Bat Algorithm Approaches for Solving the Combined Economic and Emission Dispatch Problem

Dimitrios Gonidakis, Aristidis Vlachos
Department of Informatics
University of Piraeus
80, M. Karaoli & A. Dimitriou St., 18534 Piraeus,
GREECE

ABSTRACT

Combined economic and emission dispatch (CEED) is a multi-objective optimization problem aim of which is the simultaneous minimization of operating cost and pollutant emission by allocating generation among thermal units of an electric power system. Objective functions are conflicted and several equality and inequality constraints must be satisfied. This paper uses two bat algorithm based approaches for solving CEED. One of them hybridizes bat algorithm with differential evolution strategies while the other one inserts a mutation operator into the original bat algorithm. Both methods are applied to a 10-generator sample power system. Numerical results from the proposed algorithms are compared to those obtained by other techniques in recent literature.

General Terms

mathematical programming; numerical optimization; metaheuristics; algorithms

Keywords

combined economic and emission dispatch; hybrid bat algorithm; mutation operator; multi-objective optimization

1. INTRODUCTION

Modern power generating systems are very complex and include nonlinear characteristics. Economic load dispatch (ELD) is a fundamental issue for a power system as it intends to minimize the operation cost. This problem can be handled by reaching the optimal generating power combination of the thermal units while satisfying a number of constraints.

However, ELD is not the only major problem. Electric power organizations develop strategies to reduce the atmospheric pollution comes from power plants. Use of fossil fuel causes release of sulphur dioxide (SO_2), nitrogen oxides (NO_x) and carbon dioxide (CO_2). These gassy pollutants are responsible for dangerous natural phenomena such as acid rain which harm plants, animals and humans. Thus, the problem of emission dispatch (ED) is considered as another basic concern for power systems.

All above mentioned issues cause the need to formulate the combined economic and emission dispatch (CEED). The idea behind this multi-objective mathematical programming problem is the simultaneous minimization of economic cost and pollutant emission. These two objectives are conflicting because providing cheaper fossil fuel, a generator emits larger amount of atmospheric pollutants. Furthermore, total power demand must be satisfied without exceeding the generation output limits.

Several deterministic and stochastic techniques have been reported in literature for solving CEED. One of the earlier attempts was the use of an ε-constraint method [16]. A linear programming method that combines section reduction method and third simplex method has been proposed in [6]. In [9] an elitist multi-objective evolutionary algorithm has been developed, based on the non-dominated sorting genetic algorithm - II for solving the environmental / economic dispatch problem. This study considered different objective functions for each of SO_2 and NO_x emissions. In [5] quadratic programming has been applied for various emission and economic dispatch problems. A multi-objective approach of the particle swarm optimization algorithm has been presented in [1] for environmental / economic dispatch. Simulated annealing algorithm for solving economic emission dispatch was proposed in [12] and the λ of the λ-iteration method was considered as the only decision variable. In [13] artificial bee colony algorithm has been applied to solve multi-objective environmental / economic dispatch. In [4] opposition learning method has been utilized to improve the convergence of harmony search algorithm for combined economic and emission dispatch. A variety of economic and emission dispatch problems were handled in [8] by applying an ant algorithm designed for continuous problems. Studies [5], [8], [12] and [13] use penalty factor approaches to combine two objective functions into one.

In this paper, two improved variants of the bat algorithm are proposed to solve the CEED problem. Weighted function method is used to transform the multi-objective problem into single-objective and the best solution is chosen by means of fuzzy set theory.

Rest of the paper is organized as follows: Section 2 introduces the mathematical formulation of CEED. Section 3 presents the two proposed variants of the bat algorithm. Section 4 describes the methodology followed for handling CEED. Section 5 discusses the simulation results of a test system. Results are compared to those obtained by ε-multi-objective genetic algorithm variable (εv-MOGA), hybrid ABC_PSO, multi-objective differential evolution (MODE), non-sorting genetic algorithm-II (NSGA-II), pareto differential evolution (PDE) and strength pareto evolutionary algorithm-2 (SPEA-2). Conclusion of the paper is drawn in section 6.

2. PROBLEM FORMULATION

ELD reduces fuel cost without considering polluting substances output. On the other hand, ED reduces emission pollutants by increasing fuel cost. Purpose of CEED is scheduling the generator outputs in order to bring operating cost and emission fuel into balance without violating equality and inequality constraints. The problem is formulated as follows.

