are tweaked using operators such as crossover and mutation, and are passed on to the next generation. GA is an iterative algorithm that runs for a number of generations (epochs), which, in our implementation, is controlled by a predefined number of objective function evaluations ($Eval_{max}$), and terminates if the desirable solution is found. The five GAs that are studied in this work use different genetic operators, including some recently introduced techniques, to produce new solutions for the next generation population.

A. Breeder Genetic Algorithm (BGA)

The breeder GA (BGA) [21], [22] includes three genetic operators during evolution: *elitism*, *uniform crossover* and *uniform single-point mutation*. We outline the procedure in **Algorithm 1**. Elitism is an operator that segregates a subset of individuals from the current population at the beginning of a generation. This predefined proportion of chromosomes or individuals are relatively fitter than the others in the population, thus called *elites*. Elitism aims at the survival of these highly fitted individuals to guarantee non-decreasing GA performance over time. Both crossover and mutation can create or destroy the genetic material of a chromosome. Yet, elites are passed on to the next generation without any modification. Thus, elitism allows genetic material to be kept intact through evolution.

After elitism, we use roulette wheel algorithm to select two parent chromosomes, P_1 and P_2, and apply uniform crossover

Algorithm 1: Procedure of BGA

Initialize population of individuals randomly;
while *generation count* $\leq Eval_{max}$ **do**
  Evaluate fitness of all chromosomes ;
  Perform **elitism**;
  Perform **selection** of parents for crossover;
  Perform **uniform crossover** (AmC);
  Perform **uniform single-point mutation**;
end

Algorithm 2: Procedure of FNGA

Initialize population of individuals randomly;
Initialize AM with the best available solution (chromosome);
while *generation count* $\leq Eval_{max}$ **do**
  Evaluate fitness of all chromosomes ;
  Perform global **elitism**;
  Perform **selection** of parents for crossover;
  Perform **AM-based single-point crossover** (AmC);
  Perform **uniform single-point mutation**;
end

on the parents to generate two offspring chromosomes, O_1 and O_2. In uniform crossover, the parents contribute to the offspring chromosomes in the gene (or variable) level, not in the segment level. We define a mixing ratio (α) for each variable in the chromosome which is sampled from a uniform distribution within $[-0.1, +1.1]$. Then, $O_1 = \alpha P_1 + (1 - \alpha)P_2$ and $O_2 = \alpha P_2 + (1 - \alpha)P_1$. Finally, we apply single-point uniform mutation on randomly selected candidates for mutation. Mutation is the process of randomly changing individuals of the current population to produce new individuals for the next generation. Mutation emulates the process of having random genetic diversity in nature. In uniform mutation process, we replace the value of one randomly chosen variable, also referred as mutation point ($1 \leq mutation_p \leq d$, d is the number of variables in a chromosome) in the mutation candidate with a uniform random value selected between the minimum and maximum variable values in that chromosome.

B. Fast Navigating Genetic Algorithm (FNGA)

The FNGA is a recently introduced GA variant [23] that uses *elitism*, a modified single-point crossover called *AM-based crossover*, and *single-point mutation* to introduce variation in the process of evolution, as outlined in **Algorithm 2**. In the beginning of the evolution, we first segregate a predefined number of fitter chromosomes as elites and pass these individuals to the next generation population without modification. After that, we apply the modified single-point crossover operation. In classical single-point crossover, one randomly selected locus is considered as the crossover point ($1 \leq corssover_p < (d - 1)$) and the parts of the two parent chromosomes beyond $crossover_p$ are exchanged to produced two offspring chromosomes.

