International Journal of
Intelligent Systems and Applications in Engineering

ISSN:2147-6799 www.ijisae.org Original Research Paper

Improved Artificial Cooperative Search Algorithm for Solving Non-convex Economic Dispatch Problems with Valve-point Effects

Oguz Emrah TURGUT

Accepted : 10/04/2018 Published: 28/09/2018

Abstract: This paper presents Improved Artificial Cooperative Search (IACS) algorithm for solving economic dispatch problems considering the valve point effects, ramp rate limits, transmission losses and prohibited operation zones. In order to improve the solution quality and increase the search efficiency, a novel perturbation scheme called "Global best guided chaotic local search" is proposed and incorporated into ACS algorithm. The effectiveness of the proposed IACS algorithm has been benchmarked with twelve widely known optimization test problems. In order to assess the performance of the proposed algorithm on non-convex optimization problems, four case studies related to highly nonlinear economic dispatch problems have been solved . Results retrieved from IACS algorithm have been compared with literature approaches in terms of minimum, maximum and average generation cost values. Comparison results indicate that IACS produces more economical power load than those of other optimizers available in the literature.

Keywords: *Artificial Cooperative Search, Economic Dispatch, Non-convex optimization, Ramp-rate limits, Valve-point effects*

1. Introduction

Economic load dispatch (ELD) problem plays an important role in power systems planning. ELD is a constrained optimization problem whose main objective is to minimize total fuel cost of generating units while satisfying an equality and a great deal of inequality constraints including discontinuous prohibited zones, generating unit constraints and ramp rate limits. The cost of power generation in fossil fuel plants is very high therefore optimum scheduling of generation units is needed to save possible amount of expenditure on power generation systems. Each power unit is represented by a quadratic cost function which becomes highly nonlinear, non-convex and discontinuous due to the effect of valve point loadings and prohibited operating zones. This functional behaviour generates multiple local optimum points in solution space and complicates the locating the global optimum of the ELD problem[1]. Mathematical programming techniques [2-7] have been utilized to reach the optimum solution of the ELD problem however these kind of methods have not provided feasible solutions yet and they generally get trapped in local optimum points in the search space [8]. Dynamic Programming method [9] succeeds to solve ELD problems and copes with the non-convexities occurred by valve point effect, however this method incurs high computational burden and its performance deteriorates with increasing number of generation units. Besides, Newton based methods have had trouble in handling large number of inequality constraints objected to ELD problem [10].

Due to their supreme capability on maintaining acceptable balance between exploration of the search domain and exploitation of the promising areas, metaheuristic methods such as Genetic algorithm (GA) [11-13], Gravitational Search Algorithm [14-15], Simulated Annealing (SA) [16-18], Particle Swarm Optimization (PSO) [19-22], Differential Evolution [23-26], Harmony Search (HS) [27-28], Artificial Bee Colony (ABC) [29-30], Firefly Algorithm (FA) [31-35], Teaching Learning based Optimization(TLBO) [36-37], Cuckoo Search (CS) [38-39] and Biogeography-Based Optimization (BBO) [8,40-43,76] have been recruited for solving economic dispatch problems. From the literature survey, it is seen that metaheuristic algorithms applied on ELD problem have not guaranteed to find the global optimum of the problem, however they are capable of finding near-optimal solutions. Detailed explanation of some of the studies mentioned above is given in Table 1.

Table 1 Detailed explanation of some studies in the literature

Ref.	Explanation
[11]	The Atavistic GA is applied to solve the ELD problem with valve point discontinuities. The algorithm is applied to a system with 13 generating units and found better solutions than traditional GA.
[14]	The GSA algorithm is utilized to solve the ELD problem. The results showed that the algorithm is easy to implement, robust, gives more favourable solutions with less execution time.
[16]	The SA algorithm is applied to solve the ELD problem. The transmission losses are later added to the equation. The results are compared with those found by the dynamic programming of the ELD.
[29]	The ABC algorithm is utilized to solve the ELD problem with 10, 13, 15 and 40 generating units. The results are compared with that of the other techniques reported in the literature. The ABC algorithm found more favorable results.
[31]	The FA algorithm is utilized to solve the ELD problem. Many nonlinear characteristics of the generating units have been taken into account. The results showed that the FA algorithm finds better solutions than the others.
[36]	The TLBO algorithm is suggested to solve the ELD problem. The proposed methodology is similar to the other studies in the literature. The TLBO algorithm found more favorable results than the other algorithms.

In this article, Improved Artificial Cooperative Search (IACS) is presented for successful solution of non-convex economic dispatch problems. ACS is based on the interaction between prey and predator individuals of the population while they are migrating to find possible food resources. ACS has fewer control parameters and uses different mutation and crossover strategies than other optimization algorithms [44]. In order to enhance the convergence capability of the ACS algorithm, a novel perturbation scheme called "Global best guided chaotic local search" is proposed in this study. The proposed scheme is based on the motivation of exploitation of the explored areas of the search domain and refines the so-far-obtained optimum solution by means of the global best solution vector, which guides the population individuals during iterations. By this scheme, mutated individuals move towards to

[1] *Bakircay University, Department of Mechanical Engineering, Izmir, Turkey*
* *Corresponding Author Email: oeturgut@hotmail.com*

promising areas of the search space with guidance of the recruited global best solution vector. To test the performance of the proposed methodology, four standard test systems composed of 13-, 38-, 40-, and 140- generation units have been solved by IACS algorithm. Simulation results have been compared with other metaheuristic algorithms applied on ELD problems. Comparison results in terms of statistical analysis show that the proposed IACS produces better generation cost values for each case study. To the author's best knowledge, this is the first application of ACS-based algorithms on ELD problems. The main motivation of this study is suggesting the ACS and IACS algorithms for solving the ELD problem and comparing the performances of these two algorithms. Rest of the paper is organized as follows: Mathematical modelling of economic dispatch problem is explained in Section 2. Section 3 presents the description of ACS algorithm, improvements over ACS algorithm and implementation of IACS on ELD problem. Proposed algorithm is benchmarked with a suite of twelve-optimization test problem in Section 4. Section 5 describes the application of IACS algorithm on solving non-convex economic dispatch problems with 13-, 38-, 40- ,and 140-unit generating systems and Section 6 provides the conclusion.

2. Mathematical Modelling of Economic Dispatch Problems

Economic dispatch problem aims to find optimum combination of power generation units that minimizes total fuel cost while subjected to an equality and several inequality constraints. Economic dispatch, which is a sub division of Unit Commitment (UC) problems, is an example of nonlinear programming optimization due to nonlinear characteristic of power systems [31]. Formulation of the ELD problem can be described as

$$\text{Minimize } F = \sum_{i=1}^{N} F_i(P_i) = \sum_{i=1}^{N} a_i P_i^2 + b_i P_i + c_i \tag{1}$$

where F is total generation cost to be minimized and F_i is the cost function of i^{th} generator; power output of the i^{th} generator is represented by P_i; a_i, b_i and c_i are the coefficients pertaining to i^{th} generator and N is the number of the on-line generators in the power generation system. Modelling valve point loadings is necessary to capture the losses incurred due to the throttling of partially open valves in electric power generators [45]. Introducing valve point effects into economic dispatch problem makes the objective function non-convex owing to the contribution of ripple-like effect occurred in multi valve steam turbines. This feature enhances the non-linearity of the objective function and increases the chance to be getting stuck in the local optimum points in generation cost curve. Superposition of sinusoidal function and total cost function is formulized as

$$F_i\left(P_i\right) = \sum_{i=1}^{N} a_i P_i^2 + b_i P_i + c_i + \left| e_i + \sin\left(f_i \times \left(P_{i,\min} - P_i \right)\right)\right| \tag{2}$$

where $P_{i,min}$ corresponds to lower bound of the power generation for the i^{th} generator; e_i and f_i are fuel cost coefficients of the i^{th} generator that model the valve point effect in the generation cost curve.

2.1. Power balance constraints

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

where P_D is total load demand and P_L represents transmission losses of the power generation system. The B – coefficient method [46], commonly used by the power industry to calculate transmission network losses, is formulated by the following expression

$$P_L = \sum_{i=1}^{N}\sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{0i} P_i + B_{00} \tag{4}$$

2.2. Operational limits

Power output of the each generator should be restricted between maximum and minimum limits. Following inequality constraint should be applied for each generation unit:

$$P_{i,\min} \le P_i \le P_{i,\max} \tag{5}$$

where $P_{i,min}$ and $P_{i,max}$ are minimum and maximum power output for i^{th} generating unit, respectively. When it is to consider ramp rate limits of each generator, operation bounds are modified as follows:

$$\max\left(P_{i,min}, P_i^0 - DR_i\right) \le P_i \le \min\left(P_{i,\max}, P_i^0 + UR_i\right) \tag{6}$$

In Eq. (6), P_i^0 is previous generator output power; UR_i and DR_i are respectively up and down ramp limits of the i^{th} generator in terms of MW/h.

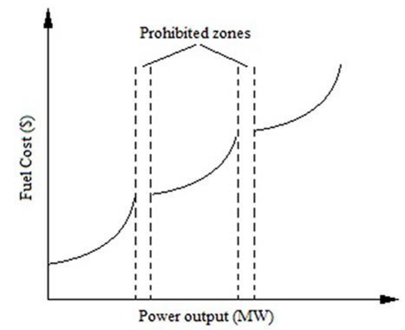


Figure 1. Fuel cost curve considering prohibited operating zones

2.3. Prohibited Zones

Due to the physical limitations of machine components or vibrations on the shaft, start and stop of coal mills that take place in their auxiliary parts, generator units have prohibited regions that make operating curves of the generator non-continuous [47]. Power output of the generators must be avoided from these areas to satisfy operation constraints. Fig. 1 shows the characteristics of the cost curves with prohibited operating zones (POZ). Mathematical representation of the constraints can be given as

$$\begin{aligned} &P_{i,\min} \le P_i \le P_{i,1}^L \\ &P_{i,k-1}^U \le P_i \le P_{i,k}^L \\ &P_{i,M}^U \le P_i \le P_{i,\max} \qquad k = 2,3,...,M \end{aligned} \tag{7}$$

where M is the number of prohibited operating zones of i^{th} generator; $P_{i,k}^L$ and $P_{i,k}^U$ are the lower and upper limits of the k^{th} prohibited zone of the i^{th} generator, correspondingly. In this article, POZ constraints are taken into consideration by applying the following procedure:
If power output of the i^{th} generator lies between upper and lower bounds of the j^{th} POZ, that is to say,

$$P_{i,j}^L < P_i < P_{i,j}^U \tag{8}$$

output of the generator is pushed to nearest boundary of the k^{th} prohibited zone by applying the following equations

$$P_{i,j}^{av} = \frac{P_{i,j}^L + P_{i,j}^U}{2} \tag{9}$$

$$P_i \leftarrow \begin{cases} P_{i,j}^L & if \;\; P_{i,j}^L < P_i < P_{i,j}^{av} \\ P_{i,j}^U & if \;\; P_{i,j}^{av} < P_i < P_{i,j}^U \end{cases} \tag{10}$$

3. Artificial Cooperative Search Algorithm

3.1. Fundamentals of the Artificial Cooperative Search

Nature has always been an inspiration for many researchers. Nowadays, many optimization algorithms has been inspired from biological, physical or chemical systems [48]. Most of the bio-inspired algorithms have been derived from the swarm intelligence concept. Swarm Intelligence (SI) is a special kind of bio-inspired algorithm deals with the behaviour of collective, multiple agents interacting with each other by following some predetermined rules. In SI algorithms, each agent may behave as an unintelligent entity, but whole system, consists of multiple agents, may show some kind of intelligent behaviour.