2.1 Fuel cost objective

Mathematically, the fuel cost of each generating unit, considering the valve-point loading effect, is defined as the sum of a quadratic function and a sinusoidal function [10]. Total fuel cost of a power generating station can be expressed as

$$F(P_i) = \sum_{i=1}^{n} \{a_i P_i^2 + b_i P_i + c_i + |d_i \, sin[e_i(P_{i,\min} - P_i)]|\} \quad (1)$$

where:
n: number of thermal units
P_i: power output of the i-the generating unit
a_i, b_i, c_i, d_i, e_i: fuel cost coefficients of the i-th generating unit.

2.2 Emission objective

The atmospheric pollutants such as sulphur dioxide (SO_2), nitrogen oxides (NO_x) and carbon dioxide (CO_2) caused by fossil-fueled thermal units can be modelled separately. However, for comparison purposes, the total emission of these pollutants is the sum of quadratic and exponential functions and can be stated as [6]:

$$E(P_i) = \sum_{i=1}^{n} [\alpha_i P_i^2 + \beta_i P_i + \gamma_i + \eta_i \exp(\delta_i P_i)] \quad (2)$$

where:
α_i, β_i, γ_i, η_i, δ_i: emission coefficients of the i-th generating unit.

2.3 Constraints

- Power balance constraint:

$$\sum_{i=1}^{n} P_i - P_D - P_L = 0 \quad (3)$$

where:
P_D: total load demand
P_L: power losses in the transmission lines.

Power losses can be represented as follows:

$$P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i B_{ij} P_j \quad (4)$$

where B_{ij} are the transmission loss coefficients.

- Generation capacity constraints:

$$P_{i,\min} \le P_i \le P_{i,\max} \quad (5)$$

where:

$P_{i,\min}$: minimum power generation of the i-th unit
$P_{i,\max}$: maximum power generation of the i-th unit

2.4 Combined economic and emission dispatch

CEED problem can be defined as

$$\text{minimize } (F, E) \quad (6)$$

subject to constraints (3) and (5).

Similarly to the majority of the real world problems, CEED involves minimization of competing objective functions. Such multi-objective problems do not find a solution that minimizes all objectives at the same time but there is a set of solutions known as Pareto-optimal or non-dominated. A Pareto optimal solution can improve an objective function but it worsens another one.

A common way to handle a multi-objective problem is to transform it into single-objective by combining the original objective functions. In this study, we use the weighted sum method. Objective function of the CEED problem is formulated as:

$$G = w_1 F + w_2 E \quad (7)$$

under the following conditions:

$$w_1 + w_2 = 1 \text{ and } w_1, w_2 \ge 0 \quad (8)$$

The most appropriate combined objective is determined by a decision making method based on cardinal priority ranking [10].

2.5 Cardinal priority ranking

The fuzzy sets are defined by equations called membership functions. The membership function represents the degree of achievement of each original objective function as a value between 0 and 1. The membership value 1 indicates a fully satisfactory objective while value 0 denotes an unsatisfactory one. By taking account of the minimum and maximum values of each objective function together with the rate of increase of membership satisfaction, the decision maker must determine the membership function $\mu(f_i)$ in a subjective manner defined as:

$$\mu(f_i) = \begin{cases} 1, & f_i \le f_{i,\min} \\ \dfrac{f_{i,\max} - f_i}{f_{i,\max} - f_{i,\min}}, & f_{i,\min} < f_i < f_{i,\max} \\ 0, & f_i \ge f_{i,\max} \end{cases} \quad (9)$$

The value of the membership function indicates how much (in the scale from 0 to 1) a non-dominated solution has satisfied the i-th objective. The sum of the membership function values $\mu(f_i)$ $(i = 1,2, \ldots L)$ for all the objectives can be computed in order to measure the 'accomplishment' of each solution in satisfying the objectives. The 'accomplishment' of each non-dominated solution can be rated with respect to all the M non-dominated solutions by normalizing its 'accomplishment' over the sum of the 'accomplishment' of the M non-dominated solutions as follows:

$$R_k = \sum_{i=1}^{L} \mu_k(f_i) \Bigg/ \sum_{k=1}^{M} \sum_{i=1}^{L} \mu_k(f_i) \quad (10)$$

The function R can represent the fuzzy cardinal priority ranking of the non-dominated solutions. The solution that attains the maximum R_k $(k = 1,2, \ldots M)$ value will be chosen as the best compromise solution [10].

3. PROPOSED ALGORITHMS

Two variants of the bat algorithm are developed for solving the CEED problem. One of the proposed techniques hybridizes bat algorithm with differential evolution and the other one applies bat algorithm enhanced with a mutation operator.

3.1 Principles

Before presenting the proposed optimization methods, we introduce the basic features of bat algorithm and differential evolution.