FNGA uses Associative Memory (AM)-based single-point crossover (AmC) [23] to enhance the constructive exploitation power of classical crossover. AM consists of two triangular memories that store the current best individual at all crossover points [23]. Unlike the classical crossover that blindly swaps the parts of parents, AmC produces two different offspring candidates for each offspring position: one taking the segment from other participating parent (classical crossover) and other using the segment available in AM. AmC takes feedback from the search space by evaluating the two potential offspring candidates and keeps the better one. Thus, the search for better solution variable in AmC operation is not limited to the other parent individual, but is rather extended to the current best solution that is stored in memory, and applied adaptively only if useful. Moreover, the AM is updated if a better solution is found from the mating partner. Thus, AM can essentially contain the best-performing variables of a solution at different crossover point through consecutive generations. In addition to the AM-based crossover, FNGA uses single-point uniform mutation to ensure diversify in its population, as described in **Section III-A**. In our implementation of these different GAs, we did not consider elites as mutation candidates to always preserve the fitter solutions in the population.

C. Twin Removal Genetic Algorithm (TRGA)

Twin Removal (TR) is an improved diversification operation, introduced in [24] for GA. The variant TRGA applies *elitism*, *single-point crossover*, *uniform single-point mutation* and *twin removal* as genetic operators, as outlined in **Algorithm 3**. It has been discussed in [25] that GA tends to produce similar chromosomes called *twins* in the population as the generation proceeds. The growth of such correlated twins inevitably debilitates the impact of mutation in producing new random solutions when the underlying landscape is complex. The TR operator can back up the reduced exploration power of mutation due to the similar chromosomes (twins) by introducing new random chromosomes in place of the similar chromosomes [26].

In TRGA, we first execute elitism (*see* **Section III-A**) and then the classical single-point crossover on the selected parents to produce offsprings for the next generation. Thus, the parents contribute to the offspring chromosomes at the segment level where the parents exchange the subset of their variables at the $crossover_p$ (*see* Section III-B) to generate the offspring. The crossover operation is followed by the single-point uniform mutation operation (*see* **Section III-A**). Finally, we exercise the TR operator on the next generation population. The TR operation is controlled by the *Chromosome Correlation Factor* (CCF) which defines the allowable similarity between chromosomes. In our implementation, we count the number of loci of the chromosome pair under comparison with identical values. For this application, we set CCF = 95% and therefore, if the similarity between two chromosomes is higher or equal than 95%, we replace the chromosome of relatively lower fitness with a new randomly generated chromosome. The performance comparison of TRGA with BGA and FNGA highlights the usefulness of TR operator (**Section IV**).

D. Kite Genetic Algorithm (KGA)

The KGA [23] combines the enhanced exploitation capacity of AM-based crossover (*see* **Section III-B**) and improved diversification power of TR (*see* **Section III-C**). Thus, KGA has advantages over both FNGA and TRGA in having balanced

Algorithm 3: Procedure of TRGA

Initialize population of individuals randomly;
while *generation count* $\leq Eval_{max}$ **do**
    Evaluate fitness of all chromosomes ;
    Perform **elitism**;
    Perform **selection** of parents for crossover;
    Perform **single-point crossover**;
    Perform **uniform single-point mutation**;
    Perform **twin removal** (TR);
end

Algorithm 4: Procedure of KGA

Initialize population of individuals randomly;
Initialize AM with the best available solution (chromosome);
while *generation count* $\leq Eval_{max}$ **do**
    Evaluate fitness of all chromosomes ;
    Perform **elitism**;
    Perform **selection** of parents for crossover;
    Perform **AM-based single-point crossover** (AmC);
    Perform **uniform single-point mutation**;
    Perform **twin removal** (TR);
end

exploitation and exploration in every generation. We show the flow of operations under procedure KGA in **Algorithm 4**. It has been laid down by the Schemata Theorem [20] that GA works by prioritizing and sustaining instances of schema with above-average-fitness. The AM-based crossover (AmC) ensures this by guiding the crossover towards better schema stored in the AM. Thus, it is more likely to exhibit similarity between chromosomes that can make the GA search static. TR plays a complimentary role by reducing similarity and introducing new random solutions. Therefore, KGA employs *elitism*, *AmC*, *single-point uniform mutation* and *TR* in one generation.