Artificial Cooperative Search (ACS), developed by Civicioglu [44] to be used in solving real-valued numerical optimization problems, is a dual-population based swarm intelligence algorithm. In nature, there are such individuals those utilize mutualism based biological interaction locations in order to sustain for their lives. Organisms involved in a mutualism based interaction locations try to take benefit of these location points. In mutualism, two types of organism living in the same habitat aims to derive mutual benefits from each other. Besides, there is another term called "cooperation" which is interaction of homogenous living beings that adopt mutualism. ACS algorithm is conceptualized on aforementioned mutual and cooperation based biological interaction of two eusocial superorganisms living in the same habitat. "Habitat" term mentioned above matches the "search domain" concept pertaining to the optimization problem.

ACS algorithm is based on the interaction between two artificial superorganisms as they interact and migrate to variety of areas to find more fruitful habitat. In nature, amount of food that can be found in a habitat depends on yearly climate changes. For that reason, many superorganisms have developed seasonal migration behaviour to find better food sources. Many species are known to set up a group called "superorganism" prior to migration. After a superorganism is formed, individuals of the superorganism start to move to better food sources by means of forming groups. In addition, many superorganisms can divide into sub-groups (sub-superorganisms) prior to migration. Many swarms use explorers to discover a habitat. Explorers discover a possible migration area, then collect information about this new explored area and share the information with the superorganism they belong to. If the superorganism decides to migrate to a new explored area, it moves to discovered area, and then this exploration process starts again and proceeds until they find productive feeding areas.

In ACS algorithm, artificial superorganisms migrating to find possible fruitful areas refer to superorganisms with random solutions under given search space. ACS algorithm is composed of two superorganisms, namely α and β, those inherit sub-superorganisms equal to the population size (N). Sub-superorganisms consist of D individuals which correspond to the dimension of the optimization problem. Prey and Predator sub-superorganisms are determined by means of α and β superorganisms. In ACS algorithm, predator sub-superorganism individuals pursue prey sub-super organism individuals while they are migrating to find productive feeding areas (optimum point of the problem). The whole iteration process in ACS algorithm can be named as "coevolution" which is based on the two superorganisms looking for the optimum solution of the problem, maintaining cooperation based biological interaction between each other.

Individuals of the i^{th} sub-superorganisms of α and β are initialized with the equation below

$$\alpha_{i,j:g} = rnd.(up_j - low_j) + low_j$$
$$\beta_{i,j:g} = rnd.(up_j - low_j) + low_j \tag{11}$$

where $i = 1,2,3,...,N$, $j = 1,2,3,...,D$ and $g = 1,2,3,...,\text{maxcycle}$. The *rnd* represents a random number selected from a uniform distribution in the range of [0,1]. The *g* value counts the iteration number. Symbols up_j and low_j show the upper and lower bounds of the search space for j^{th} dimension of the problem. Fitness values (productivity values) of the associated sub-superoganisms are calculated by using the following formula;

$$y_{i;\alpha} = f(\alpha_i)$$
$$y_{i;\beta} = f(\beta_i) \tag{12}$$

Table 1 gives the pseudo-code of the Artificial Cooperative Search algorithm equipped with the evolutionary boundary constraint mechanism that will be explained in the upcoming sections. In Table1, there are some symbolizations and abbreviatons those ease the comprehension of the conceptual descriptions. For instance, rand(0,1) stands for the representation of a uniform random number defined in the range [0,1]; permute(.) function shuffles the row elements of the population individuals; X represents the biological interaction locations for Predator and Prey individuals; R is the scale factor that determines the biological interaction speed; rndint(1,Y) generates pseudo-random integers defined between 1 and Y; determination of the passive individuals is procured by M matrix which is comprised of integers 0 and 1.

3.2. Improvements over Artificial Cooperative Search Algorithm

3.2.1. Global best guided chaotic local search

In this section, a novel local search mechanism is proposed to refine the optimal solutions corresponding to the interaction locations between prey and predator individuals and avoid being trapped in local optimum points. Inspired by the search equations of Differential Evolution [49] and Artificial Bee Colony [50] algorithms, proposed perturbation scheme takes full advantage of global best (G_{best}) vector of current population and probes around the G_{best} solution to circumvent the local optimum solutions faced on the course of iterations. Proposed scheme can be described as

$$X_{new,j} = G_{best,j} + 2.0 \cdot \left(ch_{i,j} - 0.5\right) \cdot \left(G_{best,j} - X_{i,j}\right) \tag{13}$$

where i = 1,2,3,..,N; j = 1,2,..,D and ch is chaotic variable generated by Logistic map [51]. Chaos is a deterministic, random-like mathematical phenomena occur in nonlinear systems and has a strong dependence on initial conditions [51,77]. Effective and ergodic chaotic sequences can be generated by an ordinary chaotic map on the concept of the following equation

$$x_{k+1} = f(x_k), \quad 0 < x_k < 1, \quad k = 1,2,3.... \tag{14}$$

Logistic map, which is one dimensional chaotic map and demonstrates that how complex behaviour arises from a simple deterministic system without need of any random sequence, is defined as

$$ch_{k+1} = \gamma \cdot ch_k \cdot \left(1 - ch_k\right) \tag{15}$$

where γ is a control parameter and ch is a chaotic variable as defined before. For initial conditions, ch_0 should be in the range of (0,1) and $ch_0 \notin (0.25, 0.50, 0.75)$. Chaotic behaviour of the generated sequence can be controlled by the control parameter γ, however Logistic map sequence is chaotic when γ=4.0.

3.2.2. Evolutionary boundary constraint handling scheme

Gandomi and Yang [52] developed an evolutionary scheme for boundary constraint handling. According to this proposed scheme, when population individuals goes beyond the prescribed boundaries, they are pushed into the related bounds of the optimization problem by means of a uniform random number and global best solution vector obtained so far. Proposed constraint handling scheme can be formulized as

$$x_{new,i} \leftarrow \begin{cases} \alpha \times low_i + (1-\alpha) G_{best,i} & if\ \ x_i < low_i \\ \beta \times up_i + (1-\beta) G_{best,i} & if\ \ x_i > up_i \end{cases} \tag{16}$$

where α and β are real valued number in the range of [0,1]; low_i and up_i are the i^{th} variable of the lower and upper bounds of the optimization problem, respectively; x_i is the i^{th} mutable decision variable of the related optimization problem. In the view of ACS algorithm, x_i is the biological interaction location where prey individuals are pursued by predator individuals aiming for finding more suitable areas for subsistence.

3.3. Implementation of Improved Artificial Cooperative Search Algorithm for ELD Problem

The proposed IACS algorithm will be applied on economic dispatch problems considering valve point effects, ramp rate limits and prohibited operating zones those all make the objective function of the problem non-linear, non-convex, and non-continuous. IACS is proposed for optimum scheduling of each power generation unit satisfying both equality and inequality constraints. Solution steps of the economic dispatch problem using IACS algorithm are given as follows;

Step 1: Apply upper and lower bounds; define cost coefficients, transmission loss coefficients, prohibited operating zones, valve point coefficients and ramp limits for each generation unit; initialize the chaotic sequence by using Logistic map; determine population size and maximum number of generation

Step 2: Initialize α and β superorganisms by random real valued numbers as described in Eq. (11). Remind prohibited zones by adjusting the numerical values of superorganism individuals (α and β) with using Eqs. (9) and (10). Calculate the fitness values of the α and β superorganisms with considering the valve point effects, total energy demand constraints, and transmission losses given as the following equation;

$$arg\,min \quad F = \sum_{i=1}^{N} a_i P_i^2 + b_i P_i + c_i + \left| e_i + sin\left(f_i \times \left(P_{i,min} - P_i \right) \right) \right| + \psi \left| \sum_{i=1}^{N} P_i - P_L - P_D \right| \tag{17}$$

where ψ is a problem dependant penalty coefficient which penalizes infeasible solutions. Set iteration counter to 1

Step 3: Determine the predator individuals and their respective fitness values by following the procedure given in Table 1 in the lines between 12 and 16

Step 4: Determine prey individuals by implementing the procedure given in Table 1 within the lines between 17 and 18

Step 5: Calculate scale factor (R) by the rule given in Table 1 in the lines between 19 and 23.

Step 6: Apply binary valued integer map (M) to determine passive individuals with the decision rule stated in Table 1 within the lines between 24 and 45

Step 7: Calculate the biological interaction locations with the equation given in line 47 in Table 1

Step 8: Determine the best solution (G_{best}) of the current population and fine-tune the biological interaction locations with Eq. (13). If mutated solution vectors are better than those of inferior solutions, update the perturbed solution vector. Increment the iteration counter.

Step 9: Update the biological interaction locations with the procedure given within the lines between 48 and 53.

Step 10: Apply evolutionary boundary constraint handling scheme defined in the lines between 55 and 63 in Table 1. Handle the prohibited zone constraints with (9) and (10), and process the selection update mechanism through the procedure given within the lines between 65 and 67 in Table 1

Step 11: Determine new superorganisms for next generations with the decision rule described in the lines between 68 and 72 in Table 1.

Step 12: Get the best fitness value of predator sub-superorganism. Retain the best fitness value and its corresponding design variables for next generations.

Step 13: Update the chaotic sequence generated by Logistic map as described in Eq. (15) and increment the iteration counter.