3.1.1 Bat algorithm

Bat algorithm (BA) is a swarm intelligence metaheuristic developed by Xin-She Yang [15] and inspired by the echolocation behavior of micro-bats (Microchiroptera). Bats of this suborder emit ultrasonic sound pulses with varying loudness and emission rate to detect their prey and avoid obstacles.

In BA, each virtual bat adjusts its frequency Q_i and flies randomly with a velocity v_i at position x_i. Movement of a virtual bat is controlled by its pulse emission rate r_i and loudness A_i. At time step t, a virtual bat moves according to the following equations:

$$Q_i = Q_{\min} + (Q_{\max} - Q_{\min})\beta \tag{11}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_{\text{best}})\, Q_i \tag{12}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{13}$$

where $\beta \in [0,1]$ is a random vector drawn from a uniform distribution and x_{best} is the current global best position (solution). Pseudocode 1 illustrates the original bat algorithm. Emission rate controls the local search, which is a random walk, according to the equation:

$$x_{\text{new}} = x_{\text{best}} + \epsilon\overline{A^t} \tag{14}$$

where $\epsilon \in [-1,1]$ is a random number and $\overline{A^t}$ is the average loudness of all bats at the current time step. Loudness, in conjunction with objective value, determines the acceptance of a new solution. Real bats increase pulse emission rate and decrease loudness as they approach their prey. In BA, these parameters are changed in the same way, with the following equations:

$$A_i^{t+1} = \alpha A_i^t \tag{15}$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \tag{16}$$

Pseudocode 1. Bat algorithm

Objective function $f(x)$, $x = (x_1, x_2, \dots, x_d)^T$
Initialize the bat population x_i and velocities v_i for $i = 1,2,\dots,n$
Initialize frequencies Q_i, pulse rates r_i and the loudness A_i
while ($t < Maximum\ number\ of\ iterations$)
    Generate new solutions by adjusting frequency
    Updating velocities and locations/solutions
    if ($rand\ (0,1) > r_i$)
        Select a solution among the best solutions
        Generate a local solution around the selected best solution
    end if
    Generate a new solution by flying randomly
    if [$rand < A_i$ & $f(x_i) < f(x_{\text{best}})$]
        Accept the new solutions
        Increase r_i and decrease A_i
    end if
    Rank the bats and find the current x_{best}
end while
Postprocess results and visualization

where α and γ are constants. For any $0 < \alpha < 1$ and $\gamma > 0$, we have:

$$A_i^t \longrightarrow 0\,,\quad r_i^t \longrightarrow r_i^0 \quad \text{as} \quad t \longrightarrow \infty \tag{17}$$

Typically, initial loudness A_i^0 takes values from [1,2] while initial emission rate $r_i^0 \in [0,1]$ [15]. Each bat has different values of emission rate and loudness. These parameters are updated only if the new position /solution is improved i.e. bat is moving towards the optimal solution.

3.1.2 Differential evolution

Differential evolution (DE) [14] is one of the most successful evolutionary algorithms. It creates new candidate vectors (solutions) by combining the existing ones and maintains a vector depending on the objective function value.

For each target vector x_i, a new vector is generated by adding the weighted difference between two population vectors to a third vector. This operation is called mutation and the mutated vector v_i at the t-th iteration is generated, as follows:

$$v_i^t = x_{r1}^t + SF \cdot (x_{r2}^t - x_{r3}^t) \tag{18}$$

where $i = 1,2,\dots,NP$ and NP is the population size. Vectors x_{r1}^t, x_{r2}^t, x_{r3}^t, are randomly selected with $r1, r2, r3 \in \{1,2,\dots,NP\}$ and $r1 \neq r2 \neq r3 \neq i$. $SF \in [0,2]$ is a real constant factor that controls the amplification of the differential variation $x_{r2} - x_{r3}$.

In order to increase the diversity of the population, a crossover operation is applied. Parameters of the mutated vector are mixed with parameters of the target vector and a trial vector z_i is created, as follows:

$$z_{ij}^{(t)} = \begin{cases} v_{ij}^{(t)}, randb(j) \leq CR \vee j = j_{rand} \\ x_{ij}^{(t)}, randb(j) > CR \wedge j \neq j_{rand} \end{cases} \tag{19}$$

where $randb(j) \in [0,1]$ is a uniformly distributed random number, generated new for the j-th parameter of the i-th vector. CR is the crossover constant within [0,1]. The relation $CR \wedge j \neq j_{rand}$ ensures that at least one parameter from the mutated vector is selected for the trial vector.