E. Unified Genetic Algorithm (UGA)

The UGA integrates a very recently proposed genetic operator in the literature, homologous gene replacement (hGR) [27] with the operators used in KGA. Thus, the unified GA (UGA) includes four genetic operators during evolution: *elitism with hGR*, *AM-based single-point crossover*, *uniform single-point mutation* and *twin removal*, summarized in **Algorithm 5**. The hGR works on the genes (or variables) of elites chromosomes and the elites, if possible. This operator is motivated to mimic the natural phenomena that the combination of good genes can form a fitter chromosome. The working principle of hGR operator is to identify the best gene (or variable) of each elite chromosome and replace the relatively weaker genes of the corresponding elite if these replacement improves that elite's fitness. Therefore, hGR involves the evaluation of relative fitness of the local variables to determine the best gene of an elite. To quantify the relative fitness of a gene in a chromosome (one variable in a solution), we assign a common base value equal to 0.5 to other variables to generalize the effect of other variables. The application of hGR is controlled adaptively by taking feedback from the search space to ensure non-decreasing performance of GA.

The benefits of hGR are first benchmarked in [28] where the authors used classical single-point crossover operation.

Algorithm 5: Procedure of UGA

Initialize population of individuals randomly;
Initialize AM with the best available solution (chromosome);
while *generation count* $\leq Eval_{max}$ **do**
    Evaluate fitness of all chromosomes ;
    Perform **elitism with hGR**;
    Perform **selection** of parents for crossover;
    Perform **AM-based single-point crossover** (AmC);
    Perform **uniform single-point mutation**;
    Perform **twin removal** (TR);
end

TABLE I: Results of GA variants for 6-units

Method	Best ($/hr)	Mean ($/hr)	Med. ($/hr)	Std.
BGA	15449.90979	15450.12343	15450.05756	0.193968201
FNGA	15449.96906	15451.09826	15450.77771	1.021224104
KGA	**15449.89994**	**15449.92394**	**15449.92245**	**0.017044814**
TRGA	15449.91319	15450.27628	15450.12628	0.32292185
UGA	15449.93556	15450.24261	15450.16914	0.25539524

However, in this GA variant, we employed AM-based single-point crossover (*see* **Section III-B**) to utilize its improved intensification capacity. For diversification, we used both mutation (*see* **Section III-A**) and TR operator (*see* **Section III-C**).

IV. Results

A. Case Studies

Three popular benchmarks, namely the IEEE 6-unit, 15-unit, and 10-unit systems, were used to evaluate the performance of GA variants. All algorithms were implemented in MATLAB for a population size of 50. The elite rate, crossover rate, and mutation rate were set to 10%, 80%, and 10%, respectively. Each algorithm was iterated for a maximum of $Eval_{max} = n \times 10^5$ function evaluations and was repeated 25 times.

For the 6-unit and 15-unit systems, the total load demand is 1263 MW and 2630 MW, respectively [1]. For these two cases, power balance, generator limits, ramp rate limits, and POZ (all described in Section II) are used as constraints. The system coefficients, the line loss coefficients, and the POZs are available in [1]. For the 10-unit system, VPE and MFO are used as constraints. For this scenario, the load demand is 2700 MW. The system coefficients are available in [7].

B. Discussions

Tables I, II, and III present the best, mean, median, and standard deviation of the cost ($\$/hr$) provided by the GA variants for the three case studies. Table IV presents detailed results regarding the best cost and the associated solution (unit powers) determined by the GA variants, and also by MSSA

TABLE II: Results of GA variants for 15-units

Method	Best ($/hr)	Mean ($/hr)	Med. ($/hr)	Std.
BGA	32712.03	32715.95	32715.12	2.56917
FNGA	32706.7	32717.3	32714.79	8.086838
KGA	**32704.81**	**32706.77**	32706.49	**1.593172**
TRGA	**32704.53**	32707.32	**32706.27**	2.896268
UGA	32705.52	32708.3	32707.58	2.346371

