VOL. 7, NO. 10, OCTOBER 2012 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

MODIFIED ARTIFICIAL BEE COLONY ALGORITHM FOR SOLVING ECONOMIC DISPATCH PROBLEM

R. Murugan[1] and M. R. Mohan[2]
[1]Department of Electrical and Electronics Engineering, KCG College of Technology, Karapakkam, Chennai, India
[2]Department of Electrical Engineering, Anna University, Chennai, India
E-Mail: muruganelect@rediffmail.com

ABSTRACT

A Modified Artificial Bee Colony (ABC) algorithm for Economic Dispatch (ED) problem has been proposed. The Artificial Bee Colony (ABC) algorithm which is inspired by the foraging behavior of honey bee swarm gives a solution procedure for solving economic dispatch problem. It provides solution more effective than Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). However, ABC is good at exploration but poor at exploitation; its convergence speed is also an issue in some cases. To overcome this deficiency, this paper proposes a Modified ABC algorithm (MABC). The performance of the proposed algorithm (MABC) is applied to and tested on IEEE-6 unit and IEEE- 13 unit systems. The results of the proposed algorithm are compared with that obtained by the basic ABC algorithm, lambda - iteration method to prove the validity and effectiveness of the proposed algorithm.

Keywords: economic dispatch, modified artificial bee colony, optimization.

1. INTRODUCTION

The economic dispatch problem is one of the fundamental issues in power systems in order to obtain stable, reliable and secure benefits. The objective of an ED problem is to schedule the committed generating units so as to meet the required load demand at minimum operating cost while satisfying all the equality and inequality constraints of the units [1]. To solve the ED problem, a wide variety of optimization techniques have been applied. Over the past years, a number of approaches have been developed for solving this problem using mathematical programming, i.e., lambda iteration method [2], gradient method [3], linear programming [4], Lagrangian relaxation algorithm [5], quadratic programming [6] and dynamic programming [7]. However, these methods may not be able to provide an optimal solution in large power systems because they usually get stuck at a local optimum. In these classical methods, the cost function of each generator is approximately represented by a simple quadratic function. Linear programming methods are fast and reliable; however, they have the disadvantage of being associated with the piecewise linear cost approximation. Non-linear programming methods have the known problems of convergence and algorithmic complexity. Newton-based algorithms have difficulty in handling a large number of inequality constraints [8].

In order to make the numerical methods more convenient for solving the ED problems, heuristics stochastic search algorithms such as the genetic algorithms (GA) [9-11]. Tabu Search (TS) [12], evolutionary programming (EP) [13-14], simulated annealing (SA) [15], particle swam optimization (PSO) [1], differential evolution algorithm (DE) [16-19]; harmony search [20] and Bacterial Foraging (BF) [21] have been successfully applied. However, none of the mentioned methods have guaranteed obtaining a global optimal solution in finite computational time which could be attributed to their drawbacks. SA algorithm has difficulty in tuning the related control parameters of the annealing schedule and may be too slow when applied for solving the ED problem. GA suffers from the premature convergence and, at the same time, the encoding and decoding schemes essential in the GA approach take longer time for convergence. In PSO and DE, the premature convergence may trap the algorithm into a local optimum, which may reduce their optimization ability when applied for solving the ED problem.

Recently, a new, easy-to-implement, robust evolutionary algorithm has been introduced known as Artificial Bee colony (ABC) algorithm. The Artificial bee colony (ABC) algorithm introduced in [24-27], is one approach that has been used to find an optimal solution in numerical optimization problems. This algorithm is inspired by the behavior of honey bees when seeking a quality food source. The performance of ABC algorithm has been compared with other optimization methods such as GA, differential evolution algorithm (DE), Evolution strategies (ES); Particle swarm inspired Evolutionary Algorithm (PS-EA). The comparisons were made based on various numerical benchmark functions, which consist of uni-modal and multimodal distributions. The comparison results showed that ABC can produce a more optimal solution and thus it is more effective than the other methods in several optimization techniques [28-33]. Reference [33] suggested that ABC is not only a high performance optimizer which is very easy to understand and implement, but also requires little computational bookkeeping. A problem common to all stochastic optimization methods is that a poor balance between exploration and exploitation results in a weak optimization method which will suffer either from premature convergence to local minima if excessively exploitative or will converge very slowly if the algorithm is excessively explorative. ABC is good at exploration but poor at exploitation. The exploration and exploitation are extremely important mechanisms in ABC. In ABC

ARPN Journal of Engineering and Applied Sciences

algorithm the exploration process refers to the ability of seeking for global optimum in the solution space of various unknown optimization problems, while the exploitation process refers to the ability of applying the knowledge of previous solutions to look for better solutions. This paper proposes a compounding a high-efficiency ABC algorithm to balance and accelerate the two process of ABC algorithm with the abilities of prediction and selection, which is called Modified Artificial Bee Colony algorithm (MABC).

2. PROBLEM FORMULATION

The objective of Economic Dispatch problem [1] is to allocate the most optimum real power generation level for all the available generating units in the power station that satisfies the load demand at the same time meeting all the operating constraints. The main objective function of the thermal ED problem is the fuel cost function of the thermal units expressed as:

$$C_i = \sum_{i=1}^{N} a_i P_{Gi}^2 + b_i P_{Gi} + C_i \tag{1}$$

Where a_i, b_i, c_i are the cost coefficients for the i^{th} generator, N is the number of units, P_{Gi} is the real power output of the i^{th} generator.

Subject to:

(i) Inequality constraints

The maximum active power generation of a source is limited by thermal consideration and also minimum power generation is limited by the flame instability of boiler. If the power output of a generator for optimum scheduling of the system is less than a pre-specified value P_{min}, the unit is not synchronized with the bus bar because it is not possible to generate that low value of power from that unit. Hence the generator power cannot be outside the range stated by the inequality

$$P_{Gi,\min} \le P_{Gi} \le P_{Gi,\max} \tag{2}$$

Where $P_{Gi,min}$ = lower real power generation limit of unit 'i'(MW), $P_{Gi,max}$ = upper real power generation limit of unit 'i' (MW).

(ii) Generating constraints

In order to satisfy the load demand, the sum of all the generating units on line must equal the system load plus the transmission losses. The system power balance constraint is,

$$\sum_{i=1}^{N} P_{Gi} - P_L - P_D = 0 \tag{3}$$

Where P_D = load demand, P_{Gi} = real power output produced by unit 'i'(MW), P_L = total loss in the transmission network in terms of loss coefficients.

$$P_L = P^t BP + P^t B_0 + B_{00} \tag{4}$$

Where, P^t = the vector generator loading, B = loss coefficient matrix, B_0 = losscoefficient vector, B_{00} = loss constant.

3. ARTIFICIAL BEE COLONY ALGORITHM (ABC)

In a real bee colony, some tasks are performed by specialized individuals. These specialized bees try to maximize the nectar amount stored in the hive using efficient division of labor and self-organization. The artificial bee colony (ABC) algorithm, proposed by Karaboga [24] in 2005 for real parameter optimization is an optimized algorithm which simulates the forging behavior of a bee colony. The minimal model of swarm-intelligent forage selection in a honey bee colony which the ABC algorithm simulates consists of three kinds of bees: employed bees, onlooker bees and scout bees. Half of the colony consists of employed bees, and the other half includes onlooker bees.