If the trial vector gives smaller objective value than the target vector, the trial vector replaces the target vector in the next iteration. Otherwise, the target vector remains in the population. This last operation is called selection and can be described as:

$$x_i^{(t+1)} = \begin{cases} z_i^{(t)}, f\left(x_i^{(t)}\right) > f\left(z_i^{(t)}\right) \\ x_i^{(t)}, f\left(x_i^{(t)}\right) \leq f\left(z_i^{(t)}\right) \end{cases} \tag{20}$$

There are many other variants of the DE algorithm. The strategies of each variant are denoted by the general notation DE/x/y/z, where

- x is a string that denotes base vector x_{r1} at mutation. Here, it is "rand" because x_{r1} is chosen randomly.
- y is the number of difference vectors.
- z is a string that denotes the type of crossover scheme. For the current variant it is "bin" due to binomial crossover is used [14].

Hence, the basic DE scheme that was described above is denoted as "DE/rand/1/bin".

3.2 Hybrid bat algorithm

Bat algorithm can be improved by hybridization with differential evolution [7]. Hybrid bat algorithm (HBA) uses the differential evolution operations for the local search part instead of a random walk. Pseudocode 2 illustrates HBA.

3.3 Mutated bat algorithm

In this approach, a mutation operator is embedded for more powerful exploration of the solution space. Mutation operator is described by the following equation [3]:

$$x_{\text{new}} = x_{r1} + \beta_1 \odot (x_{r2} - x_{r3}) + \beta_2 \odot (x_{\text{best}} - x_{r4}) \tag{21}$$

where x_{r1}, x_{r2}, x_{r3}, x_{r4} are mutual different solutions and different from the current solution x_i. β_1 and β_2 are random vectors drawn from a uniform distribution within (0,1). Symbol $\odot$ is the Hadamard product operator. For a pair of matrices with the same dimensions, Hadamard product creates a new matrix where each element ij is the product of the corresponding ij elements of the two original matrices. The use of mutation operator provides further exploration of the search space and can avoid trapping into local optima. Mutated bat algorithm (MBA) is described by Pseudocode 3.

4. METHODOLOGY

Bat algorithm variants include many parameters so they are eligible for extensive experimentation. For HBA, this study uses fixed loudness and pulse emission and sets velocity bounds. MBA is implemented as described above. The d-dimensional position of a bat represents the power output values of the d generators.

Initially, a matrix of n candidate solutions is created and given by

$$P = \begin{bmatrix} P_{1,1} & P_{1,2} & \cdots & P_{1,d} \\ P_{2,1} & P_{2,2} & \cdots & P_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n,1} & P_{n,2} & \cdots & P_{n,d} \end{bmatrix} \tag{22}$$

Each solution is evaluated by the objective function. Equality constraint is satisfied by adding a penalty term to the objective function. The new objective function is defined as follows:

$$G = w_1 \cdot F + w_2 \cdot E + h \cdot \left| \sum_{i=1}^{n} P_i - P_D - P_L \right| \tag{23}$$

where h is a positive constant chosen large enough so that Equation (3) holds. For the inequality constraints, if a generating limit is exceeded the power output of this thermal plant will take the value of this limit. Each algorithm continues until the maximum number of iterations is reached.

5. SIMULATION RESULTS

The proposed methods have been applied to a test system consists of ten generators at 2000 MW power demand. Generation unit data has been taken from [11] and it is shown at Appendix. Simulations were performed in MATLAB 8 environment on a PC with a 3 GHz processor. Both algorithms have run for 500 iterations with 15 bats and frequency bounds: $Q_{\min} = 0, Q_{\max} = 2$. For HBA, loudness A is 0.95, emission rate r is 0.5 and these values are the same for every bat. Differential mutation scaling factor SF and crossover constant CR of the DE scheme are equal to 0.5 and 0.8 respectively. Velocity bounds are $v_{i,\min} = -0.1 \cdot P_{i,\min}$ and $v_{i,\max} = 0.1 \cdot P_{i,\max}$. For MBA, coefficients of loudness α and emission rate γ are 0.9 and 0.01. Simulation results are compared with those presented in [2] and [11].