TABLE III: Results of GA variants for 10-units

Method	Best ($/hr)	Mean ($/hr)	Med. ($/hr)	Std.
BGA	623.9761	624.31287	625.13249	1.23677
FNGA	623.9432	624.11104	624.15378	0.1817
KGA	**623.7736**	**623.85665**	**623.87061**	0.09831
TRGA	623.8951	623.8879	623.90404	0.0855
UGA	623.8863	623.8819	623.88564	**0.07753**

[15], ASPSO [30] and BSA [7] for the 6-unit case study. Similarly, Table V presents results for the GA variants, and also CBPSO-RVM [31], BSA [7] for the 15-unit case study. The convergence plots of the GA variants for this case study are illustrated in Fig. 1. Finally, Table VI presents results for the GA variants, as well as for CBPSO-RVM [31] and BSA [7], for the 10-unit system. The convergence plots of the GA variants for this case study are presented in Fig. 2. The 10-unit case study took larger number of iterations to converge because it has multi-fuel options.

In summary, experimental results have revealed that KGA is the most consistent GA variant, both in terms of best cost and mean cost, although TRGA, FNGA, and UGA are also competitive. With respect to convergence, UGA was confirmed to be the fastest GA variant. In terms of the best cost, KGA outperformed BSA and CBPSO-RVM for the 10-unit case, and performed similarly to BSA and MSSA for the 6-unit case. BSA appears to exhibit a slight advantage over the GA variants for the 15-unit system.

V. Conclusions

In this work, several GA variants were employed to solve the ED problem. All GA variants, as well as other algorithms tested, provide a seemingly similar best cost. Nevertheless, even small savings of $0.1 - 0.2/h$ for a 10-unit system may result in considerable savings for a significantly larger system over a long period of time. In general, KGA appears to be the most consistent of the GA variants for this problem, both in terms of the best cost as well as the mean cost. The GA variants tested in this work were proven to be strong competitors to other ED solutions such as MSSA and BSA.

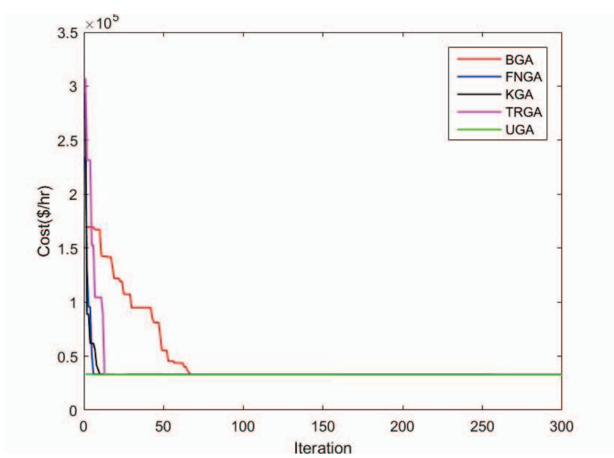

Fig. 1: Convergence plots of GA variants for 15-units

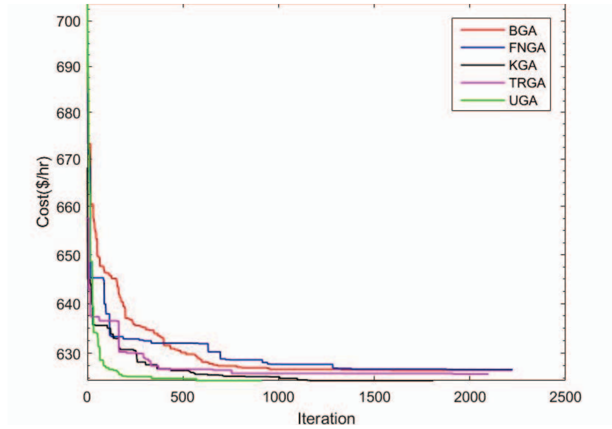

Fig. 2: Convergence plots of GA variants for 10-units

References

[1] Z.L. Gaing, "Particle Swarm Optimization to Solving the Economic Dispatch Considering the Generator Constraints," *IEEE Trans. Power Syst.* vol. 18, no. 3, pp. 1187-1195, 2003.