Employed bees are responsible for exploiting the nectar sources explored before and giving information to the waiting bees (onlooker bees) in the hive about the quality of the food sources sites which they are exploiting. Onlooker bees wait in the hive and decide on a food source to exploit based on the information shared by the employed bees. Scout either randomly searches the environment in order to find a new food source depending on an internal motivation or based on possible external clues.

This emergent intelligent behavior in foraging bees can be summarized as follows:

a) At the initial phase of the foraging process, the bees start to explore the environment randomly in order to find a food source.
b) After finding a food source, the bee becomes an employed forager and starts to exploit the discovered source. The employed bee returns to the hive with the nectar and unloads the nectar. After unloading the nectar, she can go back to her discovered source site directly or she can share information about her source site by performing a dance on the area. If her source is exhausted, she becomes a scout and starts to randomly search for a new source.
c) Onlooker bees waiting in the hive watch the dances advertising the profitable sources and choose a sources site depending on the frequency of the dance proportional to the quality of the source.

In the ABC algorithm the position of food source represents a possible solution to the optimization problem, and the nectar amount of a food source corresponds to the profitability (fitness) of associated solution. Each food

VOL. 7, NO. 10, OCTOBER 2012 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

source is exploited by only one employed bee. In other words, the number of employed bee is equal to the number of food sources existing around the hive (number of solutions in the population). The employed bee whose food source has been abandoned becomes a scout. Using the analogy between emergent intelligence in foraging of bees and the ABC algorithm, the main components of the basic ABC algorithm can be designed as detailed below.

3.1. Initialization of the parameters

The parameters of the basic ABC algorithm are the number of food sources (SN) which is equal to the number of the employed bees or onlooker bees, The colony size is 2*SN = (NP), The number of trials after which a food source is assumed to be abandoned (*limit*), and a termination criterion (MCN). In the basic ABC algorithm, the number of employed bees or the onlookers is set equal to the number of food sources in the population. In other words for every food source, there is only one employed bee.

3.2. Producing initial food source sites

If the search spaces considered being the environment of the hive that contains the food source sites, the algorithm starts with randomly producing food sources sites that correspond to the solutions in the search space. Initial food sources are produced randomly within the range of the parameters defined by equation (5).

$$X_{ij} = X_j^{min} + \text{rand}(0, 1)\,(X_j^{max} - X_j^{min}) \qquad (5)$$

Where i = 1…SN, j=1…D, SN is the number of food sources and D is the number of optimization parameters. In addition, counters which store the number of trials of solutions are reset to zero in this phase.

After initialization, the population of the food sources (solutions) is subjected to repeat cycles of the search process of the employed bees, the onlooker bees and the scout bees.

3.3. Sending employed bees to the food sources sites

As mentioned earlier, each employed bee is associated with only one food source site. Hence the number of food source site is equal to the number of employed bees.

An employed bee produces a modification on the position of the food source (solution) in her memory depending upon local information (visual information) and finds neighboring food source, and then evaluates its quality. In ABC, finding a neighboring food source is defined by equation (6).

$$V_{ij} = X_{ij} + \Phi_{ij}\,(X_{ij} - X_{kj}) \qquad (6)$$

With in the neighboring of every food source site represented by X_i, a food source V_i is determined by changing one parameter of Xi. In equation (6), j is a random in the range [1, D] and k $\in$ {1, 2…SN} is a randomly chosen index that has to be different from i. Φ_{ij} is a uniformly distributed real random number in the range [-1, 1].

As can be seen from Equation (6) as the difference between the parameters of the X_{ij} and X_{kj} decreases, the perturbation on the position X_{ij} decreases. Thus, as the search approaches to the optimal solution in the search space, the step length is adaptively reduced. If a parameter value produced by this operation exceeds its predetermined boundaries the parameter can be set to an acceptable value. If the value of the parameter exceed its boundary is set to its corresponding boundaries. If Xi > X_i^{max} then Xi = X^{max}; If Xi < X_i^{min} then $X_i = X_i^{min}$. After producing Vi within the boundaries a fitness value for a minimization problem can be calculated to the solution Vi by (7).

$$Fitness_i = \begin{cases} 1/(1+f_i) & \text{if } f_i \geq 0 \\ 1+\text{abs}(f_i) & \text{if } f_i < 0 \end{cases} \qquad (7)$$

Where f_i is cost value of the solution V_i. For maximization problems, the cost function can be directly used as a fitness function. A greedy selection is applied between X_i and Vi, the better one is selected depending on fitness values representing the nectar amount of the food sources at X_i and V_i. If the source at V_i is superior to that of X_i in terms of fitness values, the employed bees memorize the new position and forget the old one. Otherwise the previous position is kept in memory. If X_i cannot be improved its counter holding the number of trials is incremented by one, otherwise the counter is reset to zero.

3.4. Calculating probability values involved in probabilistic selection

After all employed bees complete their searches, they share their information related to the nectar amount and the positions of their sources within the onlooker bees on the dance area. This is the multiple interaction features of the artificial bees of ABC. Onlooker bees evaluate the nectar information taken from all employed bees and choose a food source site with a probability related to its nectar amount. This probabilistic selection depends on the fitness value of the solutions in the population. A fitness-base selection might be roulette wheel, ranking base, stochastic universal sampling, tournament selection etc. In basic ABC, roulette wheel selection scheme in which each slice proportional to size to the fitness value is employed in Equation (8).

$$P_i = \frac{Fitness_i}{\sum_{i=1}^{N} Fitness_i} \qquad (8)$$

3.5. Food source site selection by onlookers based on the information provided by employed bees

In the basic ABC algorithm, a random real number within the range [0, 1] is generated for each source. If the probability value (Pi in Equation (8)) associated with that source is greater than this random number then the onlooker bee produces a modification on the position of this food source site by using Equation (6) as in the case of the employed bee. After the source is evaluated, greedy selection is applied and the onlooker bee either memorizes the new position by forgetting the old one or keeps the old one. If solution X_i cannot be improved, its counter holding trial is increased by one; otherwise, the counter is reset to zero. This process is repeated until all onlookers are distributed onto food source sites.

3.6. Abandonment criteria: limit and scout production

In a cycle, after all employed bees and onlooker bees complete their searches the algorithm checks to see if there is any exhausted source to be abandoned. In order to decide if a source is to be abandoned, the counters which have been updated during search are used. If the value of the counter is greater than the control parameter of the ABC algorithm, known as the "*limit*", then the source associated with this counter is assumed to be exhausted and is abandoned.

The food source abandoned by its bee is replaced with a new food source is discovered by the scout, which represents the negative feedback mechanism and fluctuation property in the self-organisation of ABC. This is simulated by producing a site position randomly and replacing it with the abandoned one. Assume that the abandoned source is Xi, and then the scout randomly discovered a new food source to be replaced with Xi. This operation can be defined as Equation (5).

In the basic ABC, it is assumed that only one source can be exhausted in each cycle, and only one employed bee can be a scout. If more than one counter exceeds the "*limit*" values, one of the maximum ones might be chosen programmatically.

4. THE ABC ALGORITHM FOR THE ED PROBLEM

The detailed implementation of ABC algorithm to find a solution for the ED problem is given below:

Step-1: Initialization of the control parameters

The parameters of the basic ABC algorithm are the colony size (NP), the number of food sources (SN=NP/2), the limit for scout, L = SN*D, D is the dimension of the problem and a Maximum Cycle Number (MCN).