Pseudocode 2. Hybrid bat algorithm

```
Objective function f(x), x = (x_1, x_2, ..., x_d)^T
Initialize the bat population x_i and velocities v_i for i = 1,2,...,n
Initialize frequencies Q_i, pulse rates r_i and the loudness A_i
while (t < Maximum number of iterations)
    Generate new solutions by adjusting frequency
    Updating velocities and locations/solutions
    if (rand (0,1) > r_i)
        Modify the solution using "DE/rand/1/bin"
    end if
    Generate a new solution by flying randomly
    if [rand < A_i & f(x_i) < f(x_best)]
        Accept the new solutions
        Increase r_i and decrease A_i
    end if
    Rank the bats and find the current x_best
end while
Postprocess results and visualization
```

Pseudocode 3. Mutated bat algorithm

```
Objective function f(x), x = (x_1, x_2, ..., x_d)^T
Initialize the bat population x_i and velocities v_i for i = 1,2,...,n
Initialize frequencies Q_i, pulse rates r_i and the loudness A_i
while (t < Maximum number of iterations)
    Generate new solutions x_q by adjusting frequency
    Updating velocities and locations/solutions
    if (rand (0,1) > r_i)
        Select a solution among the best solutions
        Generate a local solution x_l around the selected best solution
    end if
    Generate a solution x_m using mutation operator
    Select the best solution x_i among x_q, x_l and x_m
    if [rand < A_i & f(x_i) < f(x_best)]
        Accept the new solutions
        Increase r_i and decrease A_i
    end if
    Rank the bats and find the current x_best
end while
Postprocess results and visualization
```

Results of the proposed algorithms for ELD are shown in Table 1 and they are compared to those provided by ABC_PSO hybrid algorithm and DE. Proposed algorithms achieve lower fuel costs values than those obtained using ABC_PSO and DE. Although fuel cost is the objective to be minimized, both proposed algorithms also manage to decrease pollutant emission. MBA obtains lower emission than HBA. For ED, results are shown in Table 2. HBA and MBA produce lower values for both emission and fuel cost than those obtained by ABC_PSO but these values are higher compared to DE. Figures 1 and 2 illustrate fuel cost and emission convergence of the proposed algorithms. Both HBA and MBA reach the optimal solution very fast.

Numerical results of the multi-objective CEED problem are shown in Table 3. HBA obtains higher fuel cost and lower emission than MBA. Both proposed algorithms yield lower

Table 1. Economic load dispatch comparison results

Output	Algorithm			
	HBA	**MBA**	**ABC_PSO**	**DE**
Unit 1 (MW)	55.0000	55.0000	55.0000	55.0000
Unit 2 (MW)	80.0000	80.0000	80.0000	79.8900
Unit 3 (MW)	106.6250	106.0842	106.9300	106.8253
Unit 4 (MW)	99.2860	99.4944	100.5668	102.8307
Unit 5 (MW)	82.1004	82.4612	81.4900	82.2418
Unit 6 (MW)	84.0278	83.9977	83.0110	80.4352
Unit 7 (MW)	300.0000	300.0000	300.0000	300.0000
Unit 8 (MW)	339.9983	340.0000	340.0000	340.0000
Unit 9 (MW)	470.0000	470.0000	470.0000	470.0000
Unit 10 (MW)	469.9999	469.9973	470.0000	469.8975
Losses (MW)	87.0374	87.0349	87.0344	-
Fuel cost ($/hr)	111498	111498	111500	111500
Emission (lb/hr)	4565.0	4563.2	4571.2	4581.0
CPU time (sec)	1.99	2.08	-	9.42

Table 2. Emission dispatch comparison results

Output	Algorithm			
	HBA	**MBA**	**ABC_PSO**	**DE**
Unit 1 (MW)	55.0000	55.0000	55.0000	55.0000
Unit 2 (MW)	80.0000	80.0000	80.0000	80.0000
Unit 3 (MW)	81.1342	81.1391	81.9604	80.5924
Unit 4 (MW)	81.3638	81.3567	78.8216	81.0233
Unit 5 (MW)	160.0000	160.0000	160.0000	160.0000
Unit 6 (MW)	240.0000	240.000	240.0000	240.0000
Unit 7 (MW)	294.4856	294.4801	300.0000	292.7434
Unit 8 (MW)	297.2685	297.2822	292.7800	299.1214
Unit 9 (MW)	396.7662	396.7657	401.8478	394.5147
Unit 10 (MW)	395.5769	395.5713	391.2096	398.6383
Losses (MW)	81.5952	81.5951	81.5879	-
Fuel cost ($/hr)	116412	116412	116420	116400
Emission (lb/hr)	3932.2	3932.2	3932.3	3923.4
CPU time (sec)	1.99	2.25	-	8.56