[2] V. Jadoun, N. Gupta, K.R. Niazi, A. Swarnkar, "Modulated particle swarm optimization for economic emission dispatch," *Electr. Power Energy Syst.* vol. 73, pp. 80-88, 2015.

[3] J. Dodu, P. Martin, A. Merlin, J. Pouget, "An optimal formulation and solution of short-range operating problems for a power system with flow constraints," *Proc. IEEE*, vol. 60, no. 2, pp. 54-63, 1972.

[4] J. Y. Fan, L. Zhang, "Real-time economic dispatch with line flow and emission constraints using quadratic programming," *IEEE Trans. Power Syst.*, vol. 13, no. 2, pp. 320-325, 1998.

[5] R. M. Chen, "Application of the fast Newton-Raphson economic dispatch and reactive power/voltage dispatch by sensitivity factors to optimal power flow," *IEEE Trans. Energy Convers*, vol. 10, no. 2, pp. 293-301, 1995.

[6] D.C. Walters, G.B. Sheble, "Genetic algorithm solution of economic dispatch with valve point loadings," *IEEE Trans. Power Syst.*, vol. 8, no.4, pp. 1325-1332, 1993.

[7] M. Modiri-Delshad, S. Hr. Aghay Kaboli, E.Taslimi-Renani, N. Abd Rahim, "Backtracking search algorithm for solving economic dispatch problems with valve-point effects and multiple fuel options," *Energy*, vol. 116, pp. 637-649, 2016.

[8] C.L. Chiang, "Genetic-based algorithm for power economic load dispatch," *IET Gen. Transm. Distrib.*, vol. 1, no. 2, pp. 261-269, 2007.

[9] W.M. Lin, F.S. Cheng, M.T. Tsay, "An improved Tabu search for economic dispatch with multiple minima," *IEEE Trans. Power Syst.*, vol. 17, no. 1, pp. 108-112, 2002.

[10] N. Noman, H. Iba, "Differential evolution for economic load dispatch problems," *Electr. Power Syst. Res.*, vol. 78, no. 3, pp. 1322-1331, 2008.

[11] S. Pothiya, I. Ngamroo, W. Kongprawechnon, "Ant colony optimisation for economic dispatch problem with non-smooth cost functions," *Int. J. Electr. Power Energy Syst.*, vol. 32, no. 5, pp. 478-87, 2010.

[12] M. Fesanghary, MM. Ardehali, "A novel meta-heuristic optimization methodology for solving various types of economic dispatch problem," *Energy*, vol. 34, no. 6, pp. 757-766, 2006.

[13] Y. Labbi, D. Attous, B. Mahdad, "Artificial bee colony optimization for economic dispatch with valve point effect," *Frontiers in Energy*, vol. 8, no. 4, pp. 449-458, 2014.

[14] J. Yu and V. Li, "A social spider algorithm for solving the non-convex economic load dispatch problem," *Neurocomputing*, vol. 171, pp. 955-965, 2016.

[15] W.T. Elsayed, Y.G. Hegazy, F.M. Bendary, M.S. El-bages, "Modified social spider algorithm for solving the economic dispatch problem," *Engineering Science and Technology, an International Journal,* vol. 19, no. 4, pp. 1672-1681, 2016.

[16] S. Duman, N. Yorukeren, IH. Altas, "A novel modified hybrid PSOGSA based on fuzzy logic for non-convex economic dispatch problem with valve-point effect," *Electr. Power Energy Syst.*, 2015

[17] A. Bhattacharya, P.K. Chattopadhyay, "Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch," *IEEE Trans. Power Syst,* vol. 25, no. 4, pp. 1955-1964, 2010.