Step-2: Producing initial food source sites

The initialize the power loadings $Xi=[P_1, P_2, \ldots P_D]^T$, i=1,2,….NP such that each element in the vector is determined by $P_{ij} = P_j^{min} + rand(0,1)(P_j^{max} - P_j^{min})$, j=1, 2...D with one generator as a dependent generator and evaluate the fitness value using Eq.(9) then select SN the best food source on the basis of highest fitness value as initial food sources and set the cycle=1, the trail number of each solution X_i, $trial_i$, is equal to zero.

$$\text{Fitness } (F) = \frac{1}{\left[\sum_{i=1}^{NP}(1 + F_{ti}(Pi))\right]} \tag{9}$$

F_{ti} is fuel cost of each food source.

Step-3: Sending employed bees to the food sources [SN] and assigning the nectar amount

In this step each employed bee produces a new solution V_i by using Equation (6) and computes the fitness value of the new solution using Equation (9) satisfying with all constraints. If the fitness of the new one is higher than that of the previous one, the employed bee memorizes the new position and forgets the old one; otherwise the employed bee keeps the old solution.

Step-4: Sending the onlooker bees to the food sources depending on their amount of nectar

This step required to calculate the probability value P_i of the solution X_i by means of their fitness value using Eq. (8). An onlooker bee selects a solution to update its solution depending on the probabilities and determines a neighbour solution around the chosen one. In the selection procedure for the first onlooker, a random number is generated between [0, 1] and if this number is less than P1, the solution is updated using Equation (6). Otherwise, the random number is compared with P2 and if less than that, the second solution is chosen. Otherwise, third probability of third solution is checked. This process is repeated until all onlookers have been distributed to solutions. The distributed onlooker bee updated its own solution just as the employed bees do.

Step-5: Send the scouts to the search area to discover new food sources

If the solution Xi is not improved through step 3 and 4, the $trail_i$ value of solution Xi will be increased by 1. If the $trail_i$ of the solution is more than the predetermined *"limit"* the solution Xi is considered to be an abandoned solution, meanwhile the employed bee will be changed into a scout. The scout randomly produces the new solution and then compares the fitness of new solution with that its old one. If the new solution is better than the old solution, it is replaced with the old one and set its own $trail_i$ into zero. This scout will be changed into employed bee. Otherwise, the old one is retained in the memory.

Step-6: Record the best solution

In this step, the best solution so far is recorded and increase the cycle by 1.

Step-7: Check the termination criterion

If the cycle is equal to the maximum cycle number (MCN) then the algorithm is finished; otherwise go to step-3.

The complete flowchart for ABC algorithm is shown in Figure-1.

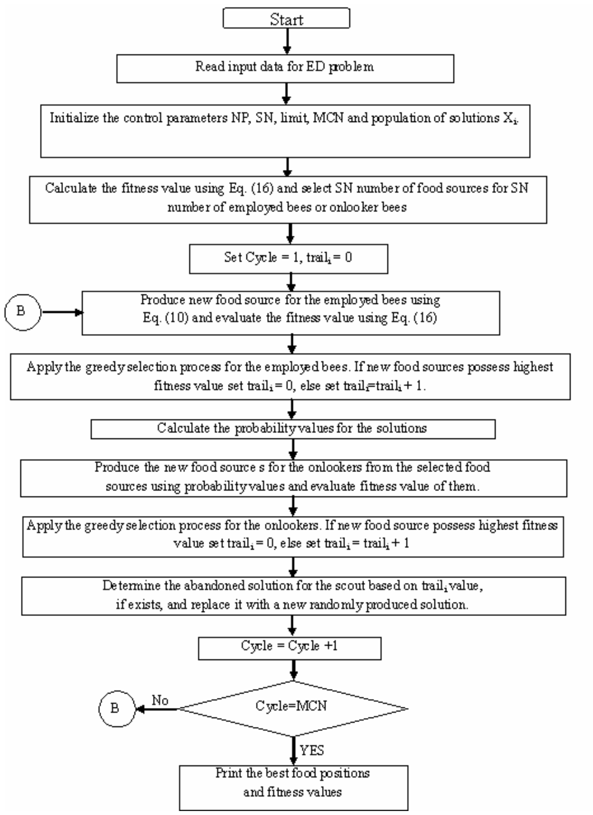

Figure-1.

5. MODIFIED ARTIFICIAL BEE COLONY ALGORITHM

5.1. Drawbacks in ABC

In ABC algorithm, the processes of the exploration and exploitation contradict with each other, so the two abilities should be well balanced for achieving good optimization performance. According to the search form of ABC algorithm which is described as Equation (6), a candidate solution would be generated by moving the previous one towards another solution selected randomly from the population. The ABC algorithm has already proved to be a very effective technique for solving global optimization. ABC is not only a high performance optimizer which is very easy to understand and implement. However, ABC could be slow to converge and sometimes trap in a local optimal solution.

5.2. Modification in ABC

In order to further improve the performances of ABC, three major changes are made by introducing the best-so-far solution, inertia weight and acceleration coefficients to modify the search process .In addition, the search form of ABC described as Equation (6) is good at exploration but poor at exploitation. Therefore, to improve the exploitation, the modification forms of the employed bees and the onlooker ones are different in the second acceleration coefficient. The improved ABC algorithm is called as I-ABC.

The operation process can be modified in the following form:

$$V_{ij} = X_{ij}W_{ij} + 2(\phi_{ij} - 0.5)(X_{ij} - X_{kj})\Phi_1 + \varphi_{ij}(X_j - X_{kj})\Phi_2 \quad (10)$$

Where V_{ij} is the new feasible solution that is a modified feasible solution depending on its previous solution $X_{ij}.W_{ij}$ is the inertia weight which controls impacts of the previous solution $X_{ij}.X_j$ is the jth parameter of the best-so-far solution, Φ_{ij} and φ_{ij} are random numbers between [0, 1], Φ_1 and Φ_2 are positive parameters that could control the maximum step size. However, if the global fitness is very large, bees are far away from the optimum values. So a big correction is needed to search the global optimum solution and then w, Φ_1 and $\Phi2$ should be bigger values. Conversely, only a small modification needed, then W, Φ_1 and Φ_2 must be smaller values. So in order to further improve the search efficiency of the bees, it is investigated to modify the parameter that is used to calculate new candidate food sources as [25]. In this investigation, inertia weight and acceleration coefficients are defined as functions of the fitness in the search process of ABC. They are proposed as follows:

$$W_{ij} = \Phi_1 = 1 \Big/ \left(1 + \exp(-Fitness(i)/ap)\right) \quad (11)$$

$$\Phi_2 = 1, \text{ if a bee is employed one} \quad (12)$$

$$\Phi_2 = 1 \Big/ (1 + \exp(-Fitness(i)/ap)), \text{ if a bee is onlooker one} \quad (13)$$

Where ap is the $Fitness(1)$ in the first iteration. In order to further balance the process of the exploration and the exploitation, the modification forms of the employed bees and the onlooker ones are different in the acceleration coefficient Φ_2. In Reference [25] it is suggested that the main advantages of I-ABC are to achieve a fast convergence speed and to find a good solution. One more algorithm is called the Gbest-guided ABC (GABC) which is having advantage in field of diversity. The solution search equation of GABC is given by the following form [34]:

$$V_{ij} = X_{ij} + 2(\phi_{ij} - 0.5)(X_{ij} - X_{kj}) + \varphi_{ij}(X_j - X_{kj}) \quad (14)$$

Where V_{ij} is the new feasible solution that is a modified feasible solution depending on its previous solution $X_{ij}.X_j$ is the jth parameter of the best - so-far

ARPN Journal of Engineering and Applied Sciences

solution, ϕ_{ij} is a random number between [0, 1], φ_{ij} [0, c], c is a nonnegative constant, which is set 1.