Step 14: Repeat Step 3 to Step 13 until termination criteria is met.

Pseudocode of artificial cooperative search algorithm

```
1   INPUT DATA: POPULATION SIZE (N), PROBLEM DIMENSION (D), MAXIMUM ITERATION
    NUMBER (MAXITER), OBJECTIVE FUNCTION F(.), PROBABILITY OF BIOLOGICAL
    INTERACTION (P), UPPER AND LOWER BOUNDS (UP AND LOW)
2   SET GLOBALMINIMUM TO 1E20 AND INITIALIZE SUPERORGANISMS (A, B)
3   FOR I = 1 TO N
4   FOR J = 1 TO D
5   A_I,J=LOW_J + (UP_J - LOW_J) X RAND_1(0,1)
6   B_I,J=LOW_J + (UP_J - LOW_J) X RAND_2(0,1)
7   END
8   FITNESS-A_I = F(A_I)
9   FITNESS-B_I = F(B_I)
10  END
11  FOR   ITER = 1 TO MAXITER
     // SELECTION PHASE
12  IF  RAND_3(0,1) < RAND_4(0,1)  THEN
13  PREDATOR = A, FITNESS-PREDATOR = FITNESS-A, KEY = 1
14  ELSE
15  PREDATOR = B, FITNESS-PREDATOR = FITNESS-B, KEY = 2
16  END
17  IF  RAND_5(0,1) < RAND_6(0,1) THEN PREY = A  ELSE  PREY = B  END
18  PREY = PERMUTE (PREY)
19  IF RAND_7(0,1) < RAND_8(0,1) THEN
20  R=4X RAND_9(0,1)X(RAND_10(0,1)- RAND_11(0,1) )
21  ELSE
22  R ~ Γ( 4X RAND_12(0,1), 1)
23  END
24  M_1:N,1:D = 1.0
25  FOR I =1 TO N
26  FOR J = 1 TO D
27  IF  RAND_13(0,1) < ( P X  RAND_14(0,1)) THEN
28  M_RNDINT(N),RNDINT(D)= 0
29  END
30  END
31  END
32  IF  RAND_15(0,1) <  ( P X RAND_16(0,1) ) THEN
33  FOR  I = 1 TO N
34  FOR J = 1 TO D
35  IF RAND_17(0,1) <  ( P X RAND_18(0,1) ) THEN
36  M_I,J = 1.0
37  ELSE
38  M_I,J = 0.0
39  END
40  END
41  END
42  END
43  FOR  I = 1 TO N
44  IF Σ M_I = D THEN  M_I,RNDINT(D) = 0 END
45  END
46  // MUTATION
47  X = PREDATOR + R X (PREY - PREDATOR )
48  FOR  I = 1 TO N
49  FOR  J = 1 TO D
50  // CROSSOVER
51  IF M_I,J > 0 THEN X_I,J= PREDATOR_I,J END
52  END
53  END
54  // BOUNDARY CONTROL
55  FOR  I = 1 TO N
56  FOR  J = 1 TO D
57  IF  (X_I,J < LOW_J)  THEN
58        X_I,J  = (RAND_19(0,1) X LOW_J ) + (1 - RAND_19(0,1)) X G_BEST,J
59  ELSE IF (X_I,J > UP_J)  THEN
60              X_I,J  = (RAND_20(0,1) X LOW_J ) + (1 – RAND_20(0,1)) X G_BEST,J
61  END
62  END
63  END
64   // SELECTION (UPDATE)
65  FOR I = 1 TO N
66  IF  F(X_I)  < FITNESS-PREDATOR_I  THEN PREDATOR_I = X_I ,FITNESS-PREDATOR_I = F(X_I) END
67  END
68  IF  KEY = 1  THEN
69       A = PREDATOR,  FITNESS-A = FITNESS-PREDATOR
70  ELSE
71        B = PREDATOR,  FITNESS- B = FITNESS-PREDATOR
72  END
73        FITNESS-BEST = ARGMIN (FITNESS - PREDATOR)  best ∈ {1, 2, 3, .., N}
74  IF  FITNESS-BEST < GLOBALMINIMUM  THEN
75       GLOBALMINIMUM = FITNESS-BEST
76       GLOBALMINIMIZER = PREDATOR-BEST
77  END
78  END
79  OUTPUT DATA : GLOBALMINIMUM = F (GLOBALMINIMIZER
```

4. Experimental Studies on Improved Artificial Cooperative Search Algorithm

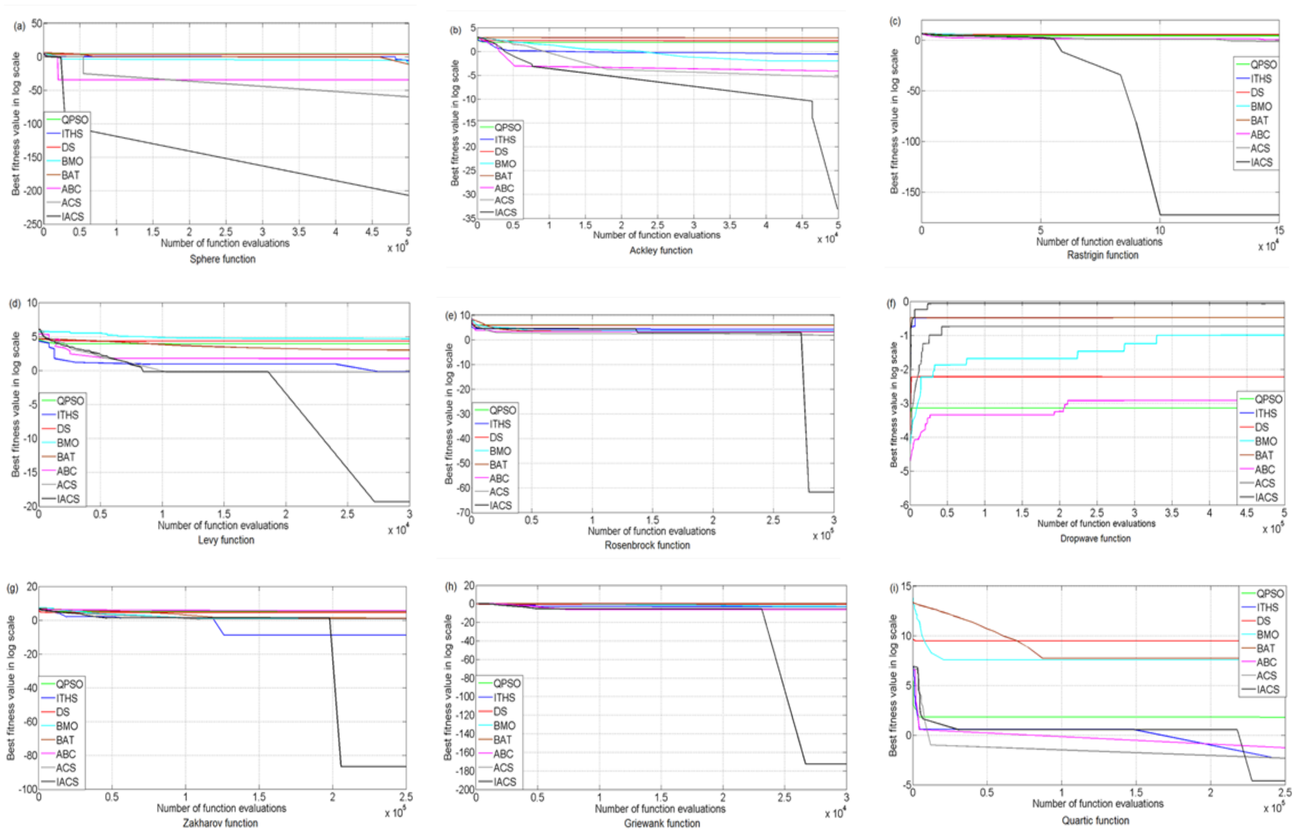
In this section, twelve widely known 50 dimensional optimization test functions have been carried out in order to assess the performance of the IACS algorithm over new emerged metaheuristics such as Quantum behaved Particle Swarm Optimization (QPSO) [53-54], Intelligent Tuned Harmony Search (ITHS) [55], Artificial Bee Colony (ABC)[10], Differential Search (DS) [56], Bird Mating Optimizer (BMO) [57], Bat Algorithm (BAT) [58] and Artificial Cooperative Search (ACS) [44]. Table 2 shows the 12 benchmark problems, which are composed of unimodal, and multimodal test functions those require expensive computational effort due to both their multidimensionality and inherent complex nature they exhibit. Due to the stochastic discrepancy, 50 consecutive algorithm runs along with 500,000 function evaluations have been performed for each test function for all mentioned algorithms. Algorithms have been developed in Java and run on Intel with 2.50 GHz CPU and 6.0 GB RAM. Table 3 gives the statistical results for all mentioned optimizers in terms of mean and standard deviation values. IACS algorithm finds global optimum of Sphere, Rastrigin, Griewank and Step functions in each algorithm run and outperforms other algorithms with regards to statistical results it attains over 11 out of 12 test functions. Concerning the best results obtained after consecutive algorithms runs, convergence performance of all above-mentioned algorithms have been compared with IACS algorithm in Fig. 2. It is clear that IACS is more quicker than other algorithms since in each test function, except for Pathologic function in Fig. 2(k), IACS is getting closer to the optimum point while others remain stagnant and are far away from the optimum.