Table 3. Combined economic and emission dispatch comparison results

Output	Algorithm							
	HBA	**MBA**	**εv-MOGA**	**ABC_PSO**	**MODE**	**PDE**	**NSGA-II**	**SPEA-2**
Unit 1 (MW)	55.0000	55.0000	54.1807	55.0000	54.9487	54.9853	51.9515	52.9761
Unit 2 (MW)	80.0000	80.0000	78.4981	80.0000	74.5821	79.3803	67.2584	72.813
Unit 3 (MW)	85.0157	85.0378	84.7653	81.1400	79.4294	83.9842	73.6879	78.1128
Unit 4 (MW)	83.6358	83.6548	81.3502	84.2160	80.6875	86.5942	91.3554	83.6088
Unit 5 (MW)	141.5894	141.3312	138.0526	138.3377	136.8551	144.4386	134.0522	137.2432
Unit 6 (MW)	161.6944	161.3887	166.2667	167.5086	172.6393	165.7756	174.9504	172.9188
Unit 7 (MW)	300.0000	299.9998	295.4660	296.8338	283.8233	283.2122	289.435	287.2023
Unit 8 (MW)	315.4369	315.4383	326.7642	311.5824	316.3407	312.7709	314.0556	326.4023
Unit 9 (MW)	429.7206	429.9759	428.9338	420.3363	448.5923	440.1135	455.6978	448.8814
Unit 10 (MW)	431.8299	432.1132	429.6309	449.1598	436.4287	432.6783	431.8054	423.9025
Losses (MW)	83.9228	83.9395	-	84.1736	-	-	-	-
Fuel cost ($/hr)	113389	113375	113422	113420	113480	113510	113540	113520
Emission (lb/hr)	4117.6	4119.0	4120.5	4120.1	4124.9	4111.4	4130.2	4109.1
CPU time (sec)	2.3006	2.9339	3.80	-	3.82	4.23	6.02	7.53

values for both fuel cost and emission output compared to εv-MOGA, ABC_PSO, MODE and NSGA-II. In comparison with PDE and SPEA-2, both bat algorithm variants yield lower fuel cost and higher emission. The distributions of 15 non dominated solutions for HBA and MBA are shown in Figures 3 and 4 respectively. As expected, an objective function cannot be improved without degradation of the other one.

6. CONCLUSION

In this paper, two bat algorithm approaches were employed to solve the combined economic and emission dispatch problem with non-polynomial equations. Both methods are very efficient and provide competitive solutions. Despite the fact that fuel cost conflicts with emission, proposed methods decrease both objectives in most cases. In addition, they achieve very fast convergence without suffering from the cost of high computational time. These facts make HBA and MBA more powerful than other successful algorithms for the solution of the multi-objective CEED problem.

Comparing the proposed method to each other, in HBA a bigger number of parameters needs to be determined while MBA consumes greater computational time. However, a good parameter setting can be obtained after experimentation and both algorithms can improve their performance so as to become more powerful than other modern optimization methods.