In order to combine the bright sides of I-ABC, ABC and GABC, the paper proposes a high-efficiency hybrid ABC algorithm which has the abilities of prediction and selection. This hybrid optimization method is called as Modified Artificial Bee Colony algorithm (MABC). In MABC algorithm; there are three different solution search equations. The first one is Equation (6), which is solution modification form of the original ABC algorithm. The second one is solution search equation of I-ABC as given in Equation (10). The third one is solution search equation of GABC as given in Equation (14).

In initialization, MABC like ABC starts by associating all employed bees with randomly generated food sources. After initialization, the population of the food sources is subject to repeated cycles of the search processes of the employed bees, the onlooker bees and the scout bees. The main difference between MABC and anyone of ABC, I-ABC and GABC is how bees get the candidate solutions. In MABC, an employed bee firstly works out three new solutions by three different solution search equations, and then chooses and determines the best one as the candidate solution. Here, due to calculating the candidate solution before the employed bee decide where they should go to explore, the process of calculating new food position is called 'predict'. After the bees 'predict' new candidate solution by three different solution search equations, they select the best one from the three solutions as the candidate solution.

If the fitness values of the candidate solution is better than the best fitness value achieved so far, then the employed bee's moves to this new food source and synchronously abandons the old one, otherwise it remains the previous food source in its mind. When all employed bees have finished this process, they share the fitness information with the onlookers, each of which selects a food source according to probability given in Equation (8). As in the case of the employed bee, an onlooker 'predicts' three modification on the position in her memory, and then selects the best one as the candidate source and checks the fitness value of the candidate source. Providing that the fitness value of the candidate source is better than that of the previous one, the bee would memorize the new position and forget the old one.

In MABC, the three solution search equations are independently calculated, but influence each other by the chosen best solution. On the whole, the MABC has inherited the bright sides of the other three algorithms. The diagrammatic representation of solving ED problem using MABC algorithm is shown in flowchart Figure-2.

6. TEST RESULTS AND ANALYSIS

The MABC algorithm is combination of the ABC, GABC and I-ABC algorithms for ED problem is tested on two test systems.

a) First system has the 6- units [22] have a total load of 900MW and constraints are given by Eqs. (2) and (3).
b) In the second case, the thirteen units [23], a total load of 1925MW and constraints given by Equations (2) and (3). Power losses have also been considered for these two systems using the B-matrix.

The performance of both the systems using MABC is compared with lambda iteration and ABC algorithm and is observed that MABC produces superior results and out performance lambda iteration and ABC algorithm in terms of convergence behavior, solution quality, consistency and computational efficiency.

ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

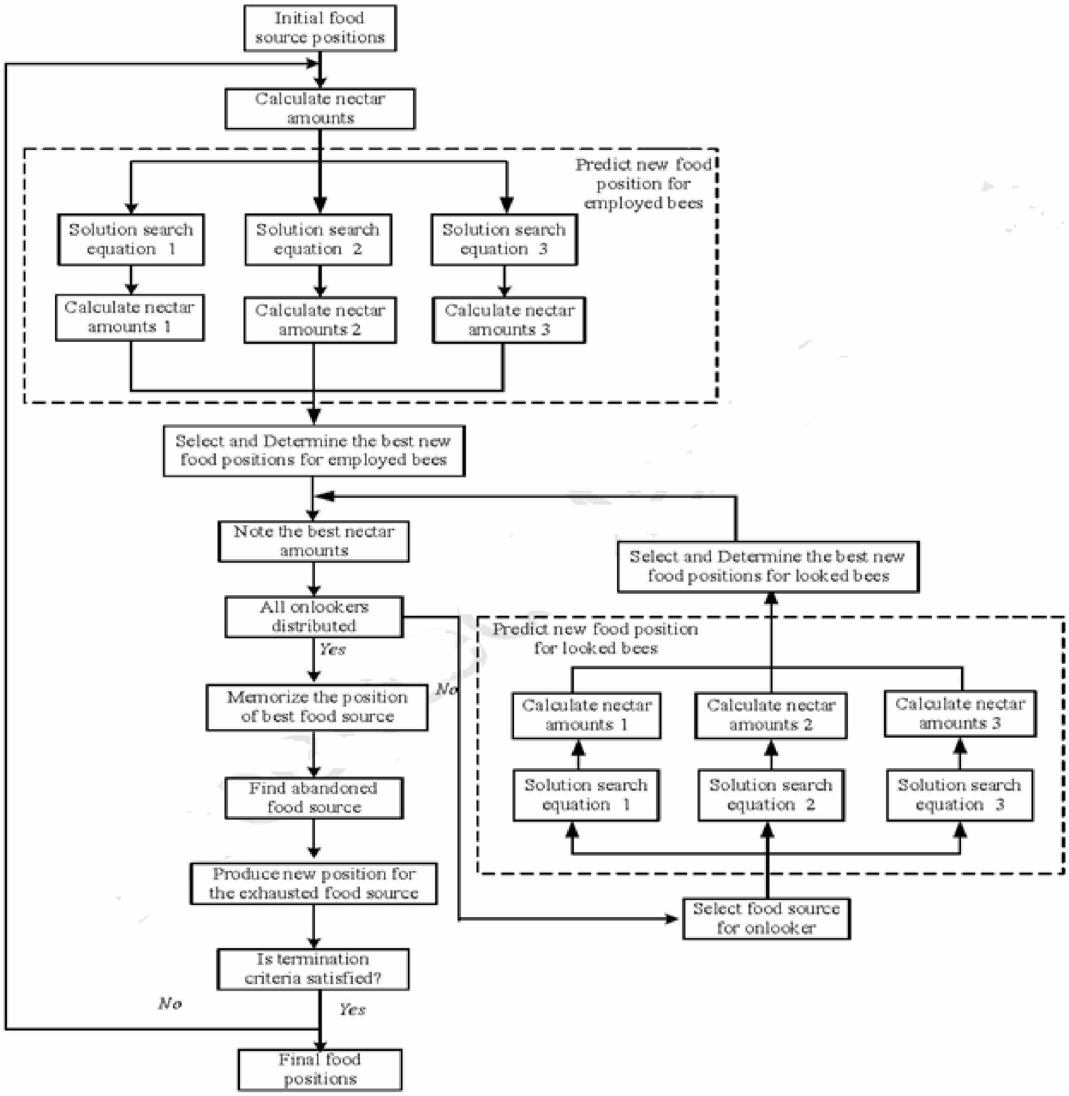

Figure-2.