Table 2. Mathematical representations of the benchmark functions

No	Range	Dim. (N)	Function	Formulation	Optimum
f_1	$-5.12 \leq x_i \leq 5.12$	50	Sphere	$f(x)=\sum_{i=1}^{N} x_i^2$	0.0
f_2	$-10.0 \leq x_i \leq 10.0$	50	Ackley	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right) - \exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos\left(2\pi x_i\right)\right) + 20 + e$	0.0
f_3	$-5.12 \leq x_i \leq 5.12$	50	Rastrigin	$f(x) = 10N + \sum_{i=1}^{N}\left[x_i^2 - 10\cos\left(2\pi x_i\right)\right]$	0.0
f_4	$-10.0 \leq x_i \leq 10.0$	50	Levy	$f(x) = \sin^2(\pi y_1) + \sum_{i=2}^{N-1}\left[\begin{array}{c}(y_i-1)^2\\ \left(1+10.0\sin^2(\pi y_i+1.0)\right)\end{array}\right] + \left(y_N-1\right)^2\left(1.0+10.0\sin^2(2\pi y_N)\right)$, $y_i = 1.0 + \frac{x_i - 1.0}{4},\ i=1,2,..,N$	0.0
f_5	$-2.0 \leq x_i \leq 2.0$	50	Rosenbrock	$f(x) = \sum_{i=1}^{N-1}\left[100(x_i^2 - x_{i+1}) + (x_i-1)^2\right]$	0.0
f_6	$-5.12 \leq x_i \leq 5.12$	50	Dropwave	$f(x) = -\frac{1.0 + \cos\left(12.0\sqrt{\sum_{i=1}^{N} x_i^2}\right)}{\left(2.0 + 0.5\sum_{i=1}^{N} x_i^2\right)}$	-1.0
f_7	$-10.0 \leq x_i \leq 10.0$	50	Zakharov	$f(x) = \sum_{i=1}^{N} x_i^2 + \left(\sum_{i=1}^{N} 0.5ix_i\right)^2 + \left(\sum_{i=1}^{N} 0.5ix_i\right)^4$	0.0
f_8	$-600.0 \leq x_i \leq 600.0$	50	Griewank	$f(x) = \sum_{i=1}^{N}\frac{x_i^2}{4000} - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0.0
f_9	$-1.28 \leq x_i \leq 1.28$	50	Quartic	$f(x) = \sum_{i=1}^{N} ix_i^4 + random(0,1)$	0.0
f_{10}	$-600.0 \leq x_i \leq 600.0$	50	Step	$f(x) = \sum_{i=1}^{N}\left(\lfloor x_i + 0.5\rfloor\right)^2$	0.0
f_{11}	$-600.0 \leq x_i \leq 600.0$	50	Pathologic	$f(x) = \sum_{i=1}^{N-1}\left(0.5 + \frac{\sin^2\left(\sqrt{100x_i^2 + x_{i+1}^2}\right) - 0.5}{1 + 0.001\left(x_i^2 - 2x_ix_{i+1} + x_{i+1}^2\right)^2}\right)^2$	0.0
f_{12}	$-50.0 \leq x_i \leq 50.0$	50	Alpine	$f(x) = \sum_{i=1}^{N}\left\lvert x_i\sin\left(x_i\right) + 0.1x_i\right\rvert$	0.0

Table 3. Statistical results for 50 Dimensional benchmark test functions

	Mean D.±Standard D. Sphere	**Mean D.±Standard D. Ackley**	**Mean D.±Standard D. Rastrigin**	**Mean D.±Standard D. Levy**
QPSO	9.52E+01±4.92E+01	8.11E+00±8.63E+00	2.04E+02±2.17E+02	5.93E+01±3.07E+01
ITHS	2.32E-02±2.09E-02	7.07E-02±2.87E-02	4.75E+01±3.88E+01	4.22E-02±3.58E-02
ABC	9.49E-16±1.70E-16	5.04E-14±6.66E-15	0.00E+00±0.00E+00	3.86E-09±1.03E-16
DS	1.61E+01±3.18E+01	1.03E+01±1.10E+00	3.17E+02±2.32E+01	2.51E+02±5.59E+00
BMO	3.74E-04±3.96E-04	2.36E-02±6.92E-03	2.06E+02±3.97E+01	3.47E+01±2.53E+01
BAT	3.82E-05±3.35E-05	1.73E+01±1.43E+00	3.68E+02±4.32E+01	9.30E+01±2.08E+01
ACS	4.12E-25±3.28E-45	1.36E-14±3.21E-15	0.00E+00±0.00E+00	3.86E-09±1.43E-23
IACS	0.00E+00±0.00E+00	4.44E-15±0.00E+00	0.00E+00±0.00E+00	3.86E-09±4.07E-24
	MeanD.±StandardD. Rosenbrock	**MeanD.±StandardD. Dropwave**	**MeanD.±StandardD. Zakharov**	**MeanD.±StandardD. Griewank**
QPSO	8.52E+02±4.17E+02	-4.53E-02±2.18E-02	2.22E+02±2.79E+02	8.15E-01±4.64E-01
ITHS	3.82E+01±2.48E-01	-4.34E-01±1.71E-01	5.25E-02±1.23E-01	1.39E-03±1.55E-03
ABC	2.20E+01±9.45E+00	-5.51E-02±2.96E-02	2.91E+02±4.10E+01	1.49E-16±1.72E-16
DS	2.54E+02±5.88E+01	-1.56E-01±2.92E-02	1.37E+02±3.4E+01	8.88E-01±6.64E-02
BMO	6.11E+01±3.24E+01	-3.32E-01±5.10E-02	1.75E+00±1.45E+00	9.08E-03±8.93E-03
BAT	2.20E+01±7.48E+01	-5.56E-01±2.25E-01	1.03E-04±5.78E-05	8.33E-03±2.43E-02
ACS	2.20E+01±5.71E+00	-4.77E-01±5.90E-02	3.45E-02±1.18E-02	0.00E+00±0.00E+00
IACS	1.44E-06±1.54E-06	-6.34E-01±1.42E-01	1.81E-15±3.62E-31	0.00E+00±0.00E+00
	MeanD.±StandardD. Quartic	**MeanD.±StandardD. Step**	**MeanD.±StandardD. Pathologic**	**MeanD.±StandardD. Alpine**
QPSO	1.03E+01±7.83E+00	3.74E+01±1.23E+01	1.66E+00±4.78E-01	1.31E+01±3.31E+00
ITHS	2.00E-02±1.69E-01	1.57E-02±1.24E-02	9.76E-01±2.80E-01	1.34E-01±6.61E-03
ABC	6.30E-02±2.51E-02	1.05E-16±1.52E-16	1.75E-03±1.60E-03	1.95E-05±2.73E-05
DS	2.56E+04±1.03E+04	3.80E+01±7.75E+00	6.93E+00±5.27E-01	3.06E+01±3.98E+00
BMO	7.78E-01±1.83E-01	7.52E-04±1.93E-04	8.35E+00±5.02E-01	1.04E+01±2.82E+00
BAT	5.31E+01±1.88E+01	3.98E-05±1.44E-05	4.29E+00±1.62E+00	2.43E+01±8.13E+00
ACS	3.84E-02±5.66E-03	0.00E+00±0.00E+00	1.80E-02±4.21E-03	1.37E-08±3.92E-08
IACS	3.66E-02±7.01E-03	0.00E+00±0.00E+00	1.58E-02±5.01E-03	7.95E-09±1.68E-08



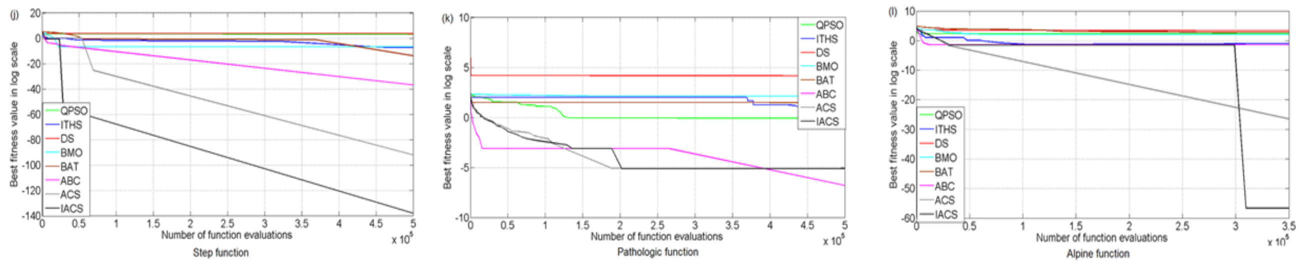


Figure 2. Convergence profiles of the optimizers for (a) Sphere function, (b) Ackley function, (c) Rastrigin function, (d) Levy function, (e) Rosenbrock function, (f) Dropwave function, (g) Zakharov function, (h) Griewank function, (i) Quartic function, (j) Step function, (k) Pathologic function, and (l) Alpine function

5. Simulation and Analysis

In this section, IACS algorithm has been assigned to 13-, 38-, 40-, and 140- unit generation systems to verify its applicability and feasibility on ELD problems. IACS is implemented using Java executing Pentium Core i5 CPU @ 2.5 GHz and 6.0 GB RAM on a personal computer.

5.1 Case Study 1: 13- Unit Test System

This case deals with 13-generating units which takes into account valve effects and prohibited zones without considering transmission losses. As number of generation sites increases, complexity of the system is improved owing to the non-linear characteristic of valve point loading effects. This non-linear behavior increases the number of local optimum therefore finding global optimum of the problem significantly becomes a challenging process. In this case, total load demand of 1800.0 MW and 2520.0 MW test systems are studied. Due to the stochastic characteristic of metaheuristic algorithms, 50 trial runs have been performed along with 50,000 function evaluations for both IACS and ACS algorithms. Problem data for 1800.0 MW test system can be found in Sinha et al. [59]. Table 5 reports the statistical results obtained by HGA[60], FA[31], BF-NM [61], MDE[62], IPSO-TVAC [63], SDE [64], MsEBBO [8], MsEBBO/sin [8], MsEBBO/mig [8], and ACS-based algorithms for 1800.0 MW test system. From Table 4, it is seen that minimum fuel cost value (17,954.091 $/h) obtained by IACS is lower than those acquired by other methods available in the literature. Table 5 lists the best generation cost results of above-mentioned literature approaches and their corresponding power generation rates. Table 6 reports statistical analysis obtained by FAMPSO[65], HHS [66], ACO [1], ICA-PSO [67], IPSO-TVAC [63] and ACS-based algorithms for 2520.0 MW test system. Table 6 clarifies that however little improvement have been made by IACS over ACS algorithm on minimum generation cost value, IACS not only attains the best result among the other algorithms but also it surpasses the remaining algorithms in terms of robustness. This behavior indicates the supremacy of the proposed algorithm. Table 7 lists the comparison of the optimal solution in the literature.