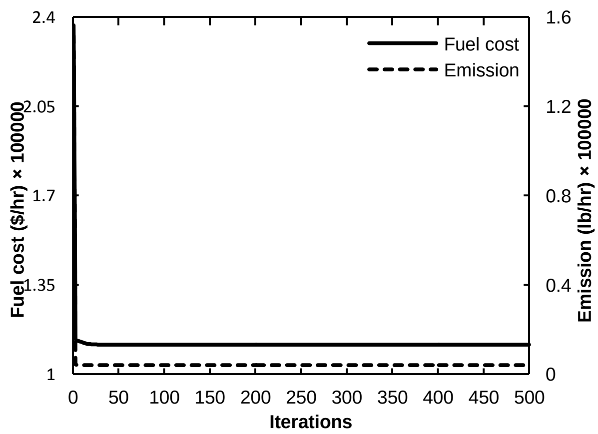

Figure 1. Fuel cost and emission convergence for HBA

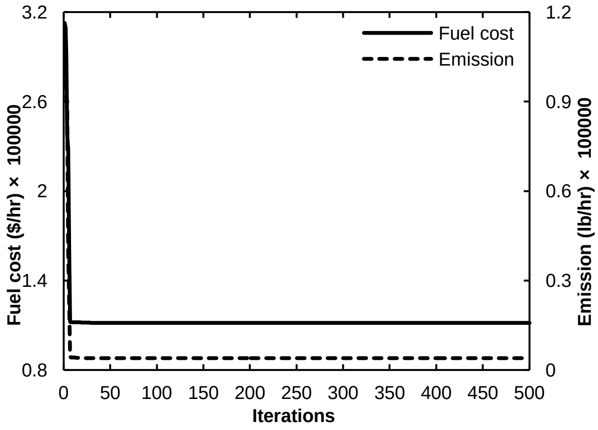

Figure 2. Fuel cost and emission convergence for MBA

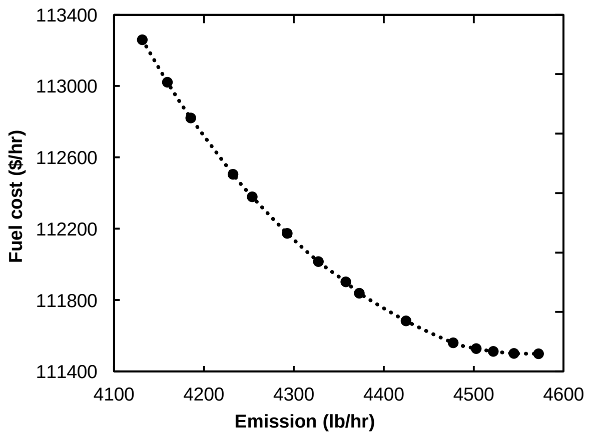

Figure 3. Pareto optimal front for HBA

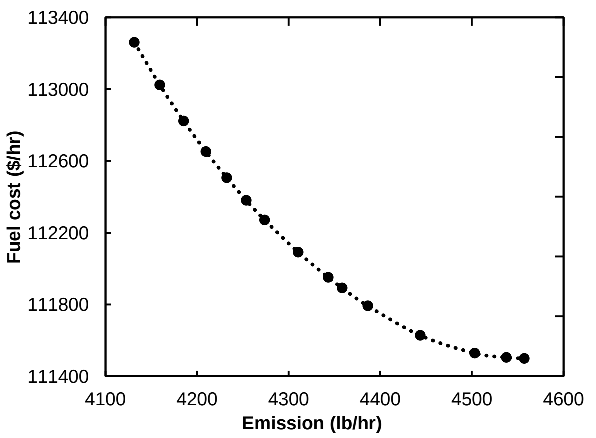

Figure 4. Pareto optimal front for MBA

7. REFERENCES

[1] Abido, M. A., 2009. “Multiobjective particle swarm optimization for environmental/economic dispatch problem”, Electric Power Systems Research, Vol. 79, No. 7, pp. 1105-1113.

[2] Afzalan, E. and Joorabian, M., 2013. “Emission, reserve and economic load dispatch problem with non-smooth and non-convex cost functions using epsilon-multi-objective genetic algorithm variable”, International Journal of Electrical Power & Energy Systems, Vol. 52, pp. 55-67.

[3] Azizipanah-Abarghooee., R and Niknam, T., 2012. “A new improved Bat Algorithm for fuzzy interactive multiobjective economic / emission dispatch with load and wind power uncertainty”, Uncertainty Modeling in Knowledge Engineering and Decision Making, pp. 388-393.

[4] Chatterjee, A., Ghoshal, S. P. and Mukherjee, V., 2012. “Solution of combined economic and emission dispatch problems of power systems by an opposition-based harmony search algorithm”, International Journal of Electrical Power & Energy Systems, Vol. 39, No. 1, pp. 9-20.

[5] Danaraj, R. M. S. and Gajendran, F., 2005. “Quadratic Programming Solution to Emission and Economic Dispatch Problems”, Journal of Institution of Engineers (India), Vol. 86, pp. 129-132.

[6] Farag, A., Al-Baiyat, S. and Cheng, T., 1995. “Economic load dispatch multiobjective optimization procedures using linear programming techniques”, IEEE Transactions on Power Systems, Vol. 10, No. 2, pp. 731-738.

[7] Fister, I., Fister, D. and Yang, X. S., 2013. “A Hybrid Bat Algorithm”, Elektrotehniški vestnik, Vol. 80, No. 1-2, pp. 1-7.

[8] Karakonstantis, I. and Vlachos, A., 2015. “Ant Colony Optimization for Continuous Domains applied to Emission and Economic Dispatch Problems”, Journal of Information and Optimization Sciences, Vol. 36, No. 1-2, pp. 23-42.

[9] King, R. T. F. A. and Rughooputh, H.C.S., 2003. “Elitist Multiobjective Evolutionary Algorithm for Environmental/Economic Dispatch”, Evolutionary Computation, 2003. CEC '03. The 2003 Congress on, Vol. 2, pp. 1108-1114.

[10] Kothari, D. P. and Dhillon, J. S., 2011. Power system optimization (2nd ed.), Prentice-Hall of India.

[11] Manteaw, E. D. and Odero, N. A., 2012. “Combined Economic and Emission Dispatch Solution Using ABC_PSO Hybrid Algorithm with Valve Point Loading Effect”, International Journal of Scientific and Research Publications, Vol. 2, No. 12, 9 pages.