6.1. Comparison of MABC with ABC algorithm

6.1.1. Influence of colony size and cycle

The colony size and cycle are another important issue in stochastic search methods. Too large a colony size and cycles makes an algorithm slow and computationally inefficient, while a very small colony size and cycle may not capable of searching a minimum, particularly in complex multimodal problems. The optimal colony size and cycle depends on problem dimension. Larger the dimension, larger is the colony size and cycle required to achieve good results. Tests were carried out for colony size of 20, 40 and 60 with variation of cycles from 100, 200 and 300 for the above mentioned two generation systems. Tables 1 and 2 are lists the performance the two methods for different colony size and cycle sets for the 6 unit and 13 unit systems respectively. With increase in colony size and cycle; a steady improvement in minimum and average costs was noticed. Moreover, out of 30 trails the number of solution hits towards global optimal solution also noticed. A colony size of 40 and cycle 300 were found to be optimum for ABC and for MABC colony size 20 with cycle 300 were found to give best results for 6 unit system. The colony size of 60 and cycle 300 were found to be optimum for ABC and for MABC colony size 40 with cycle 200 or 300 were found to give best results for 13 unit systems. Here MABC requires fewer numbers of control parameters since at these stages MABC more dominating in the view of more number of global solution hits, minimum average cost and computational time.

ARPN Journal of Engineering and Applied Sciences

Table-1. Influence of colony size for 6 unit system.

S. No.	Colony size	cycle	ABC varient	Minmum cost (Rs/hr)	F_{mean}	Maximum cost (Rs/hr)	Average cost (Rs/hr)
1	20	100	ABC	47047	0	47071	47053.3300
			MABC	47045	2	47059	47047.4880
		200	ABC	47045	2	47048	47049.2000
			MABC	47045	19	47046	47045.3660
		300	ABC	47045	21	47048	47047.2333
			MABC	47045	29	47046	47045.3300
2	40	100	ABC	47046	0	47061	47049.532 0
			MABC	47045	23	47046	47045.2660
		200	ABC	47045	5	47047	47047.5000
			MABC	47045	30	47045	47045.0000
		300	ABC	47045	24	47046	47046.2000
			MABC	47045	30	47045	47045.0000
3	60	100	ABC	47046	0	47058	47051.3300
			MABC	47045	30	47045	47045.0000
		200	ABC	47045	9	47046	47046.7330
			MABC	47045	30	47045	47045.0000
		300	ABC	47045	30	47045	47045.0000
			MABC	47045	30	47045	47045.0000

F_{min} - Number of hits towards global optimum.

Table-2. Influence of colony size for 13 unit system.

S. No.	Colony size	cycle	ABC varient	Minmum cost (Rs/hr)	F_{mean}	Maximum cost (Rs/hr)	Average cost (Rs/hr)
1	20	100	ABC	19351	0	19368	19355.066
			MABC	19349	2	19352	19350.2667
		200	ABC	19349	1	19353	19351.2330
			MABC	19349	11	19350	19349.6660
		300	ABC	19349	5	19351	19350.8000
			MABC	19349	26	19350	19349.2000
2	40	100	ABC	19351	0	19361	19352.9330
			MABC	19349	2	19352	19350.5330
		200	ABC	19350	0	19350	19350.0333
			MABC	19349	24	19350	19349.2000
		300	ABC	19349	10	19350	19349.0200
			MABC	19349	30	19349	19349.0000
3	60	100	ABC	19350	0	19355	19352.6600
			MABC	19349	21	19350	19349.3300
		200	ABC	19349	3	19352	19351.2000
			MABC	19349	30	19349	19349.000
		300	ABC	19349	8	19357	19353.0000
			MABC	19349	30	19349	19349.0000

F_{min} - Number of hits towards global optimum.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

6.1.2. Convergence characteristics

The control parameters such as colony size, food sources, limit, and cycle are having very much influence on the convergences behaviour of the ABC and MABC algorithms. The convergence behaviour of the two methods was tested for 30 trails employing the same evaluation function with above mentioned optimum control parameters of ABC and MABC respectively. The results for ABC and MABC algorithms for one trail are shown in the Figures 3 and 4 for 6, 13 unit systems convergence characteristics with different colony size and same cycles and Figures 5 and 6 shows the convergence characteristics of 6, 13 unit systems with different cycles and same colony size. These Figures shows that the MABC takes less number of cycles for all kind of low control parameters but ABC takes more number of cycles and most of the trails it takes 250 to 300 cycles to obtain the optimum cost. Note the ABC requires large amount control parameters to obtain convergence point. These convergences cycles of ABC are 4 to 5 times more than our proposed MABC algorithm; hence the proposed algorithm reduces the convergence time drastically. Table-3 shows the comparison of Convergence characteristics of two algorithms for 6 unit system, from Table-3 we can conclude that MABC requires less time to obtain the optimum cost.

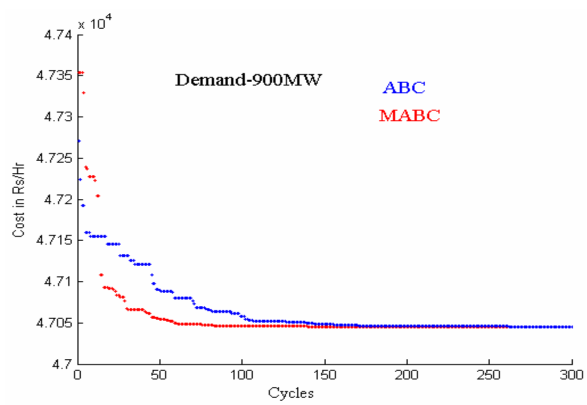

Figure-3.

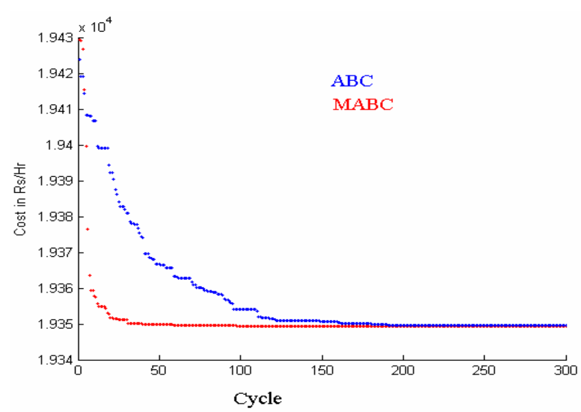

Figure-4.

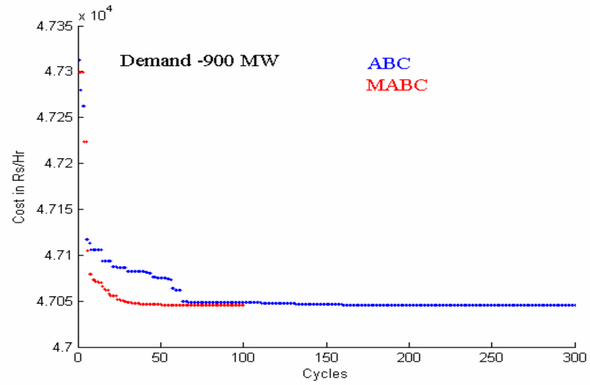

Figure-5.

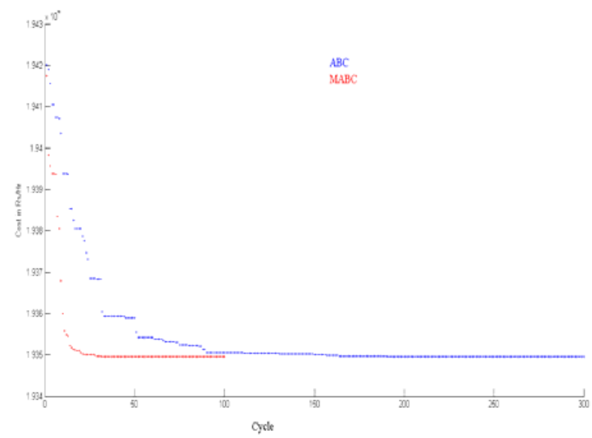

Figure-6.