Table 4 Statistical analysis for the 13-unit test system with total load demand of 1800 MW

	Minimum cost ($/h)	Maximum cost ($/h)	Average cost ($/h)	Standart dev.
MsEBBO/mig [8]	17,963.8317	17,972.8427	17,969.6001	4.3367
HGA [60]	17,963.83	N/A	17,988.04	N/A
FA [31]	17,963.83	18,168.80	18,029.16	148.54
MsEBBO/sin [8]	17,963.8292	17,972.8105	17,967.0705	4.1876
MsEBBO [8]	17,963.8292	17,969.0323	17,964.0468	1.9215
BF-NM [61]	17 960.4998	N/A	17 969.8569	2.0538
MDE [62]	17,960.39	17,969.09	17,967.19	N/A
IPSO-TVAC [63]	17,960.3703	N/A	N/A	N/A
SDE [64]	17,960.37	N/A	N/A	N/A
ACS	17,960.362	17,969.57	17,965.89	3.13
IACS	17,954.091	17,968.13	17,961.56	4.76

N/A means "not available"

Table 5 Best power outputs for the 13-unit test system with total load demand of 1800 MW

Unit (MW)	MsEBBO/sin [8]	MsEBBO [8]	MDE [62]	IPSO-TVAC [63]	SDE[64]	ACS	IACS
P_1	628.3185	628.3185	628.318	628.3185	628.32	628.3184	538.5593
P_2	222.7492	149.5997	149.594	149.5996	149.60	149.5995	224.3994
P_3	149.5997	222.7492	222.758	222.7489	222.75	222.7494	149.5996
P_4	109.8666	109.8666	109.8665	109.8666	109.87	109.8665	109.8665
P_5	109.8666	60.0000	109.8665	109.8666	109.87	60.0000	109.8665
P_6	109.8666	109.8666	109.8665	109.8666	109.87	109.8665	109.8665
P_7	109.8666	109.8666	109.8665	109.8666	60.00	109.8665	109.8665
P_8	109.8666	109.8666	60.0000	109.8666	109.87	109.8665	109.8665
P_9	60.0000	109.8666	109.8665	60.0000	109.87	109.8665	109.8665
P_{10}	40.0000	40.0000	40.0000	40.0000	40.00	40.0000	40.0000
P_{11}	40.0000	40.0000	40.0000	40.0000	40.00	40.0000	77.3999
P_{12}	55.0000	55.0000	55.0000	55.0000	55.00	55.0000	55.0000
P_{13}	55.0000	55.0000	55.0000	55.0000	55.00	55.0000	55.0000
P_{Total}	1800.00	1800.00	1800.00	1800.00	1800.00	1800.00	1800.00
Total cost ($/h)	17,963.82	17,963.82	17,960.39	17960.3703	17,960.37	17,960.36	**17,954.091**

Table 6 Comparison of the statistical results for the 13-unit test system with total load of 2520.0 MW

	Minimum Cost ($/h)	Maximum cost ($/h)	Average cost ($/h)	Standart dev.
FAMPSO [65]	24,169.9176	24,169.9176	24,169.9176	N/A
HHS [66]	24,169.90	24,196.9	24,169.9	N/A
ACO [1]	24,169.63	24,195.91	24,182.79	7.86
ICA-PSO [67]	24,168.91	24,184.92	24,175.24	N/A
IPSO-TVAC [63]	24,166.8	24,169.41	24,167.37	N/A
ACS	24,164.058	24,177.86	24,165.72	4.046
IACS	**24,164.046**	24,164.046	24,164.046	2.45E-8

N/A means "not available"

Table 7 Best power outputs for the 13-unit system with total load of 2520.0 MW

Unit (MW)	FAMPSO [65]	HHS [66]	ACO [1]	ICA-PSO [67]	IPSO-TVAC [63]	ACS	IACS
P1	628.3185	628.3185	628.32	628.32	628.319	628.3185	628.3185
P2	299.1993	299.1993	299.06	299.19	299.199	299.1993	299.1993
P3	299.1993	299.1993	299.17	294.51	295.878	294.4839	294.4821
P4	159.7331	159.7331	159.73	159.73	159.265	159.7331	159.7331
P5	159.7331	159.7331	159.73	159.73	159.73	159.7331	159.7331
P6	159.7331	159.7331	159.73	159.73	159.73	159.7331	159.7331
P7	159.7331	159.7331	159.73	159.73	159.73	159.7331	159.7331
P8	159.7331	159.7331	159.73	159.73	159.73	159.7331	159.7331
P9	159.7331	159.7331	159.73	159.73	159.73	159.7331	159.7331
P10	77.3999	77.3999	75.47	114.80	77.363	77.3999	77.3999
P11	77.3999	77.3999	77.33	77.40	77.397	77.3999	77.3999
P12	87.6845	87.6845	92.10	55.00	92.397	92.3999	92.3999
P13	92.3999	92.3999	90.59	92.40	91.517	92.3999	92.3999
PTotal (MW)	2520.0	2520.0	2520.0	2520.0	2520.0	2520.0	2520.0
Total cost ($/h)	24,169.9176	24,169.9	24,169.63	24,168.91	24,168.8	24,164.058	24,164.046

5.2 Case Study 2: 38-Unit Test System

A test system with 38-generation units is considered to test the actual performance of IACS algorithm on non-convex problems. Fuel costs are represented by quadratic cost functions, transmission losses are not taken into account and total power demand is set to 6000.0 MW for this case. System parameters are taken from Liang and Glover [9]. Table 8 compares the optimum results extracted by DE/BBO [68], PSO-TVAC [69], NPSO [69] and ACS based algorithms. Table 10 gives the statistical analysis for aforementioned algorithms in terms of minimum, maximum and average cost values. As seen from Table 10, IACS algorithm not only finds lower fuel cost values than the other methods but also it is so robust and consistent such that the worst fuel cost value obtained by IACS is much better than the best fuel cost value acquired by ACS. Both algorithms shows similar convergence characteristics until 32,653 function evaluations then both of them remain in stagnation till the end of iterations. However, ACS algorithm is trapped in the local optimum with the corresponding fuel cost value of 9,417,205.08 ($/h). Fig. 3 illustrates the set of optimum solutions obtained after 50 algorithm run for this case.

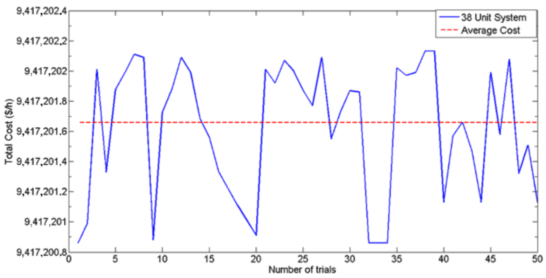


Figure 3. Set of optimum results obtained by IACS for 38-unit test system

Table 8 Comparison of the best results for 38-unit system

Output (MW)	IACS	ACS	DE/BBO [68]	PSO-TVAC [69]	NPSO [69]
P_1	426.5920	426.5925	426.6060	443.659	550.000
P_2	426.5920	426.5925	426.6060	342.956	512.263
P_3	429.6491	429.6497	429.6631	433.117	485.733
P_4	429.6591	429.6596	429.6631	500.000	391.083
P_5	429.6491	429.6496	429.6631	410.539	443.846
P_6	429.6491	429.6496	429.6631	482.864	358.398
P_7	429.6491	429.6496	429.6631	409.483	415.729
P_8	429.6491	429.6496	429.6631	446.079	320.816
P_9	114.0000	114.0000	114.0000	119.566	115.347
P_{10}	114.0000	114.0000	114.0000	137.274	204.422
P_{11}	119.7621	119.7626	119.7680	138.933	114.000
P_{12}	127.0666	127.0686	127.0728	155.401	249.197
P_{13}	110.0000	110.0000	110.0000	121.719	118.886
P_{14}	90.0000	90.0000	90.0000	90.924	102.802
P_{15}	82.0000	82.0000	82.0000	97.941	89.039
P_{16}	120.0000	120.0000	120.0000	128.106	120.000
P_{17}	159.5963	159.5973	159.5980	189.108	156.562
P_{18}	65.0000	65.0000	65.0000	65.000	84.265
P_{19}	65.0000	65.0000	65.0000	65.000	65.041
P_{20}	271.9999	271.9999	272.0000	267.422	151.104
P_{21}	271.9999	271.9999	272.0000	221.383	226.344
P_{22}	259.9999	259.9999	260.0000	130.804	209.298
P_{23}	130.6431	130.6431	130.6486	124.269	85.719
P_{24}	10.0000	10.0000	10.0000	11.535	10.000
P_{25}	113.3012	113.3012	113.3050	77.103	60.000
P_{26}	88.0647	88.0647	88.0669	55.018	90.489
P_{27}	37.5037	37.5035	37.5051	75.000	39.670
P_{28}	20.0000	20.0000	20.0000	21.682	20.000
P_{29}	20.0000	20.0000	20.0000	29.829	20.995
P_{30}	20.0000	20.0000	20.0000	20.326	22.810
P_{31}	20.0000	20.0000	20.0000	20.000	20.000
P_{32}	20.0000	20.0000	20.0000	21.840	20.416
P_{33}	25.0000	25.0000	25.0000	25.620	25.000
P_{34}	18.0000	18.0000	18.0000	24.261	21.319
P_{35}	8.0000	8.0000	8.0000	9.667	9.122
P_{36}	25.0000	25.0000	25.0000	25.000	25.184
P_{37}	21.7815	21.7818	21.7820	31.642	20.000
P_{38}	21.1929	21.1928	21.0621	29.935	25.104
($/h)	**9,417,200.86**	9,417,205.98	9,417,235.786	9,500,448.30	9,516,448.31

Table 10 Statistical results and comparison for the 38 unit system.

	Minimum cost ($/h)	Maximum cost ($/h)	Average cost ($/h)
SPSO [69]	9,543,984.77	N/A	N/A
PSO-Crazy[69]	9,520,024.60	N/A	N/A
NPSO [69]	9,516,448.31	N/A	N/A
PSO-TVAC [69]	9,500,448.30	N/A	N/A
DE-BBO [68]	9,417,235.78	N/A	N/A
ACS	9,417,205.98	9,417,206.28	9,417,206.19
IACS	**9,417,200.86**	9,417,202.13	9,417,201.66

N/A means "not available"

5.3 Case Study 3: 40-Unit Test System

In this section, the proposed algorithm has been applied on a power system consists of 40 generating units incorporating valve loading effects. Total power demand is set to 10,500 MW for this case. Input parameters of the cost functions for all generating units are referred to the case study given in Coelho and Mariani [5]. 50 trial runs along with 500,000 function evaluations have been performed due to the stochastic nature of the proposed metaheuristic algorithm. Optimal fuel cost values obtained by FA [31], PS [10], BBO [68], SOH-PSO [70] along with corresponding generator loads have been reported in Table 11. Table 12 gives the detailed comparison of the proposed algorithm and literature studies in terms of minimum, maximum and mean of the generation cost value. Table 12 clarifies that proposed IACS algorithm surpasses other remaining methods in terms of minimum fuel cost values. Table 13 reports the convergence frequency of the best results obtained by ACS, IACS and other optimizers available in the literature . IACS and ACS algorithms have obtained fuel cost values between 120.0 x 10^3 ($/h) and 121.5 x 10^3 ($/h) in each algorithm run which shows their superiority over other optimization methods in terms of solution accuracy and robustness. Similar evolution characteristics have been observed for both algorithms however ACS remains stable after 7813 function evaluations and converges to local optimum point. Fig. 4 shows the sequence of optimum results generated after 50 algorithm runs.