[12] Sasikala, J. and Ramaswamy, M., 2010. “Optimal λ based Economic Emission Dispatch using Simulated Annealing”, International Journal of Computer Applications, Vol. 1, No. 10, pp. 55-63.

[13] Sonmez, Y., 2011. “Multi-objective environmental / economic dispatch solution with penalty factor using artificial bee colony algorithm”, Scientific Research and Essays, Vol. 6, No. 13, pp. 2824-2831.

[14] Storn, R. and Price K., 1997. “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces”, Journal of Global Optimization, Vol. 11, No. 4, pp. 341–359.

[15] Yang, X. S., 2010. “A New Metaheuristic Bat-Inspired Algorithm”, Nature Inspired Cooperative Strategies for Optimization (NISCO 2010): Studies in Computational Intelligence, Vol. 284, pp. 65-74.

[16] Yokoyama, R., Bae, S., Morita, T. and Sasaki, H., 1988. “Multiobjective optimal generation dispatch based on probability security criteria”, IEEE Transactions on Power Systems, Vol. 3, No.1, pp. 317-324.

8. APPENDIX

Table A1. Fuel cost coefficients and power limits

Generating unit	a_i ($/MW^{2}hr)	b_i ($/MWhr)	c_i ($/hr)	d_i ($/hr)	e_i (rad/MW)	$P_{i,\min}$ (MW)	$P_{i,\max}$ (MW)
1	0.12951	40.5407	1000.403	33	0.0174	10	55
2	0.10908	39.5804	950.606	25	0.0178	20	80
3	0.12511	36.5104	900.705	32	0.0162	47	120
4	0.12111	39.5104	800.705	30	0.0168	20	130
5	0.15247	38.5390	756.799	30	0.0148	50	160
6	0.10587	46.1592	451.325	20	0.0163	70	240
7	0.03546	38.3055	1243.531	20	0.0152	60	300
8	0.02803	40.3965	1049.998	30	0.0128	70	340
9	0.02111	36.3278	1658.569	60	0.0136	135	470
10	0.01799	38.2704	1356.659	40	0.0141	150	470

Table A2. Emission coefficients

Generating unit	α_i (lb/MW^{2}hr)	β_i (lb/MWhr)	γ_i (lb/hr)	η_i (lb/hr)	δ_i (1/MW)
1	0.04702	-3.9864	360.0012	0.25475	0.01234
2	0.04652	-3.9524	350.0012	0.25475	0.01234
3	0.04652	-3.9023	330.0056	0.25163	0.01215
4	0.04652	-3.9023	330.0056	0.25163	0.01215
5	0.00420	0.3277	13.8593	0.24970	0.01200
6	0.00420	0.3277	13.8593	0.24970	0.01200
7	0.00680	-0.5455	40.2699	0.24800	0.01290
8	0.00680	-0.5455	40.2699	0.24990	0.01203
9	0.00460	-0.5112	42.8955	0.25470	0.01234
10	0.00460	-0.5112	42.8955	0.25470	0.01234

The transmission loss coefficient matrix is

$$B = 10^{-5} \times \begin{bmatrix} 4.9 & 1.4 & 1.5 & 1.5 & 1.6 & 1.7 & 1.7 & 1.8 & 1.9 & 2.0 \\ 1.4 & 4.5 & 1.6 & 1.6 & 1.7 & 1.5 & 1.5 & 1.6 & 1.8 & 1.8 \\ 1.5 & 1.6 & 3.9 & 1.0 & 1.2 & 1.2 & 1.4 & 1.4 & 1.6 & 1.6 \\ 1.5 & 1.6 & 1.0 & 4.0 & 1.4 & 1.0 & 1.1 & 1.2 & 1.4 & 1.5 \\ 1.6 & 1.7 & 1.2 & 1.4 & 3.5 & 1.1 & 1.3 & 1.3 & 1.5 & 1.6 \\ 1.7 & 1.5 & 1.2 & 1.0 & 1.1 & 3.6 & 1.2 & 1.2 & 1.4 & 1.5 \\ 1.7 & 1.5 & 1.4 & 1.1 & 1.3 & 1.2 & 3.8 & 1.6 & 1.6 & 1.8 \\ 1.8 & 1.6 & 1.4 & 1.2 & 1.3 & 1.2 & 1.6 & 4.0 & 1.5 & 1.6 \\ 1.9 & 1.8 & 1.6 & 1.4 & 1.5 & 1.4 & 1.6 & 1.5 & 4.2 & 1.9 \\ 2.0 & 1.8 & 1.6 & 1.5 & 1.6 & 1.5 & 1.8 & 1.6 & 1.9 & 4.4 \end{bmatrix}$$