Table-3. Convergence characteristics for 6 unit system.

S. No.	Colony size	cycle	ABC varient	Time
1	20	100	ABC	3.3890
			MABC	1.5124
		200	ABC	9.5197
			MABC	3.2424
		300	ABC	6.4706
			MABC	5.5560
2	40	100	ABC	1.5988
			MABC	2.1778
		200	ABC	8.6362
			MABC	4.7436
		300	ABC	5.8031
			MABC	7.5684
3	60	100	ABC	2.047 0
			MABC	2.8750
		200	ABC	3.853 0
			MABC	6.2406
		300	ABC	8.4722
			MABC	9.8672

6.1.3. Solution quality

The minimum, maximum and average costs obtained out of 30 trails for ABC and MABC algorithms are given in Tables 1 and 2 for various control parameters. It can be see that the minimum cost of ABC is more than the MABC and no hits towards the global optimum solution for the control parameters such as food source 20, 40, 60 with cycle 100 and less number of global minimum hits for food source 20 and 300 and high average cost. Therefore the MABC achieve minimum cost even with less control parameters as well as average cost produced by MABC algorithm is least compared with ABC algorithm for all colony size and all variations of cycle, emphasizing the better solution quality of the method. To analyze the solutions more closely, the dynamic convergence behaviour of the two methods was also studied by calculating the mean value, standard deviation and number of hits towards global optimal solution of each of food source in the colony after each run. The mean value μ and standard deviation σ are defined as:

$$\mu = \frac{\sum_{i=1}^{SN} f(x_i)}{SN} \tag{25}$$

$$\sigma = \sqrt{\frac{1}{SN}\sum_{i=1}^{SN}(f(x_i)-\mu)^2} \tag{26}$$

SN is number of food source here and f (x_i) is the objective function value at end of each cycle. Figures 7 and 8 shows the plot of standard deviation for 6 and 13 unit systems of ABC, MABC algorithms with above mentioned optimum control parameters. The plot of mean value is shown in Figures 9 and 10 for 6, 13 unit systems with above mentioned control parameters for ABC and MABC algorithms respectively. The MABC algorithm records a clear superiority over the ABC algorithm and produces better dynamic convergence as the mean cost and the standard deviation of the colony reduces continuously even for low value of control parameters. The ABC algorithm show premature convergence at low value control parameters and do not achieve minima.

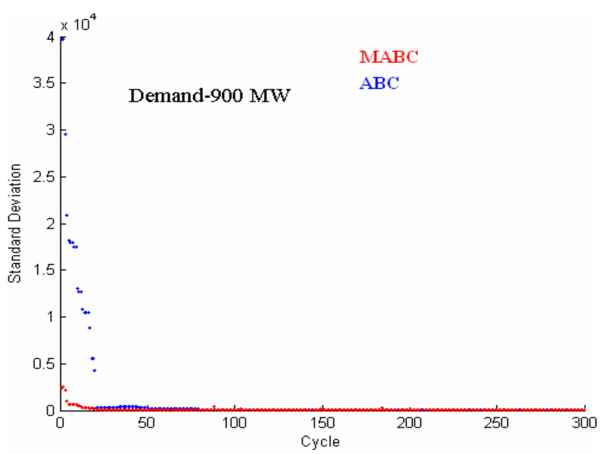

Figure-7.

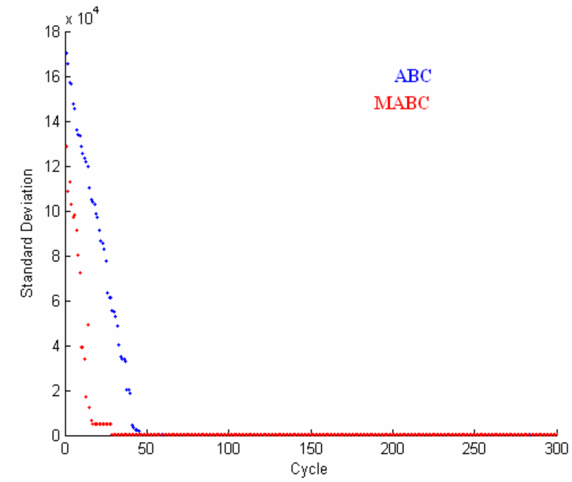

Figure-8.

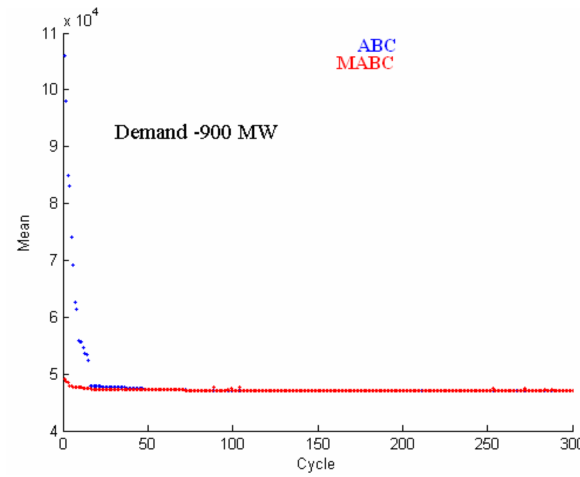

Figure-9.

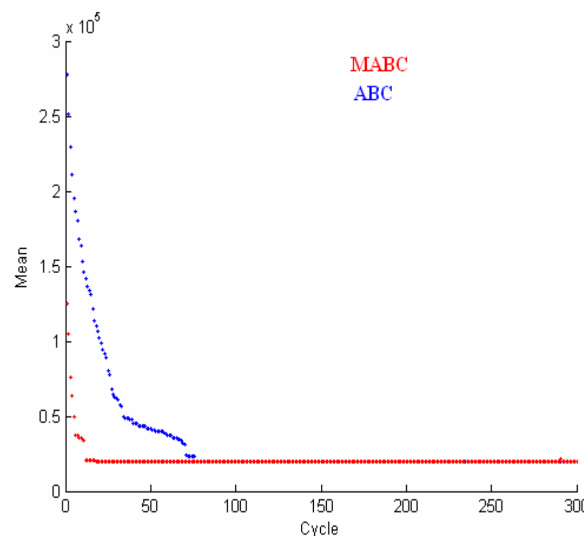

Figure-10.

ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

6.1.4. Computational efficiency

Tables 4 and 5 present the best cost achieved by the ABC, MABC algorithms and lambda iteration method for the two test cases, while satisfying the constraints. It can be seen that MABC is computationally quite efficient as the CPU time required is less compared with all three methods and producing same cost as ABC with less control parameters. Hence, the proposed method improves the convergence speed.

6.1.5. Robustness

Due to the inherent randomness involved, the performance of heuristic search based optimization algorithms cannot be judged by the result of a single trial. Therefore, many trials with different initial colony size were carried out to test the robustness/consistency of MABC algorithm with ABC algorithm. The lowest cost for each of 30 trails has been plotted in Figures 11 and 12 for the above mentioned optimum control parameters such for 6, 13 unit systems. Figures 13-16 shows combined cost convergence plot of 6, 13 unit systems for all 30 trails of ABC and MABC from which it can be seen that MABC algorithm produces lowest production cost most consistently as compared with ABC in all runs. From the above discussions it is clear that our proposed MABC algorithm towards the global convergences making it more efficient, robust and consistent.