Table 11 Optimal dispatch results for 40 unit systems for total power demand of 10,500 MW

Output (MW)	**IACS**	**ACS**	**FA [31]**	**PS [10]**	**BBO [68]**	**SOH-PSO [70]**
P_1	110.8687	110.7998	110.8099	110.8051	110.8158	110.80
P_2	110.0013	110.7998	110.8059	110.8051	111.0896	110.80
P_3	97.3999	97.3999	97.4023	97.4023	97.4026	97.40
P_4	179.7331	179.7331	179.7332	179.7332	179.7549	179.73
P_5	92.4706	92.7561	92.7070	92.7070	88.2083	87.80
P_6	139.9999	139.9999	140.0000	140.0000	139.9886	140.00
P_7	259.5996	259.5996	259.6004	259.6004	259.5935	259.60
P_8	284.5996	284.5996	284.6004	284.6004	284.6174	284.60
P_9	284.5996	284.5996	284.6004	284.6004	284.6479	284.60
P_{10}	130.0000	130.0000	130.0028	130.0028	130.0298	130.00
P_{11}	168.7998	168.7998	168.8008	168.8008	94.01459	94.00
P_{12}	168.7998	168.7998	168.8008	168.8008	94.26367	94.00
P_{13}	214.7597	214.7597	214.7606	214.7606	304.5153	304.52
P_{14}	394.2793	394.2793	304.5204	304.5204	394.2642	304.52
P_{15}	394.2793	394.2793	394.2801	394.2801	304.5057	394.28
P_{16}	304.5195	304.5195	394.2801	394.2801	394.2472	394.28
P_{17}	489.2793	489.2793	489.2801	489.2801	489.3273	489.28
P_{18}	489.2793	489.2793	489.2801	489.2801	489.3047	489.28
P_{19}	511.2793	511.2793	511.2817	511.2817	511.3087	511.28
P_{20}	511.2793	511.2793	511.2817	511.2817	511.2495	511.27
P_{21}	523.2793	523.2793	523.2793	523.2793	523.3217	523.28
P_{22}	523.2793	523.2793	523.2793	523.2793	523.3144	523.28
P_{23}	523.2793	523.2793	523.2832	523.2832	523.3629	523.28
P_{24}	523.2793	523.2793	523.2832	523.2832	523.2883	523.28
P_{25}	523.2793	523.2793	523.2793	523.2793	523.2989	523.28
P_{26}	523.2793	523.2793	523.2793	523.2793	523.2802	523.28
P_{27}	10.0000	10.0000	10.0000	10.0008	10.0281	10.00
P_{28}	10.0000	10.0000	10.0000	10.0028	10.0032	10.00
P_{29}	10.0000	10.0000	10.0000	10.0028	10.0288	10.00
P_{30}	87.8169	87.8012	87.8008	87.8008	88.1459	97.00
P_{31}	189.9999	189.9999	189.9989	189.9989	189.9913	190.00
P_{32}	189.9999	189.9999	189.9989	189.9989	189.9888	190.00
P_{33}	189.9999	189.9999	189.9989	189.9989	189.9998	190.00
P_{34}	164.7998	164.8036	164.8036	164.8036	164.8452	185.20
P_{35}	164.7998	164.8036	164.8036	164.8036	192.9876	164.80
P_{36}	164.7998	164.8036	164.8036	164.8036	199.9876	200.00
P_{37}	109.9999	109.9999	110.0000	109.9989	109.9941	110.00
P_{38}	109.9999	109.9999	110.0000	109.9989	109.9992	110.00
P_{39}	109.9999	109.9999	110.0000	109.9989	109.9833	110.00
P_{40}	511.2793	511.2793	511.2794	511.2817	511.2794	511.28
($/h)	**121,371.5603**	121,414.6091	121,415.0522	121,415.14	121,479.5029	121,501.14

Table 12 Statistical results for the 40 unit test system

	Minimum cost ($/h)	Maximum cost ($/h)	Average cost ($/h)
PSO [73]	121,735.47	123,467.40	122.513.91
NPSO-LRS [75]	121,664.43	122,981.59	122,209.31
SOH-PSO [70]	121,501.14	122,446.30	121,853.57
BBO [68]	121,479.50	121,688.66	121,512.06
PS [10]	121,415.14	125,486.29	122,332.65
FA [31]	121,415.05	121,424.56	121,416.57
ACS	121,414.60	121,468.63	121,426.73
IACS	**121,371.56**	121,450.32	121,423.33

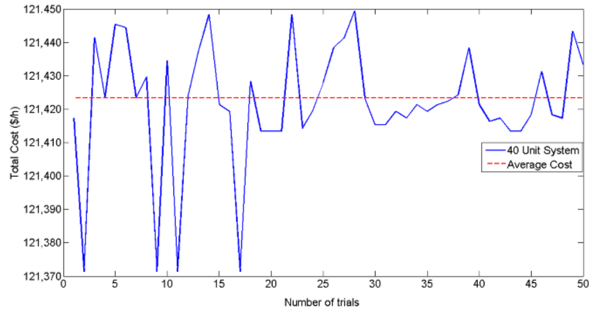


Figure 4. Sequence of optimum results obtained by IACS for 40-unit test system

Table 13 Convergence frequency of the algorithms for 40 generator system

	Range of total generation cost (x 10^3 , \$/h)						
	120.0 – 121.5	121.5 – 122.5	122.5 – 123.0	123.0 - 123.5	123.5 – 124.0	124.0 - 124.5	124.5 – 125.0
IACS	**50**	0	0	0	0	0	0
ACS	**50**	0	0	0	0	0	0
BBO [68]	38	12	0	0	0	0	0
QPSO [72]	2	27	20	1	0	0	0
SOH-PSO [70]	0	50	0	0	0	0	0
NPSO-LRS [75]	0	40	10	0	0	0	0
NPSO [73]	0	37	13	0	0	0	0
PSO-LRS [73]	0	26	17	7	0	0	0
CBPSO-RVM [74]	41	8	1	0	0	0	0
IFEP [59]	0	0	11	25	9	2	2

5.4 Case Study 4: Large Scale Application on Korea Power System

To test the efficiency of the proposed algorithm on a large scale application, numerical experiments have been conducted on the Korean power system which consists of 140 thermal generating units with ramp rate limits, valve point effects and prohibited operating zones. Total power load for this system is set to 49,342 MW. Input data for this case is obtained from Park et al. [47]. 1,000,000 function evaluations have been made owing to the high dimensionality of the problem. Due to the stochastic discrepancy, both ACS and IACS have performed 50 consecutive algorithm runs. Park et al. [47] proposed four PSO based algorithms including CTPSO, CSPSO, COPSO, and CCPSO for successful solution of this case study. In addition, Dalvand et al.[71] propounded Group Search Optimizer (GSO) and Continuous Quick Group Search Optimizer(CQGSO) in order to solve large scale Korea power system problem. Coelho et al.[26] proposed Differential Evolution algorithm combined with truncated Levy flight random walks and population diversity measure (DEL) to solve this case study. Table 14 compares the optimum solutions acquired by literature studies discussed above with the best results of ACS and IACS algorithms. As reported in Table 14, IACS algorithm surpasses other algorithms concerning the minimum fuel cost values. Table 15 gives the best results obtained by both ACS and IACS algorithms. According to the Table 15, IACS algorithm succeeds in finding much better generation cost values than ACS algorithm with the minimum fuel cost of 1,657,956.80 \$/h. Although both algorithms follow similar trends until the end of iterations, ACS is first to saturate and get trapped in a local optimum solution. After all, it can be concluded that IACS proves its efficiency in solving high dimensional optimization problems.

Table 14 Statistical results for the 140 unit Korean power generation system

Methods	Minimum cost (\$/h)	Maximum cost (\$/h)	Average cost (\$/h)	Std.dev.
CTPSO [47]	1,657,962.73	1,658,002.79	1,657,964.06	7.3150
CSPSO [47]	1,657,962.73	1,657,962.85	1,657,962.74	0.0235
COPSO [47]	1,657,962.73	1,657,962.73	1,657,962.73	0.0002
CCPSO [47]	1,657,962.73	1,657,962.73	1,657,962.73	0.0000
GSO [71]	1,728,151.16	1,753,229.56	1,745,514.99	N/A
CQGSO [71]	1,657,962.72	1,657,962.77	1,657,962.74	N/A
DEL [26]	1,657,962.7166	1,651,518.6719	1,658,001.7003	57.9836
ACS	1,658,002.72	1,658,756.32	1,658,101.39	219.30
IACS	**1,657,956.80**	1,657,966.74	1,657,960.85	5.6328

N/A means “not available”

Table 15 Best results of the ACS and IACS algorithms for 140-unit Korean power generation system