TABLE IV: Comparison results for the 6-unit system

Generation	BGA	FNGA	KGA	TRGA	UGA	MSSA	APSO	BSA
P1 (MW)	447.5560503	448.5251124	447.6611636	447.0228188	447.4926549	447.5029	446.66857	447.4902
P2 (MW)	172.9237691	173.6749594	173.5994024	172.7145796	173.8888666	173.3186	173.155594	173.3308
P3 (MW)	263.7451965	264.9752614	262.9094742	264.5056047	262.8151961	263.463	262.825958	263.4559
P4 (MW)	139.4709937	138.6599312	139.1772992	138.975938	138.1555654	139.0656	143.468614	139.0602
P5 (MW)	164.8319461	164.8087921	165.5761515	165.7457395	166.1256904	165.473	163.91395	165.4804
P6 (MW)	87.41794403	85.32395972	87.03216628	87.00090973	87.50021122	87.1349	85.343745	87.1409
Total Generations (MW)	1275.9459	1275.968016	1275.955657	1275.96559	1275.978185	1275.958	1275.37643	1275.9583
PL (MW)	12.94541391	12.96764091	12.95558197	12.96539486	12.97662296	12.958	12.421628	12.9583
Total generation cost ($/h)	15449.90979	15449.96906	**15449.89994**	15449.91319	15449.93556	**15449.8995**	15449.99	**15449.8995**

TABLE V: Comparison results for the 15-unit system

Generation	BGA	FNGA	KGA	TRGA	UGA	CBPSO-RVM	BSA
P1 (MW)	454.9993	455.0000	455.0000	455.0000	454.9889	455.0000	455.0000
P2 (MW)	379.5807	380.0000	380.0000	380.0000	380.0000	380.0100	380.0000
P3 (MW)	129.9073	129.9979	130.0000	130.0000	130.0000	130.0000	130.0000
P4 (MW)	129.9933	129.9943	130.0000	130.0000	130.0000	126.5228	130.0000
P5 (MW)	169.9111	168.8164	170.0000	169.9735	169.9876	170.1312	170.0000
P6 (MW)	459.9703	459.9957	460.0000	460.0000	460.0000	460.0000	460.0000
P7 (MW)	429.9738	430.0000	430.0000	430.0000	430.0000	428.2836	430.0000
P8 (MW)	79.8044	61.96802	77.0887	69.3805	60.0000	60.0000	71.6368
P9 (MW)	82.91083	70.05567	52.94777	61.29818	77.11828	25.0000	59.0234
P10 (MW)	127.6979	160.0000	160.0000	159.9730	153.4569	159.7893	160.0000
P11 (MW)	79.58085	79.99906	80.0000	80.0000	80.0000	80.0000	80.0000
P12 (MW)	79.58233	79.75555	80.0000	80.0000	80.0000	80.0000	80.0000
P13 (MW)	25.18185	25.0000	25.00248	25.02635	25.0000	33.7037	25.0001
P14 (MW)	16.06463	15.0000	15.0000	15.0000	15.0000	55.0000	15.0001
P15 (MW)	15.21293	15.11412	15.60567	15.0000	15.0000	15.0000	15.0005
Total Generations (MW)	2660.371	2660.697	2660.645	2660.6520	2660.552	2658.323	2660.661
PL (MW)	30.37138	30.69576	30.64399	30.65153	30.54953	28.36553	30.6609
Total generation cost ($/h)	32712.03	32706.7	**32704.81**	**32704.53**	32705.52	32976.68	**32704.45**