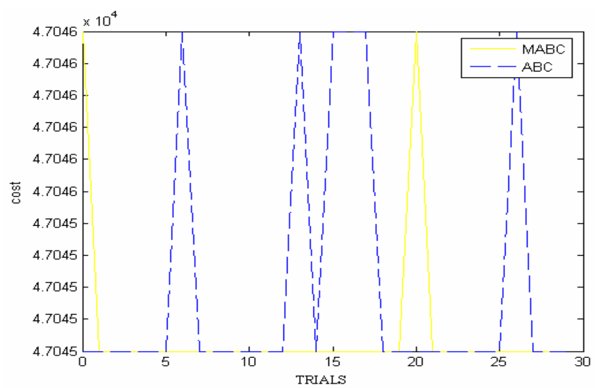

Figure-11.

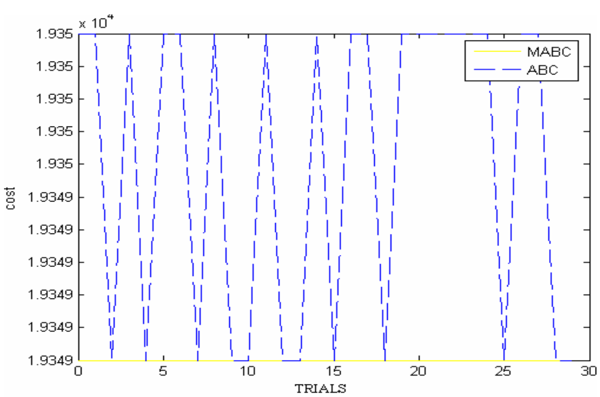

Figure-12.

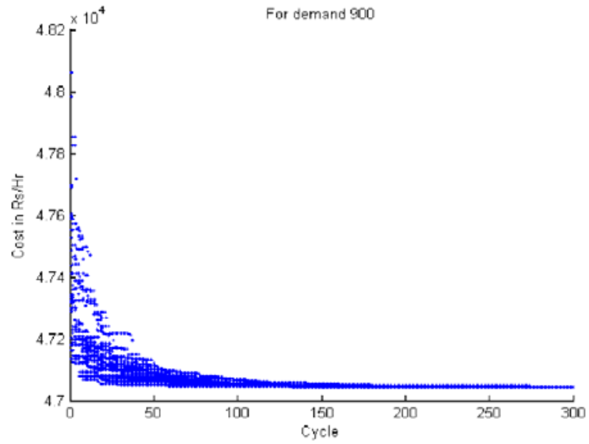

Figure-13.

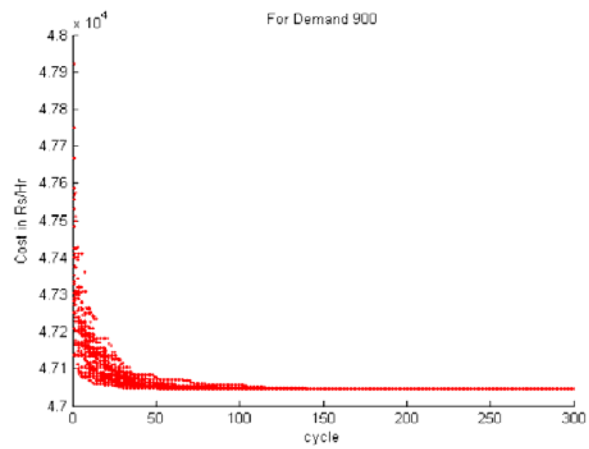

Figure-14.

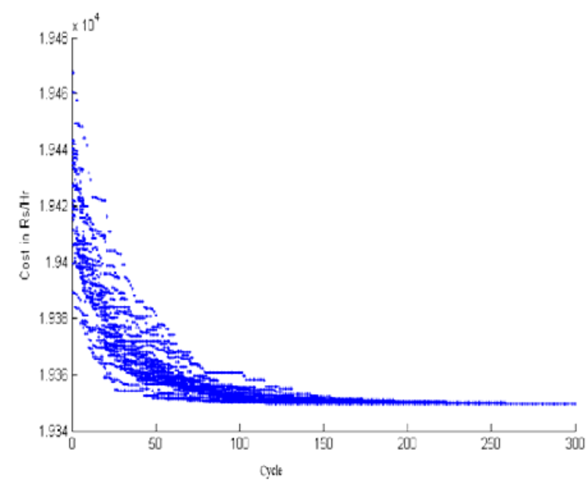

Figure-15.

VOL. 7, NO. 10, OCTOBER 2012 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

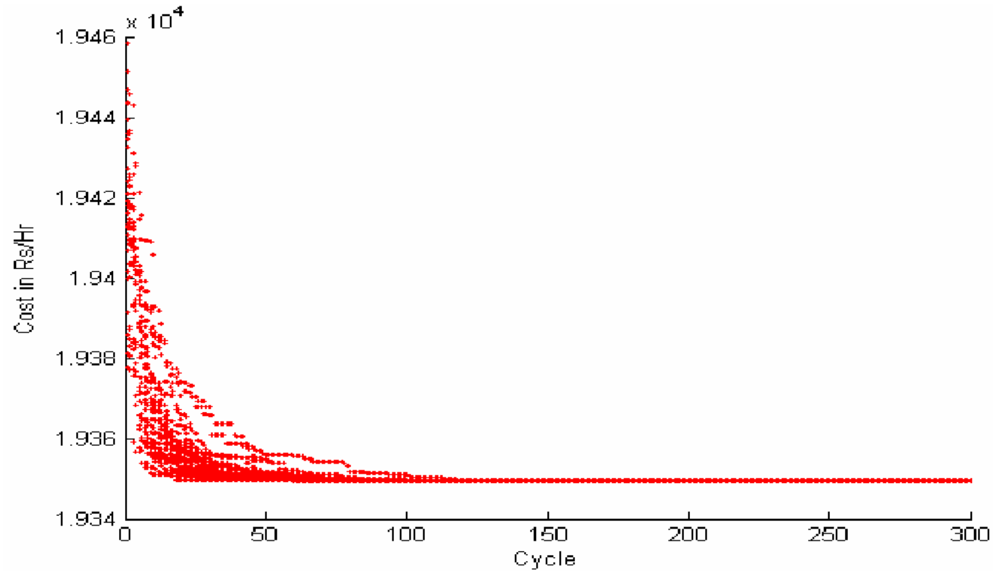

Figure-16.

Table-4. Generator output for least cost for 6 unit system.

Unit power output	Lambda	ABC	MABC
P1(MW)	36.89	36.95	36.76
P2(MW)	21.07	21.08	21.10
P3(MW)	163.90	163.29	164.36
P4(MW)	153.20	153.78	153.53
P5(MW)	284.13	283.67	284.13
P6(MW)	272.69	273.22	272.08
Total power demand (MW)	900	900	900
Total loss	31.98	31.99	31.97
Total cost(Rs/hr)	47130.48	47045.29	47045.25
CPU time(s)	15.5170	6.9000	5.3120

ARPN Journal of Engineering and Applied Sciences

Table-5. Generator output for least cost for 13 unit system.