Unit	ACS	IACS	Unit	ACS	IACS	Unit	ACS	IACS	Unit	ACS	IACS
C_01	118.9999	118.9999	C_36	499.9999	499.9999	L_29	141.6208	141.6247	N_15	880.8999	880.8999
C_02	163.9999	163.9999	C_37	240.9999	240.9999	L_30	325.4955	388.3854	N_16	873.6999	873.6999
C_03	189.9999	189.9999	C_38	240.9999	240.9999	L_31	195.0000	195.0000	N_17	877.3999	877.3999
C_04	189.9999	189.9999	C_39	773.9999	773.9999	L_32	223.7982	190.2913	N_18	871.6999	871.6999
C_05	168.5398	168.5398	C_40	768.9999	768.9999	L_33	234.1251	217.1671	N_19	864.8000	864.7999
C_06	189.9999	189.9999	LNG_1	3.0000	3.0000	L_34	256.8416	259.3935	N_20	881.9999	881.9999
C_07	489.9999	489.9999	LNG_2	3.0000	3.0000	L_35	399.1424	383.9276	OIL_01	94.0000	94.0000
C_08	489.9999	489.9999	L_1	249.9999	249.9999	L_36	330.0000	330.0000	OIL_02	94.0000	94.0000
C_09	495.9999	495.9999	L_2	249.9999	249.9999	L_37	530.9999	530.9999	OIL_03	94.0000	94.0000
C_10	495.9999	495.9999	L_3	249.9999	249.9999	L_38	530.9999	530.9999	OIL_04	244.0000	244.0000
C_11	495.9999	495.9999	L_4	249.9999	249.9999	L_39	541.9999	541.9999	OIL_05	244.0000	244.0000
C_12	495.9999	495.9999	L_5	249.9999	249.9999	L_40	56.0000	56.0000	OIL_06	244.0000	244.0000
C_13	505.9999	505.9999	L_6	249.9999	249.9999	L_41	115.0000	115.0000	OIL_07	95.0000	95.0000
C_14	508.9999	508.9999	L_7	249.9999	249.9999	L_42	115.0000	115.0000	OIL_08	95.0000	95.0000
C_15	505.9999	505.9999	L_8	249.9999	249.9999	L_43	115.0000	115.0000	OIL_09	116.0000	116.0000
C_16	504.9999	504.9999	L_9	165.0000	165.0000	L_44	207.0000	207.0000	OIL_10	175.0000	175.0000
C_17	505.9999	505.9999	L_10	165.0000	165.0131	L_45	207.0000	207.0000	OIL_11	2.0000	2.0000
C_18	505.9999	505.9999	L_11	165.0000	165.0000	L_46	175.0000	175.0000	OIL_12	4.0000	4.0000
C_19	504.9999	504.9999	L_12	165.0000	165.0000	L_47	175.0000	175.0000	OIL_13	15.0000	15.0000
C_20	504.9999	504.9999	L_13	180.0000	180.0000	L_48	180.4266	180.4568	OIL_14	9.0000	9.0000
C_21	504.9999	504.9999	L_14	180.0000	180.0000	L_49	175.0000	175.0000	OIL_15	12.0000	12.0000
C_22	504.9999	504.9999	L_15	103.0000	103.0000	N_01	575.3999	575.3999	OIL_16	10.0000	10.0000
C_23	504.9999	504.9999	L_16	198.0000	198.0000	N_02	547.5000	547.4999	OIL_17	112.0000	112.0000
C_24	504.9999	504.9999	L_17	311.9999	311.9999	N_03	836.7999	836.7999	OIL_18	4.0000	4.0000
C_25	536.9999	536.9999	L_18	308.5520	308.6128	N_04	837.4999	837.4999	OIL_19	5.0000	5.0000
C_26	536.9999	536.9999	L_19	163.0000	163.0000	N_05	681.9999	681.9999	OIL_20	5.0000	5.0000
C_27	548.9999	548.9999	L_20	95.0000	95.0000	N_06	719.9999	719.9999	OIL_21	50.0000	50.0000
C_28	548.9999	548.9999	L_21	510.9999	510.9999	N_07	717.9999	719.9999	OIL_22	5.0000	5.0000
C_29	500.9999	500.9999	L_22	510.9999	510.9999	N_08	719.9999	719.9999	OIL_23	42.0000	42.0000
C_30	498.9999	498.9999	L_23	489.9999	489.9999	N_09	963.9999	963.9999	OIL_24	42.0000	42.0000
C_31	505.9999	505.9999	L_24	256.9179	256.9110	N_10	957.9999	957.9999	OIL_25	41.0000	41.0000
C_32	505.9999	505.9999	L_25	489.9999	489.9999	N_11	947.8999	947.8999	OIL_26	17.0000	17.0000
C_33	505.9999	505.9999	L_26	489.9999	489.9999	N_12	933.9999	933.9999	OIL_27	7.0000	7.0000
C_34	505.9999	505.9999	L_27	130.0000	130.0000	N_13	934.9999	934.9999	OIL_28	7.0000	7.0000
C_35	499.9999	499.9999	L_28	339.4395	339.4395	N_14	876.4999	876.4999	OIL_29	26.0000	26.0000
Total power (MW)										**49,342**	**49,342**
Minimum fuel cost ($)										**1,658,002**	**1,657,956**

C=COAL, N=NUCLEAR, L=L

6. Conclusion

This paper introduces Improved Artificial Cooperative Search (IACS) algorithm for successful solution of economic dispatch problems regarding valve point effects, ramp rate limits, transmission losses and prohibited operation zones those all make the operation cost curve non-continuous, non-convex, and highly non-linear. ACS is dual-population based metaheuristic algorithm, which is based on the interaction between two superorganisms as they are migrating and searching more fruitful areas for subsistence. ACS has a fewer control parameters and uses advanced perturbation strategies that ease its implementation on any optimization problem. In order to improve the solution quality and increase convergence rate, a novel local search strategy is incorporated into ACS algorithm. In addition, an evolutionary boundary constraint-handling scheme is utilized to restrict population individuals into the prescribed limits. In order to verify the applicability of the proposed algorithm on multi-dimensional optimization test problems, IACS is benchmarked with twelve widely known benchmark functions. The results obtained from IACS, ACS and some other metaheuristics in the literature showed that IACS performs better than the others. Also, IACS converges faster than the other algorithms. Then, IACS has been applied to 13-, 38- , 40- and 140- unit systems and obtained results over 50 algorithm runs have been compared with recently published ED optimizers. Outcomes of the comparisons indicate that IACS algorithm is very effective and efficient in finding the optimum solution of the economic dispatch problem and can be nominated as an alternative method for solving non-linear ELD problems as well as multi-dimensional real world optimization problems. For a future work, IACS will be implemented on environmental/economical dispatch (EED) problems.

References

[1] S. Pothiya, I. Ngamroo, W. Kongprawechnon, "Ant colony optimization for economic dispatch problem with non-smooth cost functions," Int. J. Elec. Power., vol.32, pp.478-87, 2010.

[2] J.C. Dodu, P. Martin, A. Merlin, J. Pouget, "An optimal formulation and solution of short-range operating problems for a power system with flow constraints," In: Proceedings of the IEEE, vol. 60, pp. 53-54, 1972

[3] R.A. Jabr, A.H. Coonick, B.J. Cory, "A homogeneous linear programming algorithm for the security constrained economic dispatch problem," IEEE T. Power. Syst., vol.15, pp. 930-937, 2000.

[4] C.L. Chen, S.C. Wang, "Branch-and-bound scheduling for thermal generating units," IEEE T. Energy. Conver., vol. 8, pp. 184-86, 1993.

[5] L.d.S. Coelho, V.C. Mariani, "Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect," IEEE T. Power Syst., vol.21, pp.989-96, 2006.

[6] J. Nanda, L. Hari, M.L. Kothari, "Economic emission load dispatch with line flow constraints using a classical technique," IEEE Proc. Generat. Transm. Distrib., vol.141, pp.1-10, 1994.

[7] A.A. El-Keib, H. Ma, J.L. Hart, "Environmentally constrained economic dispatch using the lagrangian relaxation method," IEEE T. Power. Syst., vol.9, pp.1723-1727, 1994.