TABLE VI: Comparison results for the 10-unit system

Generation	BGA	FNGA	KGA	TRGA	UGA	CBPSO-RVM	BSA
P1(MW)	218.2629	218.1646	217.188	214.4453	217.9038	219.2073	218.5777
P2(MW)	213.1529	209.9432	212.17	212.4232	212.6456	210.2203	211.2153
P3(MW)	282.6736	283.5296	277.7917	278.8647	284.5811	278.5456	279.5619
P4(MW)	241.2989	241.8433	239.4108	240.2289	239.5394	239.3704	239.5024
P5(MW)	284.2452	281.9754	278.7414	276.1498	276.3653	276.412	279.9724
P6(MW)	236.8393	242.0798	240.4689	240.5948	237.1091	240.5797	241.1174
P7(MW)	291.8698	287.0128	287.721	289.7905	290.8168	292.3267	289.7965
P8(MW)	235.7892	242.7772	240.359	241.7059	239.5454	237.7557	240.5785
P9(MW)	416.8246	423.4688	429.5014	427.3029	427.7358	429.4008	426.8873
P10(MW)	279.0435	269.2094	276.6584	278.5045	273.7903	276.1815	272.7907
Total Generations(MW)	2700.000	2700.004	2700.011	2700.010	2700.033	2700.000	2700.0001
Total generation cost($/h)	623.9761	623.9432	**623.7736**	623.8951	623.8863	623.9588	623.9016

[18] B. Mahdad, K. Srairi K, "Interactive gravitational search algorithm and pattern search algorithms for practical dynamic economic dispatch," *Int. Trans. Electr. Energy Syst.*, vol. 25, pp. 2289-2309, 2015.

[19] S. Iqbal, M.T. Hoque, "AMLGA: A Genetic Algorithm with Adaptive and Memory-Assisted Local Operators", *Technical paper TR2016/2*, http://cs.uno.edu/~tamjid/TechReport/AMLGA_TR20162.pdf.

[20] J. Holland, "Adaptation in natural and artificial systems," *Univ. of Michigan Press*, pp. 1975-1992.

[21] H. M´uhlenbein and D. Schlierkamp-Voosen, "The science of breeding and its application to the breeder genetic algorithm (BGA)," *Evolutionary Computation*, vol. 1, no. 1, pp. 25-49, 1993.

[22] H. M´uhlenbein and D. Schlierkamp-Voosen, "The science of breeding and its application to the breeder genetic algorithm (BGA)," *Evolutionary Computation*, vol. 1, no. 4, pp. 335-60, 1993.

[23] M.T. Hoque and S. Iqbal, "Genetic Algorithm based Improved Sampling for Protein Structure Prediction," *International Journal of Bio-Inspired Computation*, vol. 9, no. 3, pp. 129-141, 2017.

[24] M.T. Hoque, M. Chetty, L. Dooley, "Critical Analysis of the Schemata Theorem: The Impact of Twins and the Effect in the Prediction of Protein Folding Using Lattice Model," *GSIT, MONASH University*, 2005.

[25] M.T. Hoque, M. Chetty, L. Dooley, "Generalized schemata theorem incorporating twin removal for protein structure prediction," *Pattern Recognition in Bioinformatics*, pp. 84-97, 2007.

[26] M.T. Hoque, M. Chetty, A. Lewis, A. Sattar, "Twin removal in genetic algorithms for protein structure prediction using low-resolution model," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 1, pp. 234-245, 2011.

[27] S. Iqbal, and M.T. Hoque MT, "A Homologous Gene Replacement based Genetic Algorithm," *Proceedings of Genetic and Evolutionary Computation Conference Companion*, ACM, 2016.

[28] S. Iqbal S, and M.T. Hoque MT, "hGRGA: A scalable genetic algorithm using homologous gene schema replacement," *Swarm and evolutionary computation*, vol. 34, pp. 33-49, 2017.

[29] D.C Secui, "A modified Symbiotic Organisms Search algorithm for large scale economic dispatch problem with valve-point effects," *Energy*, vol. 113, pp. 366-384, 2016.

[30] B.K. Panigrahi, V. Ravikumari, Sanjoy Das, "Adaptive particle swarm optimization approach for static and dynamic economic load dispatch," *Energy Conversion and Management*, vol. 49, pp. 1407-1415, 2008.

[31] H. Lu, P. Sriyanyong, YH. Song, "Experimental study of a new hybrid PSO with mutation for economic dispatch with non-smooth cost function," *Int. J. Electr. Power Energy Syst.*, vol. 32, no. 9, pp. 921-935, 2010.