Unit power output	Lambda	ABC	MABC
P1(MW)	119.71	118.92	120.16
P2(MW)	101.35	114.74	117.17
P3(MW)	142.03	141.90	142.01
P4(MW)	135.81	134.56	135.78
P5(MW)	130.40	129.08	130.18
P6(MW)	143.29	143.34	143.36
P7(MW)	40.0000	40.0000	40.0000
P8(MW)	40.0000	40.0000	40.0000
P9(MW)	55.0000	55.0000	55.0000
P10(MW)	55.0000	55.0000	55.0000
P11(MW)	426.68	468.16	462.69
P12(MW)	282.14	288.19	284.36
P13(MW)	260.36	237.49	240.55
Total power demand (MW)	1925	1925	1925
Total loss	41.60	41.37	41.25
Total cost(Rs/hr)	20524.94	19349.52	19349.44
CPU time(s)	19.95	7.0000	6.0000

6. CONCLUSIONS

This paper has presented an MABC algorithm for solving economic dispatch problem. In MABC, the three search equations are independently calculated, but influence each other by the chosen best solution. The results indicate that the global search ability of MABC has been improved. In addition; the proposed MABC algorithm shows the very fast convergence. On the whole, the MABC could be thought as the combination of the bright sides of the ABC, GABC and I-ABC algorithms.

REFERENCES

[1] J.B. Park, K.S. Lee, J.R. Shin and K.Y. Lee. 2005. A particle swarm optimization for economic dispatch with Nonsmooth cost functions. IEEE Trans. Power Syst. 20(1): 34-42.

[2] C. CL and W. SC. 1993. Branch-and bound scheduling for thermal generating units. IEEE Trans. Energy Convers. 8(2): 184-189.

[3] J.C. Dudu, P. Martin, A. Marlin and J. Pouget. 1972. An optimal formulation and solution of short-range operating problems for a power system with flow constraints, Proc. IEEE. 60(1): 54-63.

[4] R.A. Jabr, A.H. Coonick and B.J. Cory. 2000. A homogeneous linear programming algorithm for the security constrained economic dispatch problem. IEEE Trans. Power Syst. 15(3): 930-937.

[5] J.F. Bard. 1988. Short-term scheduling of thermal-electric generators using Lagrangian relaxation, Oper. Res. 36(5): 756-766.

[6] J.Y. Fan and L. Zhang. 1998. Real-time economic dispatch with line flow and emission constraints using quadratic programming. IEEE Trans. Power Syst. 13(2): 320-325.

[7] S. Pothiya, I. Ngamroo and W. Kongprawechnon. 2007. Application of multiple tabu search algorithm to solve dynamic economic dispatch considering generator constraints, Energy Convers. Manage. 49(6): 506-516.

[8] J.S. Al-Sumait, A.K. AL-Othman and J.K. Sykulski. 2007. Application of pattern search method to power system valve-point economic load dispatch. Electr. Power Energy Syst. 29(10): 720-730.

[9] J.O. Kim, D.J. Shin, J.N. Park and C. Singh. 2002. Atavistic genetic algorithm for economic dispatch with valve point effect. Electr. Power Syst. Res. 62(3): 201-207.

[10] C.-L. Chiang. 2005. Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels. IEEE Trans. Power Syst. 20(4): 1690-1699.

[11] S. Baskar, P. Subbaraj and M.V.C. Rao. 2003. Hybrid real coded genetic algorithm solution to economic dispatch problem. Comput. Electr. Eng. 29(3): 407-419.

[12] W.-M. Lin, F.-S. Cheng and M.-T. Tsay. 2002. An improved tabu search for economic dispatch with multiple minima. IEEE Trans. Power Syst. 17(1): 108-112.

[13] H.T. Yang, P.C. Yang and C.L. Huang. 1996. Evolutionary programming based economic dispatch for units with non-smooth fuel cost functions. IEEE Trans. Power Syst. 11(1): 112-118.

[14] N. Sinha, R. Chakrabati and P.K. Chattopadhyay. 2003. Evolutionary programming techniques for economic load dispatch, IEEE Trans. Evol. Comput. 7(1): 83-94.

[15] K.P. Wong and C.C. Fong. 1993. Simulated annealing based economic dispatch algorithm. IEE Proc. C. 140(6): 509-515.

[16] S.-K. Wang, J.-P. Chiou and C.-W. Liu. 2007. Non-smooth/non-convex economic dispatch by a novel hybrid differential evolution algorithm. IET Gener. Trans. Distrib. 1(5): 793-803.

[17] L.S. Coelho, A.D.V. Almeida and V.C. Mariani. 2008. Cultural differential evolution approach to optimize the economic dispatch of electrical energy using thermal generators. In: Proceeding of 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Hamburg, Germany. pp. 1378-1383.

[18] L.S. Coelho and V.C. Mariani. 2007. Improved differential evolution algorithms for handling economic dispatch optimization with generator constraints. Energy Convers. Manage. 48(5): 1631-1639.

[19] N. Noman and H. Iba. 2008. Differential evolution for economic load dispatch problems. Electr. Power Syst. Res. 78(3): 1322-1331.

[20] L.S. Coelho and V.C. Mariani. 2009. An improved harmony search algorithm for power economic load dispatch. Energy Convers. Manage. 50(10): 2522-2526.

[21] B.K. Panigrahi and V.R. Pandi. 2008. Bacterial foraging optimization: Nelder-Mead hybrid algorithm for economic load dispatch. IET Gener. Trans. Distrib. 2(4): 556-565.

[22] Saumendra Sarangi. 2009. A thesis of “particle swarm optimization applied to Economic Dispatch Problem.

[23] P. Aravindhabau and K.R. Nayar. 2002. Economic dispatch based on optimal lambda using radial basis Function network. Electrical Power and Energy Systems. 24: 551-556.

[24] D. Karaboga. 2005. An idea based on honey bee swarm for numerical optimization. Technical Report_TR06, Ericyes University, Engineering faculty, Computer Engineering department, Turkey.

[25] Basturk B and Karaboga D. 2006. An artificial bee colony (ABC) algorithm for numerical function optimization. In: Proceeding of IEEE Swarm Intell. Symp. Indianapolis.

[26] Karaboga D and Busturk B.A. 2007. Powerful and efficient algorithm for numerical function optimization: artificial bee colony optimization. Journal of global optimization. 39: 459-471.

[27] D. Karaboga. 2005. An idea based on honey bee swarm for numerical optimization. Technical Report_TR06, Ericyes University, Engineering faculty, Computer Engineering department, Turkey.

[28] D. Karaboga and B. akay. 2009. A comparative study of artificial bee colony, applied mathematics and computation. 214: 108-132.

[29] D. Karaboga and B. Basturk. 2008. On the performance of artificial bee colony (ABC) algorithm Applied soft computing. 8: 687-697.

[30] A. Singh. 2009. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. Applied soft computing. 9: 625-631.

[31] F. Kang and j. Li O.XU. 2009. Structural inverse analysis by hybrid simplex artificial bee colony algorithms, computers and structures. 87: 861-70.

[32] N. Karaboga. 2009. A new design method based on artificial bee colony algorithm for digital IIR filters. Journal of the Franklin institute. 346: 328-48.

[33] Guo Qiang Li, Peifeng Niu and Xingjun Xiao. 2012. Development and investigation of efficient artificial bee colony algorithm for numerical function optimization. Applied soft computing. 12: 320-332.

[34] Guopu Zhu and Sam Kwong. 2010. Gbest-guided artificial bee colony algorithm for numerical function optimization. Applied Mathematics and computation. 217: 3166-3173.