[8] G. Xiong, D. Shi, X. Duan, "Multi-strategy ensemble biogeography-based optimization for economic dispatch problems," Appl. Energy, vol. 111, pp. 801–811, 2013.
[9] Z.X. Liang, J.D. Glover, "A zoom feature for a dynamic programming solution to economic dispatch including transmission losses," IEEE T. Power. Syst., vol.7, pp.544-550, 1992.
[10] J.S. Al-Sumait, A.K. AL-Othman, J.K. Sykulski, "Application of pattern search method to power system valve-point economic load dispatch," Int. J. Elec. Power., vol.11, pp. 720-30, 2007.
[11] J.O. Kim, D.J. Shin, J.N. Park, C. Singh, "Atavistic genetic algorithm for economic dispatch with valve point effect," Int. J. Elec. Power., vol. 62, pp.201-207, 2002.
[12] C.L. Chiang, "Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels," IEEE T. Power. Syst., vol.20, pp. 1690-1699, 2005.
[13] S. Baskar, P, Subbaraj, M.V.C. Rao, "Hybrid real coded genetic algorithm solution to economic dispatch problem," Comput. Electr. Eng., vol. 29, pp.407-419, 2003.
[14] R.K. Swain, N.C. Sahu, P.K. Hota, "Gravitational search algorithm for optimal economic dispatch," Procedia Technol., vol.6, pp.411-19, 2012.
[15] S. Duman, A.B. Arsoy, N. Yörükeren, "Solution of economic dispatch problem using gravitational search algorithm," In: 7th International Conference on Electrical and Electronics Engineering, Bursa, Turkey, pp.54-59, 2011.
[16] K.P. Wong, C.C. Fong, "Simulated annealing based economic dispatch algorithm.," IEEE Conference Proceedings, vol.140, pp.509-515, 1993.
[17] K.K. Vishwakarma, H.M. Dubey, M. Pandit, B.K. Panigrahi, "Simulated annealing approach for solving economic load dispatch problems with valve point loading effects," Int. J. Eng. Sci. Technol., vol. 4, pp. 60-72, 2012
[18] C.K. Panigrahi, P.K. Chattopadhyay, R.N. Chakrabarti, M. Basu, "Simulated annealing technique for dynamic economic dispatch," Electr. Pow. Compo. Sys, vol.34, pp. 577-586, 2006.
[19] D.N. Jeyakumar, T. Jayabarathi, T. Raghunathan, "Particle swarm optimization for various types of economic dispatch problems," Int. J. Elec. Power., vol . 28, pp. 36-42, 2006.
[20] C.K. Panigrahi, V.R. Pandi, S. Das, "Adaptive particle swarm optimization approach for static and dynamic economic load dispatch," Energy. Convers. Manage., vol. 49, pp.1407-1415, 2008.
[21] J.B. Park, "A particle swarm optimization for economic dispatch with non-smooth cost functions," IEEE T. Power. Syst. vol.20, pp.34-42, 2005.
[22] V. Mahidhar, G. S. Reddy, "Economic load dispatch with valve point effects and ramp rates using new approach in PSO," IJERT, vol.1, pp.1-6, 2012.
[23] S.K. Wang, J.P. Chiou, C.W. Liu, "Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm," IET Gener. Transm. Dis., vol.1, pp.793-803, 2007.
[24] L.d.S. Coelho, A.D.V. Almeida, V.C. Mariani. "Cultural differential evolution approach to optimize the economic dispatch of electrical energy using thermal generators," In: Proc. 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Hamburg, Germany, pp.1378–83, 2008.
[25] L.d.S Coelho, V.C. Mariani, "Improved differential evolution algorithms for handling economic dispatch optimization with generator constraints," Energy Convers. Manage., vol. 48, pp.1631–39, 2007.
[26] L.d.S. Coelho, T.C. Bora, V.C. Mariani, "Differential evolution based on truncated Levy-type flights and population diversity measure to solve economic load dispatch problems," Electr. Pow. Compo. Sys, vol.57, pp.178-188, 2014.
[27] L.d.S. Coelho, V.C. Mariani, "An improved harmony search algorithm for power economic load dispatch," Energy Convers. Manage., vol.50, pp. 2522– 2526, 2009.
[28] P. Chakraborty, G.G. Roy, B.K. Panigrahi, R.C. Bansal, A. Mohapatra, "Dynamic economic dispatch using harmony search algorithm with modified differential mutation operator," Electr. Eng., vol.94, pp.197-205, 2012.
[29] S. Hemamalini, S.P. Simon, "Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions," Electr. Pow. Compo. Sys., vol.38, pp.786-803, 2010.
[30] M. Basu, "Artificial bee colony optimization for multi-area economic dispatch," Int. J. Elec. Power. 2013, vol.49, pp.181–87.
[31] X.S. Yang, S.S.S. Hosseini, A.H. Gandomi, "Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect" Appl. Soft. Comput., vol.12, pp.1180–1186, 2012.
[32] M.H. Sulaiman, M.W. Mustafa, Z.N. Zakaria, O. Aliman, S.R. Abdul Rahim, "Firefly algorithm technique for solving economic dispatch problem," IEEE International Power Engineering and Optimization Conference (PEOCO2012), Melaka, Malaysia, pp.90-95, 2012.
[33] M.H. Sulaiman, H. Daniyal, M.W. Mustafa, "Modified firefly algorithm in solving economic dispatch problems with practical constraints.," IEEE International Conference on Power and Energy (PECon), Kota Kinabalu Sabah, Malaysia, pp.157-61, 2012.
[34] A. Rathinam, R. Phukan, "Solution to economic load dispatch problem based on firefly algorithm and its comparison with BFO,CBFO-S and CBFO-Hybrid," Swarm, Evolutionary, and Memetic Computing Lecture Notes in Computer Science 7677, 57-65, 2012.
[35] T. Nikham, R. Azizipanah-Abarghooee, A. Roosta, "Reserved constrained dynamic economic dispatch: A new fast self-adaptive modified firefly algorithm." IEEE Syst. J. vol.6, pp.635-46, 2012.
[36] B.S. Ganesh, A.S. Reddy, "Teaching learning based optimization for economic dispatch problem with valve point loading effect," IJEAR, vol.4, pp. 9-15, 2014.
[37] T. Nikham, R. Azizipanah-Abarghooee, J. Aghaei, "A new modified teaching-learning algorithm for reserve constrained dynamic economic dispatch," IEEE T. Power Syst., vol.28, pp.749-63, 2013.
[38] M. Basu, A. Chowdhury, "Cuckoo search algorithm for economic dispatch," Energy, vol. 60, pp. 99-108, 2013.
[39] D.N. Vo, P. Schegner, W. Ongsakul, "Cuckoo search algorithm for non-convex economic dispatch," IET Gener. Transm. Dis., vol.7, pp. 645–54, 2013.
[40] P.K. Roy, S.P. Ghoshal, S.S. Thakur, "Biogeography-based optimization for economic load dispatch problems," Electr. Pow. Compo. Sys. vol.38, pp.166-181, 2009.
[41] B, Khokhar, K.P. Singh Parmar, .S Dahiya, "Application of biogeography-based optimization for economic dispatch problems," IJCA, vol.47, pp.25-30, 2012.
[42] S. Aravind, R. Scaria, "Biogeography-based optimization for the solution of the combined heat and power economic dispatch problem," IJEIT vol.3, pp.429-32, 2013.
[43] A. Bhattacharya, P.K..Chattopadhyay, "Solution of economic power dispatch problems using oppositional biogeography-based optimization," Electr. Pow. Compo. Sys., vol.38, pp.1139-60, 2010.
[44] P, Civicioglu, "Artificial cooperative search algorithm for numerical optimization problems," Inform. Sciences., vol.229, pp. 58–76, 2013.
[45] E.S. Fraga, L. Yang, L.G. Papageorgiou, "On the modelling of valve point loadings for power electricity dispatch," Appl. Energy., vol. 9, pp.301-303, 2012.

[46] P.H. Chen, H.C. Chang, "Large-scale economic dispatch by genetic algorithm," IEEE T. Power. Syst., vol.10, pp.1919–1926, 1995.
[47] J.B. Park, Y.W. Jeong, J.R. Shin, K.Y. Lee. "An improved particle swarm optimization for non-convex economic dispatch problems," IEEE T. Power. Syst. vol.25, pp.156-166, 2010.
[48] I. Fister Jr, X.S. Yang, I. Fister, J. Brest, D. Fister, "A brief review of nature-inspired algorithms for optimization," Elektroteh. Vestn., vol.80, pp.1-7, 2013.
[49] R. Storn, K. Price, "Differential evolution: a simple evolution strategy for fast optimization," Dr Dobbs. J. vol.22, pp.18–24, 1997.
[50] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report-Tr06t, Erciyes University, Engineering faculty, Computer Engineering Department, Turkey, 2005.
[51] R.M. May, "Simple mathematical models with very complicated dynamics," Nature, vol.261, pp.59-67, 1976.
[52] A.H. Gandomi, X.S. Yang, "Evolutionary boundary constraint handling scheme," Neural. Comput. Appl. vol.21, pp.1449-1462, 2012.
[53] J. Sun, B. Feng, W. Xu, "Particle Swarm Optimization with particles having quantum behavior," In: Proceedings of congress on evolutionary computation, Portland, OR, USA, pp.325-331, 2004.
[54] J. Sun, W. Xu, B. Feng, "Adaptive parameter control for quantum behaved particle swarm optimization on individual level," In: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Big Island, HI, USA, pp.3049-54, 2005.
[55] P. Yadav, R. Kumar, S.K. Panda, C.S. Chang, "An intelligent tuned harmony search algorithm for optimization." Inform. Sciences, vol.196, pp.47-72, 2012.
[56] P. Civicioglu, "Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm," Comput. Geosci. vol. 46, pp.229-247, 2012.
[57] A. Askerzadeh, "Bird Mating Optimizer: an optimization algorithm inspired by bird mating strategies," Commun. Nonlinear. Sci. Numer. Simul., vol.19, pp.1213-1228, 2014.
[58] X.S. Yang, "A new metaheuristic bat-inspired algorithm,"In: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. J. R. Gonzales et.al.), Studies in Computational Intelligence, Springer, Berlin, 284, pp. 65-74, 2010.
[59] N. Sinha, R. Chakrabarti, P.K. Chattopadhyay, "Evolutionary programming techniques for economic load dispatch," IEEE Trans. Evol. Comput., vol.7, pp.83-94, 2003.
[60] D. He, F. Wang, Z. Mao, "A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effect," Int. J. Elec. Power. vol.30, pp.31-38, 2008.
[61] C.K. Panigrahi, V.R. Pandi, "Bacterial foraging optimization: Nelder–Mead hybrid algorithm for economic load dispatch," IET Gener. Transm. Dis., vol.2, pp.556–65, 2008.
[62] N. Amjady, H. Sharifzadeh, "Solution of non-convex economic dispatch problem considering valve loading effect by a new Modified Differential Evolution algorithm," Int. J. Elec. Power., vol.32, pp.893-903, 2010.
[63] B. Mohammadi-Ivatlooa, A. Rabiee, A. Soroud, M. Ehsana, "Iteration PSO with time varying acceleration coefficients for solving non-convex economic dispatch problems," Int. J. Elec. Power., vol.42, pp.508-516, 2012.
[64] S.A. Reddy, K. Vaisakh, "Shuffled differential evolution for large scale economic dispatch," Electr. Pow. Syst. Res., vol.96, pp. 237-24, 2013.
[65] T. Nikham, H.D. Mojarrad, M. Nayeripour, "A new fuzzy adaptive particle swarm optimization for non-smooth economic dispatch," Energy, vol.35, pp.1764-78, 2010.
[66] M. Fesanghary, M.M. Ardehali, "A novel meta-heuristic optimization methodology for solving various types of economic dispatch problem," Energy, vol.34, pp.757-766, 2009.
[67] J.G. Vlachogiannis, K.Y. Lee, "Economic load dispatch – a comparative study on heuristic optimization techniques with an improved coordinated aggregation based PSO," IEEE T. Power. Syst. vol.24, pp.991-1001, 2009.
[68] A. Bhattacharya, P.K. Chattopadhyay, "Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch," IEEE T. Power. Syst., vol.25, pp.1955–1964, 2010.
[69] K.T. Chaturvedi, M. Pandit, L. Srivastava, "Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch," Int. J. Elec. Power., vol.31, pp.249-257, 2009.
[70] K.T. Chaturvedi, M. Pandit, "Self-organizing hierarchical particle swarm optimization for non-convex economic dispatch," IEEE T. Power. Syst., vol. 23, pp.1079–1087, 2008.
[71] M.M. Dalvand, B.M. Ivatloo, A. Najafi, A. Rabiee, "Continuous quick group search optimizer for solving non-convex economic dispatch problems," Electr. Pow. Syst. Res., vol.93, pp. 93–105, 2012.
[72] K. Meng, H.G. Wang, Z.Y. Dong, K.P. Wong , "Quantum-inspired particle swarm optimization for valve-point economic load dispatch," IEEE T. Power. Syst. vol. 25, pp.215-222, 2010.
[73] A.I. Selvakumar, K. Thanushkodi, "Anti-predatory particle swarm optimization solution to non-convex economic dispatch problems," Electr. Pow. Syst. Res., vol.78, pp.2–10, 2008.
[74] H.Lu, P. Sriyanyong, Y.H. Song, T. Dillon, "Experimental study of a new Hybrid PSO with mutation for economic dispatch with non-smooth cost function," Int. J. Elec. Power, vol.32, pp.921–935, 2010.
[75] A.I. Selvakumar, K. Thanushkodi, "A new particle swarm optimization solution to non-convex economic dispatch problem," IEEE T. Power. Syst., vol.22, pp.42-51, 2007.
[76] A. Bhattacharya, P.K. Chattopadhyay, "Solving complex economic load dispatch problems using biogeography-based optimization," Expert. Syst. Appl., vol.37, pp.3605-3615, 2010.
[77] LdS Coelho, V.C. Mariani. "Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization.," Expert. Syst. Appl., vol.34,pp. 1905-1913. 